



Программное обеспечение
«СКАЛА-Р МОНИТОРИНГ».
Руководство по установке

RU.ИРФЛ.00014-01.96.01

Москва
02/06/2023

Содержание

Аннотация.....	3
Идентификационные данные.....	4
Описание системы	5
Архитектура и основные компоненты системы.....	5
Клиент.....	5
Сервер.....	6
Агент.....	6
Уровень подготовки администратора для работы с ПО «СКАЛА-Р МОНИТОРИНГ»	8
Технические требования к оборудованию	9
Требование к клиентскому оборудованию.....	9
Технические требования к серверному оборудованию	9
Установка ПО «СКАЛА-Р МОНИТОРИНГ»	10
Установка серверной части ПО «СКАЛА-Р МОНИТОРИНГ»	10
Настройка серверной части ПО «СКАЛА-Р МОНИТОРИНГ»	10
Установка агента мониторинга ПО «СКАЛА-Р МОНИТОРИНГ»	18
Автоматическая установка агента.....	18
Установка агента на различные Unix/Linux системы.....	19
Установка агента на Mac OS X.....	20
Установка агента на Windows	20
Настройка агента	20
Термины и определения.....	23

Аннотация

Данный документ описывает процедуру установки серверной и агентской частей программного обеспечения (ПО) СКАЛА-Р МОНИТОРИНГ.

Документ предназначен для администратора системы мониторинга.

Идентификационные данные

Идентификационные данные ПО	Программа для ЭВМ «СКАЛА-Р МОНИТОРИНГ»
Название документа	«Программное обеспечение «СКАЛА-Р МОНИТОРИНГ». Руководство по установке»
Версия документа	1.0
Обозначение документа	RU.НРФЛ.00014-01.96.01
Автор документа	ООО «БАЗИС»

Описание системы

ПО "СКАЛА-Р МОНИТОРИНГ" — представляет собой платформу, предназначенную для постоянного и статического наблюдения и контроля состояния показателей работы сети, оборудования, приложений и сервисов, а также является основной системой мониторинга для платформ виртуализации развернутых с помощью ПО "СКАЛА-Р МОНИТОРИНГ".

ПО "СКАЛА-Р МОНИТОРИНГ" имеет многочисленный набор средств визуализации объектов и их состояний, методы управления критериями смены состояний, средства оповещения и возможности подключения средств автоматической обработки при смене состояний.

Объектом мониторинга могут быть любые физические или логические объекты как самой платформы виртуализации, так и бизнес-приложений, развернутых на платформе – например, память, процессор, файловая система, процесс или программа, количество пользователей, очередь файлов на обработку, объем обработанного трафика, выручка и иные финансовое показатели, значение температуры или химического состава газа или жидкости.

Отличительной особенностью платформы являются возможности хранения оригинальных, не модифицированных значений показателей за значительные промежутки времени с обеспечением высокой скорости записи и доступа к данным, что позволяет производить быстрый и качественный анализ ситуаций в настоящем и прошлом, строить математически обоснованные прогнозы развития ситуации в будущем в рамках контроля как отдельных объектов мониторинга, так и в разрезе показателей и объектов платформы виртуализации различных конфигураций.

Работа ПО "СКАЛА-Р МОНИТОРИНГ" и его настройка осуществляется через Веб-интерфейс, который визуализирует состояние наблюдаемых объектов и обеспечивает доступ к информации об их состоянии.

Архитектура и основные компоненты системы

Архитектура системы содержит три уровня:

- Клиент - тонкий web-клиент системы и клиенты для мобильных операционных систем Android и iOS;
- Сервер - централизованный сервер, на котором хранится и анализируется информация, полученная от агентов, а затем передается клиенту. Также здесь содержится информация об учётных записях пользователей;
- Агент - множество агентов системы, установленных на узлах инфраструктуры и собирающих информацию по ним.

Схема архитектуры системы представлена на рисунке (Рисунок 1).

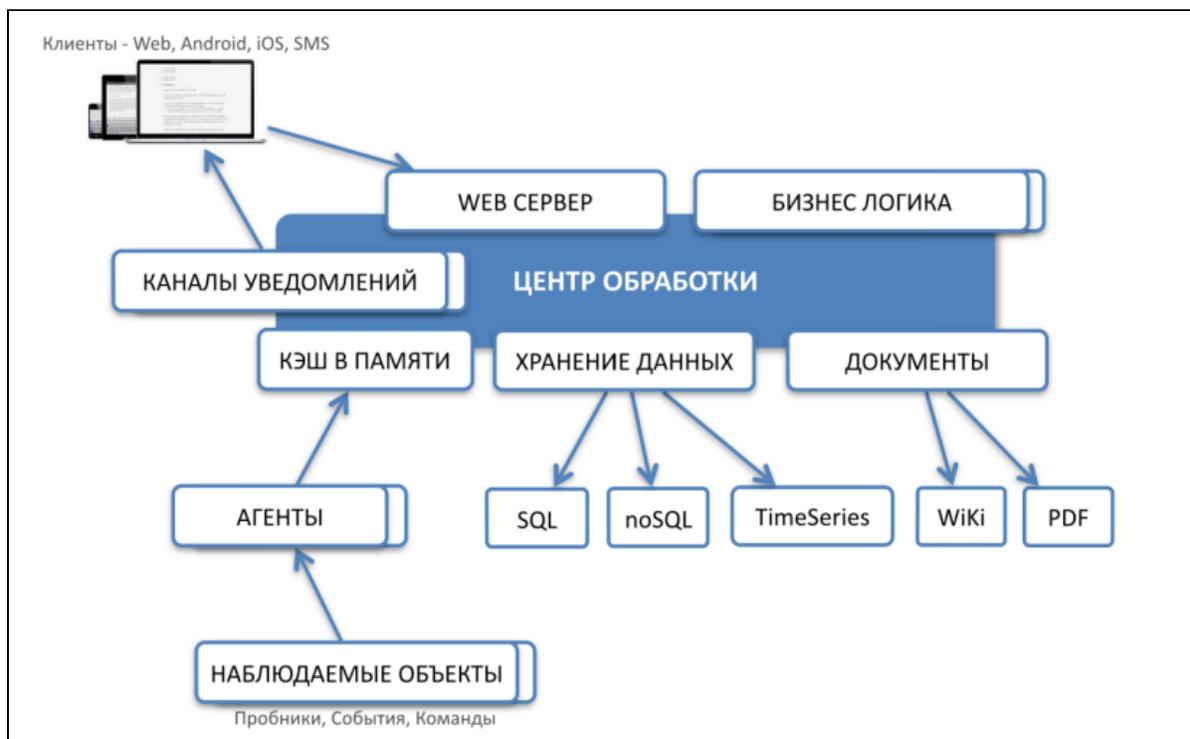


Рисунок 1. Схема архитектуры системы СКАЛА-Р МОНИТОРИНГ

Клиент

Клиент системы - тонкий web-клиент, работающий внутри браузера Google Chrome версии не ниже 58.0. Стабильность работы клиента системы в браузерах Яндекс, Safari, Opera, FireFox не гарантируется.

Для мобильных операционных систем Android и iOS разработаны приложения, поддерживающие работу с платформой для пользователей, находящихся в движении.

Сервер

Сервер включает в себя следующие компоненты:

- Центр обработки (Center) – это логический компонент, объединяющий в себе хранилище данных, кэш, REST-сервер, HTTP-сервер и бизнес-логику.
- Web-сервер – это HTTP-сервер, обслуживающий запросы клиента. Он отдаёт статические элементы web-интерфейса (HTML, JavaScript, CSS), а также проксирует запросы от клиента к REST-серверу. В качестве web-сервера используется NGINX.
- REST-сервер – серверный компонент, обрабатывающий REST-запросы от JavaScript-компонентов на тонком клиенте. Через REST API клиент получает всю информацию об инфраструктуре, а также производит манипуляции с инфраструктурой. REST-сервер реализован в виде отдельного NodeJS-приложения.
- Сервер данных производит анализ поступивших данных. В частности, управляет логикой смены состояний у объектов и связей. Сервер данных реализован в виде отдельного NodeJS-приложения.
- Документы (Хранилище документов) – структура директорий в файловой системе, предназначенная для хранения загруженных документов.
- Кэш в памяти (In-Memory кэш) – NoSQL-хранилище данных типа «ключ-значение». Хранит базу данных в оперативной памяти, благодаря чему возможен быстрый доступ к данным и их быстрая обработка. В качестве In-Memory кэша используется Redis.
- Хранилище данных (База данных). Для постоянного хранения данных используется несколько механизмов:
 - информация по объектам и связям, их свойства и исторические данные о состояниях хранятся в NoSQL-СУБД – MongoDB v.4.2;
 - числовые метрики, по которым строятся графики, сохраняются в базе данных или в специализированном хранилище временных рядов, например, OpenTSDB или InfluxDB.

Администратор серверной части осуществляет работы по установке критических обновлений как системы, так и ПО, входящего в состав серверной части.

Агент

Агент – процесс, собирающий данные о наблюдаемом узле и расположенных на нём объектах, а также связях данного узла. Данные, собранные агентом, периодически отправляются в In-Memory кэш и затем анализируются сервером. Агент реализован в виде Java-приложения.

В некоторых случаях оптимально использовать безагентный (Agentless) мониторинг.

Наблюдаемый объект – элемент инфраструктуры, по которому агент собирает данные. Это может быть, как физический элемент (хост, роутер, сетевая карта, жёсткий диск), так и логический (виртуальная машина, TCP-соединение, файловая система).

Агенты могут собирать данные и об удалённом узле с помощью дистанционных методов:

- SNMP;
- SQL;
- HTTP;
- TCP/UDP.

Более подробно с понятиями "объект" и "связь" в системе можно ознакомиться в документе в разделе "Работа с объектами и связями".

Подробная схема архитектуры системы приведена на рисунке 2.

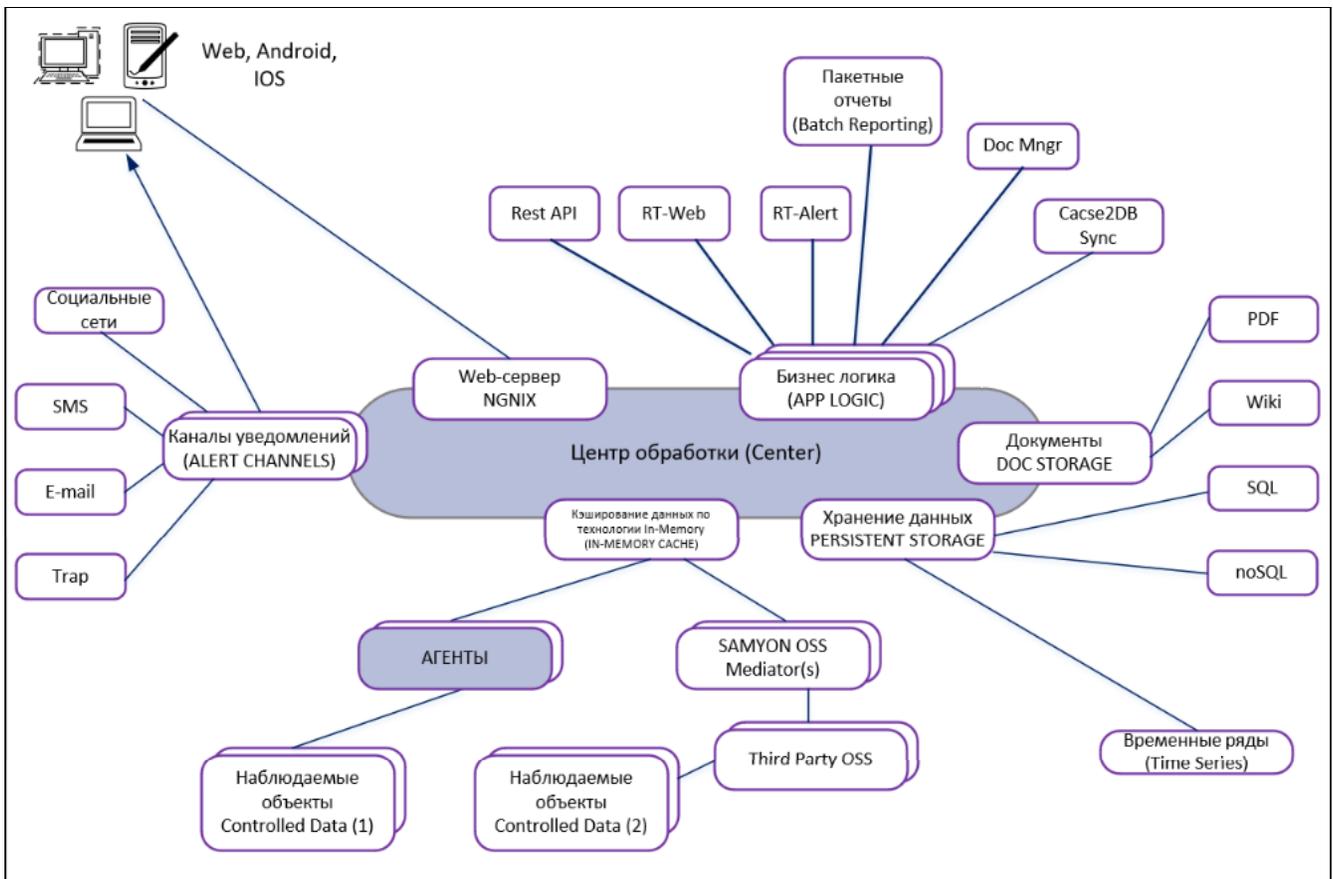


Рисунок 2. Подробная схема архитектуры системы СКАЛА-Р МОНИТОРИНГ

Уровень подготовки администратора для работы с ПО «СКАЛА-Р МОНИТОРИНГ»

Администратор платформы СКАЛА-Р МОНИТОРИНГ должен иметь опыт работы с одной из ОС: MS Windows (XP/Vista/7/8/10), Ubuntu Linux, CentOS Linux, Debian Linux и Oracle 5.0, Raspberry Pi (Raspbian OS) или Mac OS X, навык работы с браузером Chrome версии не ниже 36.0.

Квалификация администратора должна позволять:

- устанавливать и конфигурировать агент системы;
- настраивать и конфигурировать сервер;
- решать прикладные задачи мониторинга.

Технические требования к оборудованию

Требование к клиентскому оборудованию

Для комфортной работы с системой рекомендуются следующие параметры:

- Размер экрана от 20 дюймов;
- Объем свободной оперативной памяти от 512 Мб;
- Скорость интернет – соединения от 2 Мбит/с;
- Устройства воспроизведения при использовании звуковых оповещений.

Технические требования к серверному оборудованию

Система может быть установлена на выделенных аппаратных или виртуальных мощностях.

Для работы сервера системы СКАЛА-Р МОНИТОРИНГ рекомендуется следующая конфигурация:

- 64-bit OS:
 - Ubuntu Linux 10.04 / 12.04 / 14.10 / 16.04;
 - Red Hat Enterprise Linux 5.5+ / 6 / 7;
 - SUSE Linux Enterprise 11 / 12;
- CPU – 2.4 GHz / quad core;
- RAM – от 4 GB;
- HDD – от 24 GB.

Для работы агентов системы рекомендуется следующая конфигурация:

- OS с Java 7/8;
- CPU – 2 GHz single core;
- RAM – 1 GB;
- HDD – OS + 2 GB.

Установка ПО «СКАЛА-Р МОНИТОРИНГ»

Установка серверной части ПО «СКАЛА-Р МОНИТОРИНГ»

Серверная часть системы СКАЛА-Р МОНИТОРИНГ поставляется в виде готового образа виртуальной машины.

Доступ к веб-интерфейсу по умолчанию: admin / 1111 (логин/пароль). Пароль можно изменить после авторизации.

По умолчанию агент инициализирует подключение к серверу по портам 6379/9092. Необходимо проверить доступность данных портов с виртуальных машин, серверов и прочих устройств, на которых будет производиться установка агента.

Настройка серверной части ПО «СКАЛА-Р МОНИТОРИНГ»

Настройка сервера системы выполняется в конфигурационном файле `/etc/saymon/saymon-server.conf`. В таблице 1 приведены описания разделов и настроек, задаваемых в данном файле.

Более подробно процедура настройки сервера описана в документе ПО «СКАЛА-Р МОНИТОРИНГ. Руководство системного администратора».

Таблица 1. Описание разделов и настроек сервера

Раздел/Параметр	Описание
Cache	В этом разделе задаются настройки сервера Redis.
cache.auth_pass	Пароль для доступа к серверу Redis.
cache.host	Адрес сервера Redis. По умолчанию - "127.0.0.1" .
cache.port	Порт сервера Redis. По умолчанию - 6379 .
Kafka	В этом разделе задаются настройки подключения к брокеру Kafka.
kafka.create_topics	Включает автоматическое создание топиков; <ul style="list-style-type: none"> · false - автоматическое создание выключено, · true - автоматическое создание включено. По умолчанию - false - автоматическое создание выключено.
kafka.fetch_latest	Включает забор последних сообщений от брокера: <ul style="list-style-type: none"> · false - забор выключен, · true - забор включен. По умолчанию - false - забор выключен.
kafka.host	Адрес брокера Kafka. По умолчанию - "localhost" .

Раздел/Параметр	Описание
kafka.logLevel	<p>Уровень логирования Kafka:</p> <ul style="list-style-type: none"> · 0 - NOTHING, · 1 - ERROR, · 2 - WARN, · 4 - INFO, · 5 - DEBUG. <p>По умолчанию - 2 - WARN.</p>
kafka.port	<p>Порт брокера Kafka.</p> <p>По умолчанию - 9092.</p>
kafka.requestTimeout	<p>Время ожидания ответа клиентом (в миллисекундах).</p> <p>По умолчанию - 30000 - 30 секунд.</p>
LDAP	<p>В этом разделе задаются параметры внешнего LDAP-сервера для авторизации.</p> <pre data-bbox="656 997 1455 1247"> "ldap" : { "url" : "ldap://127.0.0.1:389", "suffix" : "dc=example,dc=com", "login" : "cn=admin,dc=example,dc=com", "pass" : "root" } </pre>
ldap.login	<p>Логин администратора LDAP.</p>
ldap.pass	<p>Пароль администратора LDAP.</p>
ldap.suffix	<p>Корневой элемент (как правило, доменное имя организации).</p>
ldap.url	<p>Адрес LDAP-сервера.</p>
MQTT	<p>В этом разделе задаются настройки подключения к MQTT-брокеру:</p> <pre data-bbox="656 1690 1455 1876"> "mqtt" : { "broker" : "mqtt://" "username:password@localhost:1883" } </pre>
mqtt.broker	<p>Адрес и порт брокера.</p> <p>По умолчанию - "mqtt://localhost:1883"</p> <p>Для аутентификации по имени пользователя и паролю нужно указать пользовательские данные перед адресом сервера.</p>
OpenTSDB	<p>В этом разделе задаются параметры доступа к OpenTSDB.</p>
openTsdB.enabled	<p>Запись исторических данных в OpenTSDB:</p> <ul style="list-style-type: none"> · false - запись выключена; · true - запись включена. <p>По умолчанию - true - запись включена.</p>

Раздел/Параметр	Описание
openTsdB.host	Адрес хоста с OpenTSDB. По умолчанию - "localhost" .
openTsdB.port	Порт OpenTSDB. По умолчанию - 4242 .
Push notification	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"> · true - уведомления выключены; · false - уведомления включены. По умолчанию - true - уведомления выключены.
push_notification.key_path	Путь к ключу авторизации сервера Центрального Пульта на сервере Firebase. По умолчанию - "/etc/saymon/saymon-mobile-firebase-adminsdk.json" .
push_notification.on_state_change	Включает отправку уведомления при изменении состояний на случай другого источника уведомлений - MQTT-сообщений. <ul style="list-style-type: none"> · true - отправка включена, · false - отправка выключена. По умолчанию - true - отправка включена.
push_notification.timeout	Время, через которое каждому пользователю отправляется уведомление (в миллисекундах). По умолчанию - 0 - задержки нет.
push_notification.url	URL, полученный пользователем от Firebase, для принятия содержимого уведомлений с сервера. По умолчанию - https://saymon-mobile.firebaseio.com .
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"> · false - аналитика выключена; · true - аналитика включена. По умолчанию - false - 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"> · false - раскраска выключена; · true - раскраска включена. По умолчанию - false - раскраска выключена.

Раздел/Параметр	Описание
server.comet_ping_interval	Временной интервал между отправками comet-сервером сообщений (в миллисекундах). По умолчанию - 5000 - 5 секунд.
server.comet_ping_timeout	Время ожидания сообщения от comet-сервера (в миллисекундах). По умолчанию - 12000 - 12 секунд.
server.comet_port	Порт для соединения. По умолчанию - 8091 .
server.comet_secure	Включает SSL-соединение: <ul style="list-style-type: none"> · false - соединение выключено; · true - соединение включено. По умолчанию - false - 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"> · false - функционал выключен; · true - функционал включен. По умолчанию - false - функционал выключен.
server.debug	Включает debug-режим для логирования в файл /var/log/saymon/saymon-server.log: <ul style="list-style-type: none"> · false - режим выключен; · true - режим включен. По умолчанию - false - 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 - 120000 - 2 минуты.
server.default_state_id	Состояние объекта по умолчанию. По умолчанию - 7 - "Нет данных".
server.discovery_parent_id	ID объекта, в котором появляются найденные агентами сетевые устройства. По умолчанию - "1" .
server.event_log_max_bytes	Размер записей консоли в mongoDB, при достижении которого происходит ротация данных (в байтах). По умолчанию - "1 G" - 1 гигабайт.
server.extension_path	Путь к директории с серверными расширениями.

Раздел/Параметр	Описание
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 history_update_period равен 0 .
server.history_update_period	Интервал записи исторических данных (в миллисекундах). 0 - немедленная запись пришедших значений. По умолчанию - 120000 - 2 минуты.
server.notification_buffering_period	Период ожидания для сбора сообщений о смене состояний объектов и отправки группового уведомления (в миллисекундах). По умолчанию - 0 - буферизация отключена.
server.retain_expired_stat	Включает хранение последних полученных данных после их устаревания: <ul style="list-style-type: none"> · false - хранение выключено; · true - хранение включено. По умолчанию - false - хранение выключено.
server.script_trigger_timeout	Максимальное время выполнения триггера (в миллисекундах). По умолчанию - 30000 - 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"> · false - запись выключена; · true - запись включена. По умолчанию - false - запись выключена.
server.stat_local_timestamp_field_name	Имя поля, где передается время, с которым нужно сохранять данные в OpenTSDB. По умолчанию - "localTimestamp" .
server.stat_scan_period	Период проверки актуальности пришедших данных (в миллисекундах). По умолчанию - 3000 - 3 секунды.
server.voice_call_script	Путь до скрипта, осуществляющего голосовой вызов.

Раздел/Параметр	Описание
server.email	<p>В этом подразделе задаются параметры доступа к почтовому серверу:</p> <pre data-bbox="657 370 1453 935"> "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"> · true - отправка выключена; · false - отправка включена. <p>По умолчанию - true - 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 - 1000 - 1000 символов.</p>
server.email.auth	Данные для аутентификации пользователя.
server.email.auth.user	Логин пользователя.
server.email.auth.pass	Пароль пользователя.
server.email.host	Адрес почтового сервера.
server.email.port	Порт почтового сервера.
server.email.secure	<p>Включает использование TLS при подключении к серверу:</p> <ul style="list-style-type: none"> · false - использование TLS выключено; · true - использование TLS включено. <p>Значение по умолчанию зависит от порта.</p>
server.email.service	<p>Встроенный в коннектор набор служб.</p> <p>При наличии задаёт host, port, secure автоматически.</p> <p>По умолчанию - "Gmail".</p>

Раздел/Параметр	Описание
server.user	<p>В этом подразделе задаются параметры пользователей:</p> <pre data-bbox="656 338 1455 876"> "user" : { "auth_enabled": "true", "new_user_access": "all", "lang_default" : "ru", "template": { "permissions": ["manage-objects", "view-section-stat"], "objectPermissions": { "include": [], "exclude": ["5fb643ddf277b96c8401119b", "5f8dc28407e86603bfe281f8"] } }, "usersRoot": "5800d9aaac7bf0f90d3d520e" } </pre>
server.user.auth_enabled	<p>Включает самостоятельную регистрацию для пользователей:</p> <ul data-bbox="656 956 1081 1037" style="list-style-type: none"> · false - регистрация выключена; · true - регистрация включена. <p>По умолчанию - false - регистрация выключена.</p>
server.user.lang_default	<p>Язык пользователей по умолчанию:</p> <ul data-bbox="656 1185 922 1311" style="list-style-type: none"> · "en" - английский; · "it" - итальянский; · "ru" - русский. <p>По умолчанию - "en" - английский.</p>
server.user.new_user_access	<p>Права доступа к объектам для нового пользователя:</p> <ul data-bbox="656 1459 1195 1540" style="list-style-type: none"> · "all" - есть доступ ко всем объектам; · "not" - нет доступа ни к одному объекту. <p>По умолчанию - "all" - доступ ко всем объектам.</p>
server.user.template	<p>Шаблон прав нового пользователя.</p> <p>Данные параметры применяются только для пользователей, регистрирующихся самостоятельно.</p>
server.user.template.permissions	<p>Список прав доступа, доступных пользователю по умолчанию.</p>
server.user.template.objectPermissions	<p>Права пользователя на доступ к объектам.</p>
server.user.template.objectPermissions.include	<p>Список идентификаторов объектов, к которым пользователю по умолчанию доступ разрешён.</p>
server.user.template.objectPermissions.exclude	<p>Список идентификаторов объектов, к которым пользователю по умолчанию доступ запрещён.</p>
server.user.usersRoot	<p>Идентификатор корневого объекта для создаваемых новым пользователем объектов.</p> <p>По умолчанию - "1".</p>

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 /var/log/saymon/saymon-server.log: <ul style="list-style-type: none"> · false - режим выключен; · true - режим включен. По умолчанию - false - режим выключен.
resource.server.file_storage_dir	Путь к директории для хранения документов, прикрепляемых к объектам. По умолчанию - "/var/saymon/resources" .
resource.server.ip_address	Адрес Resource-сервера. По умолчанию - "127.0.0.1" .
resource.server.port	Порт Resource-сервера. По умолчанию - 8092 .
Rest_server	В этом разделе задаются параметры REST-сервера.
rest.server.base_url	Путь к API. По умолчанию - "/api" .
rest.server.colorize_log	Включает цветную раскраску лога: <ul style="list-style-type: none"> · false - раскраска выключена; · true - раскраска включена. По умолчанию - false - раскраска выключена.
rest.server.debug	Включает debug-режим для логирования в файл /var/log/saymon/saymon-server.log: <ul style="list-style-type: none"> · false - режим выключен; · true - режим включен. По умолчанию - false - режим выключен.
rest.server.document_download_url	URL к файлам, сохраненным в \$document_storage_dir . По умолчанию - http://localhost/node/api/docs
rest.server.ip_address	Адрес хоста для запуска REST-сервера. По умолчанию - "127.0.0.1" .
rest.server.num_workers	Число процессов для загрузки данных. По умолчанию - 1 .
rest.server.port	Порт REST-сервера. По умолчанию - 8090 .
rest.server.public_url	Адрес для доступа к web-интерфейсу из уведомлений.

Раздел/Параметр	Описание
rest.server. update_download_url	Путь к файлу для обновления агента. По умолчанию - http://localhost/node/api/agents/update.

Установка агента мониторинга ПО «СКАЛА-Р МОНИТОРИНГ»

Агент может быть установлен на различные операционные системы, поддерживающие Java SE версии 6, 7 и 8.

Если по каким-то причинам установка агента нежелательна или невозможна, может быть настроен пассивный (безагентный) мониторинг системы. Актуальную версию агента можно получить по запросу в компанию-поставщика или по ссылке ниже по тексту.

Автоматическая установка агента

Создайте объект класса **SAYMON AGENT**, нажав кнопку  на панели инструментов;

Наведите на созданный объект указатель мыши и нажмите появившуюся кнопку  ("Настройки агента").



Рисунок 3 - Панель инструментов

В появившемся окне скопируйте текст из поля **Команда для установки агента**.

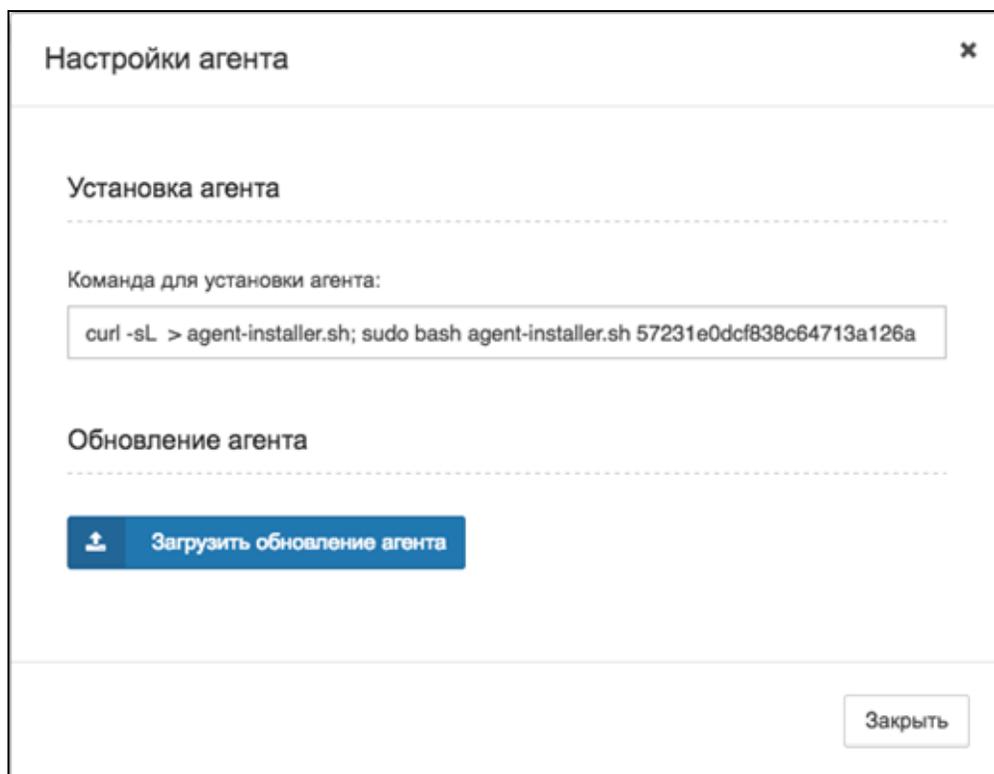


Рисунок 4 - Окно настройки агента

Выполните данную команду на необходимом сервере.

⚠ Примечание

Последние 24 символа команды - уникальный ID агента.

В течение 30 секунд агент скачивается и устанавливается.

Если установка произведена корректно, состояние объекта должно быть отражено зелёным цветом

Saymon Agent

По умолчанию агент устанавливается в одну из следующих директорий:

- Windows 32bit: C:\Program Files\SAYMON Agent;
- Windows 64bit: C:\Program Files (x86)\SAYMON Agent;
- *nix: /opt/Saymon-agent.

Установка агента на различные Unix/Linux системы

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).

Скачать инсталлятор агента для ОС Linux 64 bit или Linux 32 bit можно по ссылке:

https://saymon.info/support_ru/downloads_en/

1) Создайте системного пользователя, из-под которого будет запускаться агент. Например, для RHEL и CentOS это будет выглядеть так:

```
sudo useradd -r saymon
```

2) Скопируйте инсталлятор агента в домашнюю директорию пользователя *saymon* на целевом сервере и сделайте файл исполняемым:

- Для 64-битной ОС:

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

- Для 32-битной ОС:

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

3) Запустите скрипт:

- Для 64-битной ОС:

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

- Для 32-битной ОС:

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

4) Выполните дальнейшие инструкции в терминале.

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

6) При необходимости можно включить централизованное логирование агента через SYSLOG в файле /opt/saymon-agent/conf/logback-upstart.xml.

Log-файлы агента записываются в директорию /var/log/saymon.

7) Узнать состояние, запустить, перезапустить и остановить службу агента можно следующими командами:

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

Установка агента на Mac OS X

- 1) Создайте системного пользователя **Saymon agent**, из-под которого будет запускаться агент.
- 2) Скачайте архив **Mac OS X 64 bit** и распакуйте его в папку **/opt**.
- 3) Отредактируйте файл настроек **saymon-agent/conf/agent.properties**.
- 4) Создайте папку для хранения лог файлов:

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

5) При необходимости включите централизованное логирование агента через SYSLOG в **saymon-agent/conf/logback.xml**.

6) Добавьте права на запуск агента:

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

7) Запустите агента:

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

Лог файлы агента фиксируются в директории **/opt/saymon-agent/log**.

Установка агента на Windows

1. Скачайте инсталлятор **Windows 64 bit** или **Windows 32 bit**.
2. Откройте папку, в которой находится скачанный файл и запустите его от имени администратора.
3. Выберите место расположения папки с установкой (например, **C:\Program Files (x86)\SAYMON Agent**) и нажмите на кнопку **.**
4. Когда установка будет завершена, нажмите на кнопку **.**
5. Зайдите в папку установки агента (например, **C:\Program Files (x86)\SAYMON Agent**) и откройте файл **conf\agent.properties**.
6. Отредактируйте файл в соответствии с Конфигурацией агента. Сохраните файл в меню **Файл - Сохранить**
7. Запустите службу SAYMON Agent в меню **Пуск - Панель управления - Администрирование - Службы**.
8. После запуска службы SAYMON Agent зайдите в веб-интерфейс системы, где Вы сможете приступить к настройке мониторинга.

Лог файлы агента пишутся в папку установки агента (например, **C:\Program Files (x86)\SAYMON Agent\log**).

Настройка агента

Настройка агента выполняется в файле **.../saymon-agent/conf/agent.properties**. В таблице 2 приведены описания разделов и настроек, задаваемых в данном файле.

Более подробно процедура настройки агента описана в документе ПО «СКАЛА-Р МОНИТОРИНГ. Руководство системного администратора».

Таблица 2 - Описание разделов и настроек агента

Параметр	Описание
agent.discoveryEnabled	<p>Включает автоматический поиск агентом сетевых устройств. false - автоматический поиск выключен, true - автоматический поиск включен.</p> <p>По умолчанию - false - автоматический поиск выключен.</p> <p>Задать родительский объект для обнаруженных устройств можно с помощью параметра discovery_parent_id в разделе Server конфигурационного файла сервера.</p>
agent.id ¹	<p>Уникальный идентификатор объекта класса Saymon Agent в web-интерфейсе системы.</p> <p>При указании agent.id²=0 на сервере будет создан новый объект класса Saymon Agent, и ID этого объекта будет передан агенту. На хосте с агентом будет создан файл .../saymon-agent/storage/registration.json, где будет записан данный идентификатор.</p> <p>Чтобы заново инициировать процедуру получения ID, к примеру, если необходимо подключить агента к другому серверу, достаточно удалить файл registration.json и перезапустить агента.</p>
agent.optimizedThreadManagement	<p>Включает режим оптимизации использования потоков агентом.</p> <p>false - режим оптимизации выключен, true - режим оптимизации включен.</p> <p>По умолчанию - false - режим оптимизации выключен.</p>
agent.scriptsEnabled	<p>Включает выполнение агентом скриптов с указанным текстом;</p> <p>false - выполнение скриптов выключено, true - выполнение скриптов включено.</p> <p>По умолчанию - true - выполнение скриптов включено.</p>
agent.snmpTrapEnabled	<p>Позволяет использовать агента в качестве получателя трапов.</p> <p>false - получение трапов агентом выключено, true - получение трапов агентом включено.</p> <p>По умолчанию - false - получение трапов агентом выключено.</p> <p>О возможностях работы с SNMP-Trap читайте в разделах Настройка мониторинга и проверок и Журнал событий.</p>
agent.snmpTrapListenPort	<p>Порт для получения трапов.</p> <p>По умолчанию - 1162.</p>
agent.snmpTrapReceiverThreadPoolSize	<p>Количество одновременных потоков для получения трапов.</p> <p>По умолчанию - 4.</p>
server.host	<p>Адрес сервера системы.</p> <p>По умолчанию - 127.0.0.1</p>

¹ http://agent.id

² http://agent.id

Параметр	Описание
server.password	Пароль Redis.
server.port	Порт Kafka (9092) или Redis (6379), по которому осуществляется подключение агентов к серверу.

Термины и определения

Термин/Сокращение	Определение
API	(Application Programming Interface) — программный интерфейс, позволяющий связывать между собой различные приложения
CPU	(central processing unit, CPU) — центральное обрабатывающее устройство, часто просто процессор
HDD	(hard disk drive, жесткий диск) — это запоминающее устройство, основанное на принципе магнитной записи. На сегодняшний день HDD является одним из наиболее распространенных типов памяти, как и flash memory
host	любое устройство, предоставляющее сервисы формата «клиент-сервер» в режиме сервера по каким-либо интерфейсам и уникально определенное на этих интерфейсах
HTTP	(HyperText Transfer Protocol) протокол передачи данных, предназначенный для передачи гипертекстовых документов (документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам)
HTTP-сервер	сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными
ID	уникальный идентификатор — информация, которая может идентифицировать субъект (простыми словами — это номер или имя: число или строка символов)
LDAP	(Lightweight Directory Access Protocol) — облегченный протокол доступа к каталогам, открытый стандартизированный протокол, применяемый для работы с различными реализациям служб каталогов, в том числе и Active Directory. Сервер службы каталогов (далее — LDAP-сервер) в основном применяются для централизованного хранения учетных записей. LDAP-сервер представляет собой иерархическую базу данных
MySQL	сервер баз данных, используемый в разных приложениях. SQL означает язык структурированных запросов - (S)tructured (Q)uery (L)anguage, который MySQL использует для коммуникации с другими программами
port	целое неотрицательное число, записываемое в заголовках протоколов транспортного уровня сетевой модели OSI (TCP, UDP, SCTP, DCCP)
RAM	(Random Access Memory, с англ. — «Запоминающее устройство с произвольным доступом») — оперативная память, один из видов компьютерной памяти
SNMP	(англ. Simple Network Management Protocol — простой протокол сетевого управления) — стандартный интернет-протокол для управления устройствами в IP-сетях на основе архитектур TCP/UDP ³

³ <https://ru.wikipedia.org/wiki/UDP>

SNMP-Trap		это тип единицы данных протокола SNMP (PDU). В отличие от других типов PDU, с помощью SNMP trap агент может отправить не запрошенное сообщение менеджеру для уведомления о важном событии
SQL		(от англ. Structured Query Language — «язык структурированных запросов») — декларативный язык программирования ⁴ , применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных
SSL-соединение		протокол безопасности, создающий зашифрованное соединение между веб-сервером и веб-браузером
SYSLOG		(англ. system log — системный журнал) — стандарт отправки и регистрации сообщений о происходящих в системе событиях (то есть создания событийных журналов), использующийся в компьютерных сетях, работающих по протоколу IP
URL		(Uniform Resource Locator) — система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса
Web-сервер		HTTP-сервер, обслуживающий запросы клиента. Он отдаёт статические элементы web-интерфейса (HTML, JavaScript, CSS), а также проксирует запросы от клиента к REST-серверу. В качестве web-сервера используется NGINX
Авторизация		(англ. authorization — разрешение, уполномочивание) — предоставление определенному лицу или группе лиц прав на выполнение определенных действий, а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий
Аутентификация		процедура проверки подлинности, например проверка подлинности пользователя путем сравнения введенного им пароля с паролем, сохраненным в базе данных
Буферизация		метод организации обмена, в частности, ввода и вывода данных в компьютере и других вычислительных устройств, который подразумевает использование буфера для временного хранения данных
Виртуальная машина		(VM, от англ. virtual machine) – программная и/или аппаратная система, эмулирующая аппаратное обеспечение компьютера (guest — гостевая платформа) и исполняющая программы для guest-платформы на host-платформе (host — хост-платформа, платформа-хозяин) или виртуализирующая некоторую платформу и создающая на ней среды, изолирующие друг от друга программы и даже операционные системы
Документы (Хранилище документов)		структура директорий в файловой системе, предназначенная для хранения загруженных документов
Инцидент		нежелательные события или группа событий, которые привели или могут привести к нарушению функционирования информационного ресурса

⁴ https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F

Клиент	тонкий web-клиент системы и клиенты для мобильных операционных систем Android и iOS
Конфигурационный файл	бинарный или текстовый файл, содержащий информацию, которая определяет поведение приложения, компьютера или сетевого устройства
Логин	идентификатор пользователя (учётной записи) в компьютерных системах
Объект	физическое устройство (сервер, процессор, маршрутизатор), программный модуль (база данных, Web-сервер) или более высокоуровневые объекты, относящиеся к бизнес-процессам (услуга, платформа), в пределах которого осуществляется мониторинг с целью сбора информации, её анализа и контроля за состоянием объекта
Ротация данных	полная очистка базы данных, на которую происходит переключение
Сервер	централизованный сервер, на котором хранится и анализируется информация, полученная от агентов, а затем передаётся клиенту. Также здесь содержится информация об учётных записях пользователей
СУБД	система управления базами данных - комплекс программ, позволяющих создать базу данных и манипулировать данными (вставлять, обновлять, удалять и выбирать)
Центр обработки (Center)	логический компонент, объединяющий в себе хранилище данных, кэш, REST-сервер, HTTP-сервер и бизнес-логику