

# РУКОВОДСТВО АДМИНИСТРАТОРА ВИЗИОН

Документ	Описание	Ссылка на PDF
Руководство администратора	v1.5	<a href="#">PDF</a>

## 1 ОБЩИЕ СВЕДЕНИЯ

### 1.1 Наименование программы

Полное наименование: Система мониторинга ПАК семейства Скала^р: ПО «Скала^р Визион».

Краткое наименование: Визион.

### 1.2 Область применения

Основным назначением ПО «Скала^р Визион» (далее Визион) является предоставление возможности контроля жизненного цикла ПАК (машин) производства Скала^р, обслуживаемых персоналом, к квалификации которого не предъявляется серьезных требований. Осуществляется процесс мониторинга программных компонентов ПАК, сбор информации о количестве и составе управляемых объектов, метрик. Обеспечивает оповещение о событиях при сбое работы объектов мониторинга.

### 1.3 Уровень подготовки администратора

Администратор должен обладать следующими навыками:

- установка, настройка системного программного обеспечения в ОС Linux;
- просмотр и редактирование файлов с использованием команд ls, cat, less, vim, nano.

### 1.4 Системные требования

Параметр	Требование
CPU рекомендованное	
	<b>Визион.Сервер</b> от 8 ядер
	<b>Визион.Прокси</b> от 2 ядер

	<b>Визион.Агент</b> от 1 ядра
<b>RAM рекомендованное</b>	
	<b>Визион.Сервер</b> от 32 Гб;
	<b>Визион.Прокси</b> от 16 Гб
	<b>Визион.Агент</b> от 4 Гб
<b>ROM рекомендованное</b>	
	<b>Визион.Сервер</b> - в зависимости от количества ПАК на контуре и требуемой глубины хранения метрики, от 600 Гб SSD на каждый ПАК
	<b>Визион.Прокси</b> в зависимости от требуемой глубины кеширования данных от 20 Гб SSD
	<b>Визион.Агент</b> в зависимости от требуемой глубины кеширования данных от 5 Гб SSD
<b>Поддерживаемые ОС</b>	
	ОС АЛЪТ 8 СП СЕРВЕР р9, р10
	Astra Linux Special Edition 1.7.3 (Воронеж)
	РЕД ОС 7.3 (Муром)
<b>Поддерживаемые браузеры</b>	
	Google Chrome версий от 116 и выше
	Firefox начиная с 60.xx версии

Примечание:

Основной потребитель ресурсов на стороне сервера Визиона - VictoriaMetrics

Оценка требований к ресурсам (при периоде извлечения данных каждые 15 секунд для всех временных рядов):

- 1 Гб ОЗУ на 1 млн активных временных рядов;

- ядро ЦП на каждые 300 тыс. вставленных точек данных в секунду;
- менее байта на точку данных (среднем 0.2-0.5 Б/точка);
- входящий трафик 100 байт на каждую принятую точку данных;
- на долю служебных метрик самого мониторинга приходится около 7 тыс. временных рядов.

На примере S3: на не нагруженном стенде из 4 узлов хранения и 2 балансировщиков, получается порядка 30 тыс. активных временных рядов, что примерно соответствует 120 Кб на одну точку времени или 700 Мб в день при периоде извлечения данных каждые 15 секунд для всех временных рядов.

Данные оценки носят предварительный характер и могут меняться как в большую, так и меньшую сторону.

## 2 СОСТАВ КОМПОНЕНТОВ

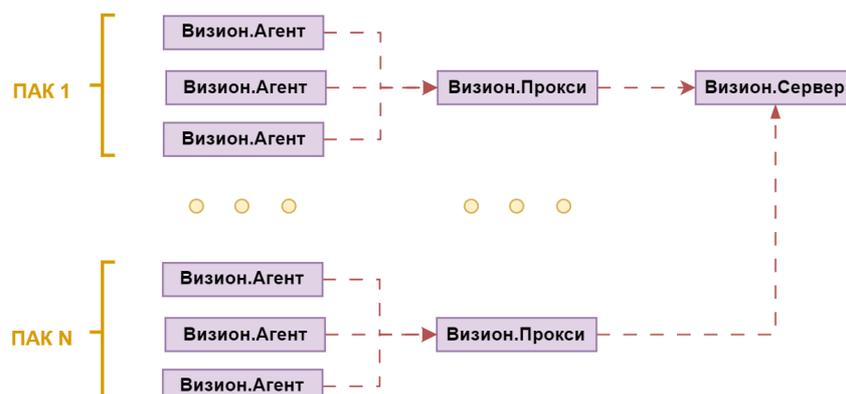
Для развертывания мониторинга на целевой инфраструктуре поставляется архив, содержащий в себе компоненты программного обеспечения:

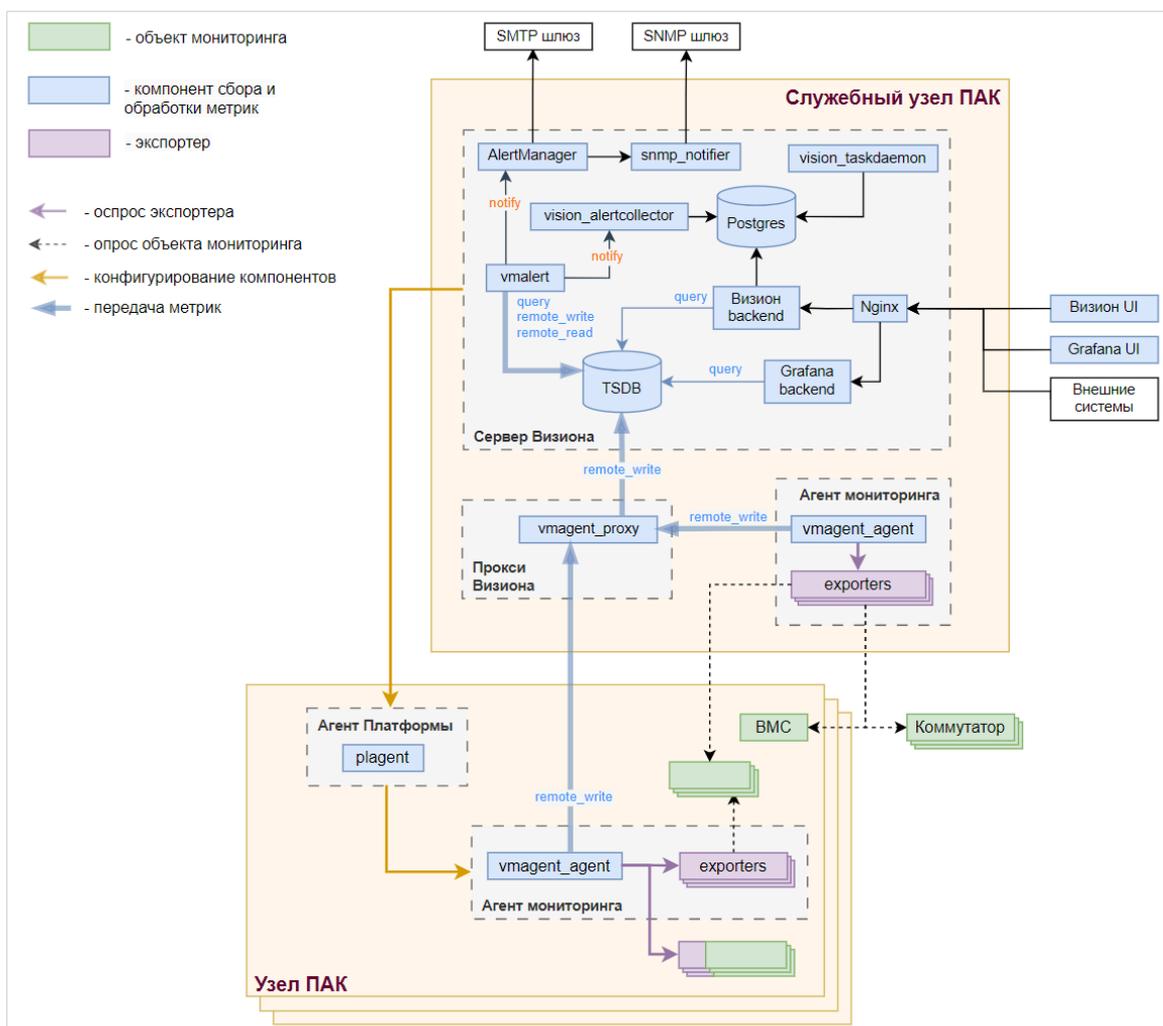
- **Визион.Сервер** - предоставляет функционал единой точки доступа к управлению мониторингом; устанавливается в одном экземпляре на контур (совокупность ПАК);
- **Визион.Прокси** - промежуточное звено, аккумулирует метрики с узлов ПАК; устанавливается по одному на каждый ПАК, возможно совмещение с сервером Визиона при использовании единой точки Визиона в каждом ПАК (Машине);
- **Визион.Агент** - агент мониторинга, обеспечивающий сбор данных с узла; устанавливается на узел ПАК или на виртуальную машину
- **Агент Платформы** - сервис, позволяющий серверу Визиона конфигурировать агенты мониторинга по HTTPS. Агент необходимо развернуть на все узлы, на которые, впоследствии, будут установлены агенты мониторинга.

Все компоненты должны находится в одной сети.

Оптимальным решением является выделение под Визион.Сервер и Визион.Прокси отдельных виртуальных серверов в рамках узлов управления Машин.

### Архитектурная схема





Для эффективного мониторинга ПАК важно, чтобы все узлы имели синхронизированное текущее время. Неправильные временные метки могут привести к неточным или вводящим в заблуждение данным мониторинга, что затруднит выявление и устранение проблем.

## 2.1 Визион.Сервер

В состав Визион.Сервера входят следующие компоненты:

- **Визион backend** - бекенд, обеспечивающий API и осуществляющий управление и координацию всех сервисов;
- **PostgreSQL** - служебная база данных;
- **VictoriaMetrics** - база данных временных рядов для хранения метрик;
- **VMAlert** - компонент, предназначенный для генерации оповещений на основе данных в Victorimetrics и правил алертинга (триггеров);
- **AlertCollector** - компонент, регистрирующий оповещения в служебную БД;

- **Grafana backend** - компонент, обеспечивающий альтернативную визуализацию собираемых метрик;
- **AlertManager** - компонент, обеспечивающий обработку, группировку и отправку оповещений (поддерживает SMTP и веб-хуки);
- **SNMP\_notifier** - вспомогательный компонент, предназначенный для отправки оповещений с помощью протокола SNMP (Simple Network Management Protocol);
- **TaskDaemon** - компонент, отвечающий за выполнение фоновых задач (например, развертывание агентов и их плагинов);
- **Nginx** - прокси-сервер, обеспечивающий получение и перенаправление запросов на компоненты Визион.Сервера.

## 2.2 Визион.Прокси

В состав Визион.Прокси входит единственный компонент - **VMagent**, обеспечивающий перенаправление поступающих метрик в Визион.Сервер.

## 2.3 Визион.Агент

В состав Визион.Агента входят следующие компоненты:

- **VMagent** - компонент, обеспечивающий сбор метрик с плагинов
- **Плагины** - дополнительно устанавливаемые модули (экспортеры и параметры скрепинга метрик), расширяющие состав, собираемых системой мониторинга, метрик.

## 2.4 Агент Платформы

Агент Платформы представлен единственным компонентом - **PLagent**.

В состав дистрибутива опционально включается плейбук развертывания агента Платформы, необходимый для взаимодействия сервера Визиона с сервером для конфигурирования агентов мониторинга. Данный агент необходимо развернуть на все узлы, на которые в последствии будут установлены Визион.Агент и Визион.Прокси.

В данный момент установка возможна на следующих ОС:

- Alt Linux Server 8 СП р9 (с9f2)
- Alt Linux Server 8 СП р10 (с10f1)
- Astra Linux 1.7.3 (Воронеж)
- РЕДОС 7.3

На целевых системах должны присутствовать следующие пакеты:

- freeipmi (если предполагается присутствие плагина ipmi на узле, как правило, это ВМ Визиона на служебном узле)
- lsblk (для плагина utlz\_exporter, необходима поддержка ключей -o и -J)

- sudo (> 1.6.9p15)

Перед началом развертывания агентов необходимо обеспечить сетевую связность между узлами и доступность серверов с хоста Визиона, а также наличие доступа к ним по протоколу SSH (порт 22).

## 2.5 Состав сервисов

Сервис ОС	Компонент	Порт	Протокол
postgresql	Визион.Сервер	5432	HTTPS
vision_core	Визион.Сервер	8088	HTTPS
nginx	Визион.Сервер	443, 80, 8092	HTTPS
grafana	Визион.Сервер	3000	HTTPS
victoriametrics_s t	Визион.Сервер	8428	HTTPS
vmalert_st	Визион.Сервер	8880	HTTPS
vision_alertcolle ctor	Визион.Сервер	10111	HTTPS
alertmanager	Визион.Сервер	9093	HTTPS
snmp_notifier	Визион.Сервер	9464	HTTPS
vision_vmagent_ проху	Визион.Прокси	8430	HTTPS
vision_vmagent_ agent	Визион.Агент	7529	HTTPS
plagent	Агент Платформы	7550, 7551	HTTPS, gRPC

## 2.6 Плагины

Плагины Визиона представляют собой механизм, с помощью которого пользователь может расширять функциональность системы мониторинга. Они дают возможность гибко настраивать процессы мониторинга в соответствии с конкретными потребностями и целями.

Каждому плагину соответствует свой тип объекта мониторинга, с которого извлекается соответствующее ему множество метрик.

Извлекаемые метрики адаптированы под характеристики и особенности объекта мониторинга. Например, для мониторинга серверов могут быть доступны метрики использования памяти и дискового пространства, в то время как для сетевых устройств - метрики пропускной способности, задержек и ошибок передачи данных.

Плагину может соответствовать комплиментарный бинарный файл (экспортер), непосредственно осуществляющий сбор метрик. При добавлении в Визион.Агент такого плагина, он автоматически устанавливается согласно местоположению агента и запускается как сервис операционной системы.

В случае, если плагин не использует экспортер, он является плагином-скреппером, сбор метрик осуществляется непосредственно с эндпоинта объекта мониторинга.

В состав Визиона входят следующие плагины:

<b>Название</b>	<b>Мониторинг</b>	<b>ПАК</b>	<b>Сервер/ВМ**</b>	<b>Сервис ОС</b>	<b>Порт***</b>
node_exporter	ОС сервера	все	все	vision_node_exporter	9101
ipmi_exporter	ВМС (IPMI)	все	ВМ с Прокси	vision_ipmi_exporter	9290
snmp_exporter	коммутатор (SNMP)	все	ВМ с Прокси	vision_snmp_exporter	9116
utlz_exporter	утилизация сервера	все	все серверы	vision_utlz_exporter	9092
utlz_exporter_v0*	утилизация сервера	все	все серверы	vision_utlz_exporter_v0	17070
one_exporter	OpenNebula	МВ.ВК	узел с OpenNebula frontend	vision_one_exporter	9621
ceph_scraper	Ceph	МВ.ВК	ВМ с Прокси		

ha_cluster_exporter	Pacemaker/Corosync	МБД.П	узел кластера	vision_ha_cluster_exporter	9664
postgres_exporter_agent	Postgres	МБД.П	узел кластера	vision_postgres_exporter_local	9187
postgres_exporter_multi	Postgres	МБД.П	узел кластера или узел с Прокси	vision_postgres_exporter_multi	9188
sql_exporter	Postgres	МБД.П	узел кластера или узел с Прокси	vision_sql_exporter	9399
spectrum_agent_scraper	агент Спектра	МБД.П	узел кластера		
supv_exporter	сервисы СУПВ	МВ.С/ВРМ	ВМ с Прокси	vision_supv_exporter	9179
haproxy_scraper_local	HAProxy	МХД.О	узлы балансировки		
s3gateway_compression_server_scraper	шлюз S3	МХД.О	узлы балансировки		
s3gateway_ostor_server_scraper	сервисы S3	МХД.О	узлы хранения		
nginx_exporter	Nginx	МХД.О	все серверы	vision_nginx_exporter	9113

vision_exporter	сервисы ОС	МХД.О/ МБД.П	все серверы	vision_exporter	7531
systemd_exporter	systemd юнитов	все	все серверы	vision_systemd_exporter	9110
process_exporter	процессы ОС	все	все серверы	vision_process_exporter	9120
kafka_exporter	Kafka	МБД.С	ВМ с Прокси	vision_kafka_exporter	9308
bash_exporter****	собственные метрики	все	все серверы	vision_bash_exporter	17055
vision_utilizer	утилизация сервера	все	все серверы	vision_utilizer	17099
vision_greenplum_sql_exporter	БД Greenplum	МБД.Г	все серверы	vision_greenplum_sql_exporter	9398
vision_clickhouse_sql_exporter	БД ClickHouse	МБД.КХ	все серверы	vision_clickhouse_sql_exporter	9397
blackbox_exporter	ICMP	все	ВМ с Прокси	vision_blackbox_exporter	9115
picodata_scraper	БД Picodata	МБД.Т	все серверы	picodata_scraper	-

\* **utlz\_exporter\_v0** по умолчанию нет необходимости устанавливать, только по запросу. Для работы ему требуется дополнительный файл конфигурации, который доставляется на хосты с помощью плейбука [playbooks/utlz\\_v0\\_config.yml](#).

\*\* На агенты каких серверов или ВМ рекомендуется добавлять плагины.

\*\*\* Порт, занимаемый экспортером по умолчанию. Порты конфигурируются в настройках плагина в интерфейсе Визиона.

\*\*\*\* Плагин `bash_exporter` устанавливается отдельно от основного дистрибутива Визиона.

## 3 УСТАНОВКА ПРОГРАММЫ

Перед началом установки необходимо убедиться в доступности узлов ПАК с хоста Визиона. Проверить доступность можно при помощи команды **ping**.

Если в выводе команды **ping** содержатся ошибки, необходимо убедиться в том, что в инсталляционной карте содержатся корректные данные, обеспечена сетевая связность между узлами и фаервол не блокирует доступ по протоколу SSH (порт 22).

В приведенных в данном руководстве примерах команд в качестве интерпретатора команд терминала используется Bash.

Все команды выполняются под привилегированным пользователем **root**. Для повышения привилегий пользователя может потребоваться использование команды **sudo**. В примерах команд опущено приглашение интерпретатора.

В данном руководстве приводятся примеры команд для версии Визиона 1.4. Номер версии используется в имени файла с архивом дистрибутива, например `vision-distrib-<version>.gz`.

На момент выполнения установки или обновления “Визиона” пользователем, версия может отличаться.

Актуальную инструкцию для сборки можно посмотреть в файле README.MD в архиве дистрибутива.

Алгоритм установки компонентов Визиона и его настройки:

1. Установка Визион.Сервер
2. Конфигурация объектов мониторинга
3. Формирование инвентори-файла
4. Установка агентов Платформы
5. Настройка плагинов

Зависимости:

- ansible >= 2.9
- sshpass
- nginx
- postgresql = 11|12

Данные пакеты и их зависимости будут установлены или обновлены при развертывании с помощью плейбуков ansible. При невозможности их автоматической установки требуется вручную установить данные пакеты. Все необходимые зависимости включены в состав дистрибутива для возможности установки Визиона без подключения к глобальной сети Internet.

## 3.1 Установка Визион.Сервер в интерактивном режиме

Дистрибутив Визиона представляет собой **tar.gz** архив. Далее описана работа с дистрибутивом.

Для установки Визиона в состав дистрибутива входит скрипт установки `setup.sh`. Установка должна выполняться на целевом узле (либо виртуальной машине) и требует привилегий суперпользователя.

1. Открыть терминал и перейти в директорию с архивом дистрибутива, например, `/root`:

```
cd /root
```

2. Распаковать архив с дистрибутивом `.tar.gz`, например:

```
tar -xzvf vision-distrib-1.4-289.tar.gz
```

3. Перейти в директорию дистрибутива, например:

```
`cd /root/vision-distrib-1.4-289
```

4. Запустить скрипт:

```
./setup.sh
```

5. Выбрать одно из действий:

- **3. Установить пакеты зависимостей из репозитория дистрибутива** в случае отсутствия пакетов в системном репозитории ОС;
- **4. Установить пакеты зависимостей из репозитория ОС.**

Будут установлены все пакеты зависимостей для соответствующих операционных систем. Так же для запуска этой операции можно вызвать команду `./setup.sh -g`

(`./setup.sh -p` в случае отсутствия пакетов в системном репозитории ОС).

При невозможности автоматической установки пакетов, требуется вручную установить пакеты из списка, содержащегося в файле

`./os_packages/<YOUR_OS>/package_list` (см. документацию на операционную систему).

По окончании установки пакетов ее лог будет сохранен в файл:

`/opt/skala-r/var/log/vision/vision_packages.log.<ДАТА_УСТАНОВКИ>`.

6. Запустить скрипт `./setup.sh` и выбрать действие “1. Установить сервер Визиона.”.

Будут установлены все необходимые компоненты, запущены требуемые сервисы.

Так же для запуска этой операции можно вызвать команду `./setup.sh -i`

В процессе установки будет предложено выбрать IP-адрес, на который в последствии будут отправляться метрики с агентов мониторинга. Как правило, он соответствует IP-адресу компонента Визион.Сервер в сети, в которой находятся все компоненты мониторинга (менеджмент-сеть).

По окончании установки лог, сформированный в процессе инсталляции, будет сохранен в файл:

`/opt/skala-r/var/log/vision/vision_install.log.<ДАТА_УСТАНОВКИ>`

## 3.2 Установка Визион.Сервер в автоматическом режиме

Поддерживается автоматическая установка с передачей IP-адреса, на который в последствии будут отправляться метрики с агентов мониторинга. Как правило, он соответствует IP-адресу сервера Визиона в сети, в которой находятся все компоненты мониторинга (менеджмент-сеть).

Установка должна выполняться на целевом сервере (виртуальной машине).

Требования при установке:

- наличие привилегии суперпользователя;
- наличие необходимых пакетов в системном репозитории ОС.

1. Открыть терминал и перейти в директорию с архивом дистрибутива, например:

```
sh cd /root
```

2. Распаковать архив с дистрибутивом, например:

```
sh tar -xzvf vision-distrib-<version>.tar.gz
```

3. Перейти в директорию дистрибутива, например:

```
sh cd /root/vision-distrib-<version>
```

4. Запустить скрипт `./setup.sh` с ключом `-s`, указав IP-адрес сервера Визиона, например:

```
sh ./setup.sh -s 192.168.1.1
```

Автоматически будут установлены пакеты зависимостей из системного репозитория ОС и непосредственно сервер Визиона (аналогично п.3.1).

## 3.3 Проверка корректности установки

### Визион.Сервер

Для проверки корректности установки:

1. Открыть терминал и перейти в директорию с установленным Визионом:

```
sh cd /opt/skala-r/vision
```

2. Запустить скрипт check\_service.sh:

```
sh ./check_service.sh
```

Критерием корректности развертывания является значение active для всех сервисов, приведенных в таблице в выводе скрипта check\_service.sh.

По завершении процесса установки должен быть доступен веб-интерфейс Визиона через браузер по его IP-адресу (вводим в адресной строке адрес вида <https://192.168.186.1>, далее выбираем пункты «Аналитические панели» и «Модуль Администратора».

## 3.4 Просмотр логов компонента

Просмотреть логи работы компонентов за интересующий период можно, используя службу журналирования логов journalctl, выполнив команду:

```
journalctl -u <имя сервиса> --since "начало интервала YYYY-MM-DD hh:mm:ss" --until "конец интервала YYYY-MM-DD hh:mm:ss"
```

Просмотреть текущие логи в режиме онлайн можно, используя службу журналирования логов journalctl, выполнив команду:

```
journalctl -fu <имя сервиса>
```

Для просмотра событий за временной период можно использовать команду:

```
journalctl -u <имя сервиса> --since "начало интервала YYYY-MM-DD hh:mm:ss" --until "конец интервала YYYY-MM-DD hh:mm:ss"
```

Например:

```
journalctl -u vision_core --since "2024-04-26 11:05:00" --until "2024-04-26 11:10:00"
```

## 3.5 Просмотр журналов Визион.Сервер

Журнал Визион.Сервера размещаются в директории `/opt/skala-r/var/log/vision/journal`, а журнал аудита - в `/opt/skala-`

```
r/var/log/vision/audit.
```

Просмотреть текущие журналы можно выполнив команды, например:

```
sh tail -n 20 /opt/skala-r/var/log/vision/journal/journal.log tail -n 20 /opt/skala-r/var/log/vision/audit/audit.log
```

## 3.6 Резервное копирование ВМ Визион

[Инструкция](#) по резервному копированию и восстановлению работоспособности ВМ Визион находится в разделе “Дополнительная документация”.

## 4 КОНФИГУРАЦИЯ ОБЪЕКТОВ МОНИТОРИНГА

После развертывания Визион.Сервер, необходимо ввести информацию об объектах мониторинга в систему. В консоли администратора мониторинга предусмотрен пользовательский интерфейс для настройки объектов. Алгоритм настройки выглядит следующим образом:

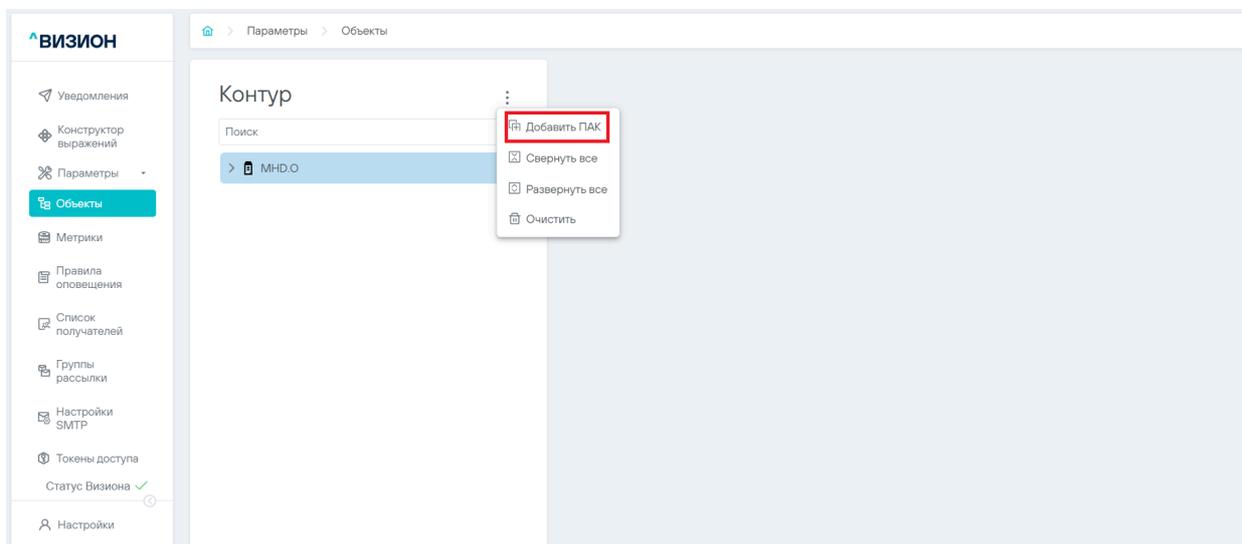
1. Перейти в веб-интерфейс Визиона, открыв Интернет-браузер, ввести в строке для адреса IP-адрес сервера Визиона <https://<vision-ip>>.
2. Перейти в раздел в боковом меню “Параметры” -> “Объекты”, в панели “Контур” нажать кнопку , выбрать “Добавить ПАК”. Ввести информацию о ПАК.
3. В случае отсутствия на добавляемом ПАК агентов Платформы, необходимо его развернуть в соответствии с разделом 6 настоящего Руководства.
4. Перейти во вкладку “Агенты” и нажать кнопку “Сконфигурировать агенты”.
5. Нажать на иконку  и включить необходимые плагины.
6. Перейти в пункт меню “Конструктор выражений”, ввести выражение `up{pak_id=" <ИМЯ_ПАК>"}` и проверить, что работают все сконфигурированные плагины (в наличии временные ряды с именем метки “job” и значениями имен агентов).

### 4.1 Подготовка к настройке

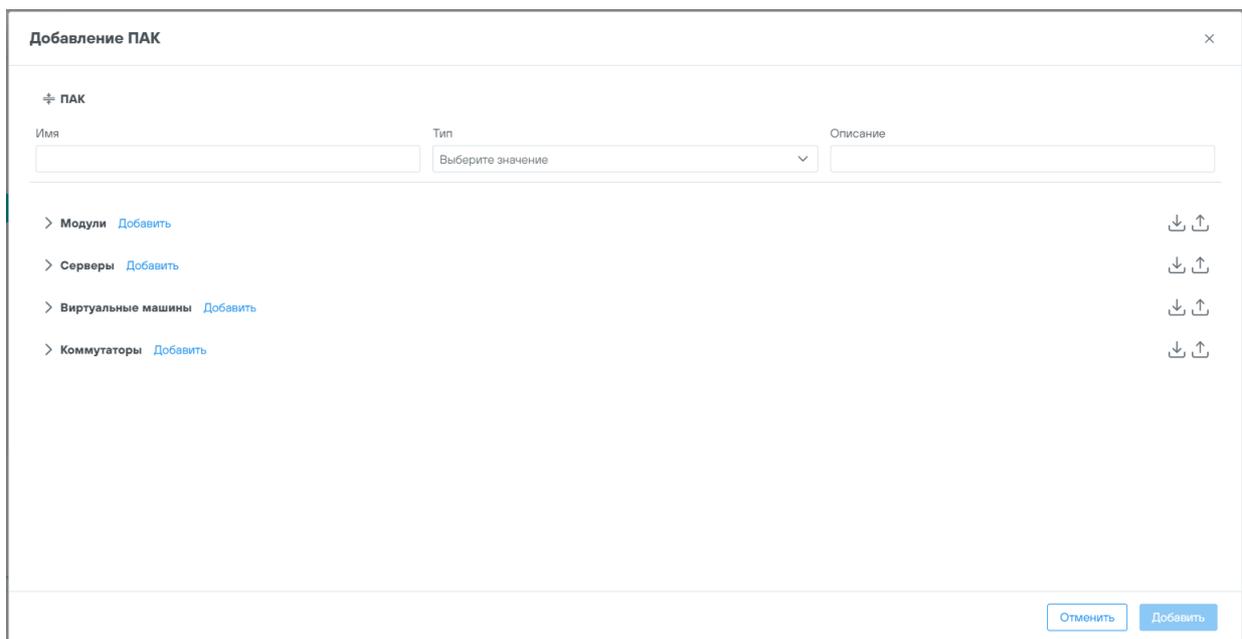
Перед настройкой требуется убедиться, что имеется инсталляционная карта с соответствующим описанием ПАК, либо все необходимые параметры и доступы для постановки объектов на мониторинг.

Перейти в веб-интерфейс Визиона, открыв Интернет браузер, и введя в строке адреса IP-адрес сервера Визиона, например, <https://<vision-ip>>.

Добавить новый ПАК: раздел в боковом меню “Параметры”->“Объекты”, в панели “Контур” нажать кнопку  и выбрать “Добавить ПАК”.



Откроется форма «Добавление ПАК». Пример интерфейса:



Требуется ввести данные во все разделы (в соответствии с инсталляционной картой) и нажать кнопку «Добавить» внизу экрана, чтобы сохранить введенную конфигурацию. В случае, если обязательные поля остались незаполненными, то при попытке сохранить они будут подсвечены цветной рамкой и соответствующим текстом и данные не будут сохранены до тех пор пока все обязательные поля не будут заполнены. Также поля будут подсвечены в случае ввода значений, которые уже используются в других ПАК, и сохранение таких данных будет невозможно.

При заполнении формы для добавления и удаления строк используются элементы управления:



При нажатии кнопки «Отменить» сохранение будет отменено и введенные данные потеряны.

Возможен альтернативный ввод параметров через импорт CSV файла. Для этого используются элементы управления:

Необходимо экспортировать настройки в CSV файл, изменить их и импортировать измененный файл.

Описание полей формы приведено в разделах ниже.

## 4.1.1 Описание функций импорта и экспорта настроек в csv-файл

В режиме создания/редактирования ПАКа имеется возможность импорта/экспорта настроек ПАКа.

В поле изменения компонентов (модули, серверы, VM, коммутаторы и т.д.) ПАКа имеются кнопки



При нажатии кнопки импорта появляется окно файлового менеджера, в котором требуется выбрать csv-файл с настройками компонента ПАКа.

Примечание: импорт осуществляется в режиме дозаписи настроек.

При нажатии кнопки экспорта осуществляется скачивание csv-файла в выбранную папку. При этом csv-файл имеет имя аналогичное имени ПАКа в формате **ПАК\_<имя\_ПАКа>**. Скачанный csv-файл содержит текущие настройки компонента ПАКа.

## 4.2 Заполнение формы «Добавление ПАК»

### 4.2.1 ПАК

- «Имя» - ввести идентификатор ПАК (в дальнейшем изменить невозможно без реинсталляции)
- «Тип» - выбрать из справочника тип ПАК в соответствии с поставкой
- «Описание» – ввести произвольный вспомогательный текст

### 4.2.2 Модуль

Заполнение информации о модулях. Тип модуля берется из инсталляционной карты. Список меняется в зависимости от предварительно выбранного типа ПАК.

- «Имя» - идентификатор модуля (в дальнейшем изменить невозможно без реинсталляции)
- «Тип» – тип модуля, в соответствии с типом ПАК. Требуется выбрать из списка
- «Описание» – ввести произвольный вспомогательный текст

Модули (1) [Добавить](#) ↓ ↑

Имя	Тип	Описание
<input type="text" value="M1"/>	<input type="text" value="Базовый модуль"/>	<input type="text" value=""/>

## 4.2.3 Сервер

Требуется ввести информацию об узлах.

- «Имя» - идентификатор сервера (в дальнейшем изменить невозможно без реинсталляции)
- «Модуль» – выбрать из заведенных ранее в пункте Модуль
- «Тип» – выбрать из справочника
- «Адрес» – адрес, по которому будет установлен агент
- «Адрес ВМС» – адрес, по которому будут производить опрос агенты мониторинга, расположенные на Визион.Прокси

Серверы (1) [Добавить](#) ↓ ↑

Имя	Модуль	Тип	Адрес	Адрес ВМС
<input type="text" value="server1"/>	<input type="text" value="M1"/>	<input type="text" value="Базовый сервер"/>	<input type="text" value="192.168.1.10"/>	<input type="text" value="192.168.1.1"/>

## 4.2.4 VM

Требуется ввести информацию о виртуальных машинах, в том числе сервисных (Визиона, Генома и т.д.).

- «Имя» - идентификатор VM (в дальнейшем изменить невозможно без реинсталляции)
- «Сервер» – принадлежность VM к физическому узлу (серверу)
- «Адрес» – адрес VM в соответствии с инсталляционной картой

Виртуальные машины (1) [Добавить](#) ↓ ↑

Имя	Сервер	Адрес
<input type="text" value="VM1"/>	<input type="text" value="server1"/>	<input type="text" value="192.168.1.15"/>

## 4.2.5 Коммутатор

Требуется ввести информацию о серверах маршрутизации (коммутаторы) из инсталляционной карты.

- «Имя» - идентификатор коммутатора (в дальнейшем изменить невозможно без реинсталляции);

- «Модуль» – принадлежность к модулю, выбрать из заведенных ранее
- «Адрес» – адрес коммутатора.

▼ Коммутаторы (1) [Добавить](#)
↓ ↑

Имя	Модуль	Адрес
switch1	M1	192.168.1.20

## 4.3 Редактирование ПАК

Редактировать ПАК: раздел в боковом меню для ПАК нажать кнопку  и выбрать “Редактировать”.

Форма редактирования ПАК:

**Редактирование ПАК** ×

---

▼ Модули (1) [Добавить](#)
↓ ↑

Имя	Тип	Описание
M1	Базовый модуль	

▼ Серверы (1) [Добавить](#)
↓ ↑

Имя	Модуль	Тип	Адрес	Адрес ВМС
server1	M1	Базовый сервер	192.168.1.11	192.168.1.12

▼ Виртуальные машины (1) [Добавить](#)
↓ ↑

Имя	Сервер	Адрес
VM1	server1	192.168.1.15

▼ Коммутаторы (1) [Добавить](#)
↓ ↑

Отменить Редактировать

В форме редактирования ПАК используются те же элементы что и в форме добавления ПАК, но нет возможности изменять имена объектов.

## 4.4 Дублирование ПАК

В случае, если требуется создать структуру ПАК, аналогичную уже существующему ПАК, можно воспользоваться функцией дублирования.

Дублировать ПАК: раздел в боковом меню для ПАК нажать кнопку  и выбрать “Дублировать”.

Форма выглядит как при добавлении ПАК, но все объекты уже созданы и остается изменить их имена и заполнить поля:

**Дублирование ПАК** ✕

---

⊞ ПАК

Имя:  Тип:  Описание:

▼ Модули (1) [Добавить](#) ↓ ↑

Имя	Тип	Описание
Дубликат_M1	Базовый модуль	

▼ Серверы (1) [Добавить](#) ↓ ↑

Имя	Модуль	Тип	Адрес	Адрес ВМС
Дубликат_server1	Дубликат_M1	Базовый сервер		

▼ Виртуальные машины (1) [Добавить](#) ↓ ↑

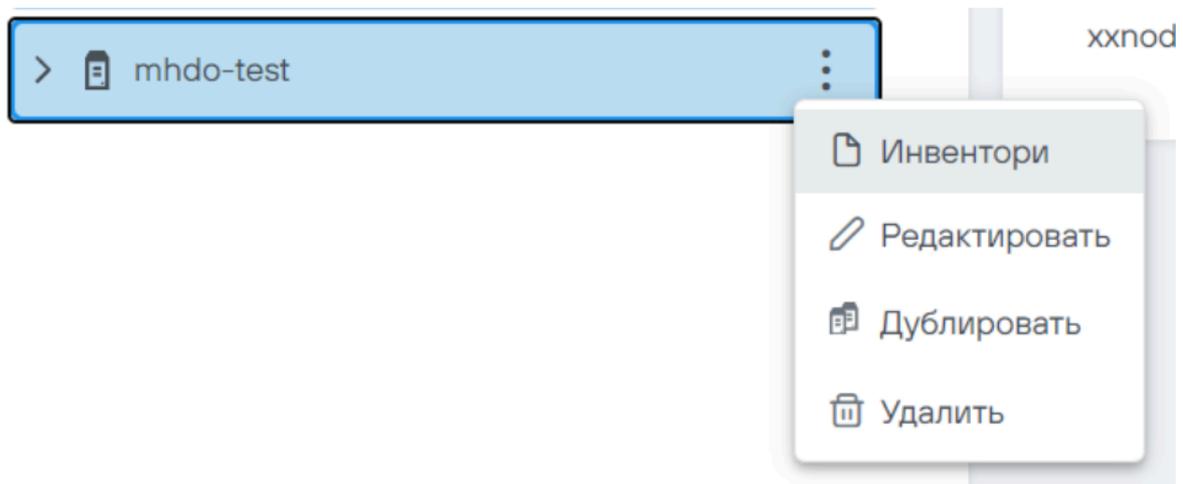
Имя	Сервер	Адрес

Возможно также удалять и добавлять объекты при необходимости.

## 5 ФОРМИРОВАНИЕ ИНВЕНТОРИ-ФАЙЛА

Для проведения сервисных операций с узлами ПАК может потребоваться так называемый инвентори-файл, описывающий перечень узлов и их технические учетные записи (ТУЗ). С использованием Ansible производится автоматизация управления конфигурациями и выполнение задач на этих узлах.

Данный файл может быть получен из веб-интерфейса Визиона для уже добавленных ПАК.



В этом случае достаточно выбрать необходимый ПАК, нажать **:**, пункт “Инвентори”. Файл-инвентори скачается, и так же будет доступен в директории `/opt/skalar/vision/server/inventory/`. Имя файла состоит из имени ПАК и даты экспорта, например, `ПАК-1_2024-06-10_10-10-10.yml`.

## 6 УСТАНОВКА АГЕНТОВ ПЛАТФОРМЫ

1. Сформировать инвентори файл для ПАК (см. п.5)
2. Отредактировать файл, например:

```
nano /opt/skala-r/vision/server/inventory/PAK-1_2024-06-10_10-10-10.yml
```

1. Заменить значения “===REPLACE===” на реальные:

- `ansible_user` и `ansible_ssh_pass` - имя и пароль учетной записи для подключения к хостам по SSH
- `ansible_become_password` - пароль учетной записи администратора для установки агента Платформы

2. Перейти в директорию дистрибутива:

```
cd ./vision-distrib-1.4-289
```

3. При необходимости, сконфигурировать установку агента Платформы при помощи файла `./plagent/_plagent_settings.yml`

4. Запустить развертывание, например:

```
ansible-playbook -i /opt/skala-r/vision/server/inventory/PAK-1_2024-04-14_23-19-46.yml ./plagent/install.yml
```

Результат работы скрипта при успешном развертывании будет иметь вид:

```
PLAY RECAP *****
PG-PRD-3-MDL-BD1-01-N-196 : ok=15  changed=6  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
PG-PRD-3-MDL-BD1-01-N-197 : ok=15  changed=6  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
PG-PRD-3-MDL-BD1-01-N-198 : ok=15  changed=6  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
PG-PRD-3-MDL-BS-01-N-199   : ok=15  changed=2  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
PG-PRD-3-MDL-BS-01-N-199-VM-GMN-192 : ok=15  changed=2  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
PG-PRD-3-MDL-BS-01-N-199-VM-SPR-191 : ok=15  changed=6  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
PG-PRD-3-MDL-BS-01-N-199-VM-VSN-200 : ok=14  changed=2  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
```

Так же в рамках автоматизации работы с агентами Платформы доступны некоторые сервисные команды. Общий вид команд для их запуска:

```
ansible-playbook -i <INVENTORY_PATH> <PLAYBOOK>
```

```
ansible-playbook -i <INVENTORY_PATH> -l <HOST> <PLAYBOOK>
```

где `<INVENTORY_PATH>` - путь до итогового файла инвентори, - выполняемое действие:

- `./plagent/install.yml` - установить агенты Платформы;
- `./plagent/status.yml` - вывести статус агентов Платформы;
- `./plagent/stop.yml` - остановить агенты Платформы;
- `./plagent/start.yml` - запустить агенты Платформы;
- `./plagent/remove.yml` - удалить агенты Платформы - имя сервера(ов) из инвентори, для которых необходимо выполнить команду, например `server1`.

## 7 НАСТРОЙКА АГЕНТОВ И УСТАНОВКА ВИЗИОН.ПРОКСИ

### 7.1 Конфигурирование агентов

Последовательность действий:

1. Перейти во вкладку “Агенты” и нажать кнопку “Сконфигурировать агенты”.

Откроется форма настроек.

2. В настройках указать:

- Расположение прокси-сервера Визиона. На выбранный узел будет установлен компонент Прокси-сервер Визиона, а также на нем требуется сконфигурировать плагины, которые обозначены типом «VM с прокси» в пункте 2.6 данного руководства.

## Расположение Визион.Прокси

i48-PG-T-1-MDL-BS-S-1-1-N-48

- Расположение агентов: по умолчанию, в форме отобразятся все серверы и VM, которые были ранее указаны в списке объектов. Для корректного расположения, на серверах должны быть предварительно установлены агенты Платформы.

**Конфигурация агентов**

Расположение Визион.Прокси  
i48-PG-T-1-MDL-BS-S-1-1-N-48

Серверы

Расположение	Статус	Плагины
i48-PG-T-1-MDL-BS-S-1-1-N-48	УСТАНОВЛЕН	Выбрано плагинов: 17
i48-PG-T-1-MDL-BS-S-1-1-N-131	СИМВЛ	

VM

Расположение	Статус
i48-PG-T-1-MDL-BS-S-1-1-N-131-202	УСТАНОВЛЕН

Плагины:

- sqj\_exporter
- ha\_cluster\_exporter
- utlz\_exporter
- postgres\_exporter\_agent
- nginx\_exporter
- spectrum\_agent\_scraper

Отмена Сохранить

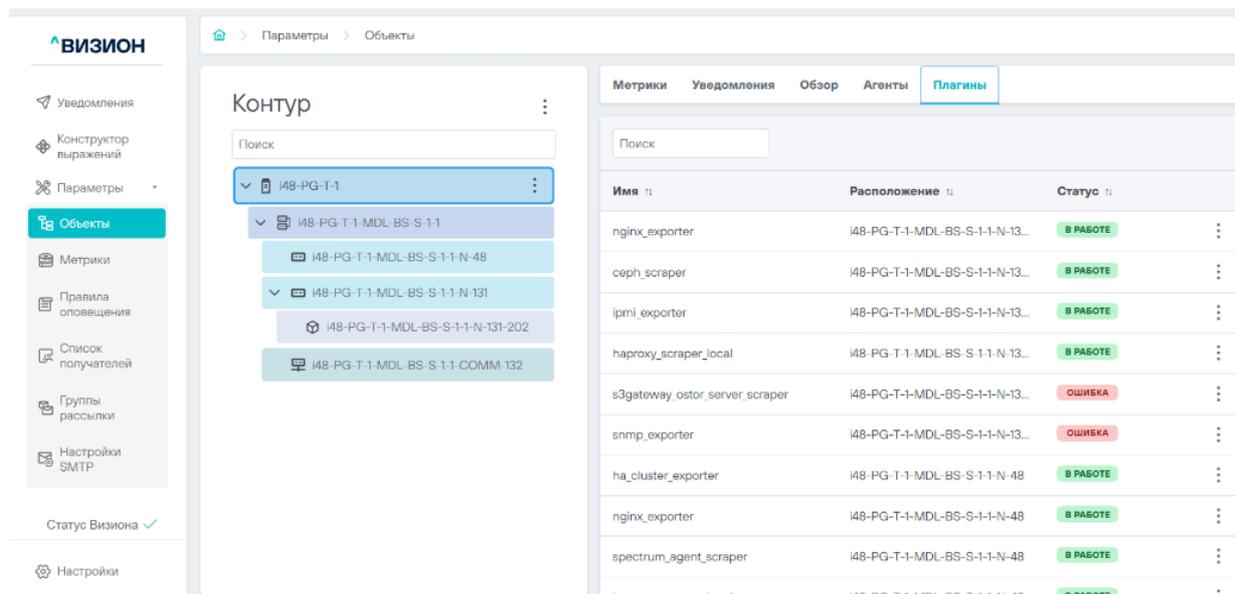
- Плагины – состав конфигурационных файлов, содержащих в себе инструкции для настройки агентов и расстановки экспортеров. На данном шаге необходимо выбрать Плагины в соответствии с типом узла. Для узла, который будет выбран как Прокси, требуется выбрать плагины, реализующие сбор метрик снаружи относительно объекта мониторинга. Рекомендуется расставлять плагины по таблице применимости, указанной в разделе 2.6 данного руководства.

Если в столбце Сервер/VM указано «VM с прокси» - плагин рекомендуется разместить на VM, выбранную Прокси. Остальные плагины конфигурируются на усмотрение пользователя, в зависимости от расположения сервисов на узлах.

После расстановки всех агентов и нажатия кнопки «Сохранить», меняются статусы агентов. В случае успеха будет указан статус «Установлен». Если агент не будет настроен, отобразится статус Ошибка. Требуется проверить логи и сетевую связность между компонентами, а также наличие агента Платформы на всех узлах, за которыми требуется осуществлять мониторинг.

## 7.2 Конфигурирование плагинов

Для конфигурирования плагинов необходимо перейти в раздел «Объекты» - на вкладку «Плагины»



Для каждого плагина предусмотрен набор настроек, которые необходимо задать в контекстном меню : - конфигурировать.

Часть плагинов используют настройки, которые были заданы при развертывании Визиона. Для тех, которые не получают доступ к данным, будет выведен статус «Ошибка». Требуется осуществить настройки.

Перед добавлением убедиться, что при установке плагина его порт не конфликтует с другим ПО.

После заполнения параметров, применяем настройки и происходит автоматическое конфигурирование.

Для проверки корректности настроек, перейдите в пункт меню «Конструктор выражений», введите выражение `up{pak_id="<ИМЯ_ПАК>"}` и проверьте что работают все сконфигурированные плагины (в наличии временные ряды с именем метки «job» и значениями имен плагинов).

Параметры всех плагинов приведены в [документе](#) .

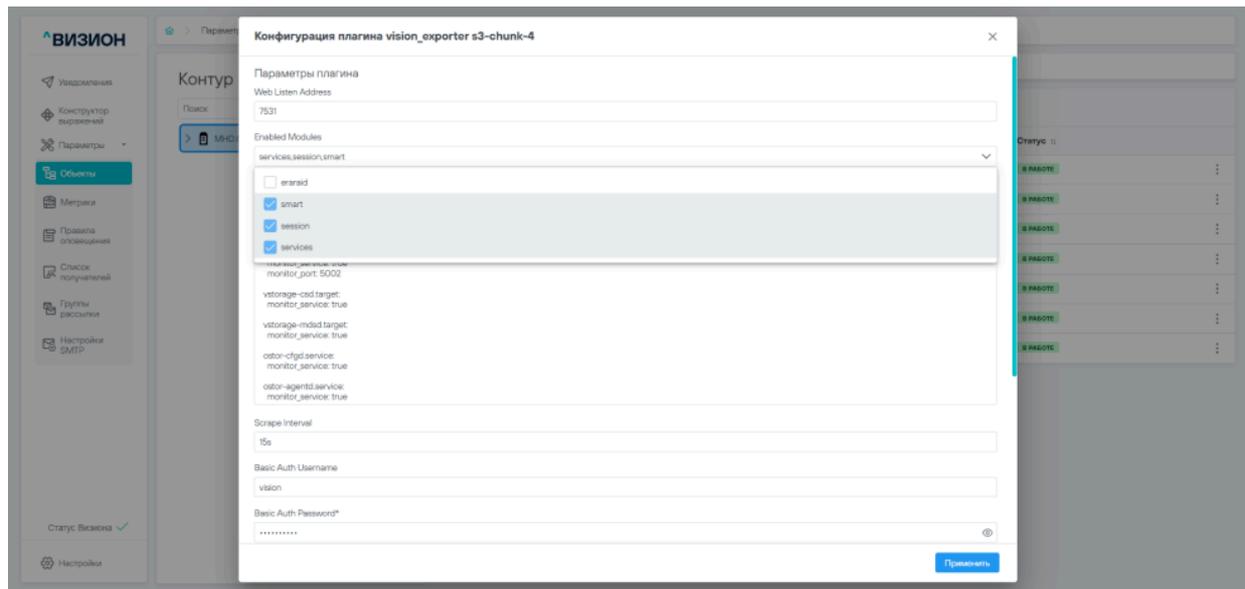
Далее будут приведены особенности конфигурации некоторых плагинов.

### 7.2.1 vision\_exporter

Плагин предназначен для сбора метрик о состоянии сервисов ОС, пользовательских сессиях, некоторой информации S.M.A.R.T, состоянии БД Greenplum, а также о состоянии дисковых массивов, организованных с помощью ПО RAIDIX ERA и лицензиях на ПО RAIDIX ERA.

В поле Enabled Modules нужно включить модули:

- **smart, session, services** - для мониторинга МХД.О
- **eraraid** - для мониторинга конфигураций, включающих программный модуль RAIDIX ERA
- **greenplum** - для мониторинга состояния БД Greenplum

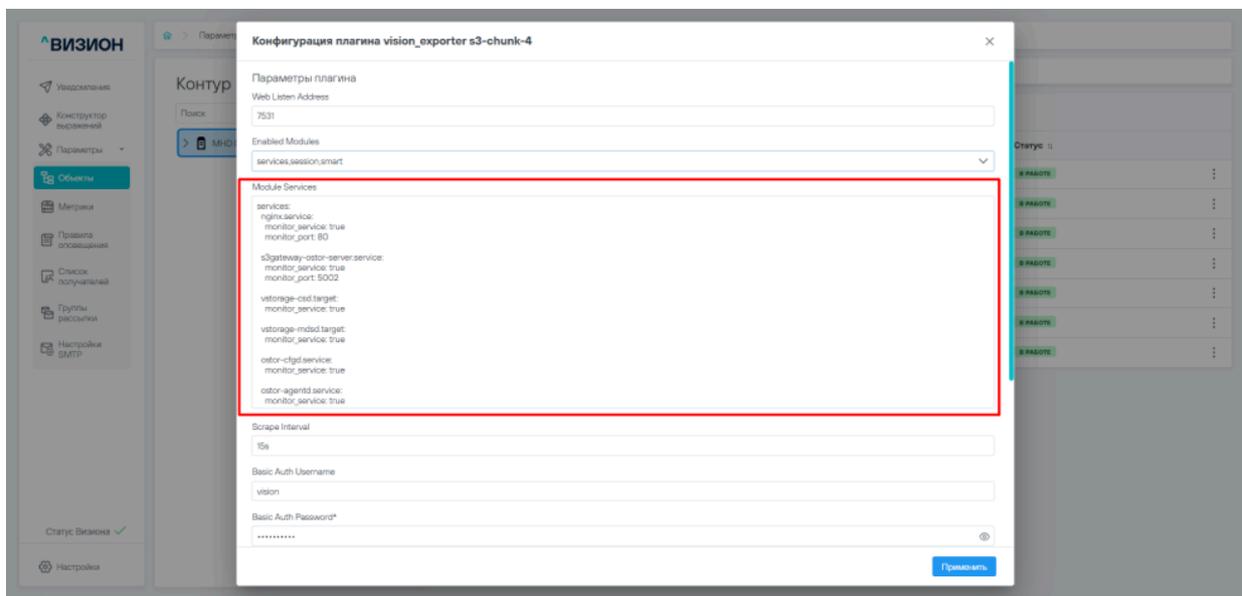


Для получения метрик eraraid в МХД.О/МБД.П нужно включить одноименный модуль в поле Enabled Modules.

Для настройки мониторинга сервисов в форме конфигурации плагина в поле **Module Services** вы можете добавить конфигурацию systemd сервисов в формате

```
services:
  SERVICENAME.service:           # имя сервиса
    monitor_service: [true|false] # Мониторинг uptime и active сервис
    monitor_port: PORT            # Мониторинг порта сервиса
    monitor_specific_config: firewallld # Мониторинг конфигурации заранее з
```





Для МХД.О (S3) рекомендуется следующая конфигурация:

- узел балансировки

```
services:  
  firewallld:  
    monitor_service: false  
    monitor_specific_config: firewallld  
  
  nginx.service:  
    monitor_service: true  
    monitor_port: 6006  
  
  haproxy.service:  
    monitor_service: true  
    monitor_port: 80  
  
  postgresql.service:  
    monitor_service: true  
    monitor_port: 5432  
  
  s3gateway-compression-server.service:  
    monitor_service: true  
    monitor_port: 5001  
  
  s3gateway-control-server.service:  
    monitor_service: true  
    monitor_port: 3000  
  
  s3gateway-proxy-server.service:
```

```
monitor_service: true
monitor_port: 4000
```

- узел хранения

```
services:
  nginx.service:
    monitor_service: true
    monitor_port: 80

  s3gateway-ostor-server.service:
    monitor_service: true
    monitor_port: 5002

  vstorage-csd.target:
    monitor_service: true

  vstorage-mdsd.target:
    monitor_service: true

  ostor-cfgd.service:
    monitor_service: true

  ostor-agentd.service:
    monitor_service: true
```

При необходимости скорректировать состав и параметры сервисов, проверьте порты работы указанных сервисов.

## 7.2.2 nginx\_exporter

Плагин предназначен для локального сбора метрик с Nginx. На данный момент используется для мониторинга компонентов МХД.О.

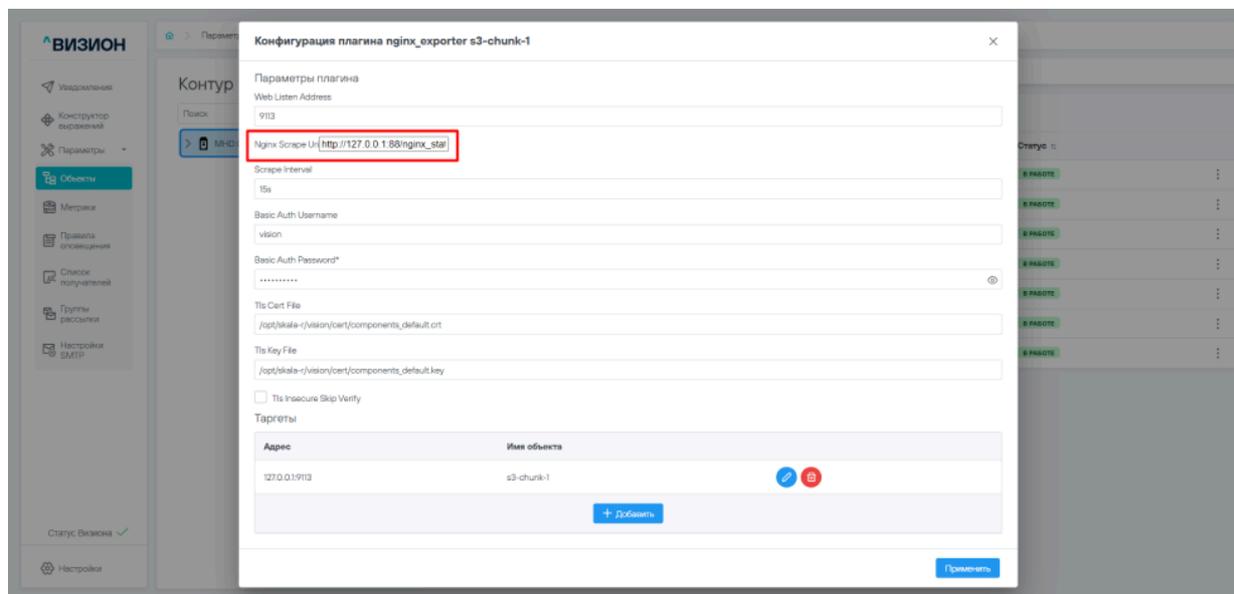
Плагин представляет собой экспортер, который получает состояние Nginx с помощью модуля `http_stub_status_module` и преобразует в метрики.

Предварительно Nginx должен быть настроен на отдачу метрик следующим образом:

1. Выполните на узле кластера S3 команду `nginx -V 2>&1 | grep -o http_stub_status_module` и убедитесь, что Nginx собран с модулем `http_stub_status_module`.
2. В файле конфигурации Nginx (по умолчанию, `/etc/nginx/nginx.conf`) в блоке `server` должен присутствовать следующий роут:

```
location = /nginx_status {
    stub_status;
}
```

В интерфейсе Визиона укажите эндпоинт в конфигурации плагина в поле **Nginx Scrape Uri**. Например, **http://127.0.0.1:88/nginx\_status**.



## 7.2.3 haproxy\_scrapers\_local

Плагин предназначен для локального сбора метрик с HAProxy. На данный момент компонент используется в МХД.О на узлах балансировки.

Перед настройкой плагина вы можете убедиться, что HAProxy скомпилирован с Prometheus Exporter с помощью команды `haproxy -vv`. В выводе вы должны увидеть **Built with the Prometheus exporter as a service**.

При наличии модуля предоставления метрик необходимо, чтобы включить его в файле конфигурации HAProxy (по умолчанию: `/etc/haproxy/haproxy.cfg`).

Пример раздела **frontend**:

```
frontend prometheus
    bind :8008
    mode http
    http-request use-service prometheus-exporter
    no log
```

В интерфейсе Визиона укажите адрес HAProxy в конфигурации плагина в разделе **Таргеты** в параметре **Адрес**. Например, **127.0.0.1:8008**, в случае установки плагина локально и предоставления метрик по порту **:8008**.

Конфигурация плагина `haproxy_scraper_local s3_gate_redOS1`

Параметры плагина

Scrape Interval  
15s

Basic Auth Username

Basic Auth Password

Tls Ca File

Tls Insecure Skip Verify

Таргеты

Адрес	Имя объекта
127.0.0.1:8008	s3_gate_redOS1

Применить

## 7.2.4 s3gateway\_ostor\_server\_scraper

Плагин предназначен для локального сбора метрик с `s3gateway_ostor_server`. На данный момент компонент используется в МХД.О на узлах хранения.

В `s3gateway_ostor_server` присутствует встроенная поддержка предоставления метрик.

В интерфейсе Визиона укажите адрес `s3gateway_ostor_server` в конфигурации плагина в разделе **Таргеты** в параметре **Адрес**. Например, `127.0.0.1:7002`, в случае установки плагина локально и предоставления метрик по порту `:7002`.

Конфигурация плагина `s3gateway_ostor_server_scraper vmhdo1`

Scrape Interval  
15s

Basic Auth Username

Basic Auth Password

Tls Ca File

Tls Insecure Skip Verify

Таргеты

Адрес	Имя объекта
127.0.0.1:7002	vmhdo1

+ Добавить

Применить

## 7.2.5 s3gateway\_compression\_server\_scraper

Плагин предназначен для локального сбора метрик с `s3gateway_compression_server` (компонент используется в МХД.О на узлах балансировки).

В s3gateway\_compression\_server присутствует встроенная поддержка предоставления метрик.

В интерфейсе Визиона укажите адрес s3gateway\_compression\_server в конфигурации плагина в разделе **Таргеты** в параметре **Адрес**. Например, **127.0.0.1:7001**, в случае установки плагина локально и предоставления метрик по порту **:7001**.

Конфигурация плагина s3gateway\_compression\_server\_scraper s3\_gate\_redOS1

Scrape Interval: 15s

Basic Auth Username: [input field]

Basic Auth Password: [input field]

Tls Ca File: [input field]

Tls Insecure Skip Verify

Таргеты

Адрес	Имя объекта	
127.0.0.1:7001	s3_gate_redOS1	[edit] [delete]

+ Добавить

Применить

## 7.2.6 node\_exporter

Особое внимание обращать на машине МВ.ДИ на конфликт портов с другим ПО.

## 7.2.7 bash\_exporter

Внимание! Плагин bash\_exporter устанавливается отдельно от основного дистрибутива Визиона.

Необходимо в настройках плагина указать путь к **bash** скрипту:

Конфигурация плагина bash\_exporter n

Параметры плагина

Web Listen Address: 17055

Bash Script Path: /opt/skala-rf/vision/test\_script.sh

Scrape Interval: 15s

Scrape Timeout: 10s

Basic Auth Username: vision

Basic Auth Password: [password field]

Применить

На файл скрипта необходимо назначить права на исполнение для пользователя **vision**.

В процессе исполнения скрипт должен отдавать значения метрик в формате **prometheus**.

## 7.2.8 systemd\_exporter

Необходимо в настройках плагина указать сервисы, которые необходимо включить в сбор метрик (**include units**) либо исключить (**exclude units**):

Конфигурация плагина **systemd\_exporter** PG-PRD-3-MDL-BS-01-N-199-VM-VSN-192

Параметры плагина

Web Listen Address  
9110

Include Units  
vision\_\*.service|grafana.service

Exclude Units

Scrape Interval  
15s

Basic Auth Username  
vision

Basic Auth Password  
.....

Применить

Если поле **include units** не заполнено и поле **exclude units** также не заполнено, то это означает, что будут собираться метрики по всем сервисам.

## 7.2.9 kafka\_exporter

Плагин обладает множественными настройками:

Конфигурация плагина **kafka\_exporter** Визιον

Параметры плагина

Web Listen Address  
9308

Kafka Servers

kafka\_servers-0\*  
192.168.184.144:9092

kafka\_servers-1\*  
192.168.184.146:9092

kafka\_servers-2\*  
192.168.184.148:9092

+ Добавить

Kafka Version  
2.0.0

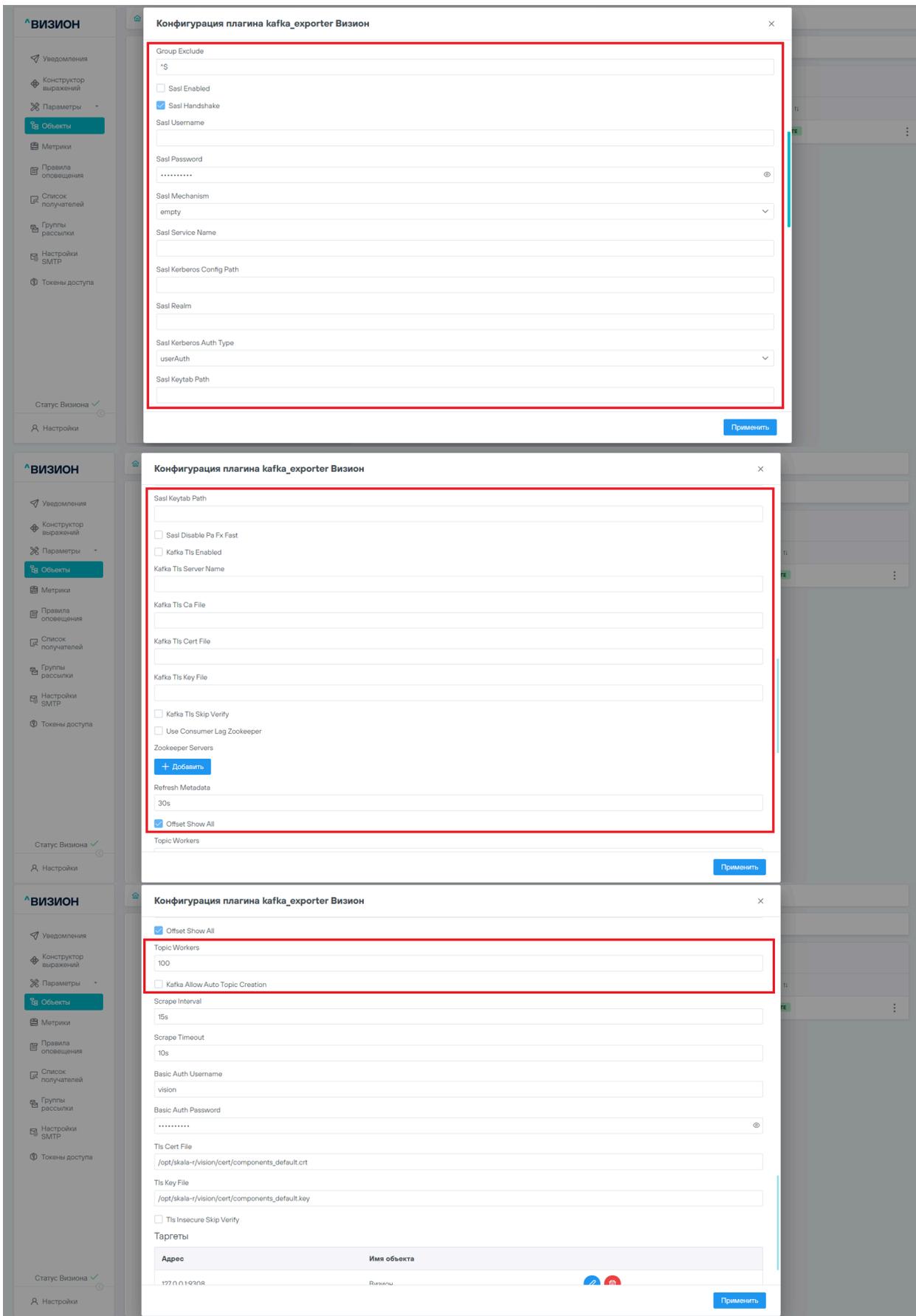
Topic Filter  
\*

Topic Exclude  
\*\$

Group Filter  
\*

Group Exclude

Применить



Описание настроек:

Имя	Описание	Дополнительные параметры поля
-----	----------	-------------------------------

Kafka Servers	Адрес (host:port) сервера Kafka.	<pre>pattern: ^[a-zA-Z0-9.-]+: ([1-9][0-9]{0,3} [1-5][0-9]{4} 6[0-4][0-9]{3} 65[0-4][0-9]{2} 655[0-2][0-9] 6553[0-5])\$minItems: 1items: {"type": "string", "pattern": "^[a-zA-Z0-9.-]+:([1-9][0-9]{0,3} [1-5][0-9]{4} 6[0-4][0-9]{3} 65[0-4][0-9]{2} 655[0-2][0-9] 6553[0-5])\$"} </pre>
Kafka Version	Версия брокера Kafka.	pattern: ^(0.
Topic Filter	Регулярное выражение, определяющее, какие топики собирать.	
Topic Exclude	Регулярное выражение, определяющее, какие топики исключить.	
Group Filter	Регулярное выражение, определяющее, какие потребительские группы собирать.	
Group Exclude	Регулярное выражение, определяющее, какие потребительские группы исключить.	
Sasl Enabled	Подключение с использованием SASL/PLAIN, по умолчанию — выключено.	

Sasl Handshake	Выключите, если используется SASL-прокси, не связанный с Kafka.	
Sasl Username	Имя пользователя SASL.	
Sasl Password	Пароль пользователя SASL.	writeOnly: Trueformat: password
Sasl Mechanism	Механизм SASL: scram-sha256, scram-sha512, gssapi или plain.	enum: ['scram-sha256', 'scram-sha512', 'gssapi', 'plain', '']
Sasl Service Name	Имя сервиса при использовании аутентификации Kerberos. Используется только при sasl_mechanism=gssapi.	
Sasl Kerberos Config Path	Путь к конфигурации Kerberos. Используется только при sasl_mechanism=gssapi.	
Sasl Realm	Realm Kerberos. Используется только при sasl_mechanism=gssapi.	
Sasl Kerberos Auth Type	Тип аутентификации Kerberos. Либо keytabAuth, либо userAuth. Используется только при sasl_mechanism=gssapi.	enum: ['keytabAuth', 'userAuth']

Sasl Keytab Path	Путь к файлу keytab для Kerberos. Используется только при <code>sasl_mechanism=gssapi</code> и <code>sasl_kerberos_auth_type=keytabAuth</code> .	
Sasl Disable Pa Fx Fast	Сконфигурировать клиент Kerberos без использования <code>PA_FX_FAST</code> , по умолчанию выключено. Используется только при <code>sasl_mechanism=gssapi</code> .	
Kafka Tls Enabled	Подключение к Kafka с TLS, по умолчанию выключено.	
Kafka Tls Server Name	Используется для проверки имени хоста в возвращаемых сертификатах, если не задан пропуск проверки сертификата. Должно быть указано имя сервера Kafka.	
Kafka Tls Ca File	Путь до CA файла для аутентификации клиента Kafka TLS (необязательно).	
Kafka Tls Cert File	Путь до файла сертификата для аутентификации клиента Kafka TLS (необязательно).	

Kafka Tls Key File	Путь до файла ключа сертификата для аутентификации клиента Kafka TLS (необязательно).	
Kafka Tls Skip Verify	Пропустить проверку сертификата.	
Use Consumer Lag Zookeeper	Сбор текущей задержки (zookeeper) ConsumerGroup в топике/разделе.	
Zookeeper Servers	Адрес (хосты) zookeeper.	<pre>pattern: ^[a-zA-Z0-9.-]+ ( :([1-9][0-9]{0,3} [1-5] [0-9]{4}) 6[0-4][0-9] {3} 65[0-4][0-9] {2} 655[0-2][0- 9] 6553[0-5]))\$items: {"type": "string", "pattern": "^[a-zA-Z0- 9.-]+( :([1-9][0-9]{0,3}  [1-5][0-9]{4}) 6[0-4][0-9] {3} 65[0-4][0-9] {2} 655[0-2][0- 9] 6553[0-5]))\$"} </pre>
Refresh Metadata	Интервал обновления метаданных.	<pre>minLength: 1maxLength: 255pattern: ^((([0- 9]+)y)?([0-9]+)w)?([0- 9]+)d)?([0-9]+)h)?([0- 9]+)m)?([0-9]+)s)?([0- 9]+)ms)? 0)\$ </pre>
Offset Show All	Показывать смещение/ задержку для всех групп потребителей. Если выключено, то показывать только подключенные группы потребителей.	

Topic Workers	Количество параллельных потоков для обработки топиков.	exclusiveMinimum: 0
Kafka Allow Auto Topic Creation	Разрешить автоматически создавать запрошенные топики, которые еще не существуют, по умолчанию выключено.	

## 7.2.10 process\_exporter

Плагин обладает возможностью конфигурации:



Параметры конфигурации плагина:

- children (по умолчанию: true) - обеспечивает, что любой процесс, который иначе не принадлежал бы к своей собственной группе, присоединяется к первой найденной группе (если такая существует) при перемещении вверх по дереву процессов.
- threads (по умолчанию: true) - показатели будут разбиты по названию потока, а также по названию группы.
- recheck (по умолчанию: false) - пересмотр имен процессов при каждом сборе метрик, так как процессы могут изменять свои имена, это может привести к тому, что процесс попадет не в ту группу, если мы впервые увидим его до того, как он получит свое собственное имя.
- recheck duration (по умолчанию: 0) - включить проверку имен процессов только на определенный период после запуска плагина. Отключает постоянную

проверку recheck

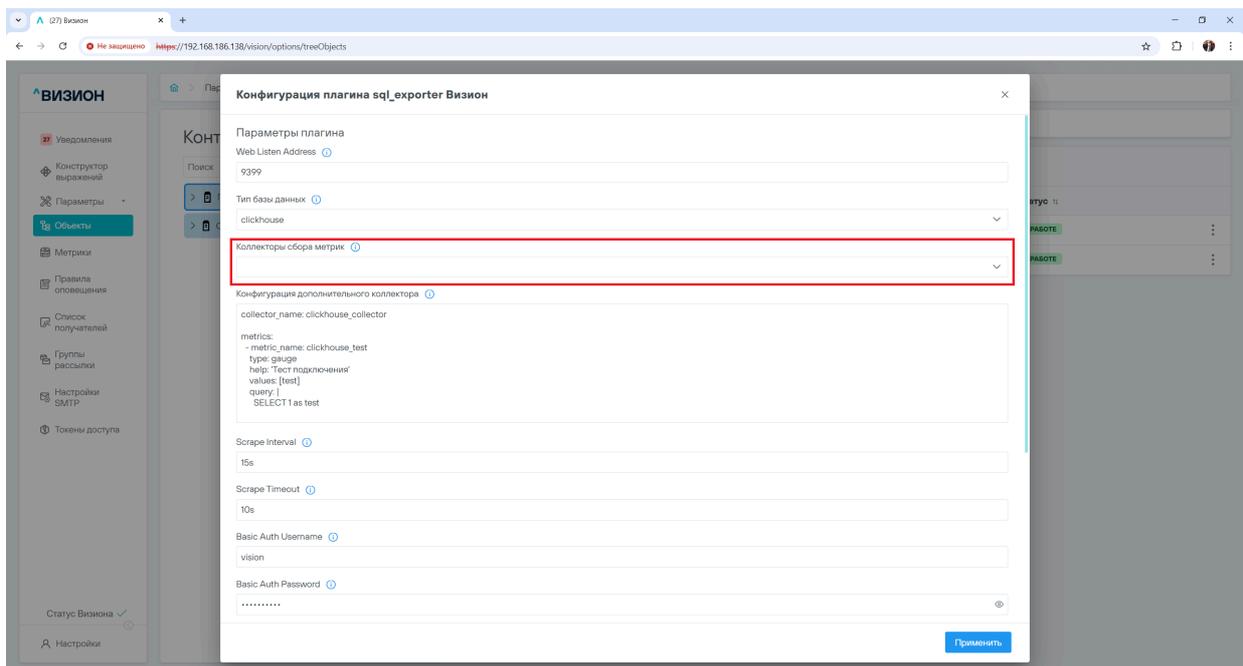
Для настройки мониторинга процессов в форме конфигурации плагина в поле Plugin Configuration вы можете добавить конфигурацию в формате yaml (пустое поле конфигурации заменится стандартной конфигурацией по умолчанию):

```
process_names:  
  - name: "{.Comm}"  
    cmdline:  
      - '+'
```

другой пример конфигурации:

```
process_names:  
  - comm:  
    - chromium-browser  
    - bash  
    - prometheus  
    - gvim  
  - exe:  
    - /sbin/upstart  
  cmdline:  
    - --user  
  name: upstart:-user
```

## 7.2.11 sql\_exporter



Для конфигурирования sql\_exporter можно воспользоваться коллектором сбора метрик: из предустановленных коллеторов можно будет выбрать наиболее подходящий для мониторинга машины.

## 7.2.12 blackbox\_exporter

Для конфигурации следует использовать раздел таргетов, где указывается адрес интересующего объекта.

Имя	Тип	Значение по умолчанию	Описание	Дополнительные параметры поля
Web Listen Address	integer	9115	Порт запуска экспортера	exclusiveMinimum: <input type="text" value="0"/> maximum: <input type="text" value="65535"/>
Scrape Interval	string	'15s'	Периодичность опроса экспортера	minLength: <input type="text" value="1"/> maxLength: <input type="text" value="255"/> pattern: <input type="text" value="^(([0-9]+)y)?([0-9]+)w)?([0-9]+)d)?([0-9]+)h)?([0-9]+)m)?([0-9]+)s)?([0-9]+)ms)? 0)\$"/>

Scrape Timeout	string	'10s'	Таймаут опроса таргетов. Значение не должно превышать значения поля "Scrape Interval"	minLength: 1 maxLength: 255 pattern: ^(([0-9]+)y)?([0-9]+w)?([0-9]+d)?([0-9]+h)?([0-9]+m)?([0-9]+s)?([0-9]+ms)? 0)\$
Basic Auth Username	string	'vision'	Имя пользователя BasicAuth для доступа к экспортеру	
Basic Auth Password	string	'skala-r'	Пароль пользователя BasicAuth для доступа к экспортеру	writeOnly: True format: password

Tls Cert File	string	'/opt/skalar/vision/cert/component_s_default.crt'	Путь до файла сертификата на узле, где установлен плагин, для запуска экспортера	
Tls Key File	string	'/opt/skalar/vision/cert/component_s_default.key'	Путь до файла ключа сертификата на узле, где установлен плагин, для запуска экспортера	
Tls Insecure Skip Verify	boolean	False	Пропустить проверку сертификата при подключении агента к источнику метрик	

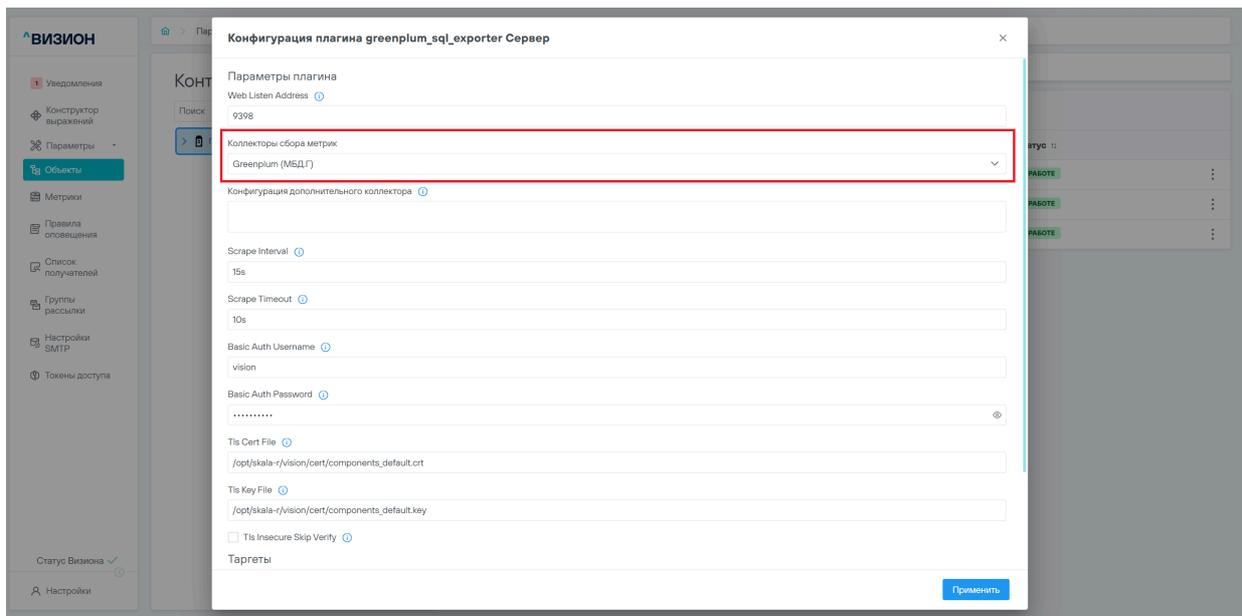
### Параметры таргета

Поддерживает несколько таргетов.

Имя	Тип	Значение по умолчанию	Описание	Дополнительные параметры поля
Адрес	string	'127.0.0.1:9115'	Адрес и порт экспортера, которому принадлежит таргет	

Имя объекта	string	'<имя-объекта>'	Допустимы е типы объектов: Сервер, Виртуальная машина	
-------------	--------	-----------------	---	--

## 7.2.13 greenplum\_sql\_exporter



Для конфигурирования greenplum\_sql\_exporter можно воспользоваться коллектором сбора метрик: из предустановленных коллеторов можно будет выбрать наиболее подходящий для мониторинга машины.

Имя	Тип	Значение по умолчанию	Описание	Дополнительные параметры поля
Web Listen Address	integer	9398	Порт запуска экспортера	exclusiveMinimum: 0 maximum: 65535

<p>Коллекторы сбора метрик</p>	<p>array</p>	<p>['Greenplum (МБД.Г)']</p>		<p>items:  <pre>{ 'enum': [ 'Greenplum (МБД.Г)' ] , 'type': 'string' }u niquelitems: True</pre> </p>
<p>Конфигурация дополнительного коллектора</p>	<p>string</p>	<p>"</p>	<p>Дополнительный коллектор для сбора метрик в формате YAML</p>	<p>format:  <pre>textarea</pre> </p>
<p>Scrape Interval</p>	<p>string</p>	<p>'15s'</p>	<p>Периодичность опроса экспортера</p>	<p>minLength:  <pre>1</pre> maxLength:  <pre>255</pre> pattern:  <pre>^((([0-9]+)y)? ([0-9]+)w)? ([0-9]+)d)? ([0-9]+)h)? ([0-9]+)m)? ([0-9]+)s)? ([0-9]+)ms)?  0)\$</pre> </p>

Scrape Timeout	string	'10s'	Таймаут опроса таргетов. Значение не должно превышать значения поля "Scrape Interval"	minLength: 1 maxLength: 255 pattern: ^(([0-9]+)y)?([0-9]+w)?([0-9]+d)?([0-9]+h)?([0-9]+m)?([0-9]+s)?([0-9]+ms)? 0)\$
Basic Auth Username	string	'vision'	Имя пользователя BasicAuth для доступа к экспортеру	
Basic Auth Password	string	'skala-r'	Пароль пользователя BasicAuth для доступа к экспортеру	writeOnly: True format: password

Tls Cert File	string	'/opt/skalar/vision/cert/component_s_default.crt'	Путь до файла сертификата на узле, где установлен плагин, для запуска экспортера	
Tls Key File	string	'/opt/skalar/vision/cert/component_s_default.key'	Путь до файла ключа сертификата на узле, где установлен плагин, для запуска экспортера	
Tls Insecure Skip Verify	boolean	False	Пропустить проверку сертификата при подключении агента к источнику метрик	

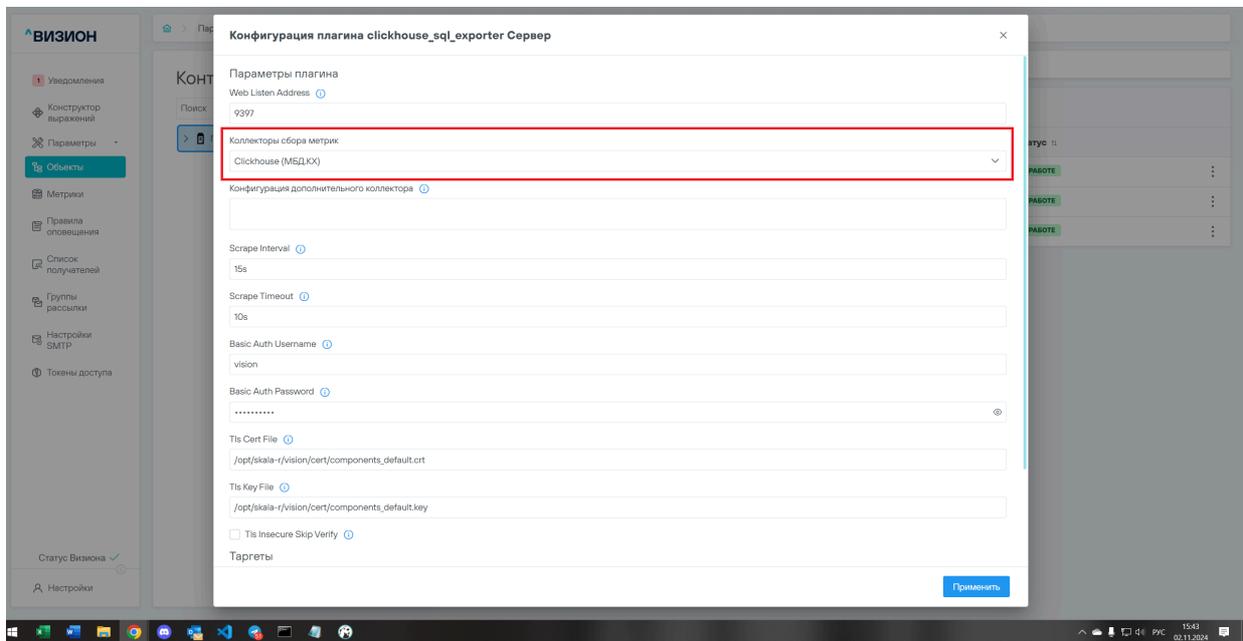
### Параметры таргета

Поддерживает несколько таргетов.

Имя	Тип	Значение по умолчанию	Описание	Дополнительные параметры поля
-----	-----	-----------------------	----------	-------------------------------

Адрес	string	'127.0.0.1:5432'	Адрес и порт сервиса с которого необходимо снять метрики	
Имя объекта	string	'<имя-объекта>'	Допустимые типы объектов: Виртуальная машина, Сервер	
Имя пользователя БД	string	'vision'	Имя пользователя для подключения к БД	
Пароль пользователя БД	string	'password'	Пароль пользователя для подключения к БД	writeOnly: <b>True</b> format: <b>password</b>
Имя базы данных	string	'postgres'	Имя базы данных для подключения	

## 7.2.14 clickhouse\_sql\_exporter



Для конфигурирования clickhouse\_sql\_exporter можно воспользоваться коллатором сбора метрик: из преднастроенных коллеторов можно будет выбрать наиболее подходящий для мониторинга машины.

Имя	Тип	Значение по умолчанию	Описание	Дополнительные параметры поля
Web Listen Address	integer	9397	Порт запуска экспортера	exclusiveMinimum: 0 maximum: 65535
Коллекторы сбора метрик	array			items: {'type': 'null'} uniqueItems: True

<p>Конфигурация дополнительного коллектора</p>	<p>string</p>	<p>"</p>	<p>Дополнительный коллектор для сбора метрик в формате YAML. Обязательно включайте параметр <code>no_prepared_statement: true</code> для всех метрик.</p>	<p>format: <code>textarea</code></p>
<p>Scrape Interval</p>	<p>string</p>	<p>'15s'</p>	<p>Периодичность опроса экспортера</p>	<p>minLength: <code>1</code> maxLength: <code>255</code> pattern: <code>^(([0-9]+)y)?([0-9]+)w)?([0-9]+)d)?([0-9]+)h)?([0-9]+)m)?([0-9]+)s)?([0-9]+)ms)? 0)\$</code></p>

Scrape Timeout	string	'10s'	Таймаут опроса таргетов. Значение не должно превышать значения поля "Scrape Interval"	minLength: <b>1</b> maxLength: <b>255</b> pattern: <b>^([0-9]+)y?([0-9]+)w?([0-9]+)d?([0-9]+)h?([0-9]+)m?([0-9]+)s?([0-9]+)ms? 0)\$</b>
Basic Auth Username	string	'vision'	Имя пользователя BasicAuth для доступа к экспортеру	
Basic Auth Password	string	'skala-r'	Пароль пользователя BasicAuth для доступа к экспортеру	writeOnly: <b>True</b> format: <b>password</b>

Tls Cert File	string	'/opt/skalar/vision/cert/component_s_default.crt'	Путь до файла сертификата на узле, где установлен плагин, для запуска экспортера	
Tls Key File	string	'/opt/skalar/vision/cert/component_s_default.key'	Путь до файла ключа сертификата на узле, где установлен плагин, для запуска экспортера	
Tls Insecure Skip Verify	boolean	False	Пропустить проверку сертификата при подключении агента к источнику метрик	

### Параметры таргета

Поддерживает несколько таргетов.

Имя	Тип	Значение по умолчанию	Описание	Дополнительные параметры поля
-----	-----	-----------------------	----------	-------------------------------

Адрес	string	'127.0.0.1:9000'	Адрес и порт сервиса с которого необходимо снять метрики	
Имя объекта	string	'<имя-объекта>'	Допустимые типы объектов: Виртуальная машина, Сервер	
Имя пользователя БД	string	'vision'	Имя пользователя для подключения к БД	
Пароль пользователя БД	string	'password'	Пароль пользователя для подключения к БД	writeOnly: <b>True</b> format: <b>password</b>
Имя базы данных	string	'postgres'	Имя базы данных для подключения	

## 7.2.15 picodata\_scraper

Следует указать в таргетах адрес и порт для сбора метрик Picodata.

## 7.3 Добавление и обновление плагинов

Выполнить следующие действия на виртуальной машине сервера Визиона:

1. Распаковать архив с новым/обновленным плагином в директорию `/opt/skala-r/vision`, например:

```
tar -xzf plugin_archive.tar.gz -C /opt/skala-r/vision
```

2. Перезапустить сервис vision\_core:

```
systemctl restart vision_core.service
```

3. Проверьте состояние vision\_core и убедитесь, что логе нет ошибок после перезапуска

```
systemctl status vision_core.service
```

```
journalctl -fu vision_core -n 20
```

## 7.4 Снятие ПАК с мониторинга (деинсталляция Визиона с узлов ПАК).

**Внимание!** Данные действия необратимы. В процессе удаления будут деинсталлированы все агенты мониторинга, их плагины, а также Визион.Прокси.

Перейти в веб-интерфейс Визиона, открыв Интернет браузер, и введя в строке адреса IP-адрес сервера Визиона <https://<vision-ip>>.

Перейти в боковом меню в раздел “Параметры”->“Объекты”, в панели “Контур” выбрать удаляемый ПАК, нажать кнопку  и выбрать “Удалить”. Согласиться с удалением, нажав кнопку “Удалить”.

## 8 НАСТРОЙКА ОТПРАВКИ УВЕДОМЛЕНИЙ ПО SMTP

Для интеграции с шлюзом SMTP, требуется осуществить настройки подключения, затем осуществить подписку пользователей на рассылку уведомлений.

Подробнее о работе с интерфейсом описано в руководстве Пользователя. В данном руководстве описан алгоритм работы:

1. Перейти в веб-интерфейс Визиона, открыв Интернет браузер, и введя в строке адреса IP-адрес сервера Визиона (<https://<vision-ip>>), после чего выбрать пункт “Модуль администратора”.
2. Перейти в раздел “Настройки SMTP” и заполнить все поля валидными значениями. Поле “Основной адрес получателя” заполняется адресом электронной почты, на которую будут приходить уведомления о сработанных правилах оповещения, для которых нет подписок конкретных групп рассылок. Нажмите на кнопку “Сохранить”.
3. Перейти в раздел “Список получателей” и создать получателя уведомлений с каналом E-mail и адресом электронной почты, на которую будут приходить уведомления о сработанных правилах оповещения.
4. Перейти в раздел “Группы рассылки”, создать группу, добавить в эту группу получателя. Добавить условие в этой группе: например, severity = critical.
5. Перейти в раздел “Правила оповещения”, создать новое правило с важностью critical (в целях тестирования добавить правило оповещения, которое точно сработает, пример: ввести в поле ввода PromQL выражение `up == 1`).

6. Перейти в раздел “Уведомления” и дождаться срабатывания правила оповещения.
7. Войти в электронную почту, указанную на шаге 3, и дождаться получения письма о сработавшем правиле оповещения из шага 5.

## 9 НАСТРОЙКА ОТПРАВКИ УВЕДОМЛЕНИЙ ПО SNMP

### 9.1 Настройка SNMP notifier

Необходимо задать параметры отправки уведомлений мониторинга по SNMP путём POST запроса к API Визиона

[https://<vision\\_host>:8092/vision/api/v1/config/snmp\\_notifier\\_config](https://<vision_host>:8092/vision/api/v1/config/snmp_notifier_config) с телом JSON со следующими полями:

Имя параметра	Описание
snmp_version	версия SNMP. Допустимые значения: “V2c”, “V3”. По умолчанию “V2c”.
snmp_destination	адресат сообщений. По умолчанию “127.0.0.1:162”.
snmp_retries	количество повторных посылок сообщения SNMP. По умолчанию 1.
snmp_trap_default_oid	OID ловушки, если он не найден в метках сработавшего правила. По умолчанию “1.3.6.1.4.1.98789”.
snmp_timeout_sec	длительность таймаута в секундах. По умолчанию 5.
snmp_community	комьюнити SNMP (только для V2c, иначе null). По умолчанию “public”.
snmp_authentication_protocol	протокол шифрования пароля (только для V3). Допустимые значения: “MD5”, “SHA”, null. По умолчанию null.

snmp_authentication_username	имя пользователя аутентификации SNMP (только для V3). Обязателен*, если указан snmp_authentication_protocol. По умолчанию null.
snmp_authentication_password	пароль аутентификации SNMP (только для V3). Обязателен*, если указан snmp_authentication_protocol. По умолчанию null.
snmp_private_protocol	протокол передачи данных SNMP (только для V3). Допустимые значения: "DES", "AES", null. По умолчанию null.
snmp_private_password	Пароль шифрования SNMP (только для V3). Обязателен*, если указан snmp_private_protocol. По умолчанию null.
snmp_security_engine_id	ID механизма безопасности SNMP. По умолчанию null.
snmp_context_engine_id	ID механизма контекста. По умолчанию null.
snmp_context_name	имя контекста. По умолчанию null.

Пример Curl для SNMP V3:

```
curl -k -X 'POST' 'https://<vision_host>:8088/vision/api/v1/config/snmp_noti
  "snmp_version": "V3",
  "snmp_destination": "127.0.0.1:162",
  "snmp_retries": 1,
  "snmp_trap_default_oid": "1.3.6.1.4.1.98789",
  "snmp_timeout_sec": 5,
  "snmp_authentication_protocol": "MD5",
  "snmp_authentication_username": "username",
  "snmp_authentication_password": "password",
  "snmp_private_protocol": "AES",
  "snmp_private_password": "password",
```

```
"snmp_security_engine_id": "xxxxxxxxxx"
}'
```

## 9.2 Настройка перечня уведомлений, отправляемых по SNMP

1. Перейти в веб-интерфейс Визиона, открыв Интернет браузер, и ввести в строке адреса IP-адрес сервера Визиона (<https://<vision-ip>>), после чего выбрать пункт “Модуль администратора”.
2. В боковом меню выбрать раздел “Параметры”, пункт: “Группы рассылки”.
3. Нажать кнопку “Добавить” и создать новую группу рассылки.
4. В открывшейся форме для группы рассылки заполнить параметры (в скобках – примеры заполнения):
  - Имя (“Группа А”)
  - Детальное описание (“Группа поддержки 1 линии”)
  - Условия отправки уведомлений (если необходимо получать абсолютно все уведомления по SNMP, установить следующие значения:
    - Метка: “severity”;
    - Оператор: выбрать все допустимые значения
  - Получать только критические:
    - Метка: “severity”;
    - Значение: “critical”)
  - Получатели (“SNMP-шлюз”)
5. Нажать кнопку “Добавить”.

## 10 НАСТРОЙКИ БЕЗОПАСНОСТИ

### 10.1 Межкомпонентное взаимодействие

В целях повышения безопасности работа между компонентами Визиона реализована через TLS и Basic Auth.

Настройка аутентификации и размещение сертификатов выполняется автоматически при развертывании компонентов Визиона.

Параметры технической учетной записи для авторизации между компонентами конфигурируются в файле **\_deploy\_settings.yml**, расположенном в корне дистрибутива - **basic\_auth\_username** и **basic\_auth\_password** (имя пользователя и пароль соответственно).

[Описание компонентов Визион](#)

[Описание конфигурационного файла vision\\_core \(config.yml\)](#)

[Описание параметров конфигурации плагинов](#)

[Описание учетных записей Визион](#)

## 11 УСТАНОВКА ОБНОВЛЕНИЙ ПРОГРАММЫ

Для установки и для обновления “Визиона” используется один и тот же пакет установочных файлов, поставляемый в виде архива.

Обновление выполняется путем установки новой версии поверх существующей.

**Обратите внимание!** В данной версии обновление с сохранением файлов конфигурации (таких, как `.secrets_vault`, `config.yml`, `grafana.yml`) не доступно и они будут перезаписаны.

Данные базы данных временных рядов, пользовательские дашборды, а также информация о зарегистрированных ПАК, конфигурации агентов и плагинов будут сохранены.

## 12 УДАЛЕНИЕ ВИЗИОНА

### 12.1 Удаление Визион.Сервер

1. Открыть терминал и перейти в директорию с дистрибутивом:

```
cd /root/vision-distrib-<version>
```

2. Запустить скрипт

```
./setup.sh
```

3. Выбрать действие “**2. Удалить сервер Визиона.**”

Будут удалены все компоненты Визиона. При необходимости можно очистить пользовательские данные (метрики Victorimetrics, данные служебной БД Vision и данные Grafana). Так же для запуска этой операции можно вызвать команду

```
./setup.sh -r.
```

По окончании удаления его лог будет сохранен в `/opt/skala-r/var/logs/vision_remove.log.<ДАТА_УДАЛЕНИЯ>`.

При необходимости удалить агенты мониторинга (см. п.12.2).

### 12.2 Ручное удаление агентов мониторинга

В некоторых случаях может потребоваться ручное удаление агентов.

1. Перейти в директорию дистрибутива, например:

```
cd /root/vision-distrib-<version>
```

2. Запустить команду, указав путь до файла инвентори (см. п. 6), например,

```
/opt/skala-r/vision/server/inventory/PAK-1_2024-06-10_12-34-56.yml
```

```
ansible-playbook -i /opt/skala-r/vision/server/inventory/PAK-1_2024-03-04_21-32-48.yml ./playbooks/agent_delete.yml
```

## 12.3 Ручное удаление агентов мониторинга старых версий

В случае удаления агентов мониторинга предыдущих версий < 1.2, необходимо:

1. Установить новую версию Визион.Сервера
2. Зарегистрировать ПАК
3. Сформировать файл инвентори
4. Выполнить шаги по ручному удалению

## 12.4 Известные проблемы при удалении Визиона и агентов мониторинга

При удалении Визиона могут возникнуть следующие проблемы:

- **Проблемы с правами доступа к временным файлам Ansible.** Если пользователь, под которым был установлен Визион, больше не существует или у текущих пользователей нет прав на временные файлы Ansible, то для корректного удаления необходимо проверить корректность прав для директорий `/tmp/.ansible`, `~/.ansible`.
- **Неостановленные процессы пользователя `vision`.** Процессы Визиона работают под пользователем `vision`, который может быть недоступен после удаления. В этом случае перед удалением необходимо убедиться, что такие процессы остановлены на всех узлах.

# 13 ХРАНЕНИЕ СЕКРЕТОВ

## 13.1 Общая концепция

В целях безопасности в Визионе организовано безопасное хранение секретов (паролей, токенов и т.п.) в защищенном виде. Данная информация храниться в Ansible Vault и считывается в момент старта компонента vision-core.

Для оперирования содержимым хранилища требуется ключ, выдаваемый разработчиком Визиона. Расположение файла хранилища - `/opt/skala-r/vision/secrets_vault`.

## 13.2 Работа с хранилищем секретов

Просмотр содержимого хранилища:

```
ansible-vault view /opt/skala-r/vision/.secrets_vault
```

Редактирование содержимого хранилища:

```
ansible-vault edit /opt/skala-r/vision/.secrets_vault
```

В обоих случаях потребуется ввести пароль от хранилища после вывода Vault password.

## 13.3 Описание хранимых секретов

`vault.auth.client_secret` - секрет, используемый для подключения сервера Визиона к Keycloak;

`vault.database.password` - пароль, для подключения к служебной БД Postgres;

`vault.general.cookie_secret_key` - ключ, используемый для работы с cookie;

`vault.plagent.token` - токен для подключения к агенту Платформы;

`vault.smtp_config.smtp_auth_password` - пароль для подключения к почтовому шлюзу для отправки сообщений о сработавших правилах оповещений;

`vault.snmp_notifier.snmp*_password` - пароли для подключения к SNMP-шлюзу для отправки сообщений о сработавших правилах оповещений посредством SNMP-трапов (для SNMP v3 протоколов шифрования паролей authentication и protocol);

`vault.*.basic_auth_password` - пароли HTTP Basic Auth соответствующих компонентов.

`vault.grafana.security.admin_password` - Пароль администратора Grafana по умолчанию, можно изменить перед первым запуском Grafana или в настройках профиля.

`vault.grafana.auth.generic_oauth.client_secret` - Секрет клиента, предоставленный приложением OAuth2 для Grafana.

`vault.grafana.metrics.basic_auth_password` - Пароль BasicAuth для получения метрик Grafana.

## 14 ПРОВЕРКА ПРОГРАММЫ

### 14.1 Отображение веб-интерфейса

По завершении процесса установки должен быть доступен веб-интерфейс ПО Визион через браузер по адресу `http://<хост Визиона>`, далее пункт Аналитические панели и Модуль Администратора.

### 14.2 Проверка работоспособности сервисов

Для проверки успешности установки каждого компонента зайти по `ssh` на VM Визион.Сервер и запустить команду:

```
cd /opt/skala-r/vision
```

```
./check_service.sh
```

В результате, вывод должен быть следующий:

```
[root@visiondemo200redos vision]# ./check_service.sh
Status  Port      Service      Status      Journal
-----  -
>>> Vision Server services
active  3000      grafana      systemctl status grafana      journalctl -fu grafana
active  8428      victoriametrics_st      systemctl status victoriametrics_st      journalctl -fu victoriametrics_st
active  9093      alertmanager      systemctl status alertmanager      journalctl -fu alertmanager
active  8880      vmalert_st      systemctl status vmalert_st      journalctl -fu vmalert_st
active  9464      snmp_notifier      systemctl status snmp_notifier      journalctl -fu snmp_notifier
active  8088      vision_core      systemctl status vision_core      journalctl -fu vision_core
active  10111     vision_alertcollector      systemctl status vision_alertcollector      journalctl -fu vision_alertcollector
active  vision_taskdaemon      systemctl status vision_taskdaemon      journalctl -fu vision_taskdaemon
active  443 80 8092  nginx      systemctl status nginx      journalctl -fu nginx
active  5432     postgresql      systemctl status postgresql      journalctl -fu postgresql
```

Критерием успешности развертывания является значение **active** для перечисленных сервисов, перечисленных в разделе 2.5 данного руководства.

При статусе равном **failed** обратиться к пункту 3 для получения дополнительной информации.

Конфигурация компонентов после развертывания находится в директориях `/opt/skala-r/etc/vision/<server|proxy|agent>/<vision_component>`

## 15 Устранение неполадок

Расположение файлов всех компонентов и плагинов унифицировано между Визион.Сервером/Прокси/Агентом.

Описание	Расположение
Исполняемые и прочие файлы	<code>/opt/skala-r/vision/&lt;server-proxy-agent&gt;/&lt;имя-компонента&gt;</code>
Файлы конфигурации	<code>/opt/skala-r/etc/vision/&lt;server-proxy-agent&gt;/&lt;имя-компонента&gt;</code>
Логи компонентов	Системный журнал и <code>/opt/skala-r/var/log/vision/&lt;server-proxy-agent&gt;/&lt;имя-компонента&gt;</code>
Хранилище секретов ^1	<code>/opt/skala-r/vision/.secrets_vault</code>
Nginx ^1	<code>/etc/nginx</code>

<sup>1</sup> только для Визион.Сервер

## 15.1 Запуск ansible-playbook

Список часто встречающихся проблем:

- **Failed to parse <путь-до-инвентори-файла>** - убедитесь в корректности yaml-файла. Например, частая проблема - несоблюдение отступов.
- **Permission denied, please try again.** - убедитесь в корректности данных для доступа к узлу.
- **Failed to connect to the host** - убедитесь в правильности адреса узла и проверьте возможность подключения к нему по ssh.

При прочих неполадках используйте запуск ansible с флагом **vvvv** для повышения детализации вывода.

## 15.2 Установка/конфигурация/удаление агентов/плагинов через интерфейс

Управление агентами мониторинга происходит через агента Платформы, который необходимо развернуть на все сервера, на которые в последствии будут установлены Визион.Агент и Визион.Прокси. Проблемы при установке/конфигурации/удалении агентов/плагинов могут быть связаны с работоспособностью агента Платформы или с постановкой задач к нему через TaskDaemon.

### 15.2.1 Не изменяется статус агентов/плагинов

Если при попытке установить/конфигурировать/удалить агенты/плагины не меняется статус сущности, то необходимо проверить работу компонента TaskDaemon.

```
systemctl status vision_taskdaemon.service
```

### 15.2.2 Статус “Ошибка” в интерфейсе

В интерфейсе в статусе агентов/плагинов вы можете увидеть статус “Ошибка”.

Для получения детального описания ошибки вы можете выполнить следующие запросы:

- для агентов

```
curl -k https://<vision-ip>:8092/vision/api/v1/component/?object_id=<имя-объ
```

- для плагинов

```
curl -k https://<vision-ip>:8092/vision/api/v1/plugin_job/?object_id=<имя-об
```

где `<имя-объекта>` - это имя проблемного сервера, виртуальной машины или ПАКА.

Тело ответа будет содержать поле `data` с массивом информации об агентах/плагинов. Найдите необходимый элемент массива. Описание последней ошибки содержится в поле `task_error_detail` агента/плагина.

**Ошибка “No route to host”** указывает на то, что компьютер не может установить соединение с целевым сервером. Возможные причины:

- Неверный адрес: убедитесь, что в интерфейсе Визиона указан корректный адрес хоста.
- Блокировка брандмауэром: проверьте настройки брандмауэра узла, чтобы убедиться, что они не блокируют соединение.
- Проблемы с маршрутизацией: возможно, между Визион.Сервером и целевым узлом нет сетевого пути.

**Ошибки “Connection refused”/“timed out”** указывают на недоступность Агента Платформы. Возможные причины: агент Платформы не установлен или не запущен. Проверьте статус службы и при необходимости перезапустите.

```
systemctl status plagent.service
```

В случае работоспособности агента Платформы, выполните curl-запрос версии к нему

- с узла, на котором он расположен:
  - по 127.0.0.1;
  - по IP узла по которому к нему должен обращаться Визион;
- с Визион.Сервера по IP проблемного узла.

```
curl -k -H 'Authorization: Bearer <your-token>' https://<ip-узла>:7550/v1/ve
```

Доступность агента Платформы с проблемного узла будет означать корректность его работы, одновременно недоступность его с Визион.Сервера будет означать проблему с сетевой связностью.

## 15.3 Агенты установлены, но метрики отсутствуют

Необходимо убедиться от какой части системы отсутствуют данные. Перейдите в веб-интерфейс VictoriaMetrics, открыв Интернет браузер, и введя в строке адреса IP-адрес сервера Визиона (`https://<vision-ip>/vision/victoriametrics/vmui`). В открывшемся интерфейсе вы сможете проверить наличие временных рядов в системе, используя PromQL-запросы.

При недоступности интерфейса VictoriaMetrics, проверьте работу сервиса `victoriametrics_st.service` и `nginx.service`.

**Внимание!** При проведении работ учитывайте, что все изменения в файлах конфигурации агентов/экспортеров выполненные вручную, могут быть перезаписаны при переустановки компонента из интерфейса.

### 15.3.1 Отсутствие данных с Прокси

PromQL-запрос: `up{ _pak_id="<ИМЯ-ПАК>", job="vmagent_proxy" }`

Если временной ряд отсутствует, то проверьте работоспособность systemd сервиса `vision_vmagent_proxy.service` на узле, где он установлен (выбирается при установке Агентов).

Если сервис имеет статус `active`, но данных в базе временных рядов нет: убедитесь, что с узла, где расположен Прокси, есть возможность отправлять данные в VictoriaMetrics по https. Например, результат такого curl должен быть `OK`:

```
curl -k https://<user>:<pass>@<ip-vision>/vision/victoriametrics/health
```

### 15.3.2 Отсутствие данных с Агентов

PromQL-запросы:

- `up{ _node_id="<ИМЯ-СЕРВЕРА>", job="vmagent_agent" }`
- `up{ _vm_id="<ИМЯ-ВМ>", job="vmagent_agent" }`

Если временной ряд отсутствует, то проверьте работоспособность systemd сервиса `vision_vmagent_agent.service` на узле, где он установлен.

Если сервис имеет статус `active`, но данных в базе временных рядов нет: убедитесь, что с узла, где расположен Агент, есть возможность отправлять данные в Прокси по https в порт 8430 (см. п 1.3). Например, результат такого curl должен быть `OK`:

```
curl -k https://<user>:<pass>@<ip-vision>:8430/health
```

### 15.3.3 Отсутствие данных с плагинов (экспортеры/скреперы)

PromQL-запросы:

- `up{<node_id>=<ИМЯ-СЕРВЕРА>, job=<ИМЯ-ПЛАГИНА>}`
- `up{<vm_id>=<ИМЯ-ВМ>, job=<ИМЯ-ПЛАГИНА>}`

Если временной ряд отсутствует, то проверьте работоспособность systemd сервиса `vision_<ИМЯ-ПЛАГИНА>.service` на узле, где он установлен (только в случае экспортера, см. п.1.4).

Если временной ряд присутствует, но его значение равно 0, то проверьте параметры плагина в интерфейсе Визиона на корректность (см. п.7.1). Например, данные авторизации, адрес сервиса/источника метрик, порт и т.п.

### 15.3.4 Превышение таймаута сбора метрик с помощью плагинов

PromQL-запрос: `scrape_duration_seconds{<pak_id>=<ИМЯ-ПАК>} > scrape_timeout_seconds{<pak_id>=<ИМЯ-ПАК>} * 0.5`

График с данным запросом отображает проблемы с продолжительностью сбора метрик.

Если временные ряды присутствуют, то необходимо в интерфейсе Визиона в параметрах соответствующих плагинов (имя плагина соответствует метке job) увеличить таймаут в поле “Scrape Timeout”, но не выше, чем “Scrape Interval”.

## 15.4 Отсутствие метрик при работоспособности всех компонентов

Убедитесь, что все узлы имеют синхронизированное текущее время. Неправильные временные метки приводят к неточным или вводящим в заблуждение данным мониторинга.

## 16 Журналирование и аудит

### 16.1 Конфигурирование логирования vision\_core

Файл конфигурации логирования vision\_core расположен в `/opt/skalarr/etc/vision/server/vision_core/log_config.yml`.

Для изменения конфигурации логирования необходимо обладать правами root.

Пример файла конфигурации

```
# Аудит
audit:
  # Уровень сообщений, которые логируются DEBUG|INFO|WARNING|ERROR|CRITICAL
  level: INFO

# Параметры форматов записи логов
formatters:

  # Формат CEF
  cef:
    # Вкл/выкл запись логов в формате CEF
    enable: true

  # Управление ротированием файлов логов
  rotate:
    filepath: "/opt/skala-r/var/log/vision/audit/cef/audit.log"
    maxBytes: 104857600 # максимальный размер файла логов
    backupCount: 5      # количество сохраненных логов "<filepath>.<n>"

  # Формат JSON (ГосТех)
  json:
    enable: true
    rotate:
      filepath: "/opt/skala-r/var/log/vision/audit/audit.log"
      maxBytes: 104857600
      backupCount: 5

# Фильтрация записей по группам/действиям/статусам событий
# Например, ["11-15", "2"] – исключить с 11 по 15 включительно и 2 ID
# Перечень срезов vision-core/docs/events
filters:
  groups: []
  actions: []
  status: []

# Журналирование, аналогично параметрам аудита
journal:
  level: INFO
  formatters:
    text:
      enable: true
      rotate:
        filepath: "/opt/skala-r/var/log/vision/journal/journal.log"
```

```
maxBytes: 104857600
backupCount: 5
filters:
  groups: []
  actions: []
  status: []
```

Все ключи являются необязательными.

Предоставляется возможность настройки логгеров `audit` и `journal`:

- `level` - уровень логирования. Допустимые значения: `DEBUG`, `INFO`, `WARNING`, `ERROR`, `CRITICAL`.
- `formatters` - параметры форматов записи логов. Допустимые ключи для `audit`: `cef`, `json`, для `journal`: `text`.
  - `enable` - включение/выключение записи логов в данном формате. Допустимые значения: `true`, `false`.
  - `rotate` - управление ротированием файлов логов
    - `filepath` - путь до файла в который будут записываться логи данного формата.
    - `maxBytes` - максимальный размер файла логов в байтах.
    - `backupCount` - количество сохраняемых файлов логов.
- `filters` - параметры фильтрации записей по группам/действиям/статусам событий в формате ключ-значение, где
  - ключи `groups`/`actions`/`status` (значения ключей приведены в разделе 16.1.1)
  - значения - массив строк, перечисляющий ID `groups`/`actions`/`status`

## 16.1.1 Значения ключей, используемых для фильтрации событий

Для исключения событий из процесса журналирования соответствующие идентификаторы (ID) должны быть перечислены в разделе `filters` файла конфигурации (см. Раздел 16.1).

### Ключи для групп событий (groups)

Имя группы	Текст подгруппы	ID подгруппы
Приложение	приложение vision_core	10
Авторизация	вход пользователя в систему	21

Авторизация	выход пользователя из системы	22
Авторизация	информация о входе в систему	23
Авторизация	информация о текущем пользователе	24
Запрос метрик (UI)	значения метрик за момент времени	31
Запрос метрик (UI)	значений метрик за период времени	32
Запрос метрик (внешний)	значений метрик за момент времени	41
Запрос метрик (внешний)	значений метрик за период времени	42
Запрос метрик (внешний)	список меток метрик	43
Запрос метрик (внешний)	значений метки метрик	44
Запрос метрик (внешний)	статуса TSDB	45
Запрос метрик (внешний)	серий метрик	46
Запрос метрик (внешний)	значений метрик через федерацию	47
Конфигурирование Визион.Сервера	параметры SMTP-шлюза для отправки уведомлений	51
Конфигурирование Визион.Сервера	параметры SNMP-notifier для отправки уведомлений	52
Импорт данных	импорт данных (список метрик и правил оповещения)	60

Самодиагностика Визион.Сервера	здоровье компонентов сервера	71
Самодиагностика Визион.Сервера	логи компонентов сервера	72
Инвентори ПАК	инвентори файла ПАКа	80
foliage	список серверов из приложения Foliage	91
foliage	список коммутаторов из приложения Foliage	92
foliage	список сетей из приложения Foliage	93
список меток и их значений	список меток метрик	101
список меток и их значений	значений метки метрик	102
Взаимодействие TaskDaemon с Визион.Сервером	уведомление от TaskDaemon	110
Объекты мониторинга	объект мониторинга	121
Объекты мониторинга	допустимые вложенные типы объектов мониторинга	122
Объекты мониторинга	ссылки на объект мониторинга	123
Объекты мониторинга	типы объектов мониторинга	124
Объекты мониторинга	справочник подтипов объектов мониторинга	125
Компоненты мониторинга	компонент мониторинга	130
Плагины	общая информация о плагине	141

Плагины	экземпляр плагина	142
Плагины	таргет плагина	143
Метрики	метрика	151
Правила оповещения	правило оповещения	161
Получатели уведомлений их группы	получатель уведомлений	171
Получатели уведомлений их группы	группа получателей уведомлений	172
Оповещения о сработавших правилах	оповещение о сработавшем правиле	181
Оповещения о сработавших правилах	количество оповещений о сработавших правилах	182
Токены	токен доступа	190
Версия Визиона	версия Визиона	200

### Ключи для действий (actions)

Имя действия	ID действия
Создание	1
Чтение	2
Редактирование	3
Удаление	4
Получение	5
Отзывается	6
Установка	7
Деинсталляция	8
Запуск	10

Остановка	11
Авторизация	14
Ротация логов	15

### Ключи для статусов событий (status)

Текст статуса	ID статуса
успешно	0
неуспешно	1
доступ запрещен	2

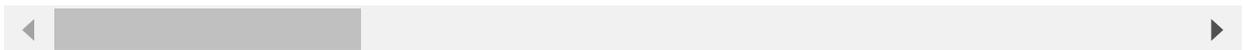
## 16.2 Описание формата логов

### 16.2.1 Аудит vision\_core

#### 16.2.1.1 Формат CEF

Пример записи

```
CEF:0|Skala-r|Vision|dev-281|20000020|Чтение: версия Визиона|3|externalId=4
pp=HTTPS dpt=8088
```



#### Заголовок CEF

Field Name	Description
CEF:Version	Версия формата CEF. Константа <b>CEF:0</b>
Device Vendor	Название производителя. Константа <b>Skala-r</b>
Device Product	Название продукта. Константа <b>Vision</b>
Device Version	Версия Визиона

Device Event Class ID	Уникальный идентификатор категории события. Формируется путём <code>&lt;группа&gt;&lt;действие&gt;&lt;статус&gt;</code>
Name	Краткое описание события.
Severity	Целое число, отражающее важность события. Число от 0 до 10, где 10 обозначает наиболее важное событие.
Extension	Список пар "ключ=значение", содержащих дополнительную информацию о событии

### Поля extension CEF

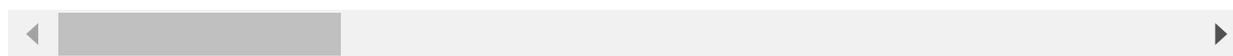
Field Name	Description
externalId	Уникальный номер строки о событии в рамках одного журнала. Сквозная инкрементация (счетчик). Обнуляется при перезапуске сервиса.
msg	Текст сообщения с детальной информацией о событии.
deviceProcessName	Полное имя процесса. Константа <code>vision_core</code>
outcome	Результат события. <code>success</code> / <code>failure</code>
start	Время возникновения события в формате Unix Timestamp.
dhost	Имя хоста получателя (Визиона)
dst	IP адрес хоста получателя (Визиона)
dpt *	Порт на стороне получателя (Визиона)
suser *	Имя пользователя, инициировавшего событие

src *	IP адрес хоста-источника
shost *	FQDN или имя хоста-источника
spt *	порт на стороне источника
app *	Протокол уровня приложения
^1 при наличии	

### 16.2.1.2 Формат JSON

Пример записи

```
{"createdAt": "1722617884851", "userNode": "vision-dev-ps.int.skala-r.tech",
```



Поле	Описание
createdAt	Время возникновения события в формате Unix Timestamp.
userNode	Имя хоста получателя (Визиона).
metamodelVersion	Константа <b>1</b> .
module	Название продукта. Константа <b>Vision</b>
name	Краткое описание события.
userLogin	Имя пользователя, инициировавшего событие.
userName	Полное имя пользователя, инициировавшего событие (если доступно).
params- name=message	Дополнительная информация о событии.
params- name=serviceVersion	Версия Визиона.

### 16.2.2 Журналирование (journal) vision\_core

## Пример записи

2024-08-02T19:58:04+0300 vision-dev-ps.int.skala-r.tech INFO Чтение: версия



- Время возникновения события
- Имя хоста получателя (Визиона).
- Уровень журнала
- Описание события

## 16.3 Расположение журналов компонентов

Компонент	Сервис ОС	Расположение журналов по умолчанию
Визион.Сервер	vision_core	Системный журнал, <code>/opt/skala-r/var/log/vision/audit/cef/audit.log</code> , <code>/opt/skala-r/var/log/vision/audit/audit.log</code> , <code>/opt/skala-r/var/log/vision/journal/journal.log</code>
Визион.Сервер	vision_taskdaemon	Системный журнал
Визион.Сервер	postgresql	Системный журнал и стандартное расположение логов PostgreSQL <code>/var/lib/pgsql/data/log/</code>
Визион.Сервер	nginx	Системный журнал и стандартное расположение логов nginx <code>/var/log/nginx/</code>

Визион.Сервер	grafana	Системный журнал и <code>/opt/skala-r/var/log/vision/server/grafana/</code>
Визион.Сервер	victoriametrics_st	Системный журнал
Визион.Сервер	vmalert_st	Системный журнал
Визион.Сервер	vision_alertcollector	Системный журнал и <code>/opt/skala-r/var/log/vision/server/vision_alertcollector/vision_alertcollector.log</code>
Визион.Сервер	alertmanager	Системный журнал
Визион.Сервер	snmp_notifier	Системный журнал
Визион.Прокси	vision_vmagent_proxy	Системный журнал и <code>/opt/skala-r/var/log/vision/proxy/vmagent_proxy/vmagent.log</code>
Агент Платформы	plagent	Системный журнал
Визион.Агент	vision_vmagent_agent	Системный журнал и <code>/opt/skala-r/var/log/vision/agent/vmagent_agent/vmagent.log</code>
Визион.Агент	vision_node_exporter	Системный журнал и <code>/opt/skala-r/var/log/vision/agent/node_exporter/node_exporter.log</code>
Визион.Агент	vision_ipmi_exporter	Системный журнал
Визион.Агент	vision_snmp_exporter	Системный журнал

Визион.Агент	vision_utlz_exporter	Системный журнал и <code>/opt/skala-r/var/log/vision/agent/utlz_exporter/utlz_exporter.log</code>
Визион.Агент	vision_utlz_exporter_v0	Системный журнал и <code>/opt/skala-r/var/log/vision/agent/utlz_exporter_v0/utlz_exporter_v0.log</code>
Визион.Агент	vision_one_exporter	Системный журнал
Визион.Агент	vision_ha_cluster_exporter	Системный журнал
Визион.Агент	vision_postgres_exporter_local	Системный журнал
Визион.Агент	vision_postgres_exporter_multi	Системный журнал
Визион.Агент	vision_sql_exporter	Системный журнал
Визион.Агент	vision_supv_exporter	Системный журнал и <code>/opt/skala-r/var/log/vision/agent/supv_exporter/supv_exporter.log</code>
Визион.Агент	vision_nginx_exporter	Системный журнал и <code>/opt/skala-r/var/log/vision/agent/nginx_exporter/nginx_exporter.log</code>
Визион.Агент	vision_exporter	Системный журнал и <code>/opt/skala-r/var/log/vision/agent/vision_exporter/vision_exporter.log</code>

Визион.Агент	bash_exporter	Системный журнал и <code>/opt/skala-r/var/log/vision/agent/bash_exporter/bash_exporter.log</code>
Визион.Агент	systemd_exporter	Системный журнал и <code>/opt/skala-r/var/log/vision/audit/agent/systemd_exporter/audit.log</code>
Визион.Агент	process_exporter	Системный журнал и <code>/opt/skala-r/var/log/vision/audit/agent/process_exporter/audit.log</code>
Визион.Агент	kafka_exporter	Системный журнал и <code>/opt/skala-r/var/log/vision/audit/agent/kafka_exporter/audit.log</code>

### 16.3.1 Аудит компонентов Визиона

Запись происходит в журнал аудита `/opt/skala-r/var/log/vision/audit/<COMPONENT>/<SERVICE>/audit.log` в формате CEF.