



## Руководство администратора

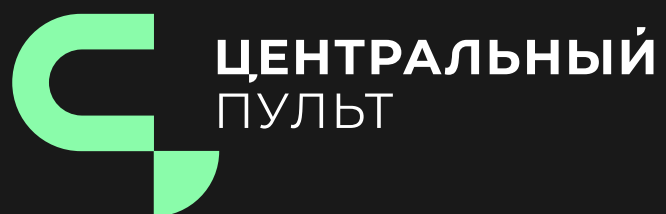
v.3.8.82 - 19.10.2021



1. Введение .....	5
1.1. Описание документа .....	6
1.2. Назначение системы .....	7
2. Системные требования .....	9
2.1. Системные требования к серверу .....	10
2.2. Системные требования к агенту .....	11
3. Принципы функционирования системы .....	13
3.1. Архитектура системы .....	14
3.2. Описание функционирования системы и её частей .....	17
4. Задачи администратора .....	19
4.1. Настройка параметров работы системы .....	20
4.1.1. Установка агента .....	21
4.1.1.1. Автоинсталляция .....	22
4.1.1.2. Linux .....	23
4.1.1.3. Mac OS X .....	24
4.1.1.4. Windows .....	25
4.1.1.5. Wiren Board 6 .....	26
4.1.2. Конфигурация агента .....	27
4.1.2.1. Подключение агента к Kafka Cluster .....	29
4.1.3. Конфигурация сервера .....	30
4.1.4. Настройка элементов web-интерфейса .....	48
4.1.5. Служба "saymon-server" .....	56
4.1.6. Просмотр websocket-нотификаций .....	57
4.1.7. Увеличение количества обработчиков SNMP-Trap .....	58
4.1.8. Сброс системы к заводским настройкам .....	59
4.1.9. Включение функционала инцидентов .....	60
4.2. Модуль аналитики .....	61
4.2.1. Подключение модуля аналитики к Центральному Пульту .....	62
4.2.2. Мониторинг работы модуля аналитики .....	63
4.3. Управление сенсорами мониторинга .....	64
4.3.1. Скрытие неиспользуемых сенсоров .....	65
4.3.2. Настройка пользовательских сенсоров .....	66

4.4. Настройка уведомлений .....	72
4.4.1. Активация функционала отправки SMS и голосовых вызовов .....	73
4.4.2. Отправка почтовых уведомлений .....	74
4.4.3. Настройка уведомлений в Telegram .....	75
4.5. Интеграция со сторонними системами .....	77
4.5.1. Интеграция с Zabbix .....	78
4.5.2. Интеграция с Grafana .....	80
4.6. Управление логированием .....	85
4.6.1. Конфигурация log-файлов .....	86
4.6.2. Конфигурация ротации log-файлов .....	87
4.6.3. Просмотр информации о Журнале событий .....	88
4.6.4. Назначение ответственного за событие .....	89
4.6.5. Установка ограничения для логирования .....	90
4.6.6. Удаление всех SNMP-Trap'ов из Журнала событий .....	91
4.6.7. Удаление логов .....	92
4.7. Управление учётными записями пользователей .....	93
4.7.1. Создание учётных записей .....	94
4.7.2. Назначение пользователям прав доступа .....	96
4.7.3. Настройка парольных политик .....	98
4.7.4. Изменение пароля от учётной записи .....	99
4.7.5. Настройка доступа к объектам .....	100
4.7.6. Блокировка учётной записи .....	101
4.7.7. Удаление учётной записи .....	103
4.8. Работа с объектами и связями .....	104
4.8.1. Создание объекта .....	105
4.8.2. Клонирование объекта .....	106
4.8.3. Удаление объекта .....	107
4.8.4. Создание ссылки на объект .....	108
4.8.5. Создание связи .....	109
4.8.6. Удаление связи .....	110
4.9. Настройка интерфейса .....	111
4.9.1. Выравнивание/расстановка объектов в стандартном виде .....	112
4.9.2. Настройка заголовка web-интерфейса .....	113
4.9.3. Перемещение/отключение фоновой иконки объекта .....	114

4.9.4. Вертикальное отображение имени объекта .....	115
4.9.5. Редактирование стилей состояний .....	116
4.10. Настройка мониторинга .....	117
4.10.1. Мониторинг основных параметров ПК .....	118
4.10.2. Мониторинг процесса памяти .....	119
4.10.3. Мониторинг изменения файлов и папок .....	120
4.10.4. Проверка доступности web-ресурса .....	121
4.10.5. Безагентный мониторинг web-сервера .....	122
4.11. Импорт и экспорт данных .....	123
5. Проблемы в работе системы и способы их решения .....	125
5.1. Недостаточно места на виртуальной машине с сервером .....	126
5.2. Отсутствие подключения агента к серверу .....	128
5.3. Проверка работы MongoDB .....	129
5.4. Проверка работы MySQL .....	130
5.5. Проверка работы Redis .....	131
5.6. 500 Internal Server Error и отсутствие графиков .....	133
5.7. Ошибка работы HTTP-проверки .....	134
Приложения .....	136
Приложение А. Права на операции .....	136



# **Введение**

### **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".

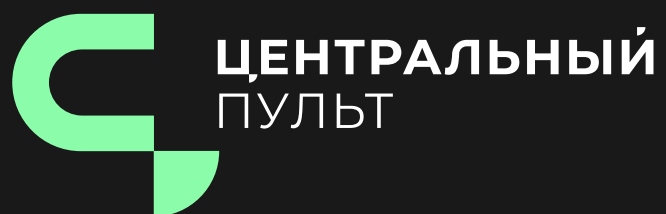
### 1.2 Назначение системы

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.





# **Системные требования**

### **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 сервера со следующими ОС:

- Ubuntu Linux 20.04;
- Ubuntu Linux 18.04.

Под сетевой конфигурацией понимаются открытые порты:

- 80 - HTTP;
- 443 - HTTPS;
- 8091 - web socket;
- 6379 - REDIS Server;
- 1162 - SNMP catcher@SAYMON Agent;
- 22 - SSH;
- 1883/8883 — MQTT/MQTTS.

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](mailto:welcome@saymon.info).

### 2.2 Системные требования к агенту

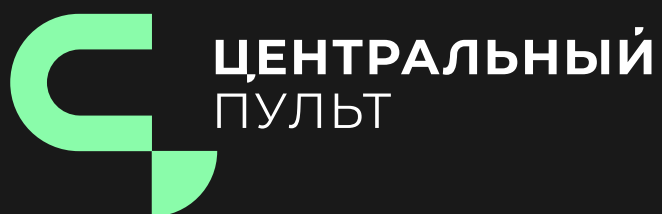
Агенты обладают кросс-платформенной совместимостью и могут быть установлены на различные операционные системы:

- Ubuntu Linux;
- Red Hat Enterprise Linux / CentOS Linux;
- Raspberry Pi;
- Mac OS X;
- 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.



# **Принципы функционирования системы**

### **3 Принципы функционирования системы**

Программное обеспечение платформы "Центральный Пульт" имеет открытые API-интерфейсы, которые обеспечивают информационную совместимость системы и возможность интеграции с другими автоматизированными системами.

### 3.1 Архитектура системы

Платформа имеет клиент-серверную архитектуру и включает в себя три основных уровня (Рис. 3.1.1):

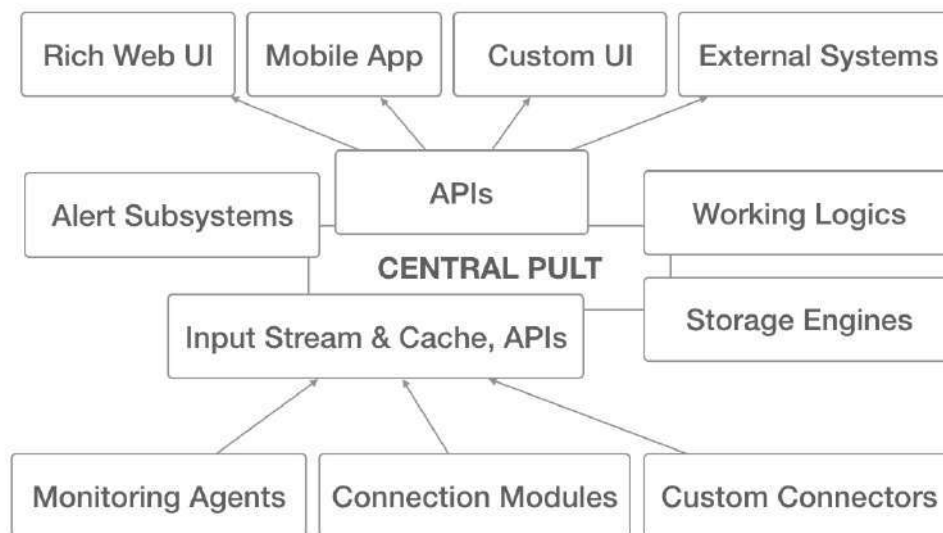


Рис. 3.1.1. Общая архитектура платформы "Центральный Пульс"

Нижний уровень предназначен для сбора данных и осуществления управления над полученной информацией, которая периодически отправляется в Input Stream & Cache для дальнейшей обработки и анализа.

Данный слой включает в себя три элемента:

- Monitoring Agents - агенты системы, собирающие с узлов мониторинга информацию.
- Connection Modules - готовое и переиспользуемое решение для сбора информации без агента.
- Custom Connectors - разработанные для клиента интерфейсы, осуществляющие мониторинг объектов без агента.

На втором уровне хранится и обрабатывается полученная от агентов и прочих интерфейсов информация, а затем передаётся клиентам в требуемом для каждого из решений виде.

Второй слой подразделяется на:

- **Working Logics** (бизнес-логики) - совокупность правил, принципов, зависимостей поведения объектов, на основе которых обрабатываются поступившие с нижнего слоя данные и сохраняются в **Storage Engines** для анализа ситуаций в настоящем и прошлом и построения математически обоснованных прогнозов в будущем.
- **Storage Engines** - отвечает за хранение данных. **Storage Engines** реализовано в виде:
  - **SQL** - формирует запросы, описывающиеся, какую информацию из **Storage Engines** необходимо получить, пути решения определяются автоматически;
  - **noSQL** - обеспечивает гибкость и согласованность системы, благодаря гарантированному завершению запроса;
  - **TimeSeries** - специализированный компонент управления базой данных временных рядов, что позволяет хранить данные с высокой скважностью.
- **Alert Subsystems** - каналы уведомлений, по которым осуществляется информирование пользователей о смене состояний объектов, разрыве соединений и других нестандартных ситуациях. При соответствующей настройке Центральный Пульс может:
  - отправлять email-уведомления,
  - отправлять SMS,
  - отправлять сообщения в Telegram,
  - совершать голосовые вызовы,
  - отображать визуальные уведомления в браузере, сопровождающиеся звуком.
- **APIs** - относятся к категории **REpresentational State Transfer (REST)**, что позволяет выполнять **RESTful**-операции на добавление, чтение, изменение и удаление информации для облачной учётной записи или инсталляции на сервере.
- **Input Stream & Cache** - хранилище данных, поступивших в сервер платформы с нижнего уровня, и их хранение. **Cache** реализован в виде сетевого журналируемого хранилища **Redis**. Взаимодействие между агентами и сервером платформы осуществляется при помощи **Kafka**:
  - собирает данные с ниже располагающегося слоя,
  - хранит данные у себя в распределённом хранилище по топикам,
  - передаёт данные серверу по запросу.



Верхний уровень отвечает за визуализацию полученных и обработанных данных, а также осуществление операций над ними конечным пользователем. В качестве средства интеграции приложений используются открытые API-интерфейсы:

- Rich Web UI - web-интерфейс платформы, является основным средством работы с системой для конечного пользователя, где при наличии определённых прав возможны изменения как всей структура, так и отдельного объекта.
- Mobile App - мобильные приложения для операционных систем Android и iOS.
- Custom UI - кастомизированные интерфейсы, созданные под специальную бизнес-задачу или проект. Функциональные возможности позволяют вносить изменения - добавлять, удалять и редактировать объекты - аналогично Rich Web UI. Web-интерфейс имеет уникальное отображение, согласно заявленным требованиям.
- External Systems - внешние системы для отображения или сбора данных, полученных средствами мониторинга платформы "Центральный Пульс".

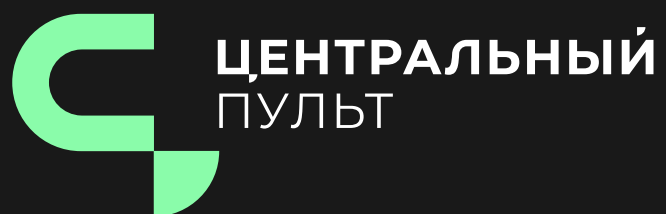
### 3.2 Описание функционирования системы и её частей

С помощью API в web-интерфейсе системы возможно реализовать большинство операций, например:

- получение списка объектов;
- получение текущего статуса объектов и истории их состояний;
- запись данных в объекты без использования агентов;
- изменение свойств объектов;
- работа с инцидентами;
- создание классов с добавлением проверок.

Система состоит из следующих логических подсистем:

- Server - централизованный сервер, на котором хранится и анализируется информация, полученная от агентов, а затем отдаётся Клиенту.
- СУБД (MongoDB, OpenTSDB) - совокупность программных средств, предназначенных для создания, использования и управления базами данных.
- Agent - множество агентов системы, установленных на узлах инфраструктуры и собирающих информацию по ним. Полученные данные периодически отправляются в кэш и затем анализируются сервером.
- Клиент - тонкий web-клиент системы и клиенты для мобильных платформ Android и iOS.



## **Задачи администратора**

### **4 Задачи администратора**

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

### 4.1 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 удаленных команд, выполняемых в случае возникновения проблем.

### 4.1.1 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 в раздел "Загрузки"  
[saymon.info/support\\_ru/downloads\\_en/](http://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.

### 4.1.1.1 Автоинсталляция

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

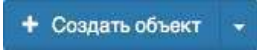

1. Открыть web-интерфейс платформы "Центральный Пульт".
2. Ввести логин и пароль учётной записи с правами на управление объектами.
3. Создать объект-агент, для этого нажать кнопку  на панели инструментов, ввести имя объекта и выбрать класс "Saymon Agent".
4. Навести курсор на созданный объект и нажать на появившуюся иконку  - Настройки агента (Рис. 4.1.1.1.1):



Рис. 4.1.1.1.1. Мониторинговый агент

5. В появившемся окне скопировать ссылку из строки "Команда для установки агента" (Рис. 4.1.1.1.2):

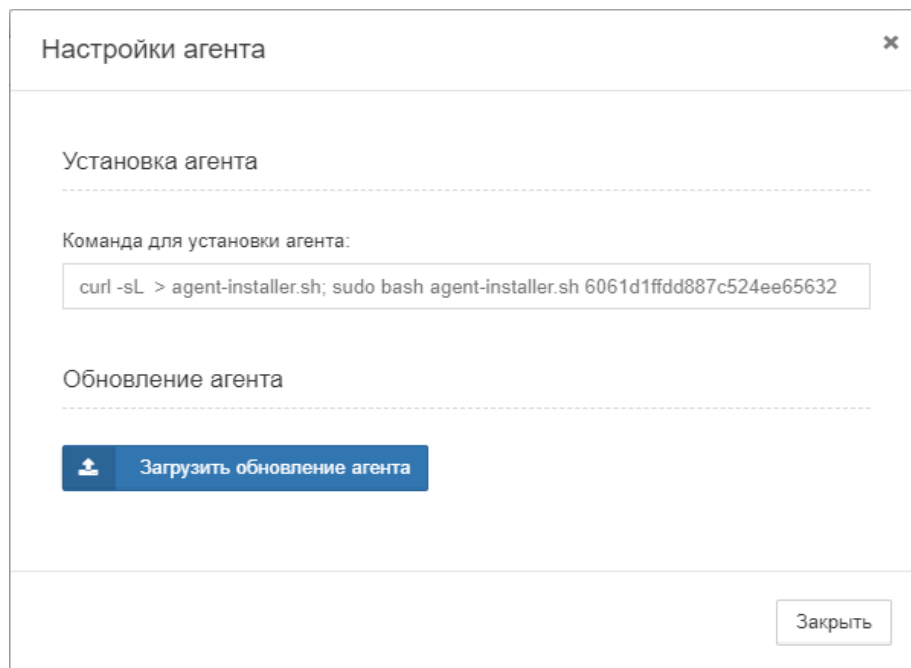
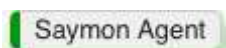


Рис. 4.1.1.1.2. Команда для установки агента

6. Выполнить эту команду в терминале на необходимом сервере.

*Примечание: номер, который дал сервер - последние 24 символа команды - совпадает с ID агента.*

В течение 30 секунд агент скачивается и устанавливается. Если установка выполнена корректно, состояние агента в web-интерфейсе будет отражено зелёным цветом:



### 4.1.1.2 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](mailto:care@saymon.info).

Для установки агента на Linux необходимо:

#### На хосте без доступа в Интернет:

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

##### 64bit:

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

##### 32bit:

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

4. Запустить скрипт:

##### 64bit:

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

##### 32bit:

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

5. Выполнить дальнейшие инструкции в терминале.

#### На хосте с доступом в Интернет:

1. Выполнить однострочник:

##### 64bit:

```
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
```

##### 32bit:

```
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 выполняется в файле  
"/opt/saymon-agent/conf/agent.properties".



### 4.1.1.3 Mac OS X

Для установки агента на Mac OS X необходимо:

1. Создать системного пользователя "Saymon agent", из-под которого будет запускаться агент.
2. [Скачать архив Mac OS X 64bit](#) и распаковать его в папку **"/opt"**.
3. Отредактировать файл конфигурации 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. При необходимости включить централизованное логирование агента через SYSLOG в файле **"saymon-agent/conf/logback.xml"**.
6. Сделать файл агента исполняемым:

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

7. Запустить агента:

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

### 4.1.14 Windows

Для установки агента на Windows необходимо:

1. [Скачать инсталлятор](#) для Windows 64bit или Windows 32bit.
2. Запустить скачанный файл от имени администратора.
3. Выбрать папку установки (например, "**C:\Program Files (x86)\SAYMON Agent**") и нажать кнопку "Install".
4. После завершения установки нажать кнопку "Close".
5. В папке установки агента из шага 3 отредактировать файл конфигурации агента "**...\conf\agent.properties**".
6. Запустить службу "SAYMON Agent".

### 4.1.15 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 регистрации 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](#).

### 4.1.2 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 (часть 1 из 2):

Параметр	Описание
<code>agent.discoveryEnabled</code>	<p>Включает автоматический поиск агентом сетевых устройств;</p> <ul style="list-style-type: none"><li>• <code>false</code> - автоматический поиск выключен,</li><li>• <code>true</code> - автоматический поиск включен.</li></ul> <p>По умолчанию - <b>false</b> - автоматический поиск выключен.</p> <p><i>Задать родительский объект для обнаруженных устройств можно с помощью параметра "<code>discovery_parent_id</code>" в разделе "<code>Server</code>" конфигурационного файла сервера.</i></p>
<code>agent.id</code>	<p>Уникальный идентификатор объекта класса "<code>Saymon Agent</code>" в web-интерфейсе системы.</p> <p><i>При указании <code>agent.id=0</code> на сервере будет создан новый объект класса "<code>Saymon Agent</code>", и ID этого объекта будет передан агенту. На хосте с агентом будет создан файл "<code>.../saymon-agent/storage/registration.json</code>", где будет записан данный идентификатор.</i></p> <p><i>Чтобы заново инициализировать процедуру получения ID, к примеру, если необходимо подключить агента к другому серверу, достаточно удалить файл "<code>registration.json</code>" и перезапустить агента.</i></p>
<code>agent.optimizedThreadManagement</code>	<p>Включает режим оптимизации использования потоков агентом;</p> <ul style="list-style-type: none"><li>• <code>false</code> - режим оптимизации выключен,</li><li>• <code>true</code> - режим оптимизации включен.</li></ul> <p>По умолчанию - <b>false</b> - режим оптимизации выключен.</p>
<code>agent.scriptsEnabled</code>	<p>Включает выполнение агентом скриптов с указанным текстом;</p> <ul style="list-style-type: none"><li>• <code>true</code> - выполнение скриптов включено.</li><li>• <code>false</code> - выполнение скриптов выключено.</li></ul> <p>По умолчанию - <b>true</b> - выполнение скриптов включено.</p>

Общие параметры конфигурационного файла агента (часть 2 из 2):

Параметр	Описание
agent.snmpTrapEnabled	Позволяет использовать агента в качестве получателя трапов; <ul style="list-style-type: none"><li>• false - получение трапов агентом выключено,</li><li>• true - получение трапов агентом включено.</li></ul> По умолчанию - <b>false</b> - получение трапов выключено.
agent.snmpTrapListenPort	Порт для получения трапов. По умолчанию - <b>1162</b> .
agent.snmpTrapReceiverThreadPoolSize	Количество одновременных потоков для получения трапов. По умолчанию - <b>4</b> .
server.host	Адрес сервера системы. По умолчанию - <b>127.0.0.1</b> .
server.password	Пароль Redis.
server.port	Порт Kafka (9092) или Redis (6379), по которому осуществляется подключение агентов к серверу.

### 4.1.2.1 Подключение агента 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.

*Примечание: При использовании Kafka Cluster поле **server.host** не должно быть пустым, так как используется для обновления агента.*

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.
kafka.security.protocol	Протокол, используемый для взаимодействия с брокерами. Возможные значения: <ul style="list-style-type: none"><li>• PLAINTEXT,</li><li>• SSL,</li><li>• SASL_PLAINTEXT,</li><li>• SASL_SSL.</li></ul>
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.

### 4.1.3 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
```

Параметры конфигурационного файла сервера (часть 1 из 16):

Раздел/Параметр	Описание
Cache	В этом разделе задаются настройки сервера Redis. <pre>"cache": {   "host": "127.0.0.1",   "port": 6379 }</pre>
cache.auth_pass	Пароль для доступа к серверу Redis.
cache.host	Адрес сервера Redis. По умолчанию - <b>"127.0.0.1"</b> .
cache.port	Порт сервера Redis. По умолчанию - <b>6379</b> .
cache.cluster	В этом подразделе задаются настройки при использовании Redis Cluster. <i>Если задействован этот подраздел, система использует указанные здесь настройки подключения, параметры "cache.host" и "cache.port" игнорируются.</i> <pre>"cache": {   "cluster": {     "nodes": [       {         "host": "127.0.0.1",         "port": 7000       },       {         "host": "127.0.0.1",         "port": 7001       }     ],     "options": {}   } }</pre>
cache.cluster.nodes	Серверы кластера, к которому необходимо подключиться.

Параметры конфигурационного файла сервера (часть 2 из 16):

Раздел/Параметр	Описание
cache.cluster.nodes.host	Адрес сервера кластера.
cache.cluster.nodes.port	Порт сервера кластера.
cache.cluster.options	Дополнительные параметры подключения к кластеру.
Kafka	В этом разделе задаются настройки подключения к брокеру Kafka.
	<pre>"kafka": {   "host": "localhost",   "port": 9092,   "requestTimeout": 30000,   "fetch_latest": false,   "logLevel": 2 }</pre> <p>Помимо перечисленных здесь параметров возможно также использовать <a href="#">дополнительные параметры подключения</a>.</p>
kafka.brokers	Пул адресов для подключения при использовании Kafka Cluster.
	<pre>"kafka": {   ...   "brokers": [     "192.168.1.10:9092",     "192.168.1.11:9092"   ],   ... }</pre> <p><i>При указании данного параметра система игнорирует параметры <b>kafka.host</b> и <b>kafka.port</b>.</i></p> <p><i>При использовании Kafka Cluster необходимо дополнительно настроить конфигурационный файл агента. Подробнее в разделе 4.1.2.1 на стр. 29.</i></p>
kafka.fetch_latest	<p>Данный параметр определяет правило обработки сообщений, поступивших в Kafka при выключенной системе:</p> <ul style="list-style-type: none"> <li>• false - обработать накопленные данные,</li> <li>• true - игнорировать накопленные данные.</li> </ul> <p>По умолчанию - <b>false</b>.</p>



Параметры конфигурационного файла сервера (часть 3 из 16):

Раздел/Параметр	Описание
kafka.host	Адрес брокера Kafka. По умолчанию - <b>"localhost"</b> .
kafka.logLevel	Уровень логирования Kafka; <ul style="list-style-type: none"><li>• 0 - NOTHING,</li><li>• 1 - ERROR,</li><li>• 2 - WARN,</li><li>• 4 - INFO,</li><li>• 5 - DEBUG.</li></ul> По умолчанию - <b>2 - WARN</b> .
kafka.port	Порт брокера Kafka. По умолчанию - <b>9092</b> .
kafka.requestTimeout	Время ожидания ответа клиентом (в миллисекундах). По умолчанию - <b>30000</b> - 30 секунд.
kafka.sasl	<b>Настройки аутентификации.</b> Параметры зависят от механизма аутентификации, используемом в Kafka. Пример настроек для механизма PLAIN/SCRAM: <pre>"kafka": {   ...   "sasl": {     "mechanism": "plain",     "username": "user",     "password": "password"   },   ... }</pre>
kafka.sasl.mechanism	Механизм аутентификации.
kafka.ssl	<b>Настройки шифрования.</b> Ключ и сертификат можно указывать как путь к файлу: <pre>"kafka": {   ...   "ssl": {     "certFile": "/opt/kafka/for-saymon-server/cert.pem",     "keyFile": "/opt/kafka/for-saymon-server/key.pem"   },   ... }</pre>

Параметры конфигурационного файла сервера (часть 4 из 16):

Раздел/Параметр	Описание
LDAP	<p>В этом разделе задаются параметры внешнего LDAP-сервера для авторизации.</p> <pre>"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 }</pre>
ldap.allow_self_signed	<p>Разрешает соединение по LDAPS с сервером, на котором установлен самоподписанный SSL-сертификат;</p> <ul style="list-style-type: none"><li>• false - соединение запрещено,</li><li>• true - соединение разрешено.</li></ul> <p>По умолчанию - <b>false</b> - соединение запрещено.</p>
ldap.login	Логин администратора LDAP.
ldap.pass	Пароль администратора LDAP.
ldap.suffix	Корневой элемент (как правило, доменное имя организации).
ldap.url	Адрес LDAP-сервера. Поддерживаются протоколы LDAP и LDAPS.
MQTT	<p>В этом разделе задаются настройки подключения к MQTT-брокеру.</p> <pre>"mqtt": {   "broker": "mqtt://username:password@localhost:1883" }</pre>
mqtt.broker	<p>Адрес и порт брокера.</p> <p>По умолчанию - <b>"mqtt://localhost:1883"</b>.</p> <p><i>Для аутентификации по имени пользователя и паролю нужно указать пользовательские данные перед адресом сервера.</i></p>

Параметры конфигурационного файла сервера (часть 5 из 16):

Раздел/Параметр	Описание
<b>OpenTSDB</b>	В этом разделе задаются параметры доступа к OpenTSDB.
openTsdB.enabled	Запись исторических данных в OpenTSDB; <ul style="list-style-type: none"><li>• false - запись выключена,</li><li>• true - запись включена.</li></ul> По умолчанию - <b>true</b> - запись включена.
openTsdB.host	Адрес хоста с OpenTSDB. По умолчанию - <b>"localhost"</b> .
openTsdB.port	Порт OpenTSDB. По умолчанию - <b>4242</b> .
<b>Push_notification</b>	В этом разделе задаются параметры push-уведомлений в мобильном приложении. Для работы с ними используется Firebase Cloud Messaging (FCM).
push_notification.disabled	Выключает уведомления; <ul style="list-style-type: none"><li>• true - уведомления выключены,</li><li>• false - уведомления включены.</li></ul> По умолчанию - <b>true</b> - уведомления выключены.
push_notification.key_path	Путь к ключу авторизации сервера Центрального Пульта на сервере Firebase. По умолчанию - <b>"/etc/saymon/saymon-mobile-firebase-adminsdk.json"</b> .
push_notification.on_state_change	Включает отправку уведомления при изменении состояний на случай другого источника уведомлений - MQTT-сообщений. <ul style="list-style-type: none"><li>• true - отправка включена,</li><li>• false - отправка выключена.</li></ul> По умолчанию - <b>true</b> - отправка включена.
push_notification.timeout	Время, через которое каждому пользователю отправляется уведомление (в миллисекундах). По умолчанию - <b>0</b> - задержки нет.
push_notification.url	URL, полученный пользователем от Firebase, для принятия содержимого уведомлений с сервера. По умолчанию - <b>"https://saymon-mobile.firebaseio.com"</b> .

Параметры конфигурационного файла сервера (часть 6 из 16):

Раздел/Параметр	Описание
Resource_server	В этом разделе задаются параметры, связанные с хранением файлов, загруженных в Центральный Пульт.
resource.server.debug	Включает debug-режим для логирования в файл <b>"/var/log/saymon/saymon-server.log"</b> ; <ul style="list-style-type: none"><li>• false - режим выключен,</li><li>• true - режим включен.</li></ul> По умолчанию - <b>false</b> - режим выключен.
resource.server.file_storage_dir	Путь к директории для хранения документов, прикрепляемых к объектам. По умолчанию - <b>"/var/saymon/resources"</b> .
resource.server.ip_address	Адрес Resource-сервера. По умолчанию - <b>"127.0.0.1"</b> .
resource.server.port	Порт Resource-сервера. По умолчанию - <b>8092</b> .
Rest_server	В этом разделе задаются параметры REST-сервера.
rest.server.base_url	Путь к API. По умолчанию - <b>"/api"</b> .
rest.server.colorize_log	Включает цветную раскраску лога; <ul style="list-style-type: none"><li>• false - раскраска выключена,</li><li>• true - раскраска включена.</li></ul> По умолчанию - <b>false</b> - раскраска выключена.
rest.server.debug	Включает debug-режим для логирования в файл <b>"/var/log/saymon/saymon-server.log"</b> ; <ul style="list-style-type: none"><li>• false - режим выключен,</li><li>• true - режим включен.</li></ul> По умолчанию - <b>false</b> - режим выключен.
rest.server.document_download_url	URL к файлам, сохраненным в <b>\$document_storage_dir</b> . По умолчанию - <b>"http://localhost/node/api/docs"</b> .
rest.server.ip_address	Адрес хоста для запуска REST-сервера. По умолчанию - <b>"127.0.0.1"</b> .

Параметры конфигурационного файла сервера (часть 7 из 16):

Раздел/Параметр	Описание
rest.server.num_workers	Число процессов для загрузки данных. По умолчанию - <b>1</b> .
rest.server.port	Порт REST-сервера. По умолчанию - <b>8090</b> .
rest.server.public_url	Адрес для доступа к web-интерфейсу из уведомлений.
rest.server.update_download_url	Путь к файлу для обновления агента. По умолчанию - <b>"http://localhost/node/api/agents/update"</b> .
rest_server.limits	<p>В этом подразделе задаются ограничения на число запросов к API.</p> <pre>"limits" : [   {     "__comment": "администратора не ограничиваем",     "user": "5048c849d7b6e40593dfce71"   },   {     "__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"   } ]</pre> <p>По умолчанию ограничения не накладываются.</p> <p>Для каждого запроса поиск правил осуществляется сверху вниз. Применяется только одно правило.</p>

Параметры конфигурационного файла сервера (часть 8 из 16):

Раздел/Параметр	Описание
rest.server.limits.ip	<p>IP-адреса, для которых устанавливается ограничение;</p> <ul style="list-style-type: none"><li>• true - свой лимит для каждого IP-адреса;</li><li>• string string[] - IP-адрес или список адресов, к которым применяется правило.</li></ul> <p>Примеры:</p> <pre>"limits": [   {     "ip": true,     ...   } ]</pre> <pre>"limits": [   {     "ip": "127.0.0.1",     ...   } ]</pre> <pre>"limits": [   {     "ip": [       "127.0.0.1",       "192.168.1.2"     ],     ...   } ]</pre>
rest.server.limits.rate	<p>Количество запросов в единицу времени. Если не указан, то ограничение не накладывается.</p> <p>Задаётся в формате <b>количество_запросов/единица_измерения_времени</b>.</p> <p>Возможные единицы времени:</p> <ul style="list-style-type: none"><li>• у - год,</li><li>• Q - квартал,</li><li>• М - месяц,</li><li>• w - неделя,</li><li>• d - день,</li><li>• h - час,</li><li>• m/minute - минута,</li><li>• s - секунда,</li><li>• ms - миллисекунда.</li></ul> <p>Пример:</p> <pre>"limits": [   {     "rate": 100/minute,     ...   } ]</pre>

Параметры конфигурационного файла сервера (часть 9 из 16):

Раздел/Параметр	Описание
rest.server.limits.uri	<p>REST-методы, к которым устанавливается ограничение;</p> <ul style="list-style-type: none"><li>• true - свой лимит для каждого метода;</li><li>• string string[] - метод или список методов, к которым применяется правило.</li></ul> <p>Примеры:</p> <pre>"limits" : [   {     "uri": true,     ...   } ]</pre> <pre>"limits" : [   {     "uri": "/ping",     ...   } ]</pre> <pre>"limits" : [   {     "uri": [       "/ping",       "/node/api/objects/:id/stat"     ],     ...   } ]</pre>

Параметры конфигурационного файла сервера (часть 10 из 16):

Раздел/Параметр	Описание
rest.server.limits.user	<p>Пользователи, для которых устанавливается ограничение;</p> <ul style="list-style-type: none"><li>• true - свой лимит для каждого авторизованного пользователя;</li><li>• false - лимит по IP-адресу для каждого авторизованного пользователя;</li><li>• string string[] - ID пользователя или пользователей, к которым применяется правило.</li></ul> <p>Примеры:</p> <pre>"limits": [   {     "user": true,     ...   } ]</pre> <pre>"limits": [   {     "user": false,     ...   } ]</pre> <pre>"limits": [   {     "user": "6048c849d7b6e40593dfce71",     ...   } ]</pre> <pre>"limits": [   {     "user": [       "57ff6853fa6db3a63d16d07b",       "6048c849d7b6e40593dfce71"     ],     ...   } ]</pre>



Параметры конфигурационного файла сервера (часть 11 из 16):

Раздел/Параметр	Описание
Server	В этом разделе задаются общие параметры сервера.
server.analytics_enabled	Включает аналитику значений метрик в процесс обработки данных; <ul style="list-style-type: none"><li>• false - аналитика выключена,</li><li>• true - аналитика включена.</li></ul> По умолчанию - <b>false</b> - аналитика выключена.
server.analytics_processes	Количество логических ядер, выделяемых для обработки аналитики временных рядов. По умолчанию используются все доступные ядра.
server.colorize_log	Включает цветную раскраску лога; <ul style="list-style-type: none"><li>• false - раскраска выключена,</li><li>• true - раскраска включена.</li></ul> По умолчанию - <b>false</b> - раскраска выключена.
server.comet_ping_interval	Временной интервал между отправками comet-сервером сообщений (в миллисекундах). По умолчанию - <b>5000</b> - 5 секунд.
server.comet_ping_timeout	Время ожидания сообщения от comet-сервера (в миллисекундах). По умолчанию - <b>12000</b> - 12 секунд.
server.comet_port	Порт для соединения. По умолчанию - <b>8091</b> .
server.comet_secure	Включает SSL-соединение; <ul style="list-style-type: none"><li>• false - соединение выключено,</li><li>• true - соединение включено.</li></ul> По умолчанию - <b>false</b> - соединение выключено.
server.comet_ssl_certificate	Путь к сертификату.
server.comet_ssl_key	Путь к ключу.
server.conditional_incidents_enabled	Включает функционал генерации инцидентов; <ul style="list-style-type: none"><li>• false - функционал выключен,</li><li>• true - функционал включен.</li></ul> По умолчанию - <b>false</b> - функционал выключен.

Параметры конфигурационного файла сервера (часть 12 из 16):

Раздел/Параметр	Описание
server.debug	Включает debug-режим для логирования в файл <b>"/var/log/saymon/saymon-server.log"</b> ; <ul style="list-style-type: none"><li>• false - режим выключен,</li><li>• true - режим включен.</li></ul> По умолчанию - <b>false</b> - режим выключен.
server.default_result_timeout	Время, через которое срабатывает условие "Нет данных от объекта" с момента создания объекта или получения последних данных (в миллисекундах). По умолчанию - <b>120000</b> - 2 минуты.
server.default_state_id	Состояние объекта по умолчанию. По умолчанию - <b>7</b> - "Нет данных".
server.discovery_parent_id	ID объекта, в котором появляются найденные агентами сетевые устройства. По умолчанию - <b>"1"</b> .
server.event_log_max_bytes	Размер записей консоли в mongoDB, при достижении которого происходит ротация данных (в байтах). По умолчанию - <b>"1 G"</b> - 1 гигабайт.
server.extension_path	Путь к директории с серверными расширениями.
server.history_temporary_storage_period	Интервал времени для буферизации метрик, по истечении которого все данные из Redis записываются в OpenTSDB (в миллисекундах). Применяется, только если параметр "history_update_period" равен 0.
server.history_update_period	Интервал записи исторических данных (в миллисекундах). 0 - немедленная запись пришедших значений. По умолчанию - <b>60000</b> - 1 минута.
server.notification_buffering_period	Период ожидания для сбора сообщений о смене состояний объектов и отправки группового уведомления (в миллисекундах). По умолчанию - <b>0</b> - буферизация отключена.

Параметры конфигурационного файла сервера (часть 13 из 16):

Раздел/Параметр	Описание
server.retain_expired_stat	Включает хранение последних полученных данных после их устаревания; <ul style="list-style-type: none"><li>• false - хранение выключено,</li><li>• true - хранение включено.</li></ul> По умолчанию - <b>false</b> - хранение выключено.
server.script_trigger_timeout	Максимальное время выполнения триггера (в миллисекундах). По умолчанию - <b>30000</b> - 30 секунд.
server.self_object_id	ID объекта, используемого для самомониторинга.
server.sms_script	Путь до скрипта, отправляющего sms-уведомления.
server.sql_history_enabled	Включает запись исторических данных в MySQL; <ul style="list-style-type: none"><li>• false - запись выключена,</li><li>• true - запись включена.</li></ul> По умолчанию - <b>false</b> - запись выключена.
server.stat_local_timestamp_field_name	Имя поля, где передается время, с которым нужно сохранять данные в OpenTSDB. По умолчанию - <b>"localTimestamp"</b> .
server.stat_scan_period	Период проверки актуальности пришедших данных (в миллисекундах). По умолчанию - <b>3000</b> - 3 секунды.
server.voice_call_script	Путь до скрипта, осуществляющего голосовой вызов.

Параметры конфигурационного файла сервера (часть 14 из 16):

Раздел/Параметр	Описание
server.email	<p>В этом подразделе задаются параметры доступа к почтовому серверу:</p> <pre>"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   }, }</pre>
server.email.disabled	<p>Выключает отправку почтовых уведомлений;</p> <ul style="list-style-type: none"><li>• true - отправка выключена,</li><li>• false - отправка включена.</li></ul> <p>По умолчанию - <b>true</b> - отправка выключена.</p>
server.email.fields	Данные об отправителе уведомлений.
server.email.fields.from	Почтовый адрес отправителя.
server.email.max_json_length	<p>Ограничение размера письма с уведомлением (в символах).</p> <p>По умолчанию - <b>1000</b> - 1000 символов.</p>
server.email.transport	Данные почтового сервера.
server.email.transport.auth	Данные для аутентификации пользователя.
server.email.transport.auth.pass	Пароль пользователя.
server.email.transport.auth.user	Логин пользователя.
server.email.transport.host	Адрес почтового сервера.
server.email.transport.port	Порт почтового сервера.

Параметры конфигурационного файла сервера (часть 15 из 16):

Раздел/Параметр	Описание
server.email.transport.secure	Включает использование TLS при подключении к серверу; <ul style="list-style-type: none"><li>• false - использование TLS выключено,</li><li>• true - использование TLS включено.</li></ul> Значение по умолчанию зависит от порта.
server.email.transport.service	Встроенный в коннектор набор служб. При наличии задаёт "host", "port", "secure" автоматически. По умолчанию - <b>"Gmail"</b> .
server.user	В этом подразделе задаются параметры пользователей: <pre>"user": {   "auth_enabled": "true",   "new_user_access": "all",   "lang_default": "ru",   "template": {     "permissions": ["manage-objects", "view-section-stat"],     "objectPermissions": {       "include": [],       "exclude": ["5fb643ddf277b96c8401119b"]     }   },   "usersRoot": "5800d9aaac7bf0f90d3d520e" }</pre>
server.user.auth_enabled	Включает самостоятельную регистрацию для пользователей; <ul style="list-style-type: none"><li>• false - регистрация выключена,</li><li>• true - регистрация включена.</li></ul> По умолчанию - <b>false</b> - регистрация выключена.
server.user.lang_default	Язык пользователей по умолчанию; <ul style="list-style-type: none"><li>• "en" - английский,</li><li>• "it" - итальянский,</li><li>• "ru" - русский.</li></ul> По умолчанию - <b>"en"</b> - английский.
server.user.new_user_access	Права доступа к объектам для нового пользователя; <ul style="list-style-type: none"><li>• "all" - есть доступ ко всем объектам,</li><li>• "not" - нет доступа ни к одному объекту.</li></ul> По умолчанию - <b>"all"</b> - доступ ко всем объектам.

Параметры конфигурационного файла сервера (часть 16 из 16):

Раздел/Параметр	Описание
server.user.template	Шаблон прав нового пользователя. <i>Данные параметры применяются только для пользователей, регистрирующихся самостоятельно.</i>
server.user.template.objectPermissions	Права пользователя на доступ к объектам.
server.user.template.objectPermissions.include	Список идентификаторов объектов, к которым пользователю по умолчанию доступ разрешён.
server.user.template.objectPermissions.exclude	Список идентификаторов объектов, к которым пользователю по умолчанию доступ запрещён.
server.user.template.permissions	Список прав на операции, доступных пользователю по умолчанию. <i>Список операций с подробным описанием содержится в Приложении А на стр. 136.</i>
server.user.usersRoot	Идентификатор корневого объекта для создаваемых новым пользователем объектов. По умолчанию - "1".

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 раздела "Server" (Рис. 4.1.3.1):

Общие настройки

Сохранять устаревшие данные от агента ☐ НЕТ ☒

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. 4.1.3.1. Общие настройки

2. "Настройки электронной почты" соответствуют параметрам подраздела "Email": (Рис. 4.1.3.2):

Настройки электронной почты

Выключено ☐ НЕТ ☒

Сервис транспорта

Имя пользователя

Пароль

Адрес отправителя

Адрес получателя

Максимальная длина JSON

Рис. 4.1.3.2. Настройки электронной почты

3. "Настройки пользователей" соответствуют параметрам подраздела "User" (Рис. 4.1.3.3):

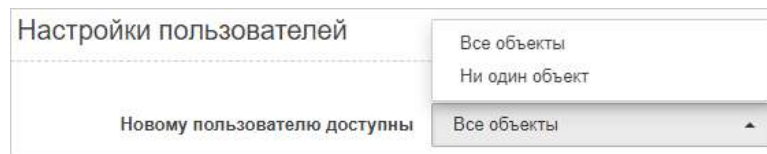


Рис. 4.1.3.3. Настройки пользователей

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
```



### 4.1.4 Настройка элементов web-интерфейса

Некоторые элементы web-интерфейса можно настроить в конфигурационном файле клиента:

**`"/usr/local/saymon/client/client-config.js"`**

В некоторых случаях файл "client-config.js" может находиться в другой директории, путь к которой может быть найден в конфигурационном файле nginx или командой:

```
sudo find / -name "client-config.js"
```

Параметры конфигурационного файла клиента (часть 1 из 8):

Раздел/Параметр	Описание
agentInstallerUrl	Ссылка для скачивания установочного скрипта для агента.
authEnabled	Включает самостоятельную регистрацию для пользователей; <ul style="list-style-type: none"><li>• false - регистрация выключена,</li><li>• true - регистрация включена.</li></ul> По умолчанию - <b>false</b> - регистрация выключена.
canEditObjectStyle	Включает возможность вручную настраивать CSS-стили для объектов и связей. При включении опции в разделе "Параметры" появляется подраздел "Стили". <ul style="list-style-type: none"><li>• false - настройка стилей выключена,</li><li>• true - настройка стилей включена.</li></ul> По умолчанию - <b>false</b> - настройка стилей включена.
collapseSections	Задаёт способ отображения графиков; <ul style="list-style-type: none"><li>• false - графики раскрыты,</li><li>• true - графики скрыты.</li></ul> По умолчанию - <b>false</b> - графики раскрыты.

Параметры конфигурационного файла клиента (часть 2 из 8):

Раздел/Параметр	Описание
Comet	<p>Наряду со стандартным HTTP, Центральный Пульт использует Comet для обмена данными между клиентом и сервером.</p> <pre>comet: {   port: 8091 }, pollInterval: 5000, pollIntervalSocket: 60000, ...</pre>
comet.port	<p>Номер порта, который слушает Comet-сервер.</p> <p>По умолчанию - <b>8091</b>.</p>
DefaultClassImage	<p>Позволяет задать стандартное изображение и его размер для объектов определенного класса.</p> <pre>defaultClassImage: {   '29': {img: '/images/cloud.svg', dim: [100, 75]} }, ...</pre>
defaultClassImage. '<number>'	Идентификатор класса, например, 29.
defaultClassImage. '<number>'.img	Путь до изображения относительно папки, где находится конфигурационный файл.
defaultClassImage. '<number>'.dim	Размер изображения (в пикселях, [ ширина, высота]).
Default EmbeddableLink	<p>Позволяет встраивать виджеты с информацией об объектах на сторонние ресурсы.</p> <pre>defaultEmbeddableLink: {   url: "http://saymon.info/",   name: "SAYMON" }, ...</pre>
defaultEmbeddableLink. name	Имя виджета.
defaultEmbeddableLink. url	URL, на который ведёт виджет.

Параметры конфигурационного файла клиента (часть 3 из 8):

Раздел/Параметр	Описание
defaultPieChartColors	<p>Цвета секторов по умолчанию для круговой диаграммы.</p> <pre>defaultPieChartColors: [   '#a90329',   '#3276b1',   '#9868B8',   '#F2C001' ], ...</pre>
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"> <li>• audit-log (История изменений),</li> <li>• config-log (Изменения конфигурации),</li> <li>• documents (Документы),</li> <li>• entity-incident-conditions (Условия генерации аварий),</li> <li>• entity-settings (Параметры),</li> <li>• entity-state-conditions (Условия перехода состояний),</li> <li>• history-graph (Графики),</li> <li>• monitoring (Мониторинг),</li> <li>• operations (Операции),</li> <li>• properties (Свойства),</li> <li>• stat (Данные),</li> <li>• stat-rules (Правила формирования данных),</li> <li>• state-history (История состояний),</li> <li>• state-triggers (Действия при смене состояний),</li> <li>• widgets (Виджеты).</li> </ul>
Documentation	<p>В этом разделе можно настроить ссылки на различную документацию для Центрального Пульта.</p> <pre>documentation: {   main: {     root: 'https://wiki.saymon.info/display/SAYMONEN',     ru: 'https://wiki.saymon.info'   },   notificationTemplates: {     root: 'https://wiki.saymon.info/display/SAYMONEN/Notification+templates',     ru: 'https://wiki.saymon.info/pages/viewpage.action?pageId=53941286'   } }, ...</pre>
documentation.main	Ссылки на пользовательскую документацию.
documentation.main.root	Ссылка на англоязычную документацию.
documentation.main.ru	Ссылка на русскоязычную документацию.

Параметры конфигурационного файла клиента (часть 4 из 8):

Раздел/Параметр	Описание
documentation. notificationTemplates	Ссылки на шаблоны оповещений.
documentation. notificationTemplates.root	Ссылка на англоязычные шаблоны оповещений.
documentation. notificationTemplates.ru	Ссылка на русскоязычные шаблоны оповещений.
enableConditionalIncidents	Включает функционал генерации инцидентов; <ul style="list-style-type: none"><li>• false - функционал выключен,</li><li>• true - функционал включен.</li></ul> По умолчанию - <b>false</b> - функционал выключен.
enableSmsTrigger	Включает использование SMS-уведомлений; <ul style="list-style-type: none"><li>• false - SMS-уведомления выключены,</li><li>• true - SMS-уведомления включены.</li></ul> По умолчанию - <b>false</b> - SMS-уведомления выключены.
enableVoiceCallTrigger	Включает использование голосовых вызовов; <ul style="list-style-type: none"><li>• false - голосовые вызовы выключены,</li><li>• true - голосовые вызовы включены.</li></ul> По умолчанию - <b>false</b> - голосовые вызовы выключены.
ForceEmptyStandardView	При выборе объекта, в том числе не содержащего вложенных объектов, Центральный Пульт по умолчанию показывает для него подробную информацию.  В параметре задаётся список идентификаторов классов, при выборе пустых объектов которых Центральный Пульт переходит в стандартный вид.  По умолчанию: <ul style="list-style-type: none"><li>• 1 (Root),</li><li>• 3 (Host),</li><li>• 13 (Node),</li><li>• 43 (Dashboard).</li></ul>






Параметры конфигурационного файла клиента (часть 5 из 8):

Раздел/Параметр	Описание
GeoMap	<p>Центральный Пульт позволяет привязывать объекты к их геоположению и просматривать их на карте.</p> <pre>geoMap: {   serverName: openstreetmap.org,   initialPosition: [30.341306, 59.930089],   initialZoom: 8 }, ...</pre>
geoMap.initialPosition	<p>Начальная позиция на карте в формате [долгота, широта]. По умолчанию - Россия, Санкт-Петербург.</p>
geoMap.initialZoom	<p>Начальный масштаб карты. По умолчанию - 8.</p>
geoMap.serverName	<p>Имя сервера, отвечающего за отображения карты. По умолчанию - <a href="https://openstreetmap.org">openstreetmap.org</a>. Также можно использовать локальный OSM-сервер.</p>
Grid	<p>В разделе задаются настройки сетки для выравнивания объектов в стандартном виде.</p> <pre>grid: {   dim: 20,   color: "rgba(128, 128, 128, 0.3)",   border: 4 }, ...</pre>
grid.border	<p>Максимальное значение отступа границы объекта от границы сетки (в пикселях). По умолчанию - 8.</p>
grid.color	<p>Цвет сетки в формате RGBA. По умолчанию - "rgba(128, 128, 128, 0.3)".</p>
grid.dim	<p>Размер сетки (в пикселях). По умолчанию - 20.</p>

Параметры конфигурационного файла клиента (часть 6 из 8):

Раздел/Параметр	Описание
GridView	<p>В разделе задаются параметры отображения объектов в виде сетки.</p> <pre> gridView: {   cols: 6,   colWidth: 240,   rowHeight: 90,   gutterSize: 5 }, ... </pre>
gridView.cols	<p>Количество объектов в строке сетки.</p> <p>При gridView.cols &gt; 0 ширина объектов меняется автоматически.</p> <p>При gridView.cols = 0 число объектов в строке сетки подстраивается под ширину экрана.</p> <p>По умолчанию - 6.</p>
gridView.colWidth	<p>Ширина объектов при gridView.cols = 0 (в пикселях).</p> <p>По умолчанию - 240.</p>
gridView.gutterSize	<p>Величина отступа между объектами (в пикселях).</p> <p>По умолчанию - 5.</p>
gridView.rowHeight	<p>Высота объектов (в пикселях).</p> <p>По умолчанию - 90.</p>
hideConnectionStatusNotifications	<p>Скрывает оповещения о текущем статусе соединения;</p> <ul style="list-style-type: none"> <li>• false - оповещения не скрыты,</li> <li>• true - оповещения скрыты.</li> </ul> <p>По умолчанию - <b>false</b> - оповещения не скрыты.</p>
History	<p>В этом разделе задаются параметры, отвечающие за отображение данных на графиках, а также активируются некоторые дополнительные элементы интерфейса.</p> <pre> history: {   slider: false,    [[1, 'hours'], '15s-avg'],   [[1, 'days'], '5m-avg'],   [[1, 'weeks'], '1h-avg'],   ['1d-avg'] ],   modelChangeAnnotations: false }, ... </pre>

Параметры конфигурационного файла клиента (часть 7 из 8):

Раздел/Параметр	Описание
history.downsampling	Параметры уменьшения размера выборки данных - downsampling.
history.modelChangeAnnotations	<p>Включает смену аннотаций на графиках;</p> <ul style="list-style-type: none"> <li>• false - смена аннотаций выключена,</li> <li>• true - смена аннотаций включена.</li> </ul> <p>По умолчанию - <b>false</b> - смена аннотаций выключена.</p>
history.slider	<p>Включает ползунок, который используется для задания интервала времени отображения данных;</p> <ul style="list-style-type: none"> <li>• false - ползунок выключен,</li> <li>• true - ползунок включен.</li> </ul> <p>По умолчанию - <b>false</b> - ползунок выключен.</p>
Languages	<p>Раздел позволяет управлять предустановленными языками web-интерфейса.</p> <pre>languages: {   en: { short: 'us', long: 'English' },   ru: { short: 'ru', long: 'Русский' },   it: { short: 'it', long: 'Italiano' } }, ...</pre>
languages.<id>	<p>Идентификатор языка. Доступны три языка:</p> <ul style="list-style-type: none"> <li>• en - английский,</li> <li>• it - итальянский,</li> <li>• ru - русский.</li> </ul>
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> <div>  - es,          - gb,          - it,          - ru,          - us.       </div>

Параметры конфигурационного файла клиента (часть 8 из 8):

Раздел/Параметр	Описание
millisecondMonitoring	Включает обновление данных каждую миллисекунду; <ul style="list-style-type: none"><li>• false - обновление выключено,</li><li>• true - обновление включено.</li></ul> По умолчанию - <b>false</b> - обновление выключено.
pollInterval	Интервал между запросами данных с REST-сервера, если соединение между клиентом и Comet-сервером недоступно (в миллисекундах). По умолчанию - 5000 - 5 секунд.
pollIntervalSocket	Интервал между запросами данных с Comet-сервера, если соединение между клиентом и Comet-сервером доступно (в миллисекундах). По умолчанию - 60000 - 1 минута.
title	Заголовок для вкладок, а также имя, отображаемое в левом верхнем углу web-интерфейса. По умолчанию - SAYMON.
themes	Этот параметр отвечает за настройку тем. По умолчанию используется тема "Light". Также доступна тема "Dark". CSS-файлы с темами находятся в директории "\$SAYMON_ROOT/target/client/css". "\$SAYMON_ROOT" - директория, где установлен Центральный Пульт. <pre>themes: {   default: { name: 'Light' },   dark: { name: 'Dark' } }, ...</pre>



### 4.15 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
```

### 4.16 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 инцидентом и выбрать пункт "Подтвердить".

В консоли отобразится websocket-нотификация.

### 4.1.7 Увеличение количества обработчиков SNMP-Trap

Для поддержания бесперебойного наблюдения за объектами и оповещения администратора Центральный Пульс 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 то количество 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
```

### 4.1.8 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.
# Lets 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
```

### 4.1.9 Включение функционала инцидентов

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

1. В конфигурационном файле сервера (**/etc/saymon/saymon-server.conf**) добавить или изменить параметр:

```
"conditional_incidents_enabled : true"
```

2. В конфигурационном файле клиента (**/usr/local/saymon/client/client-config.js**) добавить или изменить параметр:

```
"enableConditionalIncidents : true"
```

### 4.2 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
```

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

Проверить, что контейнер "saumon-analytics" запустился и работает, можно с помощью команды

```
docker ps -a
```

### 4.2.1 Подключение модуля аналитики к Центральному Пульту

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

1. В разделе "Server" конфигурационного файла сервера **`"/etc/saymon/saymon-server.conf"`** добавить следующие параметры:

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

2. В конфигурационном файле клиента **`"/usr/local/saymon/client/js/client-config.js"`** установить параметру `"enableAnalytics"` значение `"true"`:

```
...  
enableAnalytics: true,  
...
```

3. Перезапустить сервер Центрального Пульта:

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

### 4.2.2 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.



### 4.3 Управление сенсорами мониторинга

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

### 4.3.1 Скрытие неиспользуемых сенсоров

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

1. Открыть конфигурационный файл клиента `/usr/local/saymon/client/client-config.js` и добавить в него секцию **monitoring**:

```
...  
monitoring: {  
  standardTasks: ['ping', 'snmpGet']  
},  
...
```

*Примечание: Подробную информацию о файле **client-config.js** см. в разделе 4.1.4 на стр. 48.*

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

*Примечание: Идентификаторы сенсоров чувствительны к регистру символов.*

Список индикаторов 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 (ping-сенсор),
- processInfo (процесс по имени),
- remoteNetworkPort (удалённый порт),
- snmpGet (snmpGet-сенсор),
- snmpTrap (snmpTrap-сенсор),
- wmi (wmi-сенсор).

### 4.3.2 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.

*Примечание: С пользовательскими сенсорами могут работать только агенты, установленные на сервере системы.*

Чтобы сделать пользовательский скрипт мониторинговым сенсором, необходимо:

1. Задать каталог для хранения пользовательских сенсоров:

1.1. Открыть конфигурационный файл сервера `"/etc/saymon/saymon-server.conf"` и добавить в него секцию **monitoring**:

```
...  
"monitoring": {  
    "custom_tasks_path": "/opt/tasks"  
},  
...
```

*Примечание: Подробную информацию о файле **saymon-server.conf** см. в разделе 4.1.3 на стр. 30*

1.2. В параметре **monitoring.custom\_tasks\_path** указать полный путь до каталога.

2. Поместить исполняемый файл в указанный каталог.

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

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

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

*С перечнем идентификаторов стандартных сенсоров можно ознакомиться в разделе 4.3.1 на стр. 65.*

4. 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 содержимого файла настроек со всеми типами аргументов (часть 1 из 2):

```
{
  "title": "Custom probe example",
  "excludeForClasses": [3, 4],
  "icon": "fa fa-cubes fa-fw",
  "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 default 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 allows to enter multiline texts.",
      "id": "--textarea",
      "type": "textarea"
    },
    {
      "name": "Type - Checkbox",
      "description": "This is the checkbox type",
      "id": "--checkbox",
      "type": "checkbox"
    }
  ]
}
```

Пример содержимого файла настроек со всеми типами аргументов (часть 2 из 2):

```
{
  "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"
    }
  ]
}
]
```

Возможные поля файла настроек параметров (часть 1 из 3):

Поле	Описание
title	Отображаемое в web-интерфейсе имя сенсора. Опциональное поле. По умолчанию - идентификатор сенсора. <i>Если идентификатор сенсора содержит путь (исполняемый файл вложен в подпапки), имя будет отображаться с путём.</i> <i>Пример:</i> <i>путь к файлу: (путь до папки с проверками)/cisco/task_1.sh</i> <i>отображаемое им сенсора: cisco - task_1</i>
excludeForClasses	Список ID классов объектов, к которым неприменим данный сенсор. Опциональное поле. По умолчанию сенсор доступен для всех классов объектов.
icon	Иконка из набора Font Awesome, отображаемая рядом с именем проверки. Опциональное поле. По умолчанию -  - 'fa fa-cubes fa-fw'
args	Настройки аргументов, которые задаются пользователем через web-интерфейс и передаются исполняемому файлу.
args.id	Имя аргумента, передаваемое исполняемому файлу. Опциональное поле.
args.default	Значение аргумента по умолчанию, подставляемое в соответствующее поле при создании проверки. Опциональное поле.
args.description	Описание аргумента, отображаемое в web-интерфейсе. Опциональное поле.
args.name	Название аргумента, отображаемое в web-интерфейсе.
args.required	Признак обязательности заполнения значения аргумента. <ul style="list-style-type: none"><li>• false - значение опционально,</li><li>• true - значение обязательно.</li></ul> Опциональное поле. По умолчанию - <b>false</b> .

Возможные поля файла настроек параметров (часть 2 из 3):

Поле	Описание
<b>args.options</b>	<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"><li>• имя аргумента (<i>args.id</i>);</li><li>• значение аргумента (<i>args.options.value</i>).</li></ul> <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. Такие опции можно использовать для реализации пункта "Не выбрано" и разделителей в выпадающем списке.</p>

Возможные поля файла настроек параметров (часть 3 из 3):

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



### 4.4 Настройка уведомлений

При переходе объектов в определенные состояния система может:

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

Уведомления настраиваются в виде подробной информации об объекте в секции "Действия при смене состояния" (Рис. 4.4.1):

⚡ Действия при смене состояния				
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. 4.4.1. Действия при смене состояния

### 4.4.1 Активация функционала отправки SMS и голосовых вызовов

Для активации функционала необходимо выполнить описанные ниже действия:

1. В файле конфигурации клиента **"/usr/local/saymon/client/client-config.js"** установить параметрам **"enableSmsTrigger"** и **"enableVoiceCallTrigger"** значение **"true"**:

```
return {  
  ...  
  enableVoiceCallTrigger : true,  
  enableSmsTrigger : true,  
  ...  
}
```

2. В разделе **"Server"** конфигурационного файла сервера **"/etc/saymon/saymon-server.conf"** задать путь до скриптов, осуществляющих отправки SMS-уведомлений и голосовые вызовы:

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

*Примечание: Скрипт, отправляющий SMS-уведомления, получает в качестве аргументов:*

*\$1 - номер телефона получателя (как введен в триггере);  
\$2 - ID объекта, в котором сработал триггер;  
\$3 - имя объекта, в котором сработал триггер;  
\$4 - ID состояния объекта;  
\$5 - текст уведомления, настроенного в шаблонах уведомлений.*

*Скрипт, осуществляющий голосовой вызов, получает в качестве аргументов:*

*\$1 - номер телефона получателя (как введен в триггере);  
\$2 - ID объекта, в котором сработал триггер;  
\$3 - имя объекта, в котором сработал триггер;  
\$4 - ID состояния объекта.*

3. Для применения изменений перезапустить службу **"saymon-server"**:

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

### 4.4.2 Отправка почтовых уведомлений

Для настройки отправки почтовых уведомлений требуется настроить необходимые параметры доступа к почтовому серверу в подразделе "Email" раздела "Server" файла конфигурации сервера `/etc/saymon/saymon-server.conf`.

Подробности см. в п. 4.1.3 "Конфигурация сервера" в разделе "Email" на стр. 43-44.

### 4.4.3 Настройка уведомлений в Telegram

Для настройки отправки уведомлений с помощью Telegram необходимо:

#### 1. Задать бота, от которого будут поступать уведомления:

##### 1.1. Создать нового бота:

- 1.1.1. Найти в приложении Telegram контакт "BotFather".
- 1.1.2. Отправить ему сообщение "/newbot".
- 1.1.3. Задать боту отображаемое имя (name - позже возможно изменить).
- 1.1.4. Задать боту уникальное имя (username - изменить будет невозможно).
- 1.1.5. Скопировать токен бота вида  
`210979209:AAFfT2mt3oW4EK1gYqE_d3OjAJSIRLSrAL`
- 1.1.6. Отправить контакту "BotFather" сообщение "/setprivacy".
- 1.1.7. Выбрать созданного бота по его username.
- 1.1.8. Выбрать опцию Disable.

##### 1.2. Использовать существующего бота:

- 1.2.1. Найти в приложении Telegram контакт "BotFather".
- 1.2.2. Отправить ему сообщение "/mybots".
- 1.2.3. Выбрать нужного бота.
- 1.2.4. Выбрать опцию API Token.
- 1.2.5. Скопировать токен бота вида  
`210979209:AAFfT2mt3oW4EK1gYqE_d3OjAJSIRLSrAL`

#### 2. Настроить канал, чат или группу:

##### 2.1. Создать приватный канал (рекомендуется):

- 2.1.1. В приложении Telegram создать новый канал.
- 2.1.2. Открыть настройки канала и добавить бота в список администраторов.
- 2.1.3. Отправить сообщение в канал.
- 2.1.4. Перейти по ссылке в любом web-браузере, вставив в неё токен своего бота (без пробелов и знаков <>):  
`https://api.telegram.org/bot<_>/getUpdates`
- 2.1.5. Найти текст со словами "chat" и "id", например  
... "channel\_post": { "message\_id": 4, "chat": { "id": -1001156346945, "title": "SAYMON"...  
Здесь -1001156346945 - искомый ID канала.

##### 2.2. Создать чат:

- 2.2.1. Отправить боту любое сообщение.
- 2.2.2. Перейти по ссылке в любом web-браузере, вставив в неё токен своего бота (без пробелов и знаков <>):  
`https://api.telegram.org/bot<_>/getUpdates`
- 2.2.3. Найти текст со словами "chat" и "id", например  
... ": "K"}, "chat": { "id": 121399918, "first\_ ...  
Здесь 121399918 - искомый ID чата.

### 2.3. Настроить группу:

2.3.1. Добавить бота в группу.

2.3.2. Отправить боту в группу любое сообщение, начав его со знака @.

2.3.3. Перейти по ссылке в любом web-браузере, вставив в неё токен своего бота (без пробелов и знаков <>):

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

2.3.4. 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.

### 4.5 Интеграция со сторонними системами

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

### 4.5.1 Интеграция с Zabbix

Для импорта данных в 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 следующую информацию:

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

где:

- `url` - адрес 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.

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",  
    "parent_id" : "58b586d5c3a2f966427abcde"  
  }  
]
```

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" : [  
  {  
    ...  
    "polling_period": 30000,  
    ...  
  }  
]
```



### 4.5.2 Интеграция с Grafana

*Примечание: 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. 4.5.2.1):

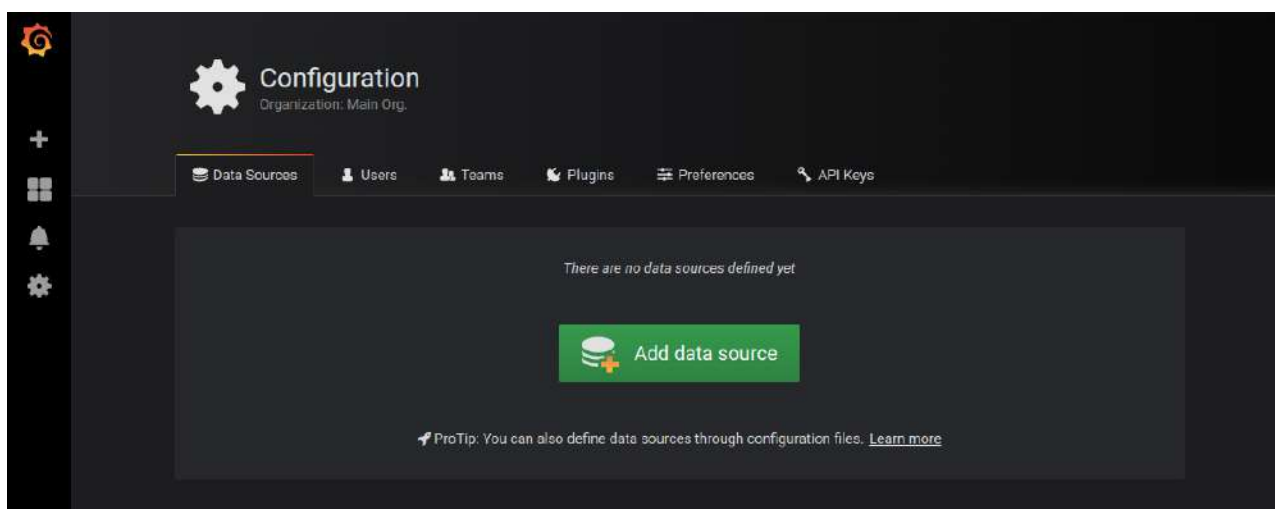


Рис. 4.5.2.1. Добавление источника данных в Grafana

3. В открывшемся окне типов источников данных выбрать "SAYMON" (Рис. 4.5.2.2):

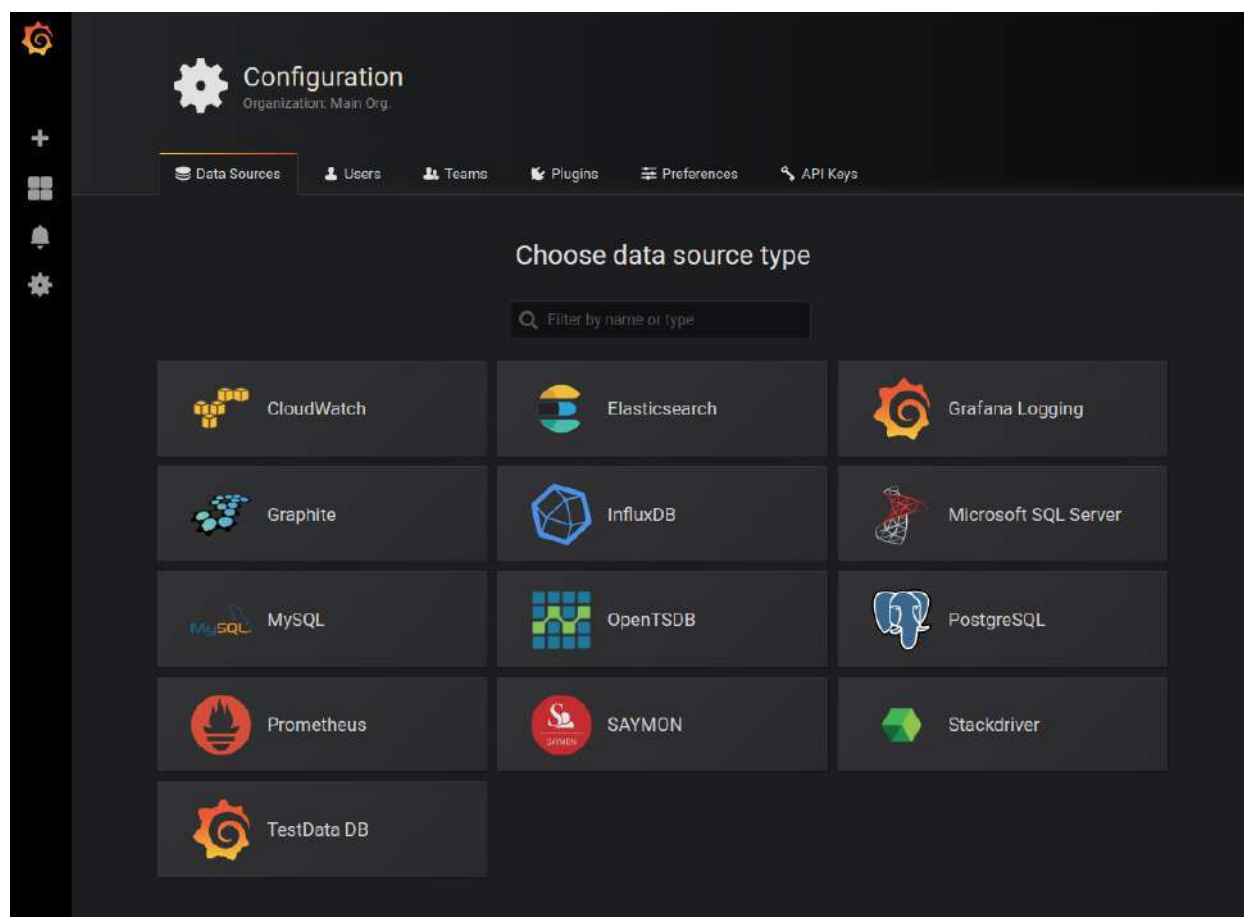


Рис. 4.5.2.2. Выбор источника данных в Grafana

4. В окне настроек необходимо указать (Рис. 4.5.2.3):

- 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.

The screenshot displays the 'Data Sources / SAYMON' configuration page in Grafana. The interface is dark-themed. At the top, there's a header with the SAYMON logo and the title 'Data Sources / SAYMON'. Below this, a 'Settings' tab is selected. The main content area is divided into sections: 'Settings' with a 'Name' field set to 'SAYMON' and a 'Default' checkbox checked; 'HTTP' section with 'URL' set to 'https://saas.saymon.info', 'Access' set to 'Server (Default)', and 'Whitelisted Cookies' with an 'Add Name' button; 'Auth' section with 'Basic Auth' checked, 'With Credentials' checked, 'TLS Client Auth' unchecked, 'With CA Cert' unchecked, and 'Skip TLS Verify' unchecked; and 'Basic Auth Details' with 'User' set to 'admin' and 'Password' masked with dots. At the bottom, there are three buttons: 'Save & Test' (green), 'Delete' (red), and 'Back' (grey).

Рис. 4.5.2.3. 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. 4.5.2.4):

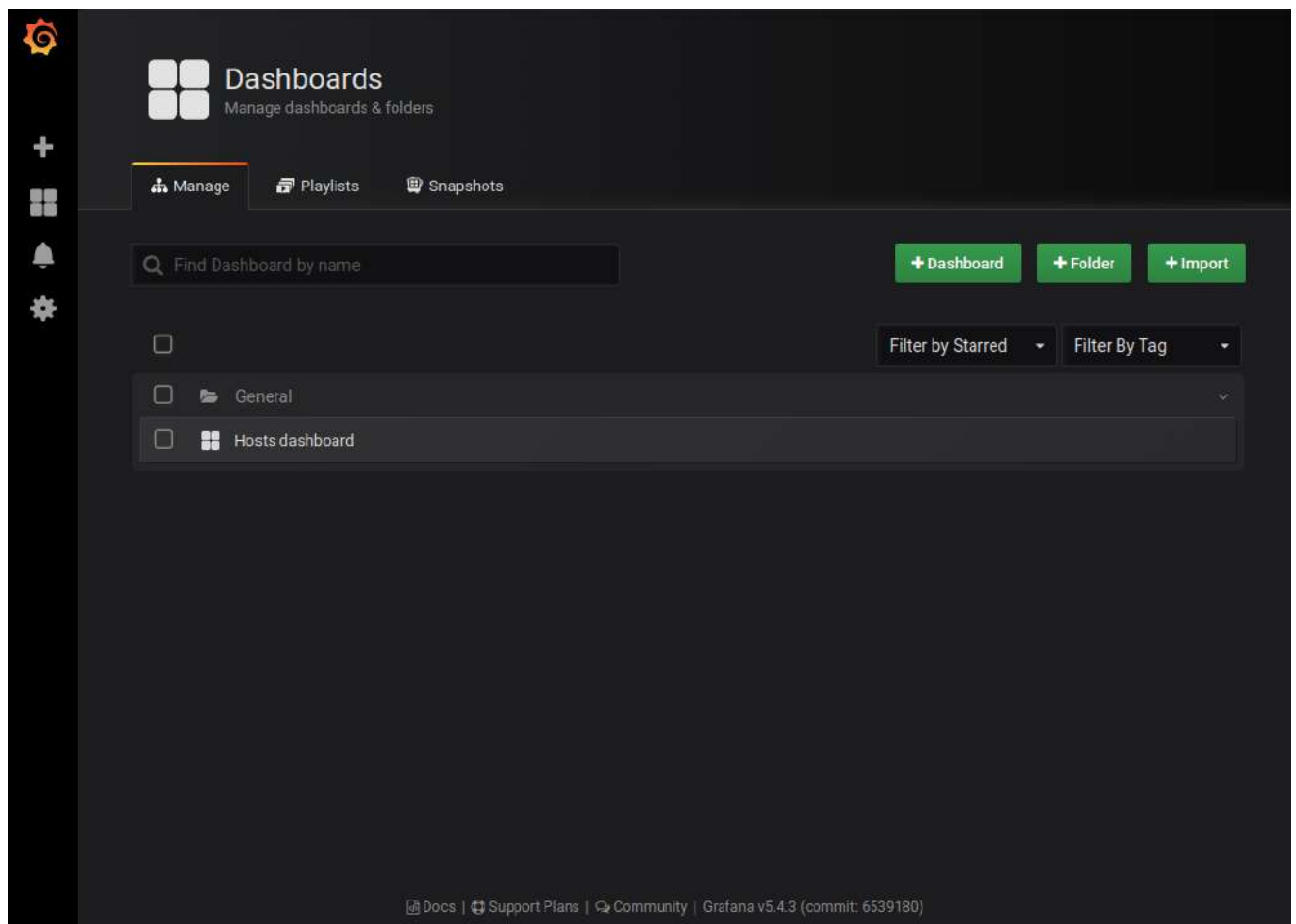


Рис. 4.5.2.4. Создание нового дашборда в Grafana

6. Выбрать тип отображения дашборда. Например, график (Рис. 4.5.2.5):

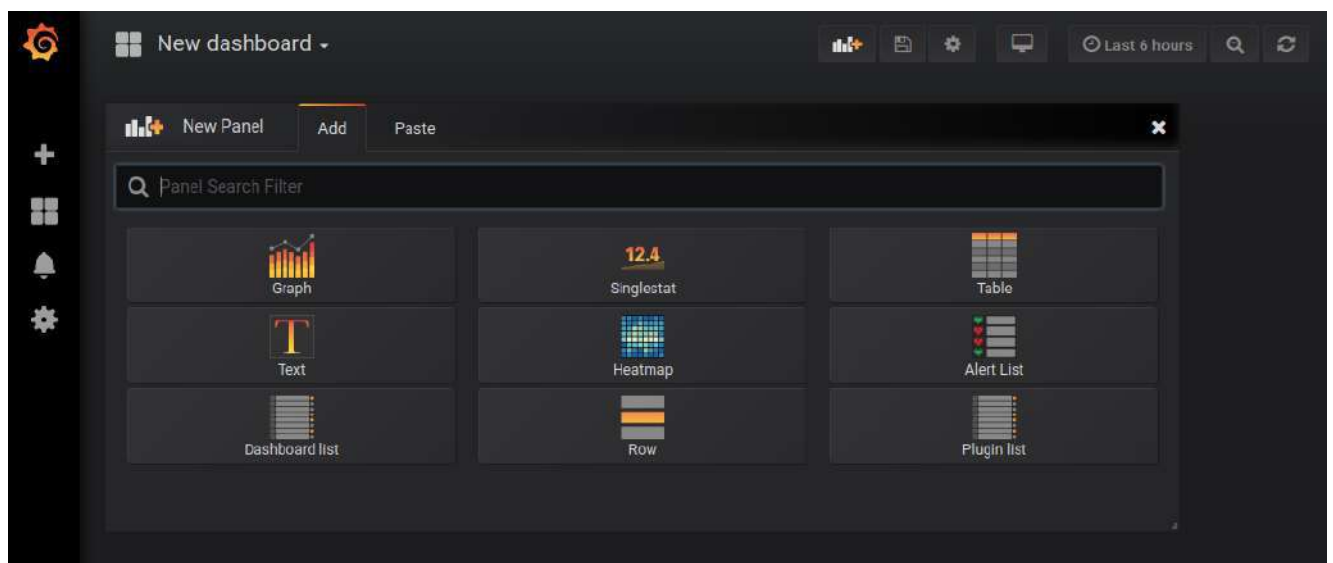


Рис. 4.5.2.5. Выбор типа отображения дашборда в Grafana

7. Вызвать контекстное меню нажатием на стрелку рядом с заголовком панели и перейти в режим редактирования.
8. В окне редактирования перейти во вкладку "Metrics" и добавить объекты, к которым у пользователя, под чьей учётной записью совершён вход, есть доступ в Центральном Пульте (Рис. 4.5.2.6). Указать:
  - имя или ID объекта;
  - метрику, которую необходимо отобразить на графике;
  - отображать ли путь до объекта.

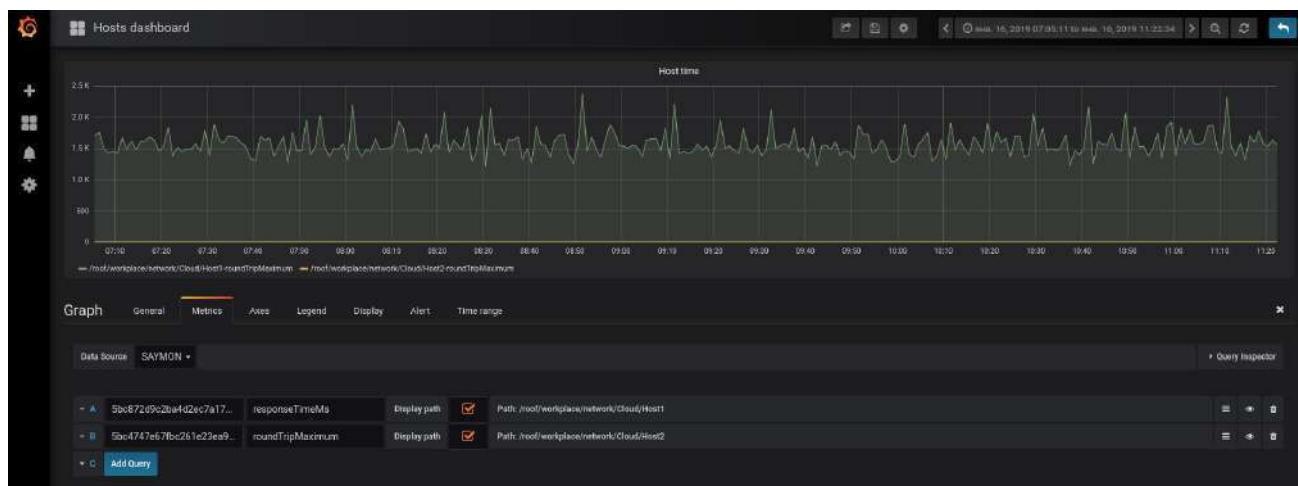


Рис. 4.5.2.6. Отображение дашборда в Grafana

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

### 4.6 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;
- Mac OS X - /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.

### 4.6.1 Конфигурация log-файлов

Конфигурация log-файлов агента осуществляется в файле **"/opt/saymon-agent/conf/logback-upstart.xml"**.

Секция настройки debug-режима:

```
<appender name="FILE-DEBUG" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>/var/log/saymon/saymon-agent.debug.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <!-- Daily rollover -->
    <fileNamePattern>/var/log/saymon/saymon-agent.debug.%d{yyyy-MM-dd}.
log.gz</fileNamePattern>
    <!-- Keep 10 days' worth of history -->
    <maxHistory>10</maxHistory>
  </rollingPolicy>
  <encoder>
    <pattern>%d{dd.MM.yyyy HH:mm:ss.SSS} [%-15thread] %-5level %logger{36} -
%msg%n</pattern>
  </encoder>
</appender>
```

где:

- `<file>/var/log/saymon/saymon-agent.debug.log</file>` – размещение log-файла;
- `<fileNamePattern>... .gz</fileNamePattern>` – указание на архивацию файлов в формат .gz;
- `<maxHistory>10</maxHistory>` – длительность хранения файлов в днях.

Для отключения debug-режима закомментировать соответствующую строку в секции ROOT следующим образом:

```
<!--<appender-ref ref="FILE-DEBUG"/>-->
```

Настройки и структура секции базового логирования аналогична секции настройки debug-режима:

```
<appender name="FILE-INFO" class="ch.qos.logback.core.rolling.RollingFileAppender">
...
</appender>
```

### 4.6.2 Конфигурация ротации log-файлов

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.



### 4.6.3 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
```

### 4.6.4 Назначение ответственного за событие

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

#### 1. Через Web UI:

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

Имя ответственного пользователя отобразится в соответствующем столбце (Рис. 4.6.4.1):

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"	

Рисунок. 4.6.4.1. 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": "..."}'
```

### 4.6.5 Установка ограничения для логирования

Центральный Пульт позволяет установить максимальный объём хранилища двумя способами:

1. Через Web UI:

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

1.2. Перейти в раздел "Журнал событий" (Рис. 4.6.5.1):

Рис. 4.6.5.1. Ограничение объёма хранилища

1.3. Заполнить поля требуемыми значениями.

1.4. 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}'
```

### 4.6.6 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
}
```

### 4.6.7 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
```

### 4.7 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-либо его группы.

Для авторизации пользователей с помощью сервера службы каталогов (Active Directory, OpenLDAP, и т.д.) необходимо добавить раздел LDAP в конфигурационный файл сервера. Подробнее см. п. 4.1.3 "Конфигурация сервера" раздел LDAP на стр. 33.

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 .

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

При использовании протокола LDAP, если пользователь принадлежит к группам, во время авторизации пользователя в системе будут автоматически созданы группы с такими же именами (если не были созданы ранее), пользователь автоматически будет в них добавлен. Права для групп необходимо настраивать отдельно.

### 4.7.1 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:

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

1.2. Перейти в раздел "Пользователи" (Рис. 4.7.1.1):

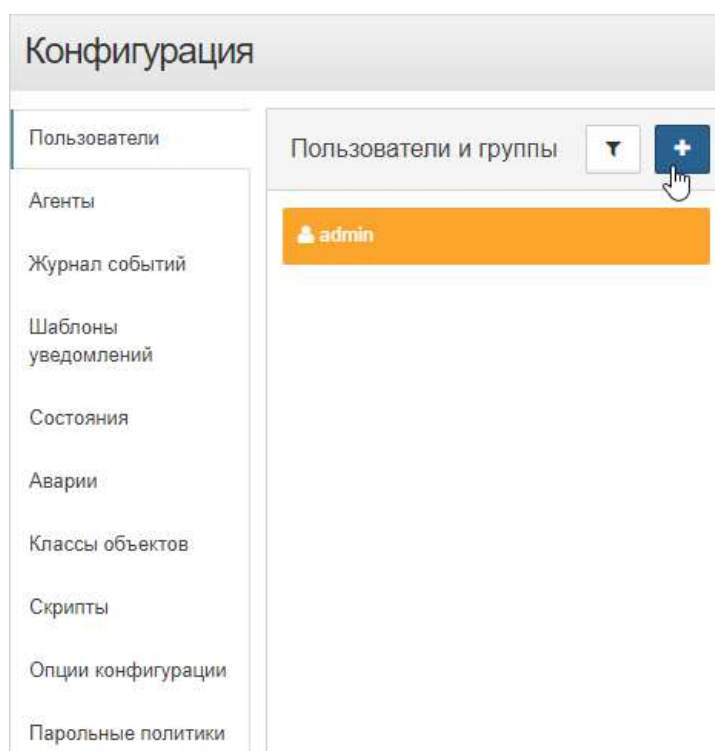


Рис. 4.7.1.1. Добавление пользователя

1.3. Нажать кнопку  и выбрать добавление пользователя.

1.4. Ввести логин пользователя, пароль и подтверждение.

1.5. 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
```



### 4.7.2 Назначение пользователям прав доступа

*Примечание: Список операций с подробным описанием содержится в Приложении А на стр. 136.*

Настройка прав пользователей осуществляется двумя способами:

1. Через Web UI:

- 1.1. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
- 1.2. Перейти в раздел "Пользователи".
- 1.3. Выбрать нужного пользователя или группу из списка.
- 1.4. Открыть вкладку "Права на операции" (Рис. 4.7.2.1):

Пользователи и группы

user

Общие Интерфейс Смена пароля **Права на операции**

Доступ к объектам Журнал событий Удалить пользователя

Разрешить всё Запретить всё

Основные права Разрешить всё Запретить всё

Пол	Операция
НЕТ	Выполнение операций
НЕТ	Поиск и групповые операции
НЕТ	Управление объектами, свойствами и документами
НЕТ	Создание объектов
НЕТ	Модификация объектов
НЕТ	Удаление объектов
НЕТ	Управление связями, свойствами и документами
НЕТ	Создание связей
НЕТ	Модификация связей
НЕТ	Удаление связей
НЕТ	Управление потоками
НЕТ	Создание потоков
НЕТ	Модификация потоков
НЕТ	Удаление потоков

Рис. 4.7.2.1. Права на операции

1.5. Отредактировать права пользователя или группы.

*Примечание: Если пользователь принадлежит к группам, права пользователя и его групп суммируются.*

### 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
```

### 4.7.3 Настройка 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. 4.7.3.1):

Конфигурация		
<b>Пользователи</b>	<b>Парольные политики</b>	<b>Редактирование политики</b>
Агенты	По умолчанию	
Журнал событий		
Шаблоны уведомлений		
Состояния		
Аварии		
Классы объектов		
Скрипты		
Опции конфигурации		
Парольные политики		

Имя		
Минимальное количество символов	<input checked="" type="checkbox"/> ДА	8
Минимальное количество символов нижнего регистра	<input checked="" type="checkbox"/> ДА	1
Минимальное количество символов в верхнем регистре	<input checked="" type="checkbox"/> ДА	1
Минимальное количество цифр	<input checked="" type="checkbox"/> ДА	1
Минимальное количество специальных символов	<input checked="" type="checkbox"/> ДА	1

✓ Применить

Рис. 4.7.3.1. Парольные политики

3. Установить необходимые требования к паролям пользователей.
4. Нажать кнопку  Применить.

### 4.7.4 Изменение пароля от учётной записи

Смена пароля пользователей осуществляется двумя способами:

1. Через Web UI:

- 1.1. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
- 1.2. Перейти в раздел "Пользователи".
- 1.3. Выбрать нужного пользователя из списка.
- 1.4. Открыть вкладку "Смена пароля" (Рис. 4.7.4.1):

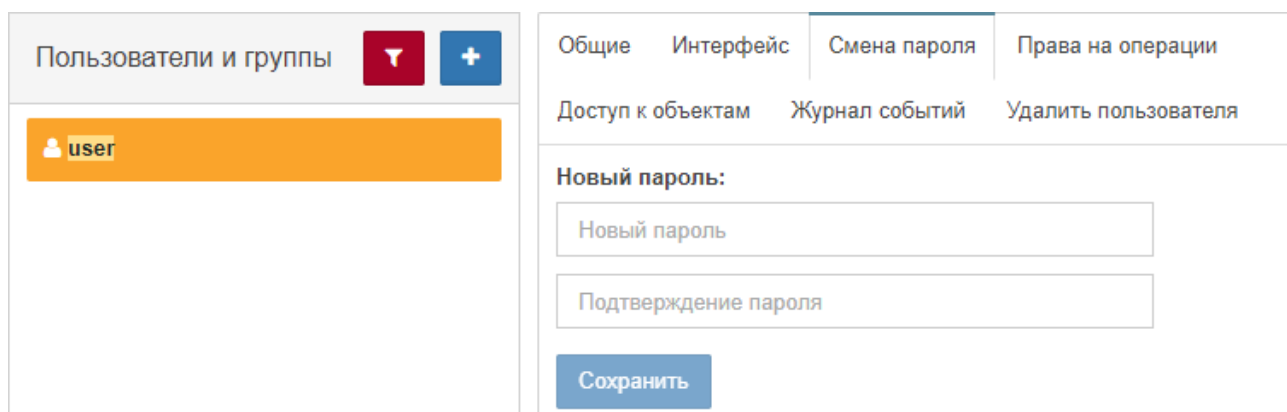


Рис. 4.7.4.1. Смена пароля пользователя

- 1.5. Ввести новый пароль пользователя и подтверждение.

*Примечание: При смене собственного пароля необходимо также ввести текущий пароль.*

- 1.6. 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
```

### 4.7.5 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. 4.7.5.1):

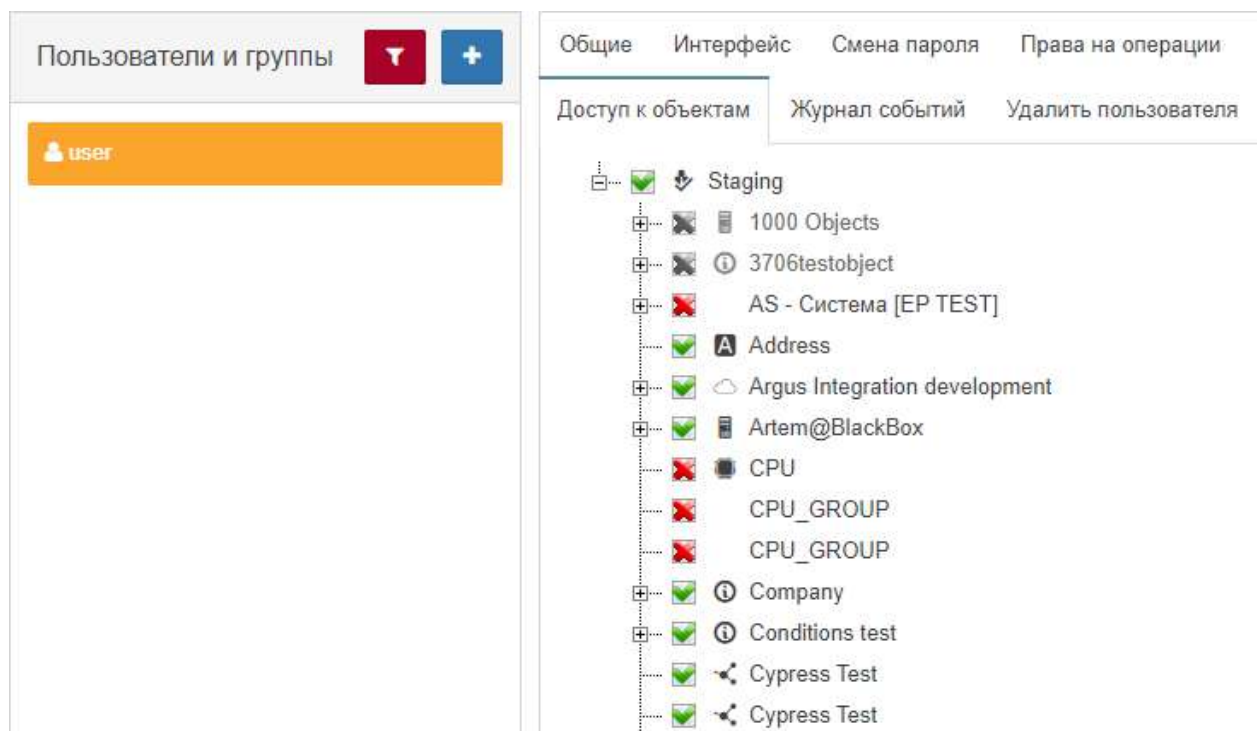


Рис. 4.7.5.1. Доступ к объектам

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

*Примечание: Объекты, недоступные для группы, становятся недоступными для всех членов группы.*

### 4.7.6 Блокировка учётной записи

Блокировка учётных записей пользователей осуществляется двумя способами:

1. Через Web UI:

- 1.1. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
- 1.2. Перейти в раздел "Пользователи".
- 1.3. Выбрать нужного пользователя из списка.
- 1.4. Открыть вкладку "Общие" (Рис. 4.7.6.1):

Общие	Интерфейс	Смена пароля	Права на операции	Доступ к объектам	Журнал событий	Удалить пользователя								
ID	58048d843bddd07007607034													
Логин	demo													
Email	<input type="text"/>													
Статус	Активен ▼													
Активен до	<input type="text" value="DD.MM.YYYY H:mm"/> <input type="button" value="📅"/> <input type="button" value="✖"/>													
Группа	-- Не выбрано -- ▼													
Ссылка для авторизации	<input type="text"/> <input type="button" value="🔄"/>													
Язык	English ▼													
Активность	<table border="1"><thead><tr><th>🕒 Время</th><th>⚡ Действие</th><th>* Тип</th><th>Разница</th></tr></thead><tbody><tr><td>17.03.2020, 16:34:27</td><td>Удалено</td><td>Session</td><td><pre>{   "userId": "58048d843bddd07007607034",   "id": "481ee9fd-6de2-4512-80eb-27e1317a5301" }</pre></td></tr></tbody></table>						🕒 Время	⚡ Действие	* Тип	Разница	17.03.2020, 16:34:27	Удалено	Session	<pre>{   "userId": "58048d843bddd07007607034",   "id": "481ee9fd-6de2-4512-80eb-27e1317a5301" }</pre>
🕒 Время	⚡ Действие	* Тип	Разница											
17.03.2020, 16:34:27	Удалено	Session	<pre>{   "userId": "58048d843bddd07007607034",   "id": "481ee9fd-6de2-4512-80eb-27e1317a5301" }</pre>											

Рис. 4.7.6.1. Общие параметры пользователя

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

В поле "Активен до" можно задать дату и время блокировки учетной записи. При наступлении указанного времени статус учётной записи будет автоматически изменён на "Заблокирован".

*Примечание: В случае, если указанное время меньше текущего, статус учётной записи немедленно изменится 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
```

### 4.7.7 Удаление учётной записи

Удаление учётных записей осуществляется двумя способами:

1. Через Web UI:

- 1.1. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
- 1.2. Перейти в раздел "Пользователи".
- 1.3. Выбрать нужного пользователя из списка.
- 1.4. Открыть вкладку "Удалить пользователя" (Рис. 4.7.7.1):

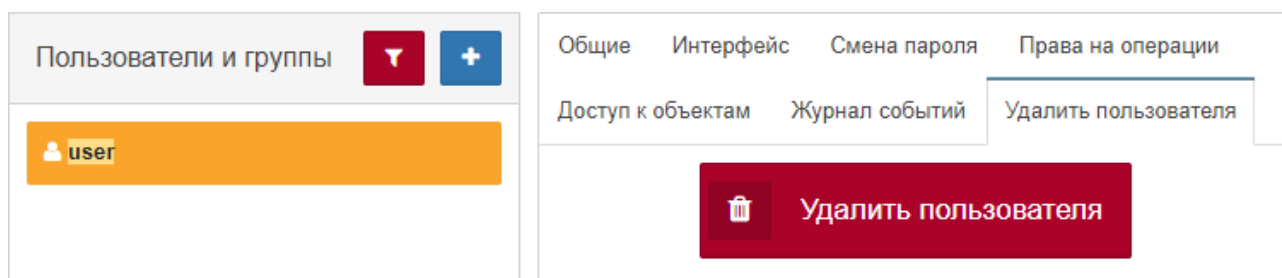


Рис. 4.7.7.1. Удаление пользовател

- 1.5. Нажать кнопку удаления пользователя.
- 1.6. 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
```

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

*Чтобы заблокировать LDAP-пользователю доступ к данным, достаточно сменить его статус на "Заблокирован".*



### 4.8 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.

### 4.8.1 Создание объекта

Создать объект можно двумя способами:

#### 1. Через Web UI:

1.1. Нажать кнопку  на панели инструментов.

1.2. В окне "Новый объект" ввести имя объекта и выбрать класс объекта (Рис. 4.8.1.1):

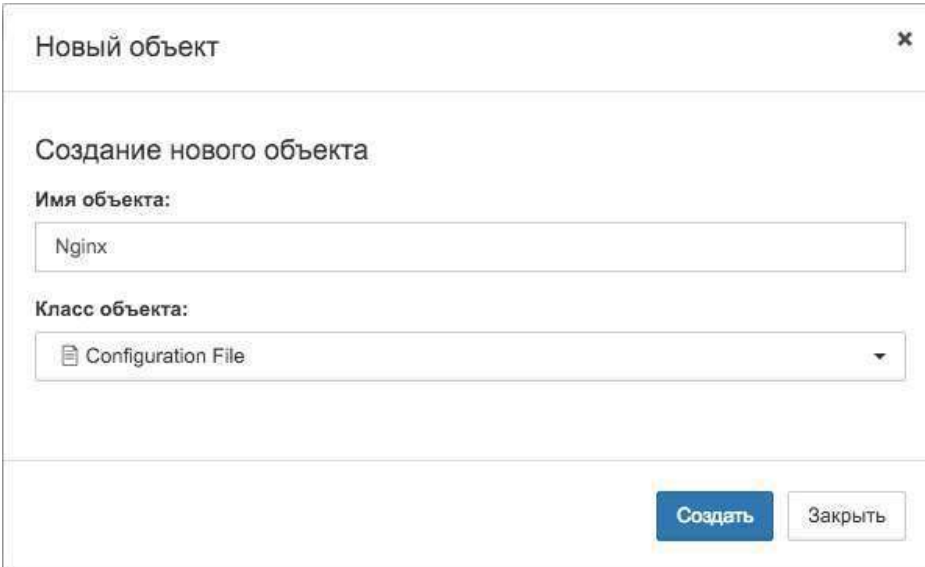


Рис. 4.8.1.1. Окно создания нового объекта

1.3. 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
```

Примечание: "parent\_id" - ID родительского объекта для создаваемого.

### 4.8.2 Клонирование объекта

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

Клонировать объект можно двумя способами:

1. Через Web UI:

- 1.1. Вызвать контекстное меню клонируемого объекта щелчком правой кнопкой мыши.
- 1.2. 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
```


### 4.8.3 Удаление объекта

Удалить объект можно тремя способами:

1. Через Web UI (контекстное меню объекта):

- 1.1. Вызвать контекстное меню удаляемого объекта щелчком правой кнопкой мыши.
- 1.2. Выбрать пункт меню "Удалить".
- 1.3. Подтвердить удаление объекта во всплывающем окне.

2. Через Web UI (режим удаления элементов):

- 2.1. Нажать кнопку  в панели хлебных крошек.
- 2.2. Нажать на такую же иконку на удаляемом объекте.
- 2.3. 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

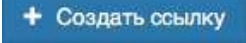
curl -X DELETE $url -u $login:$password
```

### 4.8.4 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.

Создать ссылку можно двумя способами:

#### 1. Через Web UI:

- 1.1. Нажать на кнопку  на панели инструментов.
- 1.2. В появившемся всплывающем окне "Новая ссылка" (Рис. 4.8.4.1) выбрать из выпадающего списка объект, на который создаётся ссылка:

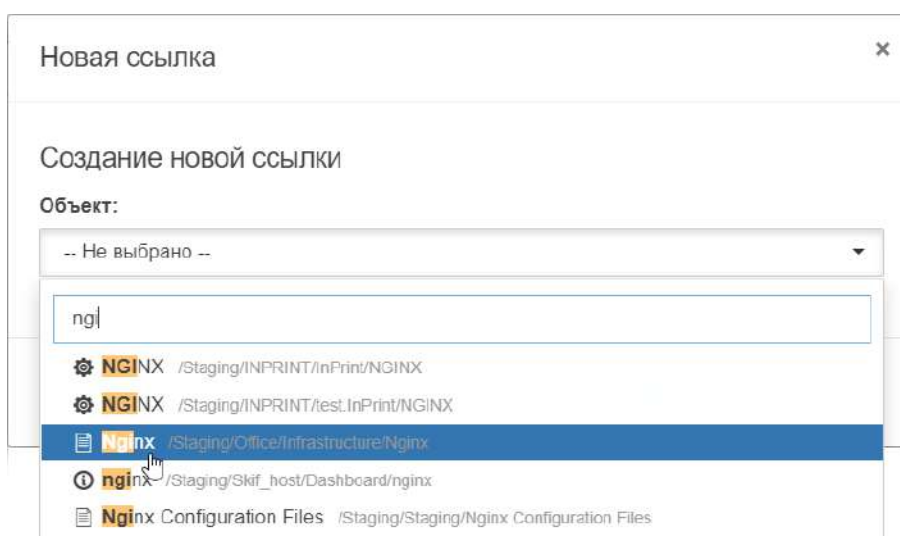


Рис. 4.8.4.1. Окно создания новой ссылки

- 1.3. Нажать на 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"}'
```

### 4.8.5 Создание связи

Связи между объектами также могут являться объектами мониторинга.

Создать связь можно двумя способами:

#### 1. Через Web UI:

- 1.1. Перейти в режим создания связей, нажав кнопку **+ Создать связь** на панели инструментов.
- 1.2. После того, как на всех объектах появится соответствующий символ , нажать на него на исходном объекте и, удерживая, переместить курсор на целевой объект (Рис. 4.8.5.1):

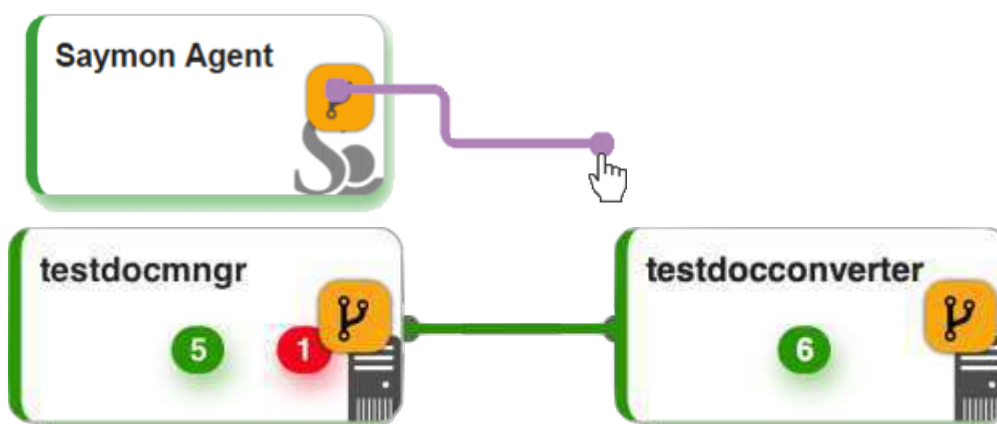


Рис. 4.8.5.1. Создание связи

- 1.3. Нажать кнопку **Выйти** 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
```

### 4.8.6 Удаление связи

Удалить связь можно двумя способами:

#### 1. Через Web UI:

- 1.1. Нажать кнопку  в панели хлебных крошек.
- 1.2. Нажать на такую же иконку на удаляемой связи.
- 1.3. 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
```


### 4.9 Настройка интерфейса

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



### 4.9.1 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 **"/usr/local/saymon/client/client-config.js"**.  
Подробнее см. в п. 4.1.4 в разделе Grid на стр. 52.

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 обновить страницу в браузере.

### 4.9.2 Настройка заголовка web-интерфейса

Для изменения заголовка в web-интерфейсе:

1. Открыть файл `"/usr/local/saymon/client/client-config.js"`.
2. В строке "title" ввести желаемое имя:

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

3. Обновить страницу браузера.

### 4.9.3 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  
  
}
```

### 4.9.4 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;  
}
```

### 4.9.5 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 файл конфигурации:

1.1. Создать файл: /usr/local/saymon/saymon.local/css/saymon.local.css

1.2. 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.

2. Через Web UI:

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

2.2. Перейти в раздел "Состояния".

2.3. В списке состояний выбрать то, которое требуется изменить.

2.4. При необходимости изменить имя в соответствующем поле.

2.5. Настроить основной цвет, цвет тени, цвет строки таблицы и фон при помощи цветовой палитры или методом ввода номера цвета.

### 4.10 Настройка мониторинга

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

Подробную информацию о каждом типе проверок см. в "Руководстве пользователя" в п. 4.2.1 на стр. 46.

Далее рассмотрены несколько примеров процесса мониторинга.

### 4.10.1 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.

### 4.10.2 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.



### 4.10.3 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.

### 4.10.4 Проверка доступности web-ресурса

Данный тип мониторинга позволяет убедиться не только в работоспособности web-сайта (статус 200 OK), но и в ограничении 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 хотя бы одного агента в инфраструктуре.
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.

### 4.10.5 Безагентный мониторинг web-сервера

Существует ряд случаев, при которых установка агента на сервере невозможна. В таких случаях рекомендовано 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 веб-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.

### 4.11 Импорт и экспорт данных

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

#### Экспорт данных

Для экспорта данных из MongoDB в архив на существующей инсталляции системы необходимо выполнить следующую команду:

```
mongodump -d saymon -o mongodb_backup
```

Для экспорта настроек конфигурации необходимо сделать копии:

- директории с конфигурационными файлами сервера **"/etc/saymon"**;
- конфигурационного файла клиента **"/usr/local/saymon/client/client-config.js"**.

*Примечание: Подробную информацию о файле **client-config.js** см. в разделе 4.1.4 на стр. 48.*

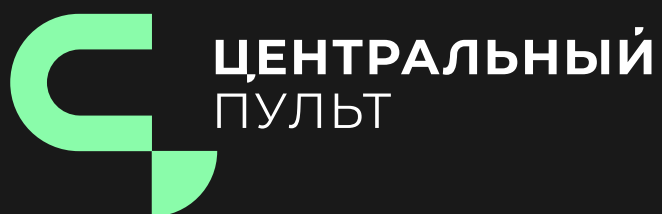
#### Импорт данных

Для восстановления или импорта данных в MongoDB на новой инсталляции системы необходимо выполнить следующую команду:

```
mongorestore --drop --noIndexRestore mongodb_backup  
echo flushall | redis-cli
```

Для импорта настроек конфигурации необходимо вставить:

- файлы из ранее сохранённой директории с конфигурационными файлами сервера в директорию **"/etc/saymon"**;
- конфигурационный файл клиента в директорию **"/usr/local/saymon/client"**.



# **Проблемы в работе системы и способы их решения**

### **5 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.

### 5.1 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** и */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 выше описанных 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
```



### 5.2 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
```

### 5.3 Проверка работы MongoDB

**Проверка наличия процесса в памяти:**

```
ps -ef| grep mongod
```

```
mongodb 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
```

### 5.4 Проверка работы MySQL

**Проверка пароля MySQL (действие на хосте с сервером):**

```
cat /etc/saymon/saymon-server.conf
```

**Просмотр секции db{}:**

```
"db" : {  
  "host" : "localhost",  
  "user" : "user",  
  "password" : "password",  
  "database" : "saymondb"  
},
```

### 5.5 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 -lnp | 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"
```

#### Проверка доступности порта для агента (действие на хосте с агентом):

```
telnet <адрес_сервера> 6379
```

```
Trying <адрес_сервера>...
```

```
Connected to <адрес_сервера>.  
Escape character is '^'].
```

### Проверка пароля Redis (действие на хосте с сервером)

1. На хосте с сервером (конфигурация Redis):

```
cat /etc/redis/redis.conf | grep requirepass  
  
requirepass Ja!MlK1&  
# If the master is password protected (using the "requirepass" configuration  
# requirepass foobared
```

2. На хосте с сервером (конфигурация Центрального Пульта):

```
cat /etc/saymon/saymon-server.conf | grep auth_pass  
  
"auth_pass" : "Ja!MlK1&"
```

3. На хосте с агентом:

```
cat /opt/saymon-agent/conf/agent.properties | grep password  
  
server.password=Ja!MlK1&
```

Пароли должны совпадать, иначе агент не сможет подключиться к серверу для отправки данных.

### 5.6 500 Internal Server Error и отсутствие графиков

#### Условие 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
```

### 5.7 Ошибка работы HTTP-проверки

#### Условие проблемы:

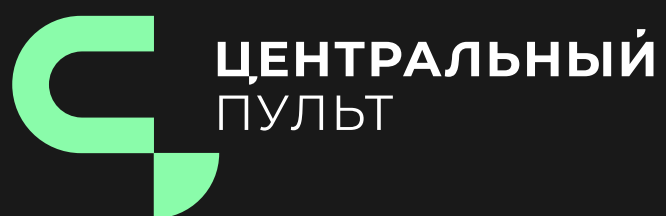
- HTTP-проверка адреса `https://xxx.xxx` не работает и возникает ошибка.

#### 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.



## **Приложения**



## Приложение А (обязательное)

## Права на операции


Основные права (часть 1 из 3):

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

## Основные права (часть 2 из 3):

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


### Основные права (часть 3 из 3):

Название	Описание	Запись в конфигурационном файле сервера
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  Применить к классу в секциях <ul style="list-style-type: none"> <li>• "Условия перехода состояний",</li> <li>• "Условия генерации аварий".</li> </ul> Включает в окне конфигурации системы вкладку "Управление классами".	"manage-classes"
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"
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	Включает в окне конфигурации системы вкладки: <ul style="list-style-type: none"><li>• Журнал событий,</li><li>• Шаблоны уведомлений,</li><li>• Аварии,</li><li>• Состояния,</li><li>• Парольные политики.</li></ul>	"manage-configuration"
Обновление агентов	Позволяет пользователю обновлять отдельных агентов кнопкой  и всех агентов на вкладке "Агенты" в окне конфигурации системы.	"upload-agent-updates"
Управление пользователями	Позволяет управлять всеми пользователями системы.	"manage-users"
Управление журналом событий	Позволяет <ul style="list-style-type: none"><li>• назначать ответственного,</li><li>• изменять текст событий,</li><li>• удалять события</li></ul> в журнале событий SNMP.	"manage-event-log"
Управление историческими данными	Позволяет <ul style="list-style-type: none"><li>• добавлять,</li><li>• редактировать,</li><li>• удалять</li></ul> комментарии на графиках.	"manage-history"
Запуск автоопределения объектов	Позволяет использовать сервисный REST-метод node/api/discovery.	"run-discovery"

[www.saymon.info](http://www.saymon.info)

2021

