

Руководство администратора Визион

Документ	Описание	Ссылка на PDF
Руководство администратора	v1.8.1	PDF

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование программы

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

Краткое наименование: Визион.

1.2 Область применения

Основным назначением ПО «Скала^Ар Визион» (далее Визион) является предоставление возможности контроля жизненного цикла ПАК (машин) производства Скала^Ар, 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	Визион. Сервер от 8 ядер
	Визион. Прокси от 2 ядер
	Визион. Агент от 1 ядра
RAM рекомендованное	
	Визион. Сервер от 32 Гб;
	Визион. Прокси от 16 Гб
	Визион. Агент от 4 Гб
ROM рекомендованное	
	Визион. Сервер - в зависимости от количества ПАК на контуре и требуемой глубины хранения метрики, от 600 Гб SSD на корневой (root, <code>/</code>) раздел
	Визион. Прокси в зависимости от требуемой глубины кеширования данных от 20 Гб SSD
	Визион. Агент в зависимости от требуемой глубины кеширования данных от 5 Гб SSD

Поддерживаемые ОС	
	ОС АЛЬТ 8 СП СЕРВЕР р9, р10
	Astra Linux Special Edition 1.7.3 (Воронеж)
	Astra Linux Special Edition 1.8 (Воронеж) – только поддержка агентов на узлах
	РЕД ОС 7.3 (Муром)
	РЕД ОС 8.0
	ALT Linux Server 10.*
Поддерживаемые браузеры	
	Google Chrome с версии 134.0.6998.166 и выше
	Firefox с версии 136.0.2 и выше

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

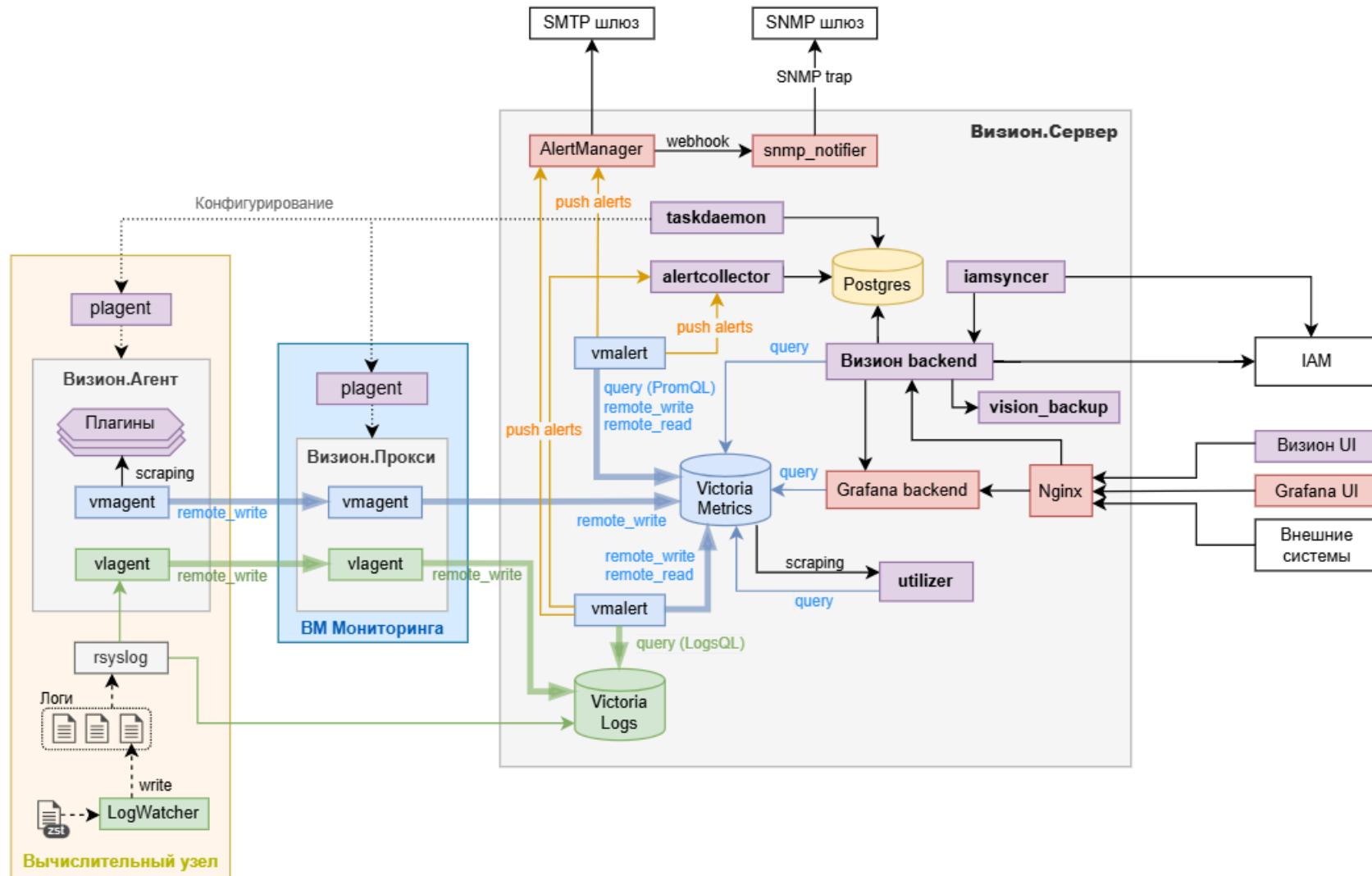
Для развертывания мониторинга на целевой инфраструктуре поставляется архив, содержащий в себе компоненты программного обеспечения:

- **Визион.Сервер** – предоставляет функциональность 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 находиться 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.
- **VMAalert** – компонент, предназначенный для генерации оповещений на основе данных в VictoriaMetrics и правил алертинга (триггеров).
- **AlertCollector** – компонент, регистрирующий оповещения в служебную БД.
- **IAMSyncer** – iamsyncer, компонент для синхронизации данных о ролях и пользователях между IAM и vision-core.
- **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.
- **Utilizer** – компонент, осуществляющий расчет утилизации CPU/RAM/ROM на основе данных от node_exporter.
- **victorialogs** – база данных для хранения логов.
- **vmalert_v1** – сервис оценки выражений VictoriaLogs (правила оповещений, метрики).

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 10.*
- Alt Linux Server 8 СП р10 (c10f1)
- Alt Linux Server 8 СП р9 (c9f2)
- Alt Linux Server СП р10 (c10f2)
- Astra Linux Special Edition 1.7.3 (Воронеж)
- РЕД ОС 7.3
- РЕД ОС 8.0

На целевых системах должны присутствовать следующие пакеты:

- `freeipmi` (если предполагается присутствие плагина `ipmi` на узле, как правило, это VM Визиона на служебном узле);
- `lsblk` (для плагина `utlz_exporter`, необходима поддержка ключей `-o` и `-J`);
- `sudo` (`> 1.6.9p15`).

Перед началом развёртывания 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	HTTPS
grafana	Визион. Сервер	3000	HTTPS
victoriametrics_st	Визион. Сервер	8428	HTTPS
victorialogs	Визион. Сервер	9428, 29514	HTTPS
vmalert_st	Визион. Сервер	8880	HTTPS
vmalert_v1	Визион. Сервер	8881	HTTPS
vision_alertcollector	Визион. Сервер	10111	HTTPS
vision_iamsyncer	Визион. Сервер	17080	HTTPS
vision_utilizer	Визион. Сервер	17099	HTTPS

vision_backup	Визион. Сервер	17050	HTTPS
alertmanager	Визион. Сервер	9093	HTTPS
snmp_notifier	Визион. Сервер	9464	HTTPS
vision_logwatcher	Визион. Агент	17078	HTTPS
vision_vmagent_proxy	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 комплементарный 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 экспорттер, он является плагином-скреппером, сбор метрик осуществляется непосредственно с эндпоинта объекта мониторинга.

В состав Визиона входят следующие плагины:

Название	Мониторинг	ПАК	Сервер/ВМ2	Сервис ОС	Порт3
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 сервера1	все	все серверы	vision_utlz_exporter_v0	17070
process_exporter	процессы ОС	все	все серверы	vision_process_exporter	9120
systemd_exporter	systemd юнитов	все	все серверы	vision_systemd_exporter	9110
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
clickhouse_sql_exporter	Postgres	МБД.КХ	узел кластера или узел с Прокси	vision_clickhouse_sql_exporter	9397
greenplum_sql_exporter	Greenplum	МБД.Г	узел кластера или узел с Прокси	vision_greenplum_sql_exporter	9398
spectrum_scraper	Спектр	МБД.П	ВМ с Прокси	-	
spectrum_agent_scraper	агент Спектра	МБД.П	узел кластера	-	

supv_exporter	сервисы СУПВ	МВ.С/ВРМ	ВМ с Прокси	vision_supv-exporter	9179
haproxy_scra_per_local	HAProxy	МХД.О	узлы балансировки	-	
s3gateway_compression_se_rver_scrapers	шлюз S3	МХД.О	узлы балансировки	-	
s3gateway_os_tor_server_s_crapers	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/ МБД.Г/МДИ.О	все серверы	vision_exporter	7531
kafka_exporter	Kafka	МБД.С	ВМ с Прокси	vision_kafka_exporter	9308
blackbox_exporter	ICMP	все	ВМ с Прокси	vision_black_box_exporter	9115
graphite_exporter	Hadoop и Kafka	МБД.С/МБД.Х	ВМ с Прокси	vision_graphite_exporter	9108

picodata_scraper	Picodata	МБД.Т	узел кластера	-	
genome_u_scraper	Геном У	все	ВМ с Прокси	-	
basis_libvirt_scrapers	Базис.DynamiX	МДИ.О	Вычислительны й узел Базис.DynamiX	-	
bash_exporter	ОС сервера	все	все	vision_bash- exporter	17055
bvs_exporter	Базис.DynamiX	МДИ.О	Узел управления Базис.Virtual Security	vision_bvs_e xporter	6565
kubernetes_exporter	Базис.DynamiX	МДИ.О	Узел управления	vision_kubernetes_exporter	9130
universal_scraper	Сбор метрик с эндпоинтов	все	все серверы	-	-
r_virtualization_exporter	Р- Виртуализация	МВ.С	Узел вычисления и хранения	vision_r_vir tualization_ exporter	9140

r_vstorage_exporter	Р-Хранилище	МВ.С	Узел вычисления и хранения	vision_r_vstorage_exporter	17090
mvs_exporter	Базис.vControl	МВ.С	ВМ с Прокси	vision_mvs_exporter	17060
mbdh_exporter	ADCM	МБД.Х	ВМ с Прокси	vision_mbdh_exporter	17023
ydb_scraper	YandexDB	МБД.Я	ВМ с Прокси	-	
avanpost_scraper	Avanpost FAM	все	ВМ с Прокси	-	
patroni_scraper	Postgres	МДИ.О	ВМ с Прокси	-	8008

1 **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](#).

2 На агенты каких серверов или ВМ рекомендуется добавлять плагины.

3 Порт, занимаемый экспортером по умолчанию. Изменяется в настройках плагина в интерфейсе Визиона.

2.6.1 Версионирование ПО плагинов

Плагин	ПО	Версия
blackbox_exporter	-	-

bvs_exporter	BVS	3.1.0
basis_libvirt_scraper	DynamiX	4.0.0
ceph_scraper	Ceph	12.0.0 и выше
clickhouse_sql_exporter	ClickHouse	C20.x и выше
genome_u_scraper	-	-
graphite_exporter	adh	3.2.4_arenadata3_b1-1
	adm	3.3.1-1
	adcm	2.0.0
	ads	3.6.2_arenadata1_b1-1
	adps	1.1.2_b1-1
greenplum_sql_exporter	Greenplum	6
	ADB	Не выше 6.x
ha_cluster_exporter	Pacemaker	1.x и выше
	Corosync	2.x и выше
haproxy_scraper_local	HAProxy	2.0 и выше
ipmi_exporter	-	-
kafka_exporter	Apache Kafka	0.10.1.0 и выше

nginx_exporter	NGINX	1.x и выше
node_exporter	-	-
one_exporter	OpenNebula	5.x и выше
picodata_scraper	Picodata	1.2.0 и выше
postgres_exporter_agent	Postgres	9.4 и выше
postgres_exporter_multi	Postgres	9.4 и выше
process_exporter	-	-
s3gateway_compression_server_scraper	Спектр S3	1.0.5 и выше
s3gateway_ostor_server_scrape	Спектр S3	1.0.5 и выше
snmp_exporter	-	-
spectrum_agent_scraper	Спектр	1.0
spectrum_scraper	Спектр	1.5
sql_exporter	Postgres	9.x и выше
supv_exporter	VMS-Backend	1.97.4
systemd_exporter	-	-
utlz_exporter	-	-

utlz_exporter_v0	-	-
vision_exporter	DynamiX	4.0.0
	RAIDIX ERA RAID	4.1.1
	Greenplum	6

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 файрвол 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 1.8.1. 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 или 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.

В процессе установки будет создана эталонная конфигурация в `/opt/skala-r/vision/server/default`

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. При установке на операционные системы Altlinux Server СП р10 (c10f2) и РЕДОС 8.0 будет предложена установка Postgres Pro 16. Необходимо обратить внимание на следующее:

- для использования Postgres Pro требуется действующая лицензия у заказчика;
- в дистрибутиве не предусмотрена возможность отката к обычной версии PostgreSQL;
- в дистрибутиве предусмотрена миграция с обычной версии PostgreSQL на Postgres Pro 16 с сохранением данных БД. Обязательно создавайте бекап БД перед миграцией.

Осуществить запуск этих операций можно с помощью ключей **setup.sh**:

- **-p** - установка пакетов зависимостей из репозитория дистрибутива;
- **-g** - установка пакетов зависимостей из репозитория ОС.

В случае 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 лог сохраняется в **/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. Осуществить запуск можно с помощью ключа **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 завершении установки лог сохраняется в **/opt/skala-r/var/log/vision/vision_install.log.<ДАТА_УСТАНОВКИ>**.

7. Запустить скрипт **./setup.sh** и выбрать действие **5. Импортировать данные для конкретного заказчика**.

Выбрать текущего заказчика/проект для которого устанавливается Визион. Если заказчика нет в списке, пункт можно пропустить.

По завершении установки лог сохраняется в 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:

```
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 компонентов Визион.Сервера:

Запустите скрипт `./setup.sh` из директории с дистрибутивом и выберите действие [6. Проверка работы сервисов](#).

Выполнить проверку можно вручную, запустив следующие команды:

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

Критерием корректности развертывания является значение `active` для всех сервисов, приведенных в таблице в выводе скрипта `check_service.sh`.

Сервис `vision_iamsyncer` не запускается автоматически после установки Визион. Подробнее в [инструкции](#)

По завершении процесса установки веб-интерфейс Визион должен быть доступен по ссылке:

`https://<vision_ip>`

где `<vision_ip>` – IP-адрес сервера Визион.

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

Журналы Визион.Сервера размещаются в директории `/opt/skala-r/var/log/vision/journal`, а журналы 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”.

3.7 Настройка сбора и приема логов в VictoriaLogs

Настройка VictoriaLogs на Визион.Сервере

Для приема логов от rsyslog VictoriaLogs должна быть запущена с указанными ниже ключами. Эти параметры необходимо задать в переменной `ENV` в файле `/opt/skala-r/etc/vision/server/victorialogs/environment.env` на узле Визион.Сервера:

```
-syslog.listenAddr.tcp=:29514
-syslog.tls=1
-syslog.tlsCertFile=/opt/skala-r/vision/cert/components_default.crt
-syslog.tlsKeyFile=/opt/skala-r/vision/cert/components_default.key
-syslog.tlsMinVersion=TLS12
-syslog.tlsCipherSuites 'TLS_AES_256_GCM_SHA384,TLS_AES_128_GCM_SHA256'
-syslog.tlsCipherSuites 'TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384'
-syslog.tlsCipherSuites 'TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256'
```

Настройка rsyslog и LogWatcher на узлах

В директории дистрибутива `rsyslog_config` представлены плейбуки, которые позволяют конфигурировать узлы для сбора и отправки логов в VictoriaLogs для ПАК МХД.О. Плейбук конфигурирует параметры `rsyslog` и устанавливает компонент `LogWatcher` для работы с `zst`-логами.

Ниже описан порядок действий для настройки `rsyslog` и `LogWatcher` для ПАК МХД.О. Для остальных ПАК допускается ручная настройка `rsyslog`.

Для настройки `rsyslog` и `LogWatcher` на узлах МХД.О необходимо выполнить следующие шаги:

1. Сформировать инвентори-файл в интерфейсе Визиона и убедиться, что в разделе `victorialogs` в переменной `host` указан IP-адрес Визион.Сервера.
2. Перейти в директорию дистрибутива;
3. Запустить плейбук с инвентори-файлом из корня директории дистрибутива:
 - Для установки:

```
ansible-playbook -i /path/to/inventory.yml rsyslog_config/deploy.yml
```

- Для удаления:

```
ansible-playbook -i /path/to/inventory.yml rsyslog_config/remove.yml
```

Плейбук `deploy.yml` необходимо запускать с узла, где установлен Визион.Сервер.

Плейбук `deploy.yml` выполняет следующие проверки:

1. Проверка наличия пакетов `rsyslog` и `rsyslog-gnutls`:
 - При отсутствии пакетов будет выполнена попытка установки из репозитория ОС.

- Если репозиторий недоступен или необходимых пакетов в нём нет, выполнение плейбука остановится. Необходимо установить пакеты вручную.

2. Проверка версии `rsyslog`.

Версия пакета должна быть не ниже значения, которое указано в переменной `rsyslog_min_version` (по умолчанию 8.1901). Если версия пакета ниже, выполнение плейбука остановится. Необходимо обновить `rsyslog` вручную.

3. Проверка наличия директивы подключения конфигураций из `/etc/rsyslog.d/*.conf` в основном конфигурационном файле `/etc/rsyslog.conf`.

Если информация отсутствует, выполнение плейбука остановится. Необходимо вручную добавить в конфигурационный файл `/etc/rsyslog.conf` строку:

```
include(file="/etc/rsyslog.d/*.conf" mode="optional")
```

4. Проверка конфигурационного файла `/etc/rsyslog.d/30-vision-vlogs.conf` с помощью команды:

```
rsyslogd -N1 -f /etc/rsyslog.d/<...>.conf
```

При неудачной проверке выполнение плейбука остановится.

Необходимо проверить:

- Наличие используемых модулей `imfile`, `imjournal` и прочих.
- Синтаксис шаблонов в директории плейбука `distrib/rsyslog_config/roles/rsyslog/templates/*.conf.j2`.

Для отключения проверки необходимо в файле `rsyslog_config/roles/rsyslog/tasks/deploy.yml` закомментировать ключ `validate`.

3.8 Конфигурирование rsyslog для отправки логов через vlagent

Для работы мониторинга логов необходимо настроить конфигурацию rsyslog для отправки логов в локальный vlagent. Подробнее в [инструкции](#)

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. В браузере перейти в веб-интерфейс Визион, доступный по ссылке:

`https://<vision_ip>`

Где `<vision_ip>` – IP-адрес сервера Визион.

2. Перейти в раздел “Объекты”.
3. В панели “Контуры” нажать кнопку :, выбрать **Добавить ПАК**.
4. Ввести информацию о ПАК.
5. В случае отсутствия на добавляемом ПАК агентов Платформы, необходимо его развернуть в соответствии с разделом 6 настоящего Руководства.
6. Перейти во вкладку “Агенты метрик” и нажать кнопку “Сконфигурировать агенты”.
7. Нажать на иконку  и включить необходимые плагины.
8. Перейти в пункт меню “Конструктор выражений”.
9. Ввести выражение:

10. Проверить, что работают все сконфигурированные плагины.

Плагин считается работающим, если в наличии временные ряды с меткой "job" и значениями имен агентов.

4.1 Подготовка к настройке

Настройка возможна двумя способами:

- Вручную.

Перед настройкой этим способом убедитесь в наличии инсталляционной карты с соответствующим описанием ПАК, либо всех необходимых параметров и доступов для постановки объектов на мониторинг.

- Экспорт данных из CSV-файлов.

Этот способ предполагает импорт конфигурации существующего ПАК в файлы формата CSV, их изменение и последующий экспорт.

4.2 Добавление ПАК

Чтобы добавить ПАК:

1. Перейдите в веб-интерфейс Визион:

- ## 1. Запустите браузер.

2. Перейдите по адресу:

<https://<vision ip>>

Здесь **<vision_ip>** – IP-адрес сервера Визион.

2. На панели навигации выберите **Параметры** → **Объекты**.
3. В панели **Контур** нажмите кнопку : и в открывшемся меню выберите **Добавить ПАК**.
4. Заполните форму добавления ПАК.

Доступность некоторых полей зависит от типа добавляемого ПАК.

- **Имя** – идентификатор ПАК.

Будьте внимательны! Идентификатор ПАК нельзя будет изменить без реинсталляции.

Имя ПАК должно быть уникальным на уровне Визион.

- **Тип** – выберите тип добавляемого ПАК в соответствии с поставкой.
- **Описание** – необязательное поле с кратким описанием ПАК.
- **Настройка правил** – если этот переключатель включен, после создания ПАК откроется окно, позволяющее включить для ПАК правила оповещения по умолчанию.
- **Модули** – нажмите кнопку **Добавить** и укажите данные о модулях, из которых состоит ПАК:
 - **Имя** – идентификатор модуля.

Будьте внимательны! Идентификатор модуля нельзя будет изменить без реинсталляции.

- **Тип** – выберите тип модуля из списка.

Список доступных типов модуля зависит от выбранного типа ПАК.

- **Описание** – необязательное описание модуля.
- **Серверы** – нажмите кнопку **Добавить** и укажите данные о серверах, входящих в состав ПАК:

- **Имя** – идентификатор сервера.

Будьте внимательны! Идентификатор сервера нельзя будет изменить без реинсталляции.

- **Модуль** – выберите модуль, к которому относится сервер.

Список доступных модулей формируется из значений, введённых в блоке **Модули**.

- **Тип** – выберите тип сервера.
 - **Адрес** – адрес для установки агента.
 - **Адрес ВМС** – адрес, по которому будут производить опрос агенты мониторинга, расположенные на Визион.Прокси.
 - **Виртуальные машины** – нажмите кнопку **Добавить** и укажите данные о ВМ, входящих в состав ПАК, в том числе сервисных ВМ (Визион, Геном и т. д.):
 - **Имя** – идентификатор ВМ.

Будьте внимательны! Идентификатор ВМ нельзя будет изменить без реинсталляции!
 - **Сервер** – принадлежность ВМ к физическому узлу (серверу).
 - **Адрес** – IP-адрес ВМ.
 - **Коммутаторы и СХД** – нажмите кнопку **Добавить** и укажите данные о коммутаторах, входящих в состав ПАК:
 - **Имя** – идентификатор коммутатора или СХД.

Будете внимательны! Идентификатор коммутатора или СХД нельзя будет изменить без реинсталляции.
 - **Модуль** – выберите модуль, к которому относится коммутатор или СХД.

Список доступных модулей формируется из значений, введённых в блоке **Модули**.
 - **Тип** – выберите тип коммутатора или СХД.
- Значение этого поля влияет на настройку **Модуль** плагина `snmp_exporter`:
- при значении **Storage** подключается мониторинг СХД Tatlin;
 - при прочих значениях подключается мониторинг коммутаторов.
- **Адрес** – IP-адрес коммутатора или СХД.

- **Ссылки на сервисы** – нажмите кнопку **Добавить** и укажите данные о сервисах:

- **Сервис** – тип сервиса.
- **URL префикс** – схема доступа к сервису, HTTP или HTTPS.
- **Адрес** – IP-адрес и порт сервиса.

Порт указывайте только в том случае, если он отличается от значения по умолчанию для выбранной схемы.

- **URL суффикс** – если сервис доступен по URL, укажите его в этом поле.

Добавленные ссылки будут доступны в виде иконок на главной странице:



Помимо ссылок, добавленных пользователем, на главной странице каждого ПАК отображаются ссылки на дашборды Grafana. Они добавляются автоматически при установке Визион и не могут быть отредактированы или удалены пользователем. Они также не отображаются на формах **Добавление ПАК** и **Редактирование ПАК**.

5. Нажмите кнопку **Добавить**.

Если при заполнении формы допущены ошибки, Визион отметит неправильно заполненные поля.

6. Если при заполнении формы **Добавление ПАК** вы включили переключатель **Настройка правил**, откроется окно **Активация правил оповещения**.

1. В блоке **Правила оповещения для ПАК** выберите системные правила оповещения.

Если для ПАК нет системных правил оповещения, раздел не будет доступен.

2. В блоке **Универсальные правила оповещения** выберите универсальные правила оповещения, подходящие для всех типов ПАК.

3. Нажмите кнопку **Сохранить**.

4.3 Редактирование ПАК

Чтобы изменить ПАК:

1. Перейдите в веб-интерфейс Визион:

1. Запустите браузер.

2. Перейдите по адресу:

`https://<vision_ip>`

Здесь `<vision_ip>` – IP-адрес сервера Визион.

2. На панели навигации выберите **Параметры** → **Объекты**.

3. В панели **Контур** найдите нужный ПАК и в строке с ним нажмите кнопку `:`.

4. В открывшемся меню выберите **Редактировать**.

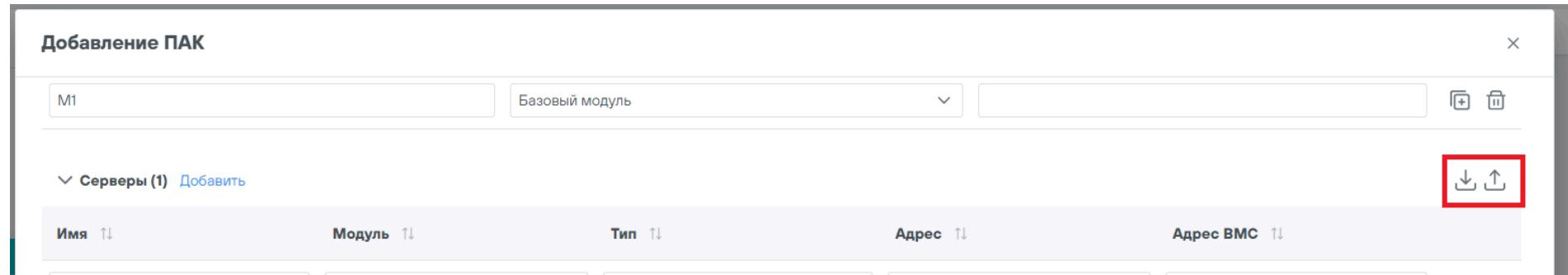
5. Внесите необходимые изменения.

Поля формы **Редактирование ПАК** идентичны полям формы **Добавление ПАК**, за исключением того, что поле **Имя** в некоторых блоках недоступно для изменения.

6. Нажмите кнопку **Редактировать**.

4.4 Импорт и экспорт настроек в CSV-файл

Формы создания и редактирования ПАК позволяют импортировать значения в CSV-файлы и экспортировать их из них. Для этого используйте кнопки **Импортировать CSV-файл** и **Импортировать из CSV-файла**, размещённые на одной линии с названием соответствующего блока настроек:



4.5 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
Дубликат_P1	МБД.П	ПАК БД Postgres

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 ↑

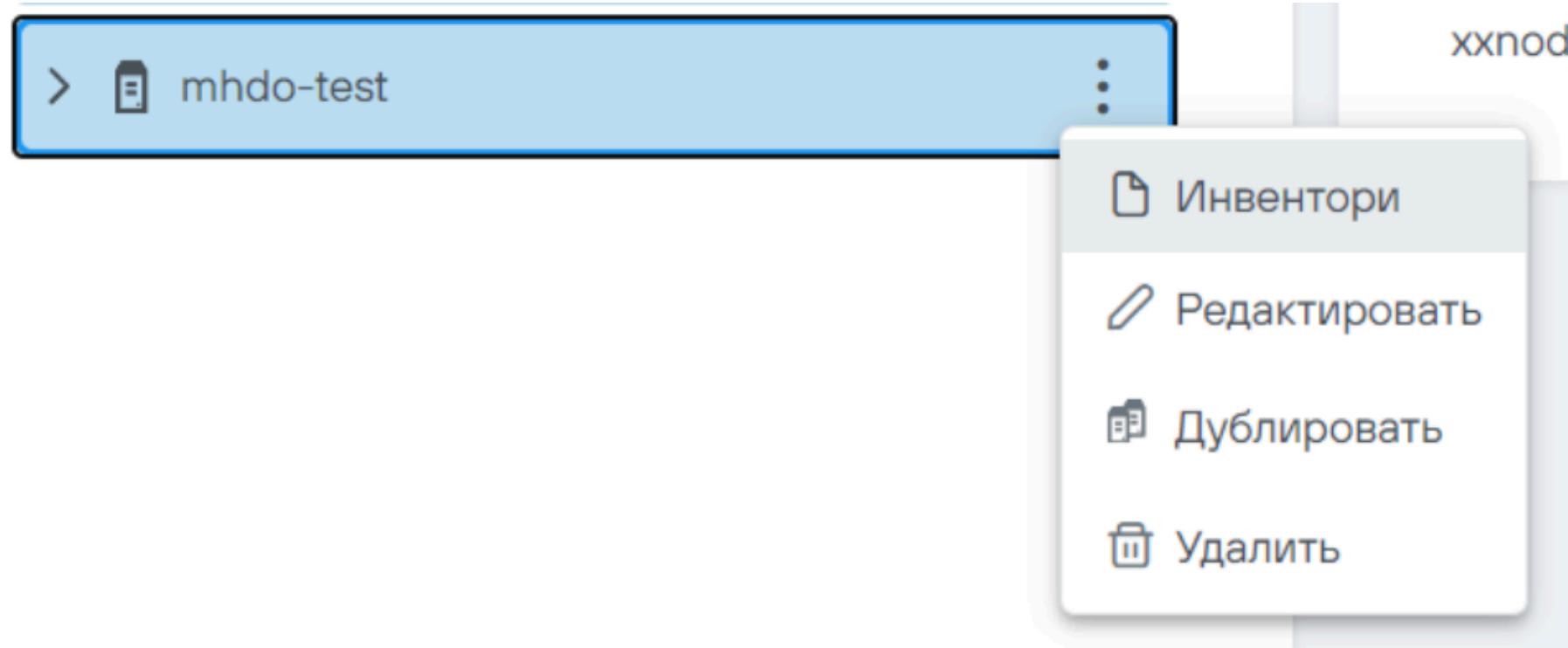
[Отменить](#) [Добавить](#)

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”. Инвентори-файл скачается, и так же будет доступен в директории `/opt/skala-r/vision/server/inventory/`. Имя файла состоит из имени ПАК и даты экспорта, например, `PAK-1_2024-06-10_10-10-10.yml`.

6 УСТАНОВКА АГЕНТОВ ПЛАТФОРМЫ

1. Сформировать инвентори-файл для ПАК (см. п.5).
2. Перейти в директорию дистрибутива:

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

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

4. Запустить установщик `setup.sh` и выбрать пункт **7. Установка plagent**:

```
./setup.sh
```

5. Выбрать номер сформированного инвентори-файла из пункта 1.

6. При необходимости, отредактировать инвентори-файл, введя `y` (если не нужно, то `n`). Указать текстовый редактор для запуска (по умолчанию nano).

В инвентори-файле заменить значения `==REPLACE==` для следующих ключей:

- Учётная запись для подключения к узлам по SSH:
 - `ansible_user`;
 - `ansible_ssh_pass`.
- Учётная запись под которой выполняется плейбук.
 - `ansible_become_user` – имя привилегированного пользователя, как правило `root`.
 - `ansible_become_password` – пароль привилегированного пользователя.
- `vars` – содержит глобальные параметры. Если у части узлов учётные данные отличаются, вы можете указать аналогичные ключи к конкретным узлам, таким образом переопределив значения глобальных.

7. По завершению установки удалить заполненный инвентори-файл.

Результат работы скрипта при успешном развёртывании 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 к итоговому инвентори-файлу;
- <PLAYBOOK> – выполняемое действие:
 - ./plagent/install.yml – установка агенты Платформы;
 - ./plagent/status.yml – вывод статуса агентов Платформы;
 - ./plagent/stop.yml – остановка агентов Платформы;
 - ./plagent/restart.yml – запуск агентов Платформы;
 - ./plagent/remove.yml – удаление агентов Платформы.
- <HOST> – имена серверов из инвентори-файла, 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 метрик” 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 «ВМ с прокси» в пункте 2.6 данного руководства.

Конфигурация агентов

×

Расположение Визион.Прокси

 ▼

Серверы

Расположение	Статус	Плагины
ser137 PROXY	УСТАНОВЛЕН	Выбрано плагинов: 36 ▼ ✓ ✗

ВМ

Расположение	Статус
Список виртуальных машин пуст	

Выбрано плагинов: 36 ▼

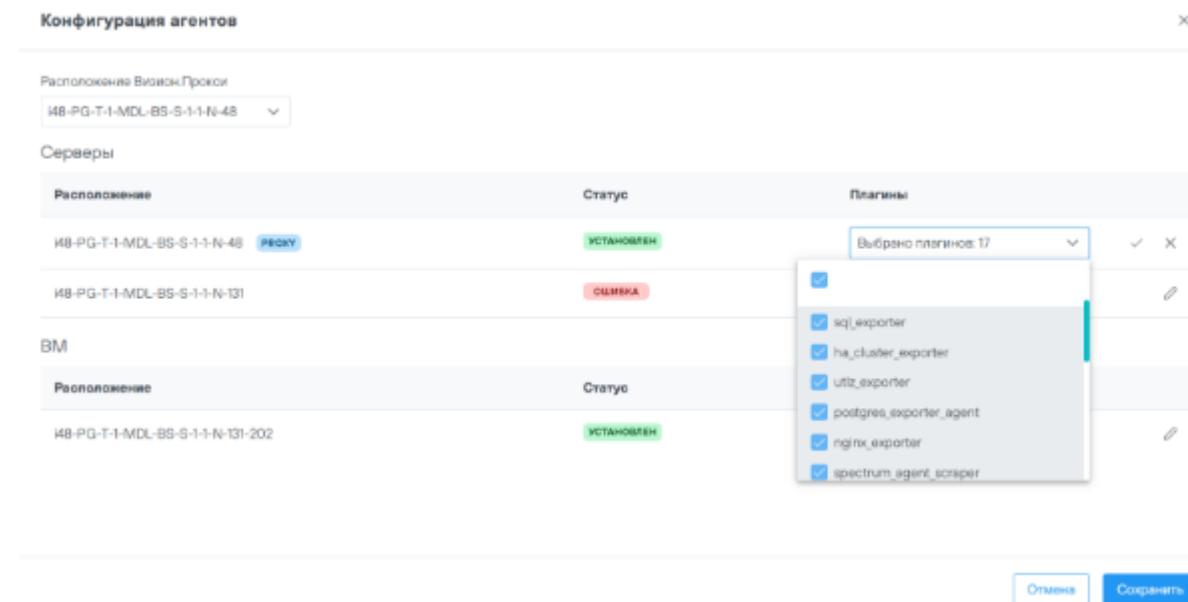
Поиск

- bash_exporter
- basis_libvirt_scrapers
- blackbox_exporter
- bvs_exporter
- ceph_scrapers
- clickhouse_sql_exporter

Отмена

Сохранить

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



- Плагины – состав конфигурационных файлов, содержащих в себе инструкции для настройки агентов и расстановки экспортёров. На данном шаге необходимо выбрать Плагины в соответствие 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/ВМ указано «ВМ с прокси» – плагин рекомендуется разместить на ВМ, выбранную Прокси. Остальные плагины конфигурируются на усмотрение пользователя, в зависимости от расположения сервисов на узлах.

После расстановки всех агентов и нажатия кнопки «Сохранить», поменяются 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»

Имя	Расположение	Статус	⋮
utlz_exporter	p1-mod1-node-205	В РАБОТЕ	⋮
node_exporter	p1-mod1-node-205	В РАБОТЕ	⋮
blackbox_exporter	p1-mod1-node-205	В РАБОТЕ	⋮
vision_exporter	p1-mod1-node-205	В РАБОТЕ	⋮
systemd_exporter	p1-mod1-node-205	В РАБОТЕ	⋮
snmp_exporter	p1-mod1-node-205	В РАБОТЕ	⋮

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 получат 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 :

1. Перейдите в пункт меню **Конструктор выражений** .
2. Введите выражение :

3. Проверьте работоспособность всех сконфигурированных плагинов (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., ERA RAID, Greenplum, Базис.DynamiX, информации о сроке действия TLS сертификатов, размера директорий, бондах, RAIDIX.

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
 - **session** - мониторинг сессий
 - **services** - мониторинг сервисов Linux
 - **dynamix** - мониторинг Basis.DynamiX с использованием API
 - **bond** - мониторинг состояния бондов
 - **certificate** - мониторинг срока истечения TLS сертификатов
 - **directories** - мониторинг размера заданных директорий
 - **raidix** - мониторинг программного компонента RAIDIX ПАК МБД.П
 - **logstat** - мониторинг состояния логирования

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 сервиса (обязательный параметр, для firewalld указывается в конфигурации)
    monitor_port: PORT                     # Мониторинг порта сервиса
    monitor_specific_config: firewalld     # Мониторинг конфигурации заранее заданным способом (указывать только для firewalld)
```



Для МХД.О (S3) рекомендуется следующая конфигурация:

- узел балансировки

```
services:
  firewalld:
    monitor_service: false
    monitor_specific_config: firewalld

  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.

Для мониторинга Basis.Dynamix следует сконфигурировать следующие поля:

Dynamix Client Id i	@
Dynamix Client Secret i	@
Dynamix Sso Host i	sso-dynamix.lab1.local	
Dynamix Api Host i	dynamix.lab1.local	

Необходимо указать адрес сервиса API и сервиса SS0. После этого следует сгенерировать токен для обращения к API и указать ID Клиента и Секрет Клиента.

Следует учитывать, что для каждого экземпляра экспортера следует генерировать уникальный токен.

Для мониторинга сертификатов следует указать адрес и порт (опционален) в соответствующем поле:

Servers i

Адрес сервера i *

dynamix.hardware01.skala:443

Адрес сервера i *

192.168.232.90

+ Добавить

Порт по умолчанию – 443. Адрес можно указать в виде доменного имени.

Для мониторинга директорий следует указать абсолютный путь интересующей директории, при наличии нескольких следует указывать с новой строчки:

Directories i

/opt/skala-r/etc

Directories Scan Period i

1m

Дополнительно есть возможность указать частоту мониторинга интересующей директории. Модуль работает через запуск команды `du -sb /path/dir/...`

Следует с осторожностью использовать данный модуль из-за стоимости операции подсчета занимаемого пространства. Не следует устанавливать в `Directories Scan Period` малые значения. Это может приводить к повышенной нагрузке, при указании больших директорий метрика может не собираться из-за ограничений таймаута.

Для получения метрик о статусе бонда и о статусе и привязке его интерфейсов необходимо включить модуль `bond`. Модуль собирает информацию локально через утилиту `ovs-appctl`.

Для получения метрик RAIDIX в МБД.П нужно включить модуль `raidix` в поле `Enabled Modules`.

Для получения метрик состояния логирования в МДИ.О нужно включить модуль `logstat` в поле `Enabled Modules`.

Enabled Modules [i](#)

logstat [v](#)

Module Services [i](#)

Module Logstat

Параметры модуля logstat

Logstat Server Type [i](#)

DX - Basis Dynamix Enterprise [v](#)

Logstat Paths [i](#)

```
/var/log/openvswitch/ovsdb-server.log
/var/log/openvswitch/ovs-vswitchd.log
/var/log/alternatives.log
/var/log/astra-history.log
/var/log/astra-safepolicy.log
/var/log/auth.log
/var/log/cron.log
/var/log/daemon.log
/var/log/dpkg.log
/var/log/fontconfig.log
/var/log/kern.log
/var/log/sudo.log
/var/log/user.log
```

Параметры модуля **logstat**:

- **Logstat Server Type**: выбор преднастроенного списка логов машины МДИ.О, Basis Dynamix Enterprise (DX) или Basis Virtual Security (BVS).
- **Logstat Paths** : поле, для ввода списка логов вручную. При заполнении данного поля преднастройка из **Logstat Server Type** будет проигнорирована.

Преднастроенный список логов DX:

- /var/log/decs3o.log
- /var/log/containers
- /var/log/haproxy.log
- /var/log/pods
- /var/log/redis

- `/var/decort/audit`
- `/var/log/decort/vms`

Преднастроенный список логов BVS:

- `/var/log/bvs`

7.2.2 `nginx_exporter`

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

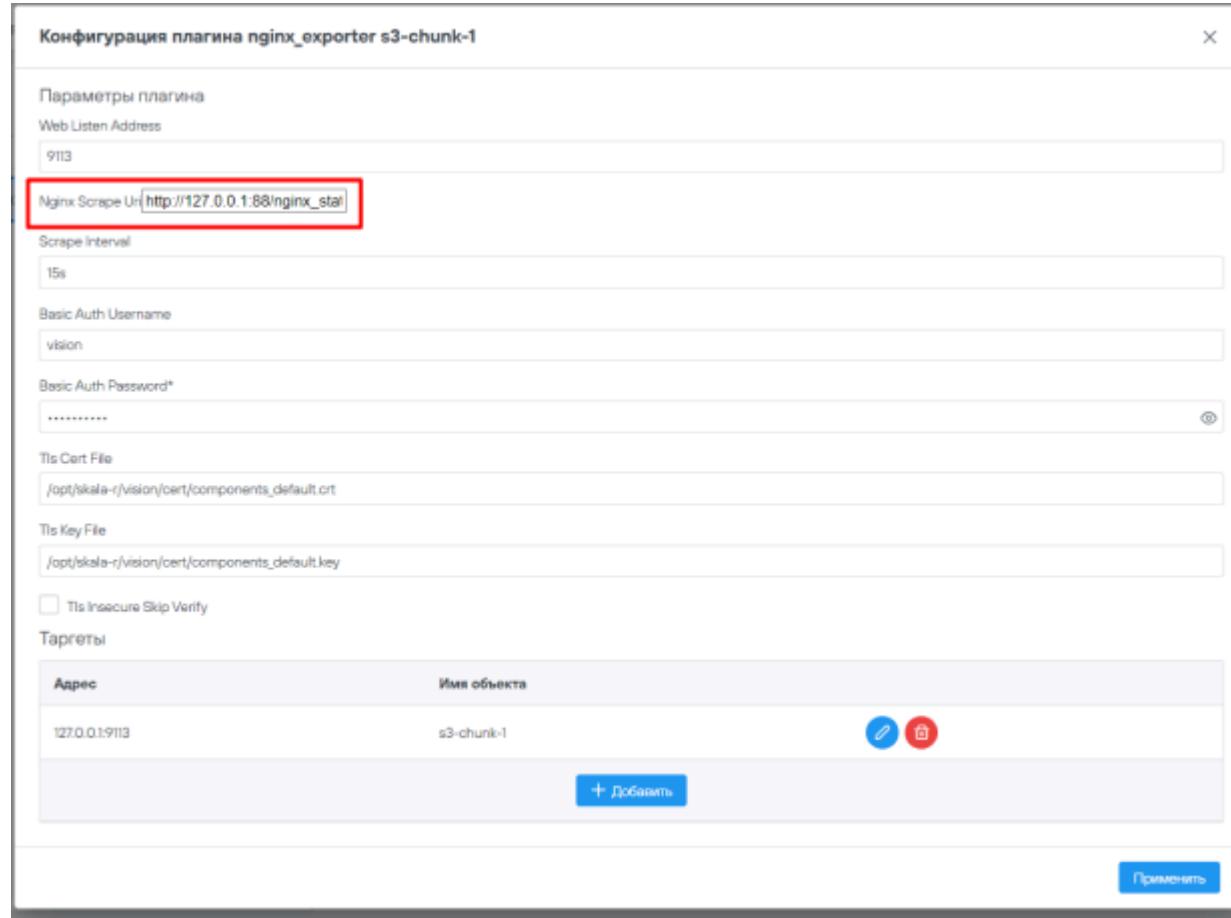
Плагин представляет собой экспортер, который получает состояние Nginx с помощью модуля `http_stub_status_module` и преобразует в метрики.

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_scrapers_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	Изменить	Удалить
127.0.0.1:8008	s3_gate_redOS1		

Применить

7.2.4 s3gateway_ostor_server_scrapers

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

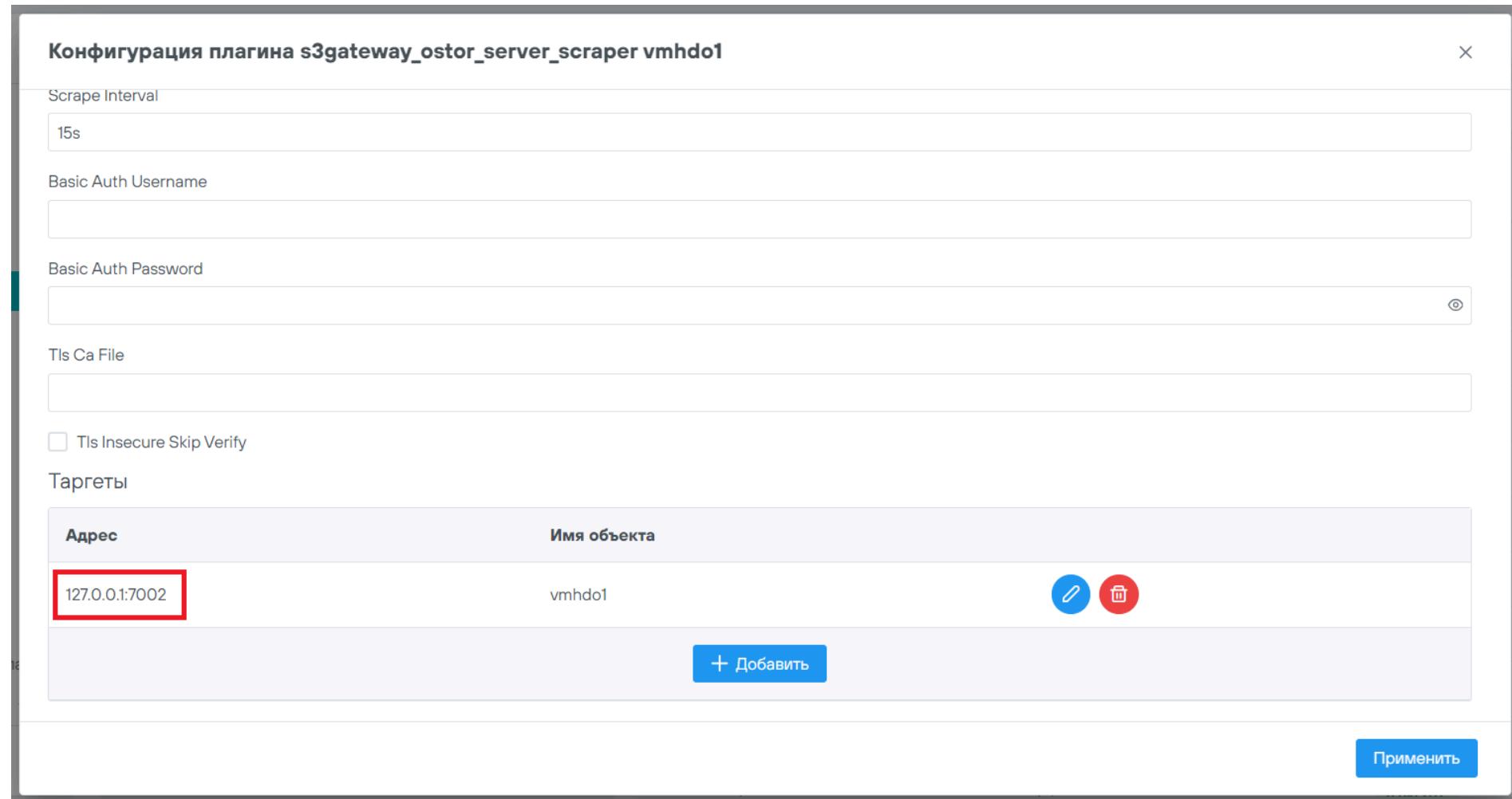
Tls Insecure Skip Verify

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_scrapers` `s3_gate_redOS1`

Scrape Interval
15s

Basic Auth Username

Basic Auth Password

Tls Ca File

Tls Insecure Skip Verify

Таргеты

Адрес	Имя объекта	Изменить	Удалить
127.0.0.1:7001	s3_gate_redOS1		

+ Добавить

Применить

7.2.6 node_exporter

Особое внимание обращать на машине МДИ.О на конфликт портов с другим ПО.

Помимо стандартных коллекторов, добавлен коллектор сбора ошибок чтения и записи дисков. Коллектор собирает информацию из `/sys/block/<device>/device/io*_cnt`. Включен по умолчанию.

7.2.7 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.

7.2.8 kafka_exporter

Плагин обладает множественными настройками:

Конфигурация плагина kafka_exporter Визион

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

Конфигурация плагина kafka_exporter Визион

×

Group Exclude

^\$

Sasl Enabled

Sasl Handshake

Sasl Username

Sasl Password

⋮

⊕

Sasl Mechanism

empty

▼

Sasl Service Name

Sasl Kerberos Config Path

Sasl Realm

Sasl Kerberos Auth Type

userAuth

▼

Sasl Keytab Path

Применить

Конфигурация плагина kafka_exporter Визион

×

Sasl Keytab Path

Sasl Disable Pa Fx Fast

Kafka Tls Enabled

Kafka Tls Server Name

Kafka Tls Ca File

Kafka Tls Cert File

Kafka Tls Key File

Kafka Tls Skip Verify

Use Consumer Lag Zookeeper

Zookeeper Servers

+ Добавить

Refresh Metadata

Offset Show All

Topic Workers

Применить

Конфигурация плагина kafka_exporter Визион

×

Offset Show All

Topic Workers

100

Kafka Allow Auto Topic Creation

Scrape Interval

15s

Scrape Timeout

10s

Basic Auth Username

vision

Basic Auth Password

.....



Tls Cert File

/opt/skala-r/vision/cert/components_default.crt

Tls Key File

/opt/skala-r/vision/cert/components_default.key

Tls Insecure Skip Verify

Таргеты

Адрес	Имя объекта	
127.0.0.1:9308	vision	

Применить

Описание настроек:

Имя	Описание	Дополнительные параметры поля
Kafka Servers	Адрес (host:port) сервера Kafka.	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])\$"}
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: True format: 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 sasl_mechanism=gssapi и sasl_kerberos_auth_type=keytabAuth.	
Sasl Disable Pa Fx Fast	Сконфигурировать клиент Kerberos без использования PA_FX_FAST, по умолчанию выключено. Используется только при sasl_mechanism=gssapi.	
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 СА 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.	<p>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]))\$</p> <p>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} 6553[0-5]))\$"} </p>

Refresh Metadata	Интервал обновления метаданных.	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)\$
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.	

7.2.9 process_exporter

Плагин обладает возможностью конфигурации:

Конфигурация плагина process_exporter test_server

×

Параметры плагина

Web Listen Address

9120

Children

Threads

Recheck

Recheck Duration

0

Plugin Configuration

```
process_names:  
  - comm:  
  - bash
```

Scrape Interval

15s

Scrape Timeout

10s

Basic Auth Username

vision

Basic Auth Password

.....



Tls Cert File

Применить

Параметры конфигурации плагина:

- **children** (по умолчанию: `true`) - данный флаг отвечает за группировку дочерних процессов в метриках родительского процесса. Дочерние процессы, которые не принадлежат своей собственной группе (не выделены в `Plugin Configuration` в отдельную мониторимую сущность), будут добавлены к группе родительского процесса.
- **threads** (по умолчанию: `true`) - данный флаг управляет сбором метрик по отдельным потокам (`threads`) процессов. Метрики будут собираться не только для групп процессов, но и для отдельных потоков этих процессов. При включенном флаге активируются метрики `namedprocess_namegroup_thread_*` – в противном случае они будут отсутствовать.
- **recheck** (по умолчанию: `false`) - флаг `-recheck` управляет тем, как часто проверяются имена процессов. По умолчанию имена процессов проверяются только при первом обнаружении процесса. При `recheck=true` каждом сборе метрик имена процессов перепроверяются. Если процесс изменил своё имя, он будет перегруппирован в соответствии с новым именем.
- **recheck duration** (по умолчанию: `0`) - включить проверку имен процессов только на определенный период после запуска плагина. Отключает постоянную проверку `recheck`

Конфигурирование:

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:
      - '.+'
```

Каждый элемент в `process_names` должен содержать один или несколько селекторов: `comm`, `exe` или `cmdline`.

- **comm:** Сравнивается с базовым именем исполняемого файла (второе поле в `/proc/<pid>/stat`). Это имя обрезается до 15 символов и не может быть изменено программой.

- *exe*: Сравнивается с `argv[0]` (первый аргумент командной строки). Если в *exe* нет слэшей (/), то сравнивается только базовое имя. Если слэши есть, то `argv[0]` должен совпасть полностью.
- *cmdline*: Применяет регулярные выражения ко всей командной строке (`/proc/<pid>/cmdline`). Все регулярные выражения должны совпасть. Если в регулярном выражении есть группы захвата (*capturing groups*), они должны быть именованными (с использованием `?P<name>`), чтобы их можно было использовать в переменной `{{.Matches}}`.

Для *comm* и *exe*: Список строк работает как логическое или. Если процесс совпадает с любой из строк, он добавляется в группу.

Для *cmdline*: Список регулярных выражений работает как логическое и. Все регулярные выражения должны совпасть.

Для извлечения процессов и их группировки (используются в *name*, попадут в *groupname*) имеются следующие шаблоны:

1. `{{.Comm}}`:

- Содержит базовое имя исполняемого файла (второе поле в `/proc/<pid>/stat`).
- Пример: Если процесс запущен из `/usr/bin/python3`, то `{{.Comm}}` будет `python3`.

2. `{{.ExeBase}}`:

- Содержит базовое имя исполняемого файла (аналогично `{{.Comm}}`), но может отличаться в зависимости от реализации).
- Пример: Для `/usr/bin/python3` это будет `python3`.

3. `{{.ExeFull}}`:

- Содержит полный путь к исполняемому файлу.
- Пример: `/usr/bin/python3`.

4. `{{.Username}}`:

- Содержит имя пользователя, от которого запущен процесс.

- Пример: `root, user.`

5. `{{.Matches}}`:

- Это карта (`map`), которая содержит все совпадения, полученные в результате применения регулярных выражений к командной строке процесса (`/proc//cmdline`).
- Пример: Если вы используете регулярное выражение для извлечения аргументов командной строки, то `{{.Matches}}` будет содержать эти аргументы.

6. `{{.PID}}`:

- Содержит PID (идентификатор процесса).
- Важно: Использование PID в шаблонах может привести к созданию групп, содержащих только один процесс, что может привести к высокой кардинальности метрик (большому количеству уникальных метрик), что не рекомендуется для Prometheus.

7. `{{.StartTime}}`:

- Содержит время начала процесса.
- Важно: Использование `{{.StartTime}}` также может привести к высокой кардинальности, так как время начала процесса уникально для каждого процесса.

8. `{{.Cgroups}}`:

- Содержит информацию о cgroups процесса (если поддерживается системой). Это полезно для идентификации контейнера, к которому принадлежит процесс.
- Пример: Если процесс запущен в Docker-контейнере, `{{.Cgroups}}` может содержать информацию о контейнере.

Примеры:

Пример exe:

```
- exe:  
  - postgres  
  - /usr/local/bin/prometheus
```

Пример сложной конфигурации с использованием регулярных выражений (выведет nginx):

```
process_names:  
  - name: "{{.ExeBase}}"  
    exe:  
      - nginx  
    cmdline:  
      - '^n'
```

7.2.10 sql_exporter

Конфигурация плагина sql_exporter Визион

×

Параметры плагина

Web Listen Address [i](#)

9399

Тип базы данных [i](#)

clickhouse

Коллекторы сбора метрик [i](#)

Конфигурация дополнительного коллектора [i](#)

collector_name: clickhouse_collector

metrics:

- metric_name: clickhouse_test

type: gauge
help: 'Тест подключения'
values: [test]
query: |
SELECT 1 as test

Scrape Interval [i](#)

15s

Scrape Timeout [i](#)

10s

Basic Auth Username [i](#)

vision

Basic Auth Password [i](#)

.....

◎

Применить

Для конфигурирования `sql_exporter` можно воспользоваться коллектором сбора метрик: из преднастроенных коллекторов можно будет выбрать наиболее подходящий для мониторинга машины.

7.2.11 `blackbox_exporter`

Плагин устанавливает экспортер и создаёт задачу сбора метрик с него компонентом `vmagent_agent`.

7.2.12 `greenplum_sql_exporter`

Конфигурация плагина greenplum_sql_exporter Сервер

×

Параметры плагина

Web Listen Address (i)

9398

Коллекторы сбора метрик

Greenplum (МБД.Г)



Конфигурация дополнительного коллектора (i)

Scrape Interval (i)

15s

Scrape Timeout (i)

10s

Basic Auth Username (i)

vision

Basic Auth Password (i)

.....



Tls Cert File (i)

/opt/skala-r/vision/cert/components_default.crt

Tls Key File (i)

/opt/skala-r/vision/cert/components_default.key

Tls Insecure Skip Verify (i)

Таргеты

Применить

Для конфигурирования `greenplum_sql_exporter` можно воспользоваться коллектором сбора метрик: из преднастроенных коллекторов можно будет выбрать наиболее подходящий для мониторинга машины.

7.2.13 `clickhouse_sql_exporter`

Конфигурация плагина clickhouse_sql_exporter Сервер

×

Параметры плагина

Web Listen Address [i](#)

9397

Коллекторы сбора метрик

Clickhouse (МБД.КХ)

▼

Конфигурация дополнительного коллектора [i](#)

Scrape Interval [i](#)

15s

Scrape Timeout [i](#)

10s

Basic Auth Username [i](#)

vision

Basic Auth Password [i](#)

.....

👁

Tls Cert File [i](#)

/opt/skala-r/vision/cert/components_default.crt

Tls Key File [i](#)

/opt/skala-r/vision/cert/components_default.key

Tls Insecure Skip Verify [i](#)

Таргеты

Применить

Для конфигурирования `clickhouse_sql_exporter` можно воспользоваться коллектором сбора метрик: из преднастроенных коллекторов можно будет выбрать наиболее подходящий для мониторинга машины.

7.2.14 `picodata_scraper`

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

7.2.15 `graphite_exporter`

Плагин устанавливает экспортер и создаёт задачу сбора метрик с него компонентом `vmagent_agent`.

Конфигурация `graphite_exporter` в UI Визиона осуществляется через соответствующее поле “Конфигурация дополнительного маппинга”, где задаются правила сопоставления и преобразования. Формат – YAML.

Основные параметры

Пример правила сопоставления

Простое правило:

```
- match: 'servers.*.cpu.*'  
  name: 'cpu_usage_$2'  
  labels:  
    server: '$1'  
    type: '$2'
```

Regex:

```
- match: 'servers\.(^\.)+\.\disk\.(^\.)+\.(^\.)+'  
  match_type: regex  
  name: 'disk_$3'
```

```
labels:  
  server: '$1'  
  device: '$2'
```

Конфигурирование

Присутствуют следующие строки:

match:

- Регулярное выражение для сопоставления метрик(и) Graphite.
- Используется для захвата частей метрики, которые можно использовать в *name* и *labels*. Также предназначен для оптимизации преобразования, позволяя использовать шаблоны для однотипных метрик. Пример:
`servers..cpu.` – будет сопоставлять метрики, такие как `servers.web01.cpu.user` или `servers.db01.cpu.system`.

name:

- Имя метрики в Prometheus.
- Можно использовать группы захвата из *match* (например, `$1`, `$2`). Пример: `cput_usage_``$2`: если вторая группа захвата – это `user`, то имя метрики будет `cput_usage_user`.

labels:

- Метки, которые будут добавлены к метрике в Prometheus.
- Ключи – это имена меток, которые задаются пользователем, значения – группы захвата из *match*. Значение также может быть фиксированным (заданным пользователем) Пример: `server: '$1'` – если первая группа захвата – это `web01`, то метка будет `server="web01"`.

match_type (опционально):

- Тип сопоставления: `glob` (по умолчанию) или `regex`.
- `glob` – простой шаблон с использованием `*`, `?`, `[...]` и др.
- `regex` – полноценное регулярное выражение

Группы захвата:

- Группы захвата (`$1`, `$2`, и т.д.) соответствуют частям метрики, захваченным в `match`.
- Пример: `servers..cpu.`: к первой звезде относится группа захвата `$1`, ко второй – `$2`.

Способ применения

Чтобы задать маппинг, следует перечислить все необходимые преобразования в UI Визиона согласно описанному формату. Дополнительно ничего вводить не следует.

Конфигурация дополнительного маппинга [\(i\)](#)

Если маппинг может конфликтовать с уже заданным, следует снять галочки с выбранных стандартных маппингов:

Маппинги метрик

Kafka,Hadoop

Kafka

Hadoop

Применяемые шаблоны в Визионе для Kafka и Hadoop

В применяемых в Визионе шаблонах используются два подхода:

- Динамическое формирование имён метрик
- Явно прописанное имя Первый подход используется преимущественно для Kafka, второй – преимущественно для Hadoop. Явно прописанное имя необходимо для более удобного использования в алертинге.

Используются следующие метки:

- `cluster`: Имя кластера (из группы захвата `${1}`).
- `host`: Имя хоста (из группы захвата `${2}`).
- `service`: Тип сервиса, прописан явно (например, `ReplicaManager`, `Broker`, `HDFS`, `Hive`)
- `rate` - Частота, связанная с измерениями внутри метрики (`OneMinuteRate`, `FiveMinuteRate`) и др.

- `collector` - Коллектор метрики

7.2.16 `snmp_exporter`

Изменение

X

Адрес*

192.168.185.217

Имя объекта*

comm217

Модуль SNMP-метрик  *

Коммутаторы



Версия SNMP  *

2



Строка сообщества  *

public

Уровень безопасности  *

noAuthNoPriv



Имя пользователя 

Пароль 



Протокол авторизации 

MD5 ▼

Протокол шифрования i

DES ▼

Отменить Изменить

Протокол шифрования i

DES ▼

Пароль шифрования i

Контекстное имя i

Отменить Изменить

Конфигурирование `snmp_exporter` применяется к конкретным таргетам. Меню конфигурации позволяет настроить мониторинг коммутаторов и СХД по SNMP версий 1, 2 и 3 с помощью различных параметров, влияющих на аутентификацию.

7.2.17 `universal_scrapers`

Плагин обладает следующими настройками для конфигурации:

Дополнительные метки [i](#)

Имя метки [i](#) *

 Delete

Значение метки [i](#) *

[+ Добавить](#)

Basic Auth Username [i](#)

Basic Auth Password [i](#)

 @

Metrics Path [i](#)

 /metrics

Scheme [i](#)

 http ▼

Tls Cert File [i](#)

Tls Insecure Skip Verify [i](#)

Изменение

×

Адрес*

127.0.0.1

Имя объекта*

self137

Дополнительные метки [i](#)

Имя метки [i](#) *

 ×

Значение метки [i](#) *

[+ Добавить](#)

[Отменить](#)

[Изменить](#)

universal_scrapers используется для:

- Сбора данных с экспортёров и эндпоинтов не из поставки Визиона
- Тестирования новых экспортёров или проверки метрик из инструментированных предложений
- Переопределения логики работы плагинов из поставки Визиона, если требуется обогатить временные ряды новыми метками

`universal_scraper` позволяет собирать данные с любых эндпоинтов, как защищённых паролем, так и доступных без аутентификации.

Для подключения к защищённым эндпоинтам следует ввести имя и пароль `BasicAuth`. При наличии можно подключиться с использованием сертификата.

Ключевая особенность скрапера – возможность обогащения временных рядов пользовательскими метками. Пользовательские метки можно применить как на уровне скрапера (ко всем таргетам), так и на уровне конкретного таргета.

Из коробки скрапер добавляет стандартные метки объектной модели того объекта, где он установлен.

В качестве примера переопределим сбор метрик `utlz_exporter` (он доступен по порту 9092). В данном случае добавлены две пользовательские метки:

Изменение

×

Адрес*

127.0.0.1:9092

Имя объекта*

self137

Дополнительные метки 

Имя метки  *

test2



Значение метки  *

qa2

Имя метки  *

label2



Значение метки  *

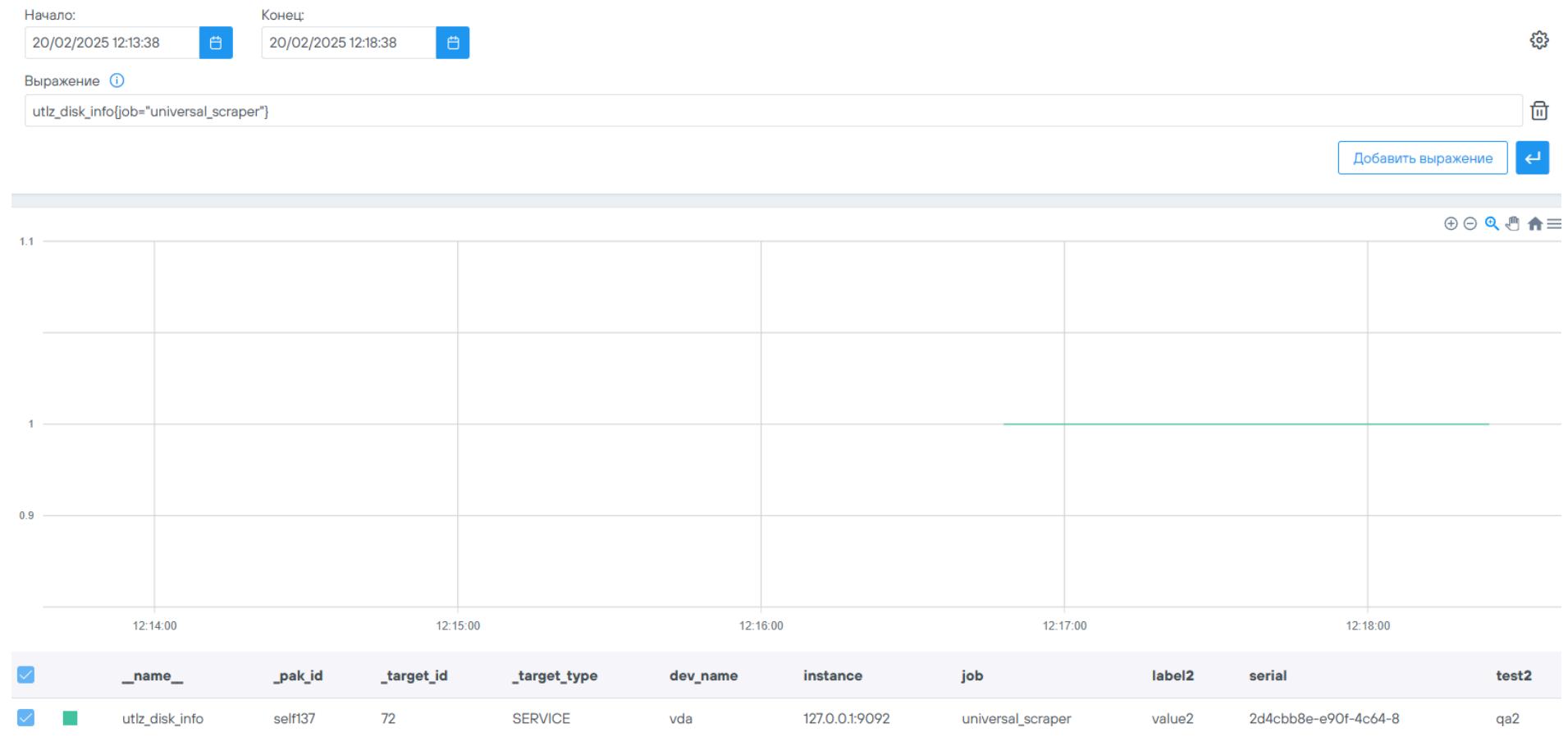
value2

 Добавить

Отменить

Изменить

Вывод временного ряда:



7.2.18 mvs_exporter

Имеются следующие поля конфигурации:

vControl Address (i)	<input type="text" value="192.168.147.27"/>
vControl Login (i)	<input type="text" value="vision"/>
vControl Password (i)	<input type="password" value="....."/> (o)

Требуется ввести данные об IP vControl, а также логин и пароль от учётной записи.

Следует регулярно чистить логи API, так как обращение к нему происходит на регулярной основе, из-за чего может возникнуть переполнение файловой системы.

7.2.19 `mbdh_exporter`

Имеются следующие поля конфигурации:

Изменение

×

Адрес*

127.0.0.1

Имя объекта*

MyPak

Username i

user

Password i

.....



Insecure i

Отменить

Изменить

Требуется внести информацию о пользователе и пароле для подключения к ADCM. Также имеется чекбокс для подключения по HTTP.

7.2.20 ydb_scrapers

Имеются следующие поля конфигурации в разделе конфигурирования экземпляра плагина:

Metrics Path [i](#)

/counters/prometheus

Tls Cert File [i](#)

/opt/skala-r/vision/cert/components_default.crt

Tls Insecure Skip Verify [i](#)

Диапазон портов [i](#)

8765, 8766

Таргеты

Адрес	Имя объекта	
192.168.147.120	hadoop21	i trash
192.168.224.101	hadoop21	i trash
192.168.185.201	hadoop21	i trash
+ Добавить		

И конфигурирования конкретного таргета:

Изменение

×

Адрес*

192.168.185.201

Имя объекта*

hadoop21

Диапазон портов 

Отменить

Изменить

Имеется возможность указать путь до метрик, а также указать диапазон портов. Порты можно указать штучно, через дефис, через запятую.

Дополнительно можно указать диапазон портов для конкретного таргета: в таком случае они перезапишут общие, для этого таргета применяются его универсальные порты.

7.2.21 patroni_scrapers

Имеются следующие поля конфигурации в разделе конфигурирования экземпляра плагина:

- **Scrape interval:** Периодичность забора метрик.

- **Scrape timeout**: Максимальное время, отводимое на забор метрик.
- **Basic Auth Username**: Имя пользователя для аутентификации при подключении к экспортеру.
- **Basic Auth Password**: Пароль пользователя для аутентификации при подключении к экспортеру.
- **Tls cert file**: сертификат сервера.
- **Tls key file**: ключ сертификата.
- **Tls insecure skip verify**: пропуск проверки сертификата при аутентификации.

Плагин позволяет указать несколько таргетов.

В настройках таргета указываются:

- Адрес – адрес таргета в формате **host[:port]**, где:
 - **host** – IP-адрес таргета.
 - **port** – опциональный порт таргета. Если значение не задано, по умолчанию используется порт **8008**.
- Имя объекта.

7.3 Добавление и обновление плагинов

Выполните следующие действия на виртуальной машине сервера Визиона:

1. Распакуйте архив с новым или обновлённым плагином в директорию **/opt/skala-r/vision**, например:

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

2. Перезапустите сервис **vision_core**:

```
systemctl restart vision_core.service
```

3. Проверьте состояние сервиса **vision_core**:

```
systemctl status vision_core.service
```

4. Убедитесь, что в логе нет ошибок после перезапуска:

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

7.4 Снятие ПАК с мониторинга (дeинсталляция Визиона с узлов ПАК).

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

В браузере перейти в веб-интерфейс Визион, доступный по ссылке:

https://<vision_ip>

Где **<vision_ip>** – IP-адрес сервера Визион.

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. В браузере перейти в веб-интерфейс Визион, доступный по ссылке:

`https://<vision_ip>`

Где `<vision_ip>` – IP-адрес сервера Визион.

2. Выбрать пункт **Модуль администратора**.
3. Перейти в раздел **Настройки отправки** на вкладку **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 кнопку **Сохранить**.
4. Перейти в раздел **Список получателей** и создать получателя уведомлений с каналом **E-mail** и адресом электронной почты, на которую будут приходить уведомления о сработавших правилах оповещения.
5. Перейти в раздел **Группы рассылки**, создать группу, добавить в эту группу получателя. Добавить условие в этой группе: например, `severity = critical`.
6. 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 `ip == 1`).
7. Перейти в раздел **Уведомления** и дождаться срабатывания правила оповещения.
8. Войти в электронную почту, указанную ранее, и дождаться получения письма о сработавшем правиле оповещения.

Настройка авторизации для отправки email-уведомлений осуществляется с помощью галки в чекбоксе “Без авторизации”. Если галка в чекбоксе установлена, то отправка email-уведомлений через SMTP будет осуществляться без авторизации. Если галка не установлена, поля “Логин” и “Пароль” обязательно должны быть заполнены; указанные параметры будут использоваться для авторизации отправляемых email-уведомлений.

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 с телом JSON со следующими полями:

Имя параметра	Описание
<code>snmp_version</code>	Версия SNMP. Допустимые значения: "V2c", "V3". По умолчанию "V2c".
<code>snmp_destination</code>	Адресат сообщений. По умолчанию "127.0.0.1:162".
<code>snmp_retries</code>	Количество повторных посылок сообщения SNMP. По умолчанию 1.
<code>snmp_trap_default_oid</code>	OID ловушки, если он не найден в метках сработавшего правила. По умолчанию "1.3.6.1.4.1.98789".
<code>snmp_timeout_sec</code>	Длительность таймаута в секундах. По умолчанию 5.
<code>snmp_community</code>	Коммюнити SNMP (только для V2c, иначе null). По умолчанию "public".
<code>snmp_authentication_protocol</code>	Протокол шифрования пароля (только для V3). Допустимые значения: "MD5", "SHA", null. По умолчанию null.

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

Пример Curl для SNMP V3:

```
curl -k -X 'POST' 'https://<vision_host>:8088/vision/api/v1/config/snmp_notifier_config' -H 'accept: application/json'
  "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. В браузере перейти в веб-интерфейс Визион, доступный по ссылке:



https://<vision_ip>

Где **<vision_ip>** – IP-адрес сервера Визион.

2. Выбрать пункт **Модуль администратора**.
3. В боковом меню выбрать раздел **Параметры**, пункт **Группы рассылки**.
4. Нажать кнопку **Добавить** и создать новую группу рассылки.
5. 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")
6. 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`, расположенным 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 `VictoriaMetrics`, 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 к инвентори-файлу (см. п. 6), например,

```
ansible-playbook \
-i /path/to/inventory.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 инвентори-файл.

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` и `~/.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 хранится 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 IAM.

vault.auth.tech_auth_password – пароль технической учетной записи IAM для использования в синхронизации пользователей в качестве получателей рассылки.

vault.auth.tech_auth_client_secret – секрет технического клиента Avanpost для использования в синхронизации пользователей в качестве получателей рассылки.

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.metrics.basic_auth_password` – пароль BasicAuth для получения метрик Grafana.

`vault.vmalert_external_integration.auth.basic_auth.password` – Basic Auth пароль для внешней интеграции alertmanager.

`vault.vmalert_external_integration.auth.bearer_token.token` – Bearer token для внешней интеграции alertmanager.

`vault.vmalert_external_integration.auth.oauth2.client_secret` – секрет клиента oauth2 для внешней интеграции alertmanager.

14 ПРОВЕРКА ПРОГРАММЫ

14.1 Отображение веб-интерфейса

По завершении процесса установки веб-интерфейс должен быть доступен по ссылке:

`https://<vision_ip>`

Где `<vision_ip>` – IP-адрес сервера Визион.

Далее пункт **Аналитические панели и Модуль Администратора**.

14.2 Проверка работоспособности сервисов

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

`cd /opt/skala-r/vision`

`./check_service.sh`

В результате, вывод должен быть следующий:

Status	Port	Service	Status	Journal
>>> Vision Server services				
active	3000	grafana	systemctl status grafana	journalctl -fu grafana
active	8428	victoriameetrics_st	systemctl status victoriameetrics_st	journalctl -fu victoriameetrics_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	/opt/skala-r/vision/<server-proxy-agent>/<имя-компоненты>
Файлы конфигурации	/opt/skala-r/etc/vision/<server-proxy-agent>/<имя-компоненты>

Логи компонентов	Системный журнал и <code>/opt/skala-r/var/log/vision/<server-proxy-agent>/<имя-компонента></code>
Хранилище секретов ^1	<code>/opt/skala-r/vision/.secrets_vault</code>
Nginx ^1	<code>/etc/nginx</code>

^1 только для Визион.Сервер

15.1 Запуск ansible-playbook

Список часто встречающихся проблем:

- `Failed to parse <путь-к-инвентори-файлу>` – 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 аргументом `-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-объекта>
```

- для плагинов

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

где **<имя-объекта>** – это имя проблемного сервера, виртуальной машины или ПАК.

Тело ответа будет содержать поле **data** с массивом информации об агентах/плагинах. 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.

Ошибка “Ошибка при выполнении плагина: exit status 1 (sudo: a password is required)” указывает на возможную ошибку в конфигурировании sudo в /etc/sudoers. Например, некорректный порядок правил, когда после включения директории с дополнительными правилами идет какое-либо общее правило.

```
#includedir /etc/sudoers.d
%wheel  ALL=(ALL)      ALL
```

```
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/version
```

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 метрик 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
```

Здесь `<vision_ip>` – IP-адрес сервера Визион.

В открывшемся интерфейсе вы сможете проверить наличие временных рядов в системе, используя PromQL-запросы.

Параметры технической учетной записи для авторизации между компонентами конфигурируются в файле `_deploy_settings.yml`, расположенному в корне дистрибутива - `basic_auth_username` и `basic_auth_password` (имя пользователя и пароль соответственно)

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 метрик

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 метрик).

Если сервис имеет статус **active**, но данных в базе временных рядов нет: убедитесь, что с узла, где расположен Прокси метрик, 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 метрик

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_vagent_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.

15.5 После изменения времени на сервере Визиона алерты продолжают приходить со старым временем

При смене времени на сервере Визиона необходимо перезапустить сервисы `vmalert_st` и `alertmanager`:

```
systemctl restart vmalert_st
systemctl restart alertmanager
```

15.6 Ошибка при работе Playbook “FAILED! => Установлена старая версия rsyslog ... (assert)”

Чтобы полностью использовать функциональность Визиона в части мониторинга логов, необходимо, чтобы был установлен пакет `rsyslog` не ниже версии 8.1901. Playbook осуществляет проверку версии пакета `rsyslog`. Если установлен пакет версии ниже 8.1901, то playbook выводит ошибку “FAILED! => Установлена старая версия `rsyslog` ... (assert)”.

Есть два способа решения:

1. самостоятельно установить пакет `rsyslog` версии 8.1901 или выше;
2. открыть файл конфигурации `rsyslog` `/etc/rsyslog.d/*.conf` и изменить значение переменной `rsyslog_min_version` на версию пакета, который установлен в ОС.

После установки пакета `rsyslog` или изменения значения в файле конфигурации необходимо повторно запустить playbook.

При выборе второго способа решения playbook сможет завершить работу без ошибок, но в таком случае функциональность мониторинга логов будет ограничена – будет недоступен динамический сбор логов.

15.7 Ошибка при работе Playbook “ERROR! no action detected in task. This often indicates a misspelled module name, or incorrect module path.”

Ошибка связана с тем, что используется старая версия пакета Ansible в ОС Astra.

Для исправления ошибки необходимо активировать виртуальное окружение со свежей версией Ansible и повторить команду:

```
source /opt/skala-r/vision/server/vision_venv/bin/activate
```

15.8 Автоматическая проверка ПО выявляет уязвимость CVE-2025-26791 DOMPurify для monaco-editor в Grafana

При проверке ПО автоматическими средствами обнаружения уязвимостей может быть обнаружена следующая уязвимость, которая связана с библиотекой DOMPurify:

Уязвимость, позволяющая обходить механизм очистки HTML-разметки шаблона в библиотеке DOMPurify. Проблема возникает из-за некорректного регулярного выражения, в котором не предусмотрена обработка уязвимости типа `incorrectly-opened-comment`.

источник: <https://ensy.zip/posts/dompurify-323-bypass/>

Обнаруженную уязвимость необходимо игнорировать, так как уязвимость CVE-2025-26791 в библиотеке DOMPurify 3.1.7 не может проявиться при использовании `monaco-editor` 0.52.2. Ни в одной функции `DOMPurify.sanitize` не передан флаг `SAFE_FOR_TEMPLATES=true`, по умолчанию флаг `SAFE_FOR_TEMPLATES=false`.

<https://github.com/microsoft/monaco-editor/issues/4844>

15.9 Агенты логов установлены, но логи отсутствуют

Необходимо убедиться от какой части системы отсутствуют данные. Перейдите в веб-интерфейс VictoriaLogs, открыв веб-браузер, и введя в строке адреса IP-адрес сервера Визиона:

https://<vision_ip>/vision/victorialogs/select/vmui

Здесь `<vision_ip>` – IP-адрес сервера Vision.

В открывшемся интерфейсе вы сможете проверить наличие логов в системе, используя LogSQL-запросы.

При недоступности интерфейса VictoriaLogs, проверьте работу сервиса `victorialogs.service` и `nginx.service`.

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

15.9.1 Отсутствие данных с Прокси логов

Проверить работоспособность компонента можно по получаемым с него метрикам. Для этого на узле также должен быть установлен агент метрик.

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

Если временной ряд отсутствует, то убедитесь, что агент метрик установлен на тот же узел и проверьте его работоспособность systemd сервиса `vision_vmagent_agent.service` на узле.

Если временной ряд равен 0, то проверьте работоспособность systemd сервиса `vision_vlagent_proxy.service` на узле, где он установлен (выбирается при установке Агентов логов).

Если сервис имеет статус `active`, но данных в базе логов и/или метрик Прокси логов нет: убедитесь, что с узла, где расположен Прокси логов, есть возможность отправлять данные в VictoriaLogs по `https`. Например, результат такого `curl` должен быть `OK`:

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

15.9.2 Отсутствие данных с Агентов логов

Проверить работоспособность компонента можно по получаемым с него метрикам. Для этого на узле также должен быть установлен агент метрик.

PromQL-запросы:

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

Если временной ряд отсутствует, то убедитесь, что агент метрик установлен на тот же узел и проверьте его работоспособность `systemd` сервиса `vision_vmagent_agent.service` на узле.

Если временной ряд равен 0, то проверьте работоспособность `systemd` сервиса `vision_vlagent_agent.service` на узле, где он установлен.

Если сервис имеет статус `active`, но данных в базе логов и/или метрик агента логов нет: убедитесь, что с узла, где расположен Агент, есть возможность отправлять данные в Прокси по `https` в порт 29511 (см. п 1.3).

Например, результат такого `curl` должен быть `OK`:

```
curl -k https://<user>:<pass>@<ip-vision>:29511/health
```

16 Журналирование и аудит

16.1 Конфигурирование логирования `vision_core`

Файл конфигурации логирования `vision_core` расположен в `/opt/skala-r/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>" по maxBytes

# Формат JSON (ГосTex)
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 ПАК	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 msg=Чтение: версия Визиона deviceProcessName=pp=HTTPS dpt=8088



Заголовок CEF

Field Name	Description
------------	-------------

CEF:Version	Версия формата CEF. Константа CEF:0
Device Vendor	Название производителя. Константа Skala-r
Device Product	Название продукта. Константа Vision
Device Version	Версия Визиона
Device Event Class ID	Уникальный идентификатор категории события. Формируется путём <группа><действие><статус>
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 vision_core

outcome	Результат события. success / failure
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", "metamodelVersion": "1", "module": "Visior
```



Поле	Описание
createdAt	Время возникновения события в формате Unix Timestamp.
userNode	Имя хоста получателя (Визиона).
metamodelVersion	Константа 1 .
module	<i>Название продукта</i> . Константа Vision
name	<i>Краткое описание события</i> .
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 Визиона

- 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>
Визион.Агент	bvs_exporter	Системный журнал и <code>/opt/skala-r/var/log/vision/audit/agent/bvs_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.

17 Ролевая модель

17.1 Концепция ролевой модели Визион

Визион допускает работу пользователей вообще без какой-либо аутентификации (любой пользователь, зашедший на веб-страницу по адресу Визион, получает полный набор полномочий), либо с аутентификацией через один из совместимых провайдеров (IAM):

- [Keycloak](#) ,
- [Avanpost FAM](#) ,

- Basis Virtual Security[¤] .

При использовании аутентификации для определения набора полномочий пользователя применяется ролевая модель.

Учётной записи пользователя на стороне провайдера аутентификации назначается одна или несколько ролей.

На стороне Визиона для каждой роли определяется набор разрешённых функций и данных (далее - полномочия).

Пользователю будут предоставлены все полномочия, определённые назначенными его учётной записи ролями.

Используется разрешительная модель полномочий: всё, что не разрешено в явном виде, запрещено.

Важно! Имя роли в провайдере аутентификации должно совпадать с именем роли в Визион

Важно! Настройка автоматизированных рассылок уведомлений в текущей версии Визион доступна только для провайдера аутентификации Keycloak .

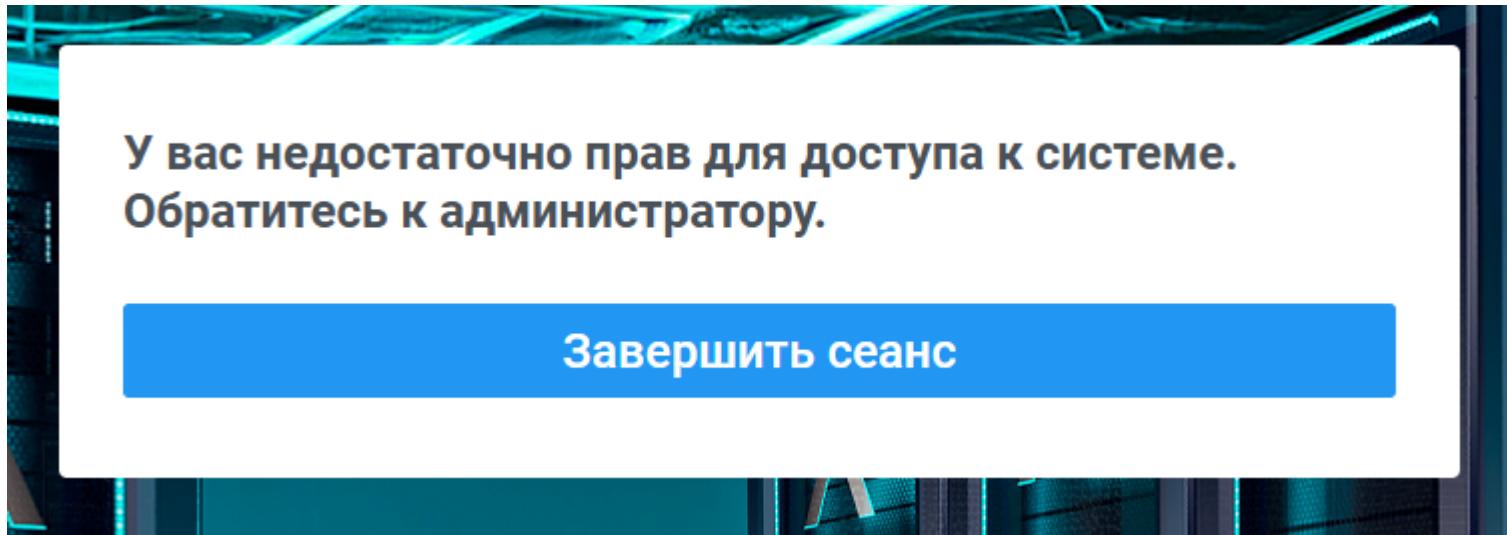
17.2 Аутентификация и авторизация пользователя

Аутентификация пользователя – процедура проверки его подлинности путём сверки введённого пароля с хранимым в базе данных провайдера аутентификации. При успешной аутентификации Визион получает от провайдера аутентификации информацию о наборе назначенных ему ролей.

Авторизация пользователя – предоставление ему полномочий на выполнение определённых действий в Визион на основе набора назначенных ролей и соответствующих этим ролям полномочий.

Для успешной авторизации пользователю должна быть назначена минимум одна роль, для которой определены полномочия в Визион. Если пользователю назначено несколько ролей, его полномочия будут определены как сумма полномочий всех назначенных ролей.

Пользователю может быть отказано в авторизации, при этом будет выведено окно:



Такое возможно, если:

- учётной записи пользователя не назначены роли на стороне провайдера аутентификации,
- роли учётной записи назначены на стороне провайдера аутентификации, но не созданы на стороне Визион,
- роли созданы на стороне Визион, но им не назначены полномочия.

В случае, если провайдер аутентификации недоступен, администратор может перевести Визион в режим обслуживания, воспользовавшись [инструкцией](#). Этот режим предназначен для временного использования и только администратором Визион.

17.3 Роли администраторов

Визион поддерживает следующие роли администраторов, полномочия которых не могут быть изменены:

Имя роли на стороне Визион	Описание	Полномочия (не могут быть изменены)
role_model	Администратор информационной безопасности Визион	Управление токенами доступа, управление ролями пользователей и настройка полномочий

obj_model	Администратор объектной модели	Управление объектами Контура (ПАКов) и плагинами
vsn_config	Администратор объектов мониторинга	Управление метриками, настройка правил оповещения, управление группами рассылки и получателями, настройка SMTP
inventory	Администратор inventory	Добавление, редактирование и удаление ссылок на сервисы ПАК - управление узлами, Arenadata, Basis и др.

При необходимости можно изменить имена ролей администраторов на стороне Визион.

Важно! Имя роли в провайдере аутентификации должно совпадать с именем роли в Визион.

Соответствие между именами ролей администраторов на стороне провайдера аутентификации и именами ролей на стороне Визион задаётся в [конфигурационном файле vision_core](#) в блоке `auth`:

```
auth:
  # ...
  admin_role_codes:
    role_model: role_model
    obj_model: obj_model
    vsn_config: vsn_config
    inventory: inventory
```

Если используются имена ролей администраторов по умолчанию, блок `admin_role_codes` может отсутствовать.

Чтобы изменения, внесённые в конфигурационный файл `config.yml`, вступили в силу, выполните команду:

```
/opt/skala-r/vision/tools/update_server_configs.sh
```

Если пользователю назначить все администраторские роли, то он становится суперадминистратором, то есть получает максимальные полномочия в Системе.

Все остальные (не административные) роли настраиваются администратором в интерфейсе пользователя Визион, как описано ниже.

17.4 Управление ролями пользователей

Управление ролевой моделью Визиона доступно только пользователям с ролью администратора информационной безопасности. Под управлением ролевой моделью понимается:

- создание ролей (раздел “17.4.1”),
- назначение полномочий на объекты Контура, плагины и дашборды (раздел “17.4.2”),
- переименование существующих ролей (раздел “17.4.3”),
- удаление ролей (раздел “17.4.4”),
- создание роли с набором полномочий, взятых из другой роли (раздел “17.4.5”).

Особенности применения полномочий, назначенных роли, описаны в разделе “17.5”. Для осуществления действий по управлению ролями администратор информационной безопасности должен перейти в раздел “Безопасность – Ролевая модель”:

ВИЗИОН

Уведомления

Дашборды

Объекты

Настройки

Метрики

Правила оповещения

Конструктор выражений

Список получателей

Группы рассылки

Настройки SMTP

Безопасность

Ролевая модель

Токены доступа

Статус Визиона

О программе

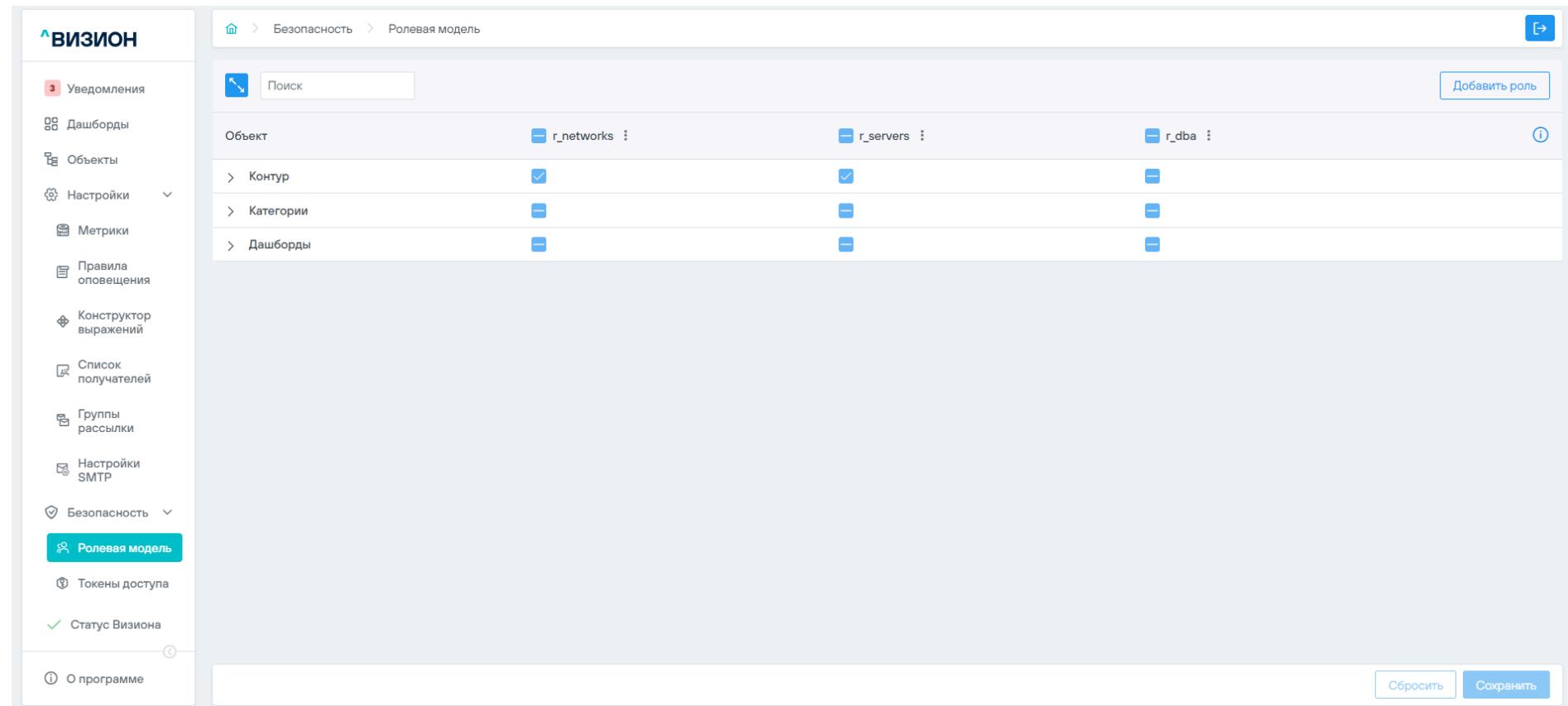
Безопасность > Ролевая модель

Поиск

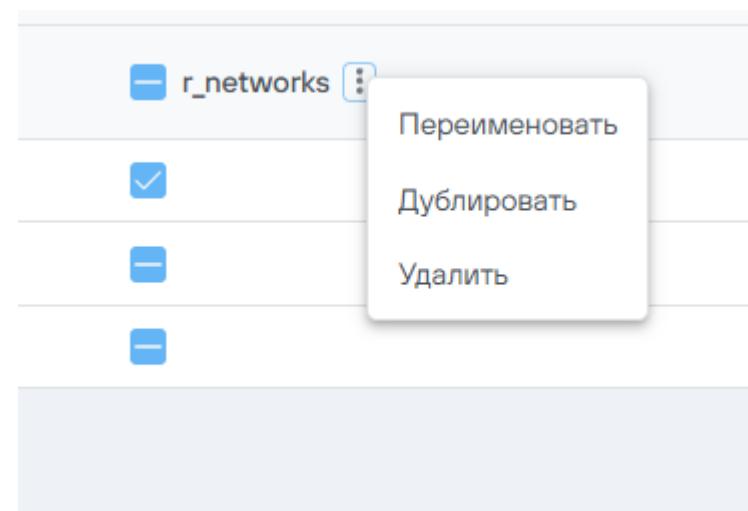
Добавить роль

Объект	r_networks	r_servers	r_db
> Контур	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
> Категории	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
> Дашборды	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Сбросить Сохранить



Справа от имени роли находится меню роли, определяющее возможные действия с ролью:



17.4.1 Добавление роли

Если вы внесли какие-либо изменения в набор полномочий ролей, перед созданием новой роли необходимо сохранить или отменить сделанные изменения.

Нажмите на кнопку “Добавить роль”. В появившемся окне введите имя роли и нажмите “Сохранить”. Имя роли должно отличаться от имен уже созданных ролей.

После создания новой роли необходимо сохранить полномочия, иначе пользователь не сможет авторизоваться.

17.4.2 Назначение полномочий для роли

Назначение полномочий для ролей пользователей осуществляется путем выбора чек-бокса на пересечении объекта и роли в таблице полномочий. Пустой чек-бокс соответствует отсутствию полномочия. После выбора полномочий необходимо нажать на кнопку “Сохранить”. Если нажать на кнопку “Сбросить”, то изменения, внесенные с момента последнего сохранения, будут отменены.

Объект	r_networks	r_servers	r_dba
Контур	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
p1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
p2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Категории	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Архив	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Функциональные сервисы	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Сеть	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Прочее	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Сервер	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Хранение данных	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Служебные сервисы	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Дашборды	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Визион	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Главная	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
МГП Г	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

17.4.3 Переименование роли

Перед переименованием роли необходимо сохранить или отменить изменения, уже внесённые на страницу настройки полномочий.

В меню роли выберите “Переименовать”. В появившемся окне введите новое имя роли и нажмите “Сохранить”. Имя роли должно отличаться от имен уже созданных ролей.

17.4.4 Удаление роли

Перед удалением роли необходимо сохранить или отменить изменения, уже внесённые на страницу настройки полномочий.

В меню роли выберите “Удалить”. В появившемся окне подтвердите удаление.

Удаление роли на стороне Визион не приведёт к удалению ролей на стороне провайдера аутентификации.

17.4.5 Дублирование роли

Перед дублированием роли необходимо сохранить или отменить изменения, уже внесённые на страницу настройки полномочий.

В меню роли выберите “Дублировать”. В появившемся окне введите имя новой роли и нажмите “Сохранить”. Имя роли должно отличаться от имен уже созданных ролей. Созданная роль будет содержать полномочия роли, которая была выбрана для дублирования.

После создания роли путем дублирования необходимо сохранить полномочия, иначе пользователь не сможет авторизоваться.

17.5 Применение полномочий

После того, как пользователь успешно авторизован, Визион предоставляет ему набор соответствующих полномочий: доступ к разделам в интерфейсе пользователя, доступ к данным, собираемым с различных объектов Контура и т. п.

17.5.1 Доступ к разделам Визион

Приведенная ниже таблица иллюстрирует наличие доступа в разделы пользовательского интерфейса Визион.

“есть” - означает, что раздел отображается в интерфейсе пользователя, при этом все пользователи имеют полномочия просмотра, но только роли администраторов имеют полномочия на изменение.

“нет” - означает, что раздел не отображается в интерфейсе пользователя и недоступен ни для просмотра, ни для внесения изменений.

Раздел “О программе” доступен пользователям с любыми ролями.

Страница	Элемент интерфейса	роль obj_model	роль role_model	роль vsn_config	роль inventory	все остальные роли (читатели)
Страница администрирования	Уведомления	есть	есть	есть	есть	есть
	Дашборды	есть	есть	есть	есть	есть
	Объекты	есть	нет	нет	нет	нет
	Объекты (Вкладка - Обзор)	есть	нет	нет	нет	нет
	Объекты (Вкладка - Метрики)	есть	нет	нет	нет	нет

	Объекты (Вкладка - Уведомления)	есть	нет	нет	нет	нет
	Объекты (Вкладка - Агенты)	есть	нет	нет	нет	нет
	Объекты (Вкладка - Плагины)	есть	нет	нет	нет	нет
	Настройки - Конструктор выражений	нет	нет	есть	нет	нет
	Настройки - Метрики	есть	есть	есть	есть	нет
	Настройки - Правила оповещения	есть	есть	есть	есть	есть
	Настройки - Список получателей	нет	нет	есть	нет	нет
	Настройки - Группы рассылки	нет	нет	есть	нет	нет

	Настройки – настройки отправки	нет	нет	есть	нет	нет
	Безопасность – Токены доступа	нет	есть	нет	нет	нет
	Безопасность – Ролевая модель	нет	есть	нет	нет	нет
	Статус Визиона	есть	есть	есть	есть	есть
	О программе – Системные	есть	есть	есть	есть	есть
	О программе – Резервная копия	нет	нет	есть	нет	нет
Стартовая страница “Визион”	Все элементы кроме верхнего меню	есть	есть	есть	есть	есть

Страница статистики уведомлений	Все элементы кроме верхнего меню	есть	есть	есть	есть	есть
Верхнее меню для Стартовой страницы и Страницы статистики уведомлений	меню "Уведомления"	есть	есть	есть	есть	есть
	меню "Дашборды"	есть	есть	есть	есть	есть
	меню "Объекты"	есть	нет	нет	нет	нет
	меню "Настройки"	есть	есть	есть	есть	есть
	меню "Безопасность"	нет	есть	нет	нет	нет

Иллюстрация доступа к общим меню для стартовой страницы Визион и страницы статистики уведомлений:

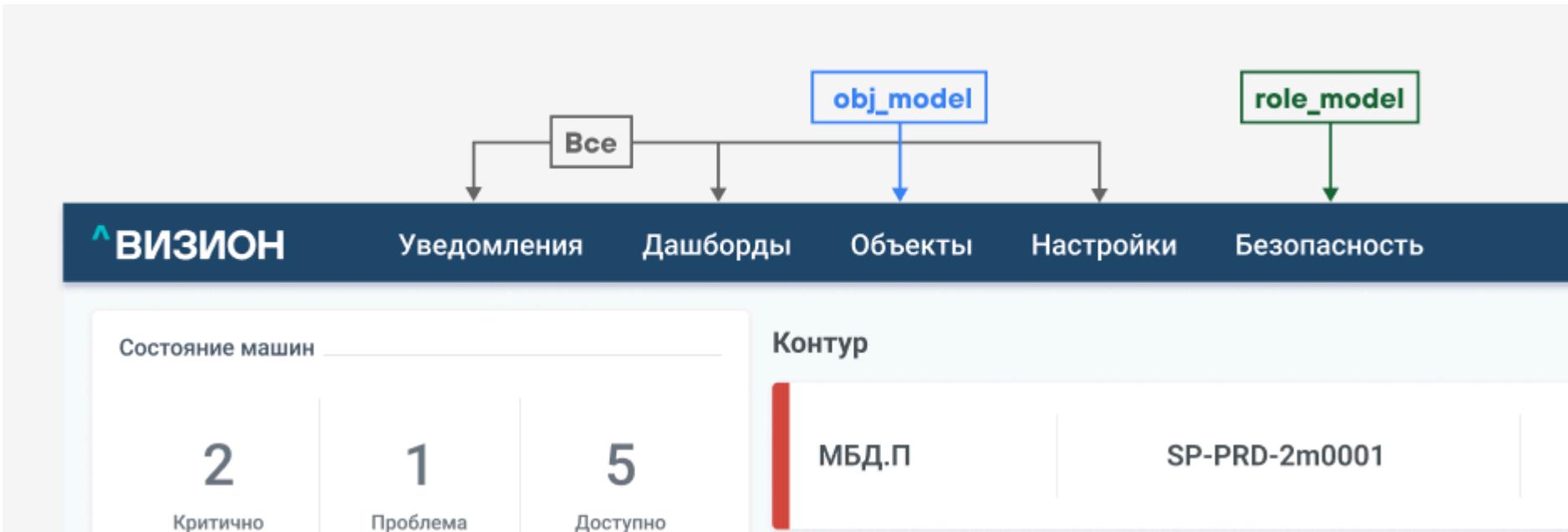
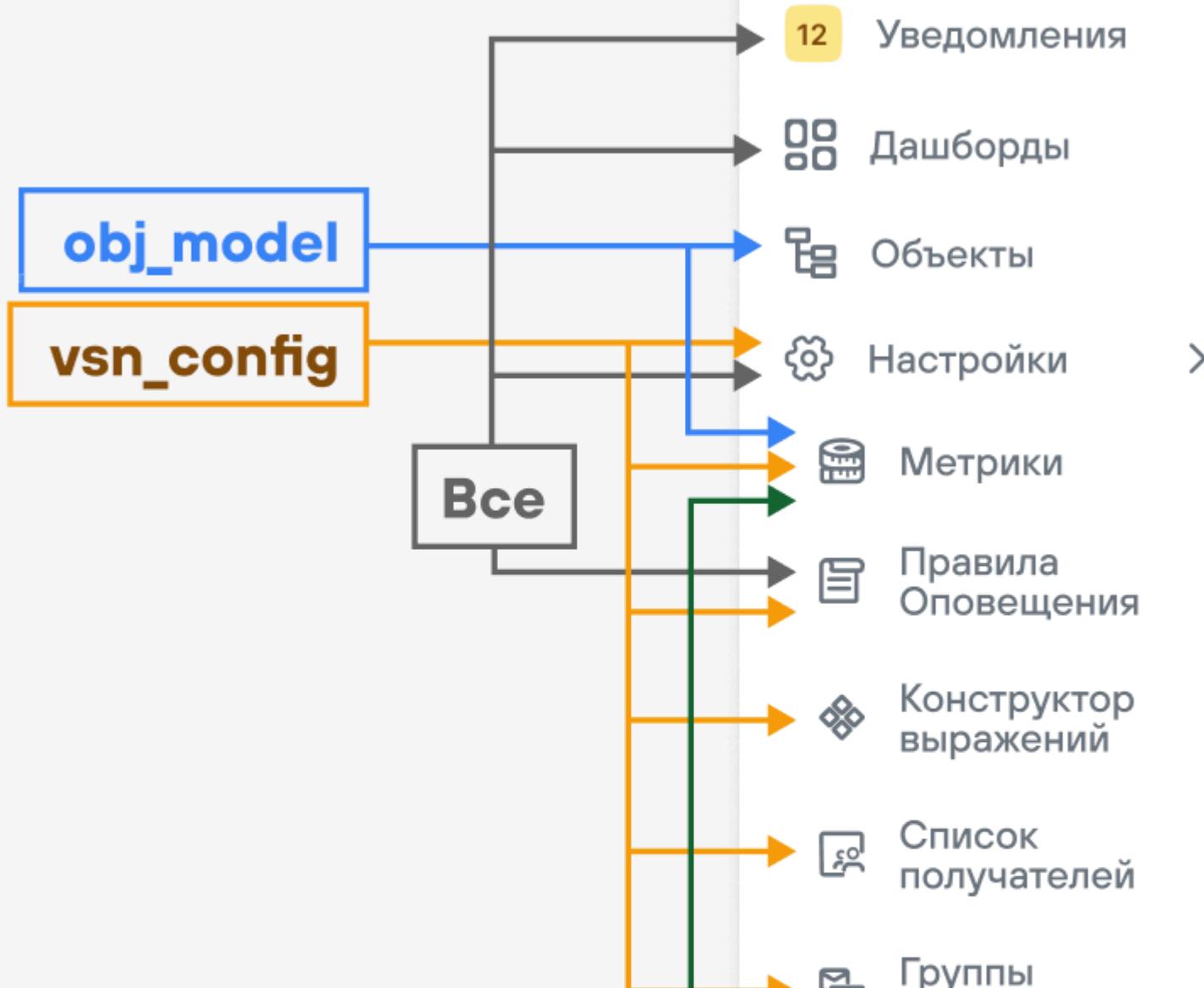
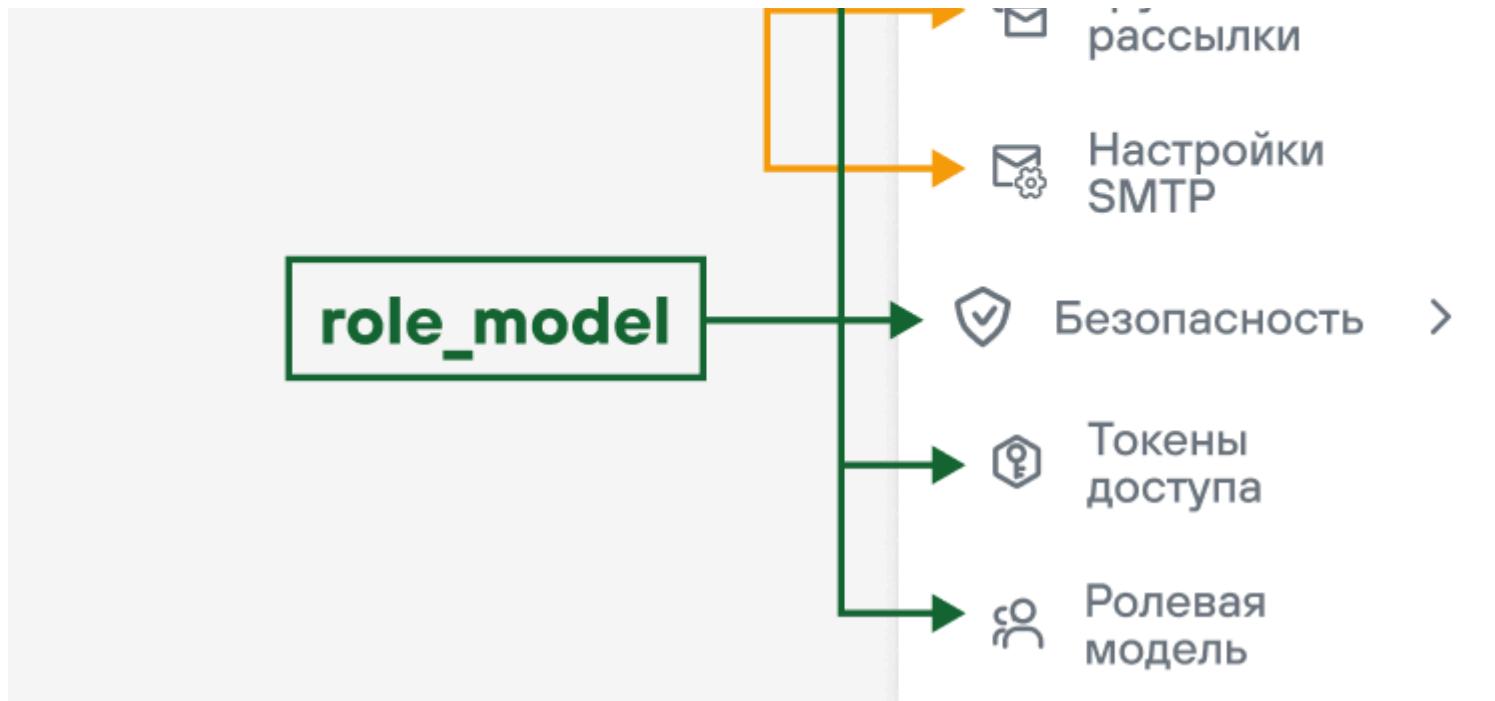


Иллюстрация доступа к разделам для страницы “Визион (администрирование)”:

ВИЗИОН





17.5.2 Список уведомлений

В разделе “Уведомления” пользователи с ролями администраторов видят весь список уведомлений, создаваемых всеми объектами Контура.

Для остальных пользователей список отображаемых уведомлений определяется комбинацией полномочий:

- на объекты Контура (ПАК) ;
- на категории алERTов .

Например, если роль имеет полномочия только на ПАК с именем “ПАК1” и на категорию алERTов “server”, пользователь, которому назначена только эта роль, увидит только уведомления касающиеся ПАК1 и его серверной инфраструктуры.

Список категорий алERTов :

Код категории	Название	Описание

server	Сервер	Мониторинг серверной инфраструктуры Контура
network	Сеть	Мониторинг сетевой инфраструктуры Контура
storage	Хранение данных	Мониторинг подсистем хранения данных - диски, дисковые массивы, дисковые полки и тп
func_service	Функциональные сервисы	Сервисы, связанные с функциональной специализацией ПАКов - БД postgres, greenplum, виртуализация и тп
utility_service	Служебные сервисы	Служебные сервисы Контура - Визион, Геном, Спектр
archive	Архив	Категория для алертов, сгенерированных до введения системы категорий алертов
other	Прочее	Категория, назначаемая пользователем на случай, если другие категории не подходят

Список категорий алертов предустановлен в Визион и не может быть изменен в интерфейсе пользователя.

17.5.3 Стартовая страница Визион

На стартовой странице пользователям с ролями администраторов доступна информация по всем ПАК Контура. Остальные пользователи увидят только те ПАК и уведомления, на которые имеют полномочия (см. раздел 17.5.2).

17.5.4 Страница статистики уведомлений

На странице статистики уведомлений пользователям с ролями администраторов доступна информация по всем ПАК Контура. Остальные пользователи увидят только статистику по тем ПАК и тем уведомлениям, на которые имеют полномочия (см. раздел 17.5.2).

17.5.5 Дашборды

Дашборды отображаются в интерфейсе Grafana. Пользователь с ролью Администратора Мониторинга (`vsn_config`) имеет административные полномочия в Grafana и может создавать или изменять дашборды. Пользователи с ролями Администратора объектной модели (`obj_model`) или Администратора ИБ (`role_model`) могут просматривать все дашборды, но не могут создавать или изменять их. Остальные пользователи имеют возможность просматривать дашборды, на которые их ролям назначены полномочия в разделе “Безопасность – Ролевая модель”.

Например, ролям `r_networks`, `r_servers` и `r_db` назначены следующие полномочия на дашборды:

Объект	r_networks	r_servers	r_dba
> Контур	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
> Категории	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Дашборды	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Визион	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Главная	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
МБД.Г	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
МБД.КХ	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
МБД.П	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
МБД.С	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
МБД.Т	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
МБД.Х	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
МВ.ВК	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
МВ.ДИ	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
МВ.С	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Тогда пользователи с ролью “r_networks” и “r_servers” увидят дашборды, входящие в группу “Главная”, а пользователи с ролью “r_dba” увидят дашборды, входящие в группу “МБД.П”.

В случае, если Администратор Мониторинга создает новые дашборды в Grafana, он должен вручную назначить им полномочие “view” для групп, соответствующих ролям Администратора ИБ (role_model) и Администратора ОМ (obj_model).

17.5.6 Функционал рассылки уведомлений

Группы рассылки уведомлений создаются автоматически при сохранении полномочий для роли (значение поля “Добавление” равно “Авто”).

ВИЗИОН

Уведомления

Дашборды

Объекты

Настройки

Метрики

Правила оповещения

Конструктор выражений

Список получателей

Группы рассылки

Настройки SMTP

Безопасность

Ролевая модель

Токены доступа

Статус Визиона

О программе

Настройки > Группы рассылки

Добавить Поиск

Имя	Описание	Добавление
obj_model		Авто
role_model		Авто
r_networks		Авто
r_servers		Авто
r_dba		Авто
manual_group_critical	Получение критических алертов	Ручное

В случае удаления роли автоматически удаляется группа рассылки для нее.

Администратор не может изменить настройки для автоматически созданной группы или удалить ее вручную.

Получатели для таких групп рассылки (их e-mail адреса) добавляются автоматически из провайдера аутентификации на периодической основе (значение поля “Добавление” равно “Авто”).

Recipient	Address	Addition	Active	More
SNMP-шлюз	https://127.0.0.1:9464/alerts	Авто	<input checked="" type="checkbox"/>	⋮
u_networks	egor6@a.ru	Авто	<input checked="" type="checkbox"/>	⋮
u_objmodel	egor21@a.ru	Авто	<input checked="" type="checkbox"/>	⋮
u_superuser	egor111@a.ru	Авто	<input checked="" type="checkbox"/>	⋮
u_servers	egor_rbac_test@a.ru	Авто	<input checked="" type="checkbox"/>	⋮
u_rolemodel	egor33@a.ru	Авто	<input checked="" type="checkbox"/>	⋮
manual_receiver	test@mmm.ru	Ручное	<input type="checkbox"/>	⋮

Администратор может вручную отключить пользователя от рассылки уведомлений используя переключатель в поле “Активно”.

Более подробно настройка провайдера аутентификации и Визион для автоматизированных рассылок уведомлений описана в [инструкции](#).

У администратора остается возможность вручную добавлять группы рассылки и получателей также как в предыдущих версиях Визион.

17.6 Поддержка устаревших ролей `reader` и `writer`

Роли `reader` и `writer`, которые использовались в предыдущих версиях Визион, не поддерживаются в новой ролевой модели. Администратору необходимо добавить недостающие роли и назначить полномочия. Например, администратор может добавить роль с именем `reader` и назначить полномочия на просмотр всех объектов Контура и все плагины, тогда пользователи с ролью `reader` смогут просматривать уведомления для объектов всего Контура.

17.7 Ролевая модель для Grafana

Grafana не требует отдельной авторизации через провайдера аутентификации, авторизация пользователя происходит через Визион. Для этого в [конфигурационном файле](#) `vision_core` должен быть выставлен параметр `enabled: true` в блоке `auth`.

Пользователь Визион с ролью администратора мониторинга (`vsn_config`) обладает ролью `administrator` в Grafana и имеет полномочия создавать и изменять дашборды. Все остальные пользователи имеют только полномочия просмотра дашбордов.

При создании новой роли в Визион автоматически создается соответствующая группа (`team`) в Grafana. В разделе “Ролевая модель” администратор ИБ видит все настроенные в Grafana дашборды и назначает полномочия на них для ролей пользователя.

Когда пользователь заходит в интерфейс Grafana, то он видит только те дашборды, на которые назначены полномочия для групп (`teams`) соответствующих его ролям в Визион.

18 Смена технологических паролей

Секреты vault

`vault.auth.client_secret`

`vault.database.password`

`vault.general.cookie_secret_key`

`vault.plagent.token`

vault.smtp_config.smtp_auth_password
vault.snmp_notifier.snmp_authentication_password
vault.snmp_notifier.snmp_private_password
vault.grafana.basic_auth_password
vault.grafana.security
vault.grafana.auth
vault.grafana.metrics
vault.snmp_notifier.basic_auth_password
vault.alertmanager.basic_auth_password
vault.victoriametrics.basic_auth_password
vault.vmagent_agent.basic_auth_password
vault.vmagent_proxy.basic_auth_password
vault.vlagent_agent.basic_auth_password
vault.vlagent_proxy.basic_auth_password
vault.vmalert.basic_auth_password
vault.auth.tech_auth_password
vault.auth.tech_auth_client_secret
vault.vision_backup.basic_auth_password
vault.victorialogs.basic_auth_password
vault.vmalert_v1.basic_auth_password
vault.vision_utilizer.basic_auth_password
vault.vision_iamsyncer.basic_auth_password

```
vault.alertcollector.basic_auth_password  
vault.vmalert_external_integration.auth  
vault.vmalert_external_integration.auth  
vault.vmalert_external_integration.auth
```

18.1 Смена пароля для postgres

Необходимо выполнить команду:

```
psql -U vision -d vision_db
```

Выполнить sql-выражение:

```
ALTER ROLE vision WITH PASSWORD 'новый пароль' ;
```

Выйти из утилиты psql: `\q`

Изменить поле в `vault vault.database.password`.

Произвести перезагрузку компонентов `vision_core`, `vision_taskdaemon`, `vision_alertcollector` путем выполнения следующих команд:

```
systemctl restart vision_core
```

```
systemctl restart vision_taskdaemon
```

```
systemctl restart vision_alertcollector
```

18.2 Смена пароля для victoriametrics

Изменить поле в `vault vault.victoriametrics.basic_auth_password`.

Выполнить скрипт `/opt/skala-r/vision/tools/update_server_configs.sh`.

Переустановить Прокси Визиона.

18.3 Смена пароля для victorialogs

Для изменения пароля предварительно необходимо выполнить команду: `/opt/skala-r/vision/server/vision_venv/bin/python3 -c "from skalar.vision.common.encode import encode_secret; print(encode_secret('новый пароль'))"`

Сменить закодированный Визионом пароль в файле `/opt/skala-r/etc/vision/server/victorialogs/environment.env`

Перезапустить компонент: `systemctl restart victorialogs`

Изменить поле в `vault.victorialogs.basic_auth_password`.

Выполнить скрипт `/opt/skala-r/vision/tools/update_server_configs.sh`.

18.4 Смена пароля для vmalert_st

Изменить поле в `vault.vmalert.basic_auth_password`.

Выполнить скрипт `/opt/skala-r/vision/tools/update_server_configs.sh`.

18.5 Смена пароля для vmalert_v1

Изменить поле в `vault.vmalert_v1.basic_auth_password`.

Выполнить скрипт `/opt/skala-r/vision/tools/update_server_configs.sh`.

18.6 Смена пароля для AlertManager

Изменить поле в `vault.alertmanager.basic_auth_password`.

Выполнить скрипт `/opt/skala-r/vision/tools/update_server_configs.sh`

18.7 Смена пароля для grafana

Изменить поле в vault.vault.grafana.metrics.basic_auth_password.

Выполнить скрипт /opt/skala-r/vision/tools/update_server_configs.sh

18.8 Смена пароля для vision_alertcollector

Изменить поле в vault.vault.alertcollector.basic_auth_password.

Выполнить скрипт /opt/skala-r/vision/tools/update_server_configs.sh

18.9 Смена пароля для snmp_notifier

Для изменения пароля предварительно необходимо выполнить команду: `/opt/skala-r/vision/server/vision_venv/bin/python3 -c "from skalar.vision.common.encode import encode_secret; print(encode_secret('новый пароль'))"`

Изменить поле в vault.vault.snmp_notifier.basic_auth_password.

Выполнить команду:

```
systemctl restart snmp_notifier
```

Выполнить скрипт /opt/skala-r/vision/tools/update_server_configs.sh.

18.10 Смена пароля для vision_utilizer

Изменить поле в vault.vault.vision_utilizer.basic_auth_password

Выполнить скрипт /opt/skala-r/vision/tools/update_server_configs.sh.

18.11 Смена пароля для vision_iamsyncer

Изменить поле в vault.vault.vision_iamsyncer.basic_auth_password

Выполнить скрипт /opt/skala-r/vision/tools/update_server_configs.sh.

18.12 Смена пароля для vision_backup

Изменить поле в vault vault.vision_backup.basic_auth_password

Выполнить скрипт /opt/skala-r/vision/tools/update_server_configs.sh.

18.12 Смена пароля для vmagent_agent

Изменить поле в vault vault.vmagent_agent.basic_auth_password

Выполнить скрипт /opt/skala-r/vision/tools/update_server_configs.sh.

При необходимости переустановить все агенты метрик

18.13 Смена пароля для vmagent_proxy

Изменить поле в vault vault.vmagent_proxy.basic_auth_password

Выполнить скрипт /opt/skala-r/vision/tools/update_server_configs.sh.

При необходимости переустановить все агенты и прокси метрик

18.14 Смена пароля для vlagent_agent

Изменить поле в vault vault.vlagent_agent.basic_auth_password

Выполнить скрипт /opt/skala-r/vision/tools/update_server_configs.sh.

При необходимости переустановить все агенты логов

18.15 Смена пароля для vlagent_proxy

Изменить поле в vault vault.vlagent_proxy.basic_auth_password

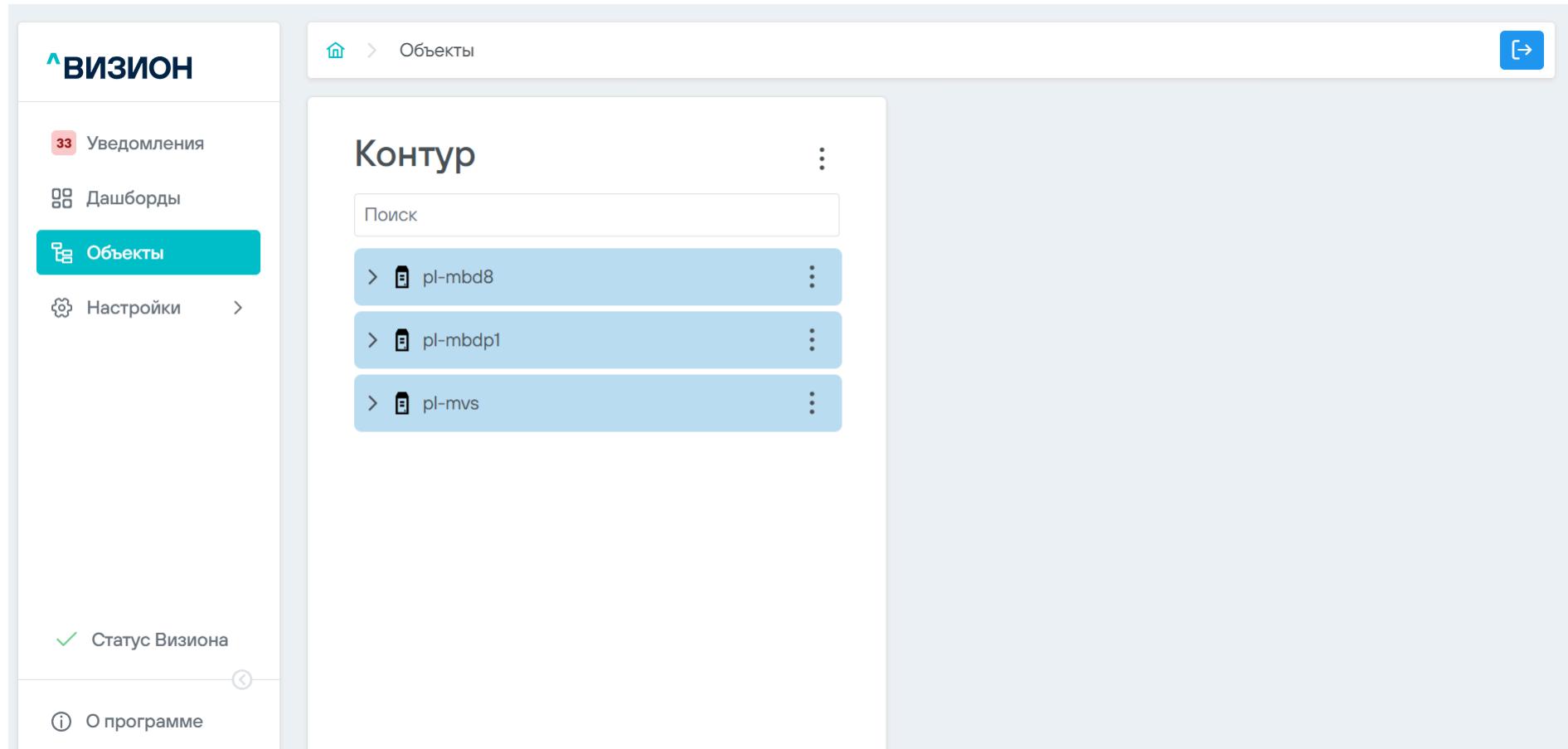
Выполнить скрипт /opt/skala-r/vision/tools/update_server_configs.sh.

При необходимости переустановить все агенты и прокси логов

18.16 Смена пароля для плагинов

Смена паролей для плагинов производится при помощи пользовательского интерфейса Визион. Необходимо авторизоваться под пользователем с правами доступа к объектной модели.

Перейти в раздел “Объекты”



В “Контуре” выбрать необходимый ПАК при помощи нажатия левой клавиши мыши на наименовании ПАК

33 Уведомления

Дашборды

Объекты

Настройки >

✓ Статус Визиона

i О программе

Контур

Поиск

pl-mbd8

pl-mbdp1

pl-mvs

Метрики Уведомления Обзор Агенты Плагины

pl-mbd8

МБД.Г

Модули

Имя	Тип	Описание
db-module		Модуль вычисления и хранения

Серверы

Имя	Модуль	Тип	Адрес	Адрес ВН
mbd8...	db-m...	Базовый сервер	192.16...	192.16...
mbd8...	db-m...	Базовый сервер	192.16...	192.16...
mbd8...	db-m...	Базовый сервер	192.16...	192.16...

Справа выбрать вкладку “Плагины”. Рядом с плагином, для которого необходимо изменить пароль, нажать на action bar(три точки) и выбрать действие “Конфигурировать”.

- 5 Уведомления
- 00 Дашборды
- Объекты**
- Настройки >
- Безопасность
- Ролевая модель
- Токены доступа

✓ Статус Визиона

Объекты > Контур

Контур

Поиск

- > Pak, pak
- > spr-g113**
- > pak-era

Имя	Расположение	Статус
sys_exporter	vision	В РАБОТЕ
postgres_exporter_m...	vision	В РАБОТЕ
graphite_exporter	vision	В РАБОТЕ
kafka_exporter	vision	В РАБОТЕ
picodata_scraper	vision	В РАБОТЕ
clickhouse_sql_expor...	vision	В РАБОТЕ
snmp_exporter	vision	В РАБОТЕ

Метрики Уведомления Обзор Агенты Плагины

Поиск

Имя ↑ Расположение ↑ Статус ↑

sys_exporter vision **В РАБОТЕ**

postgres_exporter_m... vision **В РАБОТЕ**

graphite_exporter vision **В РАБОТЕ**

kafka_exporter vision **В РАБОТЕ**

picodata_scraper vision **В РАБОТЕ**

clickhouse_sql_expor... vision **В РАБОТЕ**

snmp_exporter vision **В РАБОТЕ**

Конфигурировать
Деинсталлировать
НЕ УСТАНОВЛ.

Для смены логина и пароля плагина необходимо изменить значения в полях **Basic Auth Username** и **Basic Auth Password**

Конфигурация плагина `postgres_exporter_multi vision`

Scrape Interval i
15s

Scrape Timeout i
10s

Basic Auth Username i
vision

Basic Auth Password i
.....

Tls Cert File i
/opt/skala-r/vision/cert/components_default.crt

Tls Key File i
/opt/skala-r/vision/cert/components_default.key

Применить

После внесения изменений необходимо нажать на кнопку “Применить”

19 Диагностический дамп Визиона

19.1 Назначение

Диагностический дамп предназначен для анализа и диагностики состояния Визиона, его компонентов, агентов, прокси, а также другой системной информации на локальном или удаленном узле. Собираемый диагностический дамп очищается от пользовательских данных и секретов.

В Диагностический дамп включена следующая информация:

- Системный журнал, логи работы компонентов, агентов и прокси
- Конфигурационные файлы
- Состояние компонентов, агентов и прокси
- Информацию о состоянии памяти и кэше, процессах, стеке вызовов, CPU, дисках
- Версия ОС
- Системное время на узле

19.2 Расположение

Скрипт для сбора диагностического дампа `./run.sh` находится в дистрибутиве Визиона в папке `/diagnostic`.

19.3 Сбор диагностического дампа

Для работы диагностического дампа необходимо, чтобы скрипты `./run.sh` и `./collect.sh` были расположены в одной директории.

19.3.1 Сбор диагностического дампа с локального узла

Для запуска локальной диагностики необходимо запустить скрипт `./run.sh` на том узле, для которого необходимо собрать диагностический дамп.

19.3.2 Сбор диагностического дампа с удаленного узла

Для запуска удаленной диагностики необходимо запустить скрипт `./run.sh` с дополнительными параметрами:

`./run.sh user@host port`

`user` - имя пользователя(обязательный параметр)

`host` - адрес подключения(обязательный параметр)

`port` - адрес порта(необязательный параметр)

В случаях, когда данные для подключения указаны неверно, либо удалённый узел недоступен, в логе работы скрипта будет отображена информация о неудачной попытке подключения:

```
[root@vision_dump_alt10_152 ~]# ./run.sh root@192.168.190.197
Удаленный запуск диагностического дампа (root@192.168.190.197).
Активация соединения.
ssh: connect to host 192.168.190.197 port 22: No route to host
Не удалось активировать соединение.
```

19.4 Архив и лог работы

Диагностический дамп собирается в архив. Архив создается после завершения сбора данных и помещается в ту же директорию, в которой находится скрипт `./run.sh`.

Во время работы скрипта отображается лог его работы. После завершения работы лог `diagnostic.log` помещается в архив с остальными данными. Данные в архиве сгруппированы по папкам:

- `/configs` – конфигурационные файлы компонентов, агентов и прокси
- `/logs` – логи работы компонентов, агентов и прокси
- `/sys` – информация о состоянии памяти и кэше, CPU, дисках и т.д.
- `/vision` – файлы с информацией о процессах, сервисах, а также файл с информацией о директориях и правах на чтение, запись и исполнение в этих директориях.

20 Резервное копирование и восстановление данных

ПО Визион позволяет создавать и восстанавливать резервные копии своей конфигурации.

Восстановление можно выполнить только на той же версии Визион, где создавалась резервная копия.

Создаваемые резервные копии по умолчанию сохраняются в директорию `/opt/skala-r/var/backups/vision`.

Для получения доступа к разделу “Резервная копия” необходимо:

1. Авторизоваться в ПО Визион под учётной записью с правами администратора мониторинга Визион.
2. Перейти в раздел “О программе” и выбрать “Резервная копия”.

The screenshot shows the Vizion software interface. On the left is a sidebar with the following items:

- Уведомления (Notifications) with 57 notifications
- Дашборды (Dashboards)
- Настройки (Settings) with a dropdown arrow
- Метрики (Metrics)
- Правила оповещения (Alert rules)
- Конструктор выражений (Expression builder)
- Список получателей (List of recipients)
- Группы рассылок (Email groups)
- Статус Визиона (Vizion status) with a green checkmark
- О программе (About) with a blue button

The main content area is titled "О программе" > "Резервная копия". It shows the "Резервное копирование по расписанию" (Scheduled backup) section, which is currently active. The "Периодичность" (Frequency) is set to "12 часов" (12 hours). Below this, the "Резервные копии" (Backup copies) section lists several backup files:

- backup_1.6-537_2025-03-14_11-10.tar
- backup_1.6-537_2025-03-14_12-00.tar
- backup_1.6-537_2025-03-15_00-00.tar
- backup_1.6-537_2025-03-15_12-00.tar
- backup_1.6-537_2025-03-16_00-00.tar
- backup_1.6-537_2025-03-16_12-00.tar
- backup_1.6-537_2025-03-17_00-00.tar

Each backup file has a blue circular icon with a gear symbol to its right.

20.1 Создание резервной копии по расписанию

Для создания резервной копии по расписанию необходимо переключить ползунок рядом с записью “Резервное копирование по расписанию”.

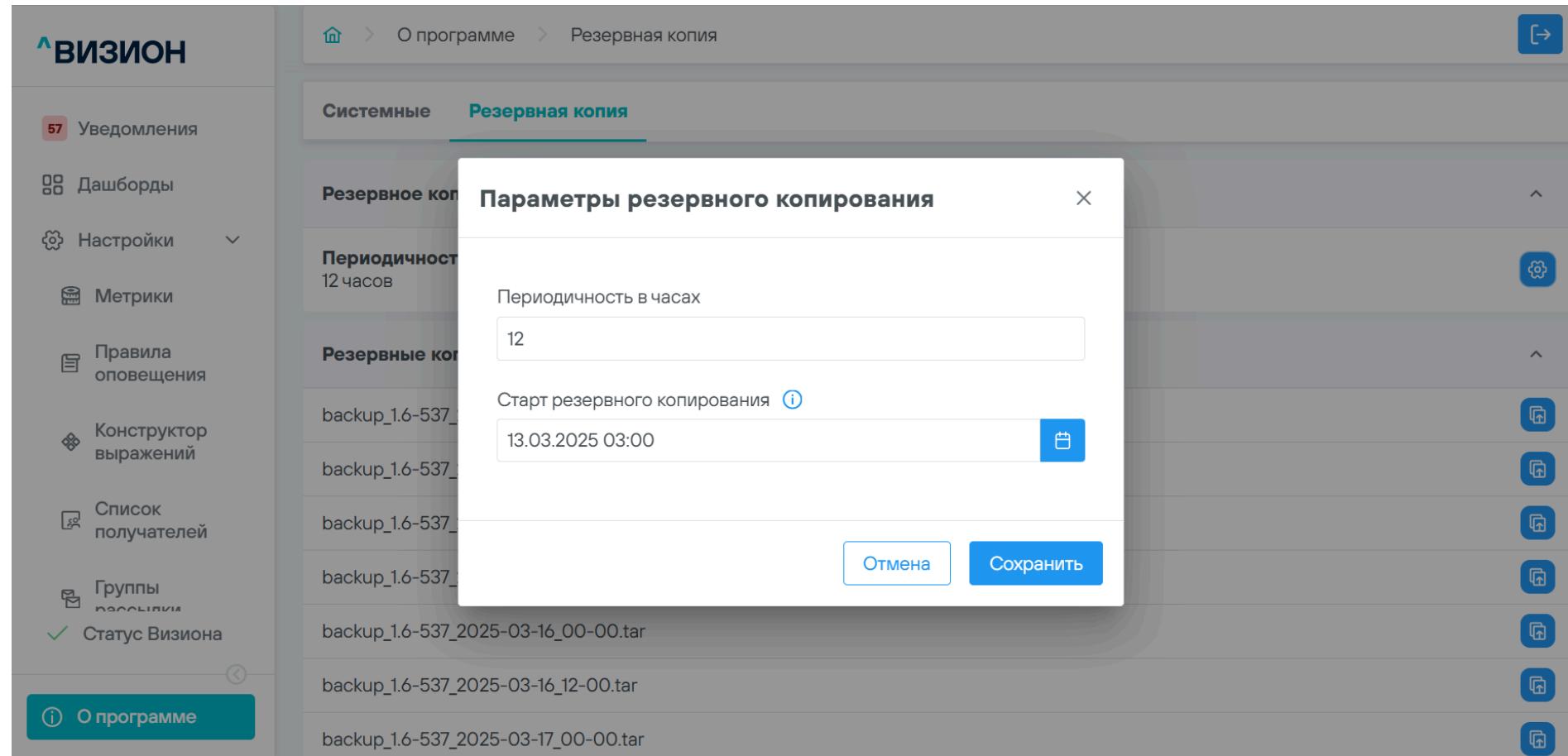
Резервное копирование по расписанию



Резервное копирование по расписанию включено

Рядом с записью “Периодичность” нажать на значок шестерёнки. Откроется модальное окно “Параметры резервного копирования”, в котором:

- значение в поле “Периодичность в часах” отвечает за то, как часто будет выполняться резервное копирование;
- значение в поле “Старт резервного копирования” отвечает за то, с какой даты будет запущено резервное копирование по расписанию.



20.2 Создание резервной копии вручную

Для создания резервной копии вручную необходимо рядом с записью “Резервные копии” нажать на кнопку “Создать резервную копию сейчас”. При нажатии на кнопку будет создана резервная копия.

57 Уведомления

Дашборды

Настройки

Метрики

Правила
оповещенияКонструктор
выраженийСписок
получателейГруппы
получателей

✓ Статус Визиона

Системные Резервная копия

Резервное копирование по расписанию



Периодичность

12 часов



Резервные копии



Создать резервную копию сейчас

backup_1.6-537_2025-03-14_11-10.tar



backup_1.6-537_2025-03-14_12-00.tar



backup_1.6-537_2025-03-15_00-00.tar



backup_1.6-537_2025-03-15_12-00.tar



backup_1.6-537_2025-03-16_00-00.tar



20.3 Восстановление данных

Для восстановления данных из резервной копии необходимо рядом с созданной резервной копией в разделе “Резервные копии” нажать на кнопку “Восстановить данные из резервной копии”.

Резервные копии



backup_1.6-537_2025-03-14_11-10.tar

Восстановить данные из резервной копии



backup_1.6-537_2025-03-14_12-00.tar



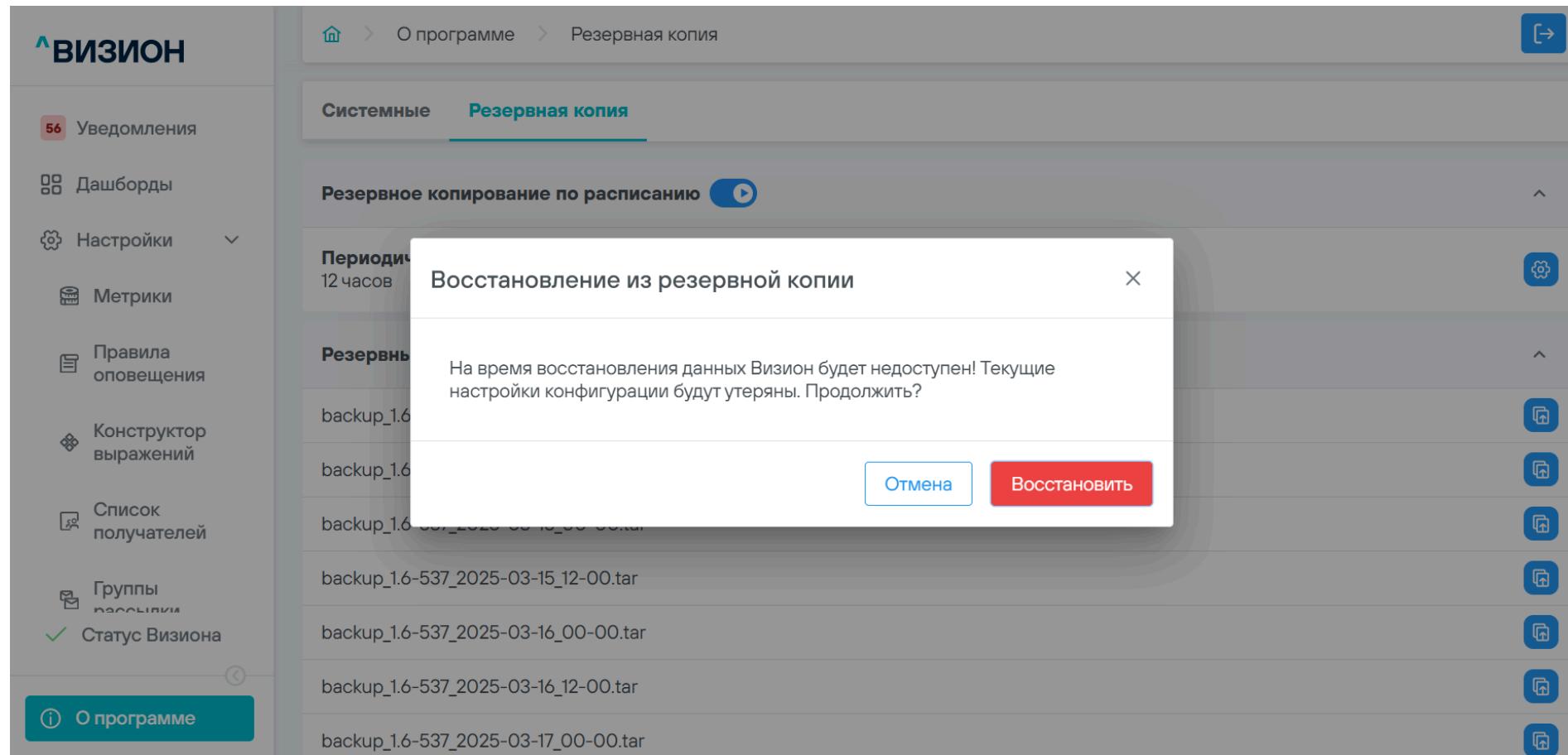
backup_1.6-537_2025-03-15_00-00.tar



backup_1.6-537_2025-03-15_12-00.tar



В появившемся модальном окне нажать на кнопку “Восстановить” .



21 Замена SSL-сертификата в Nginx

ПО Визион поставляется с самоподписанными ssl-сертификатами. Для замены ssl-сертификатов необходимо:

- добавить ключ и сертификат в директорию `/etc/nginx/ssl`
- открыть файл `/snippets/vision-ssl-params.conf` и указать пути до нового сертификата и ключа в `ssl_certificate` и `ssl_certificate_key`, соответственно

```
ssl_certificate /etc/nginx/ssl/vision.crt;  
ssl_certificate_key /etc/nginx/ssl/vision.key;
```

- перезагрузить сервис nginx:

```
systemctl restart nginx.service
```

Генерация сертификата производится стандартным способом. Пример генерации сертификата через openssl:

```
openssl req -x509 -nodes -newkey rsa:2048 -days 3650 \  
-keyout vision.key \  
-out vision.crt \  
-subj "/C=RU/L=Moscow/O=Skala-r/CN=localhost" \  
-addext "subjectAltName = DNS:localhost,IP:127.0.0.1"
```