



Скала^р Визион
(ранее Спектр.Мониторинг)

v.0.9-24

Руководство администратора

ГЛОССАРИЙ

Термины и сокращения, которые используются для описания внутреннего устройства и функций программы:

Термин	Описание, комментарий
Мониторинг	Непрерывный процесс сбора и анализа информации о значении параметров состояния объекта с целью выявления их соответствия желаемому результату
Платформа	Программная платформа Скала^р
Программно-аппаратный комплекс (ПАК)	Совокупность аппаратных компонентов (сервера, коммутаторы доступа и т. п.) и программного обеспечения, решающих прикладные задачи (машина баз данных, машина больших данных и т. д.)
Система мониторинга	Модуль Мониторинга ПАК в составе Программной платформы Скала^р
Объект мониторинга	Ресурс ПАК, о котором агент собирает данные
Объект управления	Компонент ядра системы мониторинга
Плагин объекта управления	Подключаемый программный компонент, позволяющий конфигурировать объекты управления, а также производить интеграцию со сторонними системами сбора метрик
Репозиторий ресурсов объектов управления	Наборы графических информационных панелей, справочник собираемых метрик, набор правил оповещений, исполняемые файлы агентов мониторинга
Модуль Управления кластером	Модуль платформы, обеспечивающий управление и конфигурацию кластерного ПО ПАК
Модуль единой авторизации и разграничения прав доступа	Модуль платформы, обеспечивающий авторизацию пользователей платформы и контролирующей возможность взаимодействия с ее компонентами в зависимости от роли пользователя
Пользовательский интерфейс	Интерфейс, обеспечивающий передачу информации между пользователем и Модулем Мониторинга ПАК, а также просмотр информации об объектах мониторинга

Агент сбора данных (экспортер)	Процесс ОС, собирающий данные о соответствующем объекте мониторинга и предоставляющий к ним доступ по HTTP
Временной ряд	Дискретное множество значений во времени определенного числового атрибута (параметра) объекта мониторинга
База данных временных рядов	Система хранения простых значений, привязанных к определенным моментам времени
Метрика	Временной ряд, который показывает изменение какой-либо величины во времени
Прореживание метрики	Механизм архивирования исторических данных, агрегирующий значения метрики в соответствии с политикой прореживания
Визуализация	Процесс презентации метрик в визуально удобном формате, что позволяет быстро и интуитивно интерпретировать данные
Информационная панель	Интерактивная панель с наглядной визуализацией определенных процессов или состояний, представленных в виде таблиц, диаграмм, графиков
Эффект наблюдателя	Воздействие системы мониторинга на наблюдаемые метрики
Оповещение	Процедура отправки уведомления о срабатывании предупреждения
Правило оповещения	Информация, определяющая действие по отношению к конкретному экземпляру предупреждения при его получении
Уведомление	Текстовое сообщение о предупреждении
Предупреждение	Индикатор состояния соответствующей метрики, определяющий факт срабатывания правила предупреждения
Правило предупреждения	Условие, применяемое к значению метрики и необходимое для срабатывания предупреждения

Канал уведомления	Информация о получателях и методах отправки оповещений
Сервер приложения	Централизованный сервер, осуществляющий конфигурирование и управление ядром мониторинга и агентами сбора данных, реализацию бизнес-логики, а также взаимодействие с пользовательским интерфейсом
Ядро модуля мониторинга	Совокупность взаимоувязанного ПО, обеспечивающего сбор, обработку, хранение и визуализацию информации, полученной от агентов, а так же оповещение о выходе значений метрик за допустимые пределы
База данных	Объект сервиса СУБД, в котором пользователи создают таблицы, представления, хранимые процедуры. Обычно для каждого сервиса СУБД создаётся база данных по умолчанию (PostgreSQL), но может быть несколько баз данных.
СУБД	С истема У правления Б азами Д анных. Набор программ, позволяющий организовывать, контролировать и администрировать базы данных.
Системная база данных	БД, предназначенная для хранения информации об объектах управления, конфигурации ПАК, ролевой модели доступа, конфигурации Модуля Мониторинга
AlertManager	Стороннее ПО для организации уведомлений по определенным правилам с использованием различных каналов связи
Grafana	Стороннее ПО, позволяющее отображать данные из Prometheus в виде графиков и диаграмм, организованных в информационные панели
PromQL	Язык запросов Prometheus
Victoria Metrics	Стороннее ПО, отвечающее за сбор и хранение метрик объектов

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование программы

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

Краткое наименование: ПО «Скала^р Визион».

1.2 Область применения

Основным предназначением ПО «Скала^р Визион» является предоставление возможности контроля жизненного цикла ПАК (машин) производства Скала^р, обслуживаемых персоналом, к квалификации которого не предъявляется серьезных требований. Осуществляется процесс мониторинга программных компонент ПАК, сбор информации о количестве и составе управляемых объектов, метрик, построение отчетов по этим данным, обеспечивает оповещение при сбоях работы.

1.3 Уровень подготовки администраторов

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 Лингвистическое обеспечение

Основной язык в настроенных панелях отображения метрик, уведомлений, графической информации – Русский (пользовательский).

Дополнительный язык в интерфейсах всех подсистем – Английский (пользовательский).

Основной язык интерпретации событий журналов подсистем – Английский.

Дополнительный язык интерпретации событий журналов подсистем – Русский.

1.5 Обеспечение защиты информации

Информационная безопасность ПО «Скала^р Визион» достигается путем шифрования данных и соединений сервер-клиент по протоколу SSL на всех этапах их жизненного цикла. Кроме этого, все внутренние взаимодействия компонентов ПО «Скала^р Визион» также могут быть зашифрованы с помощью протокола SSL.

2 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

2.1 Виды деятельности, функции и классы решаемых задач

Основные функции ПО «Скала^р Визион»:

- Сбор данных о конфигурации элементов ПАК (машин).
- Сбор данных, отображение, мониторинг элементов программного обеспечения, активных компонентов модулей ПАК, служебных сервисов и сервисов баз данных
- Конфигурирование метрик мониторинга, настройка предупреждений
- Конфигурирование графического отображения на информационных панелях в виде графиков, отдельных значений, диаграмм, таблиц
- Хранение метрик с возможностью настройки глубины хранения и управления жизненным циклом хранимых данных.
- Отображение в графическом пользовательском интерфейсе подробных данных об объектах мониторинга.
- Мониторинг изменений объектов контроля в режиме, близком к реальному времени
- Настройка и выполнение командного кода на стороне ПАК и на виртуальной машине Мониторинга с целью сбора данных для контроля работы объектов мониторинга
- Сопряжение конфигураций Prometheus, Victoria Metrics, Grafana и AlertManager для обеспечения кооперативного поведения;
- Установка информационных панелей для каждого типа ПАК

2.2 Характеристики и конфигурация технических средств

Тип реализующей ЭВМ: x86-64.

Минимальная конфигурация сервера управления ПО «Скала^р Визион»:

- CPU: от 2 ядер;
- RAM: от 16 Гб;
- ROM: от 600 Гб SSD для системных компонент;
- NET: от 1 Гб Ethernet.
- Internet: доступ к сети, возможно через прокси-сервер, конфигурация используется как опция и используется в контурах, где имеется и разрешён доступ к внешней по отношению к контуру сети.

2.3 Операционная среда и общесистемные программные средства

Для установки и работы ПО «Скала^р Визион» требуется операционная система ALT 8 SP Server.

3 РАЗВЕРТЫВАНИЕ

3.1 Состав дистрибутива

Дистрибутив «Скала^р Визион» представляет из себя архив, содержащий необходимые пакеты, так что установка может производиться даже на систему, не имеющую доступа к сети Internet.

Состав архива:

- Компоненты сервера Визион (vision-api, vision-core, vision-task-daemon, vision_alert_collector)
- Компоненты ядра мониторинга (alertmanager, grafana, victoriametrics, vmalert, snmp_notifier)
- UI Визиона (vision-ui)
- Компоненты агента мониторинга на узлах (vmagent, node_exporter, ha_cluster_exporter, postgres_exporter, utlz_exporter)
- Компоненты сбора метрик (серверные) (snmp_exporter, ipmi_exporter, blackbox_exporter)
- Вспомогательные файлы для разворачивания (плейбуки, скрипты, файлы конфигурации)

3.2 Подготовка

На машине, с которой выполняется разворачивание, требуются наличие следующих пакетов:

- ansible
- sshpass
- wget

Установить данные пакеты можно с помощью следующих команд (в процессе установки согласиться на установку зависимостей):

```
sudo apt-get update  
sudo apt-get install ansible wget sshpass
```

Скопировать дистрибутив Визиона в директорию /opt.

Перейти в директорию /opt командой

```
cd /opt
```

Распаковать файл с дистрибутивом Визиона командой

```
tar -xvzf ./vision-dist-altlinux8.4-v0.9-24.01.tar.gz
```

Перейти в каталог с дистрибутивом командой `cd ./vision-dist-altlinux8.4-v0.9-24.01/`
Дальнейшие команды, если не указано другое, выполняются из этой директории.

3.2 Конфигурирование развертывания

Для настройки мониторинга необходимо заполнить файл inventory с параметрами развёртывания согласно инсталляционной карте. Для примера ориентируйтесь на файл inventory.yml. Далее по тексту он будет использоваться во всех примерах.

В файле укажите:

- в разделе ``all`` ➤ ``vars`` (глобальные настройки):

- ``ansible_user`` и ``ansible_ssh_pass`` - данные для доступа по ssh для всех узлов (по-умолчанию)
- ``pak_type`` - тип ПАКа (значения из списка, например, PG для МБД.П)
- ``pak_name`` - имя ПАКа, например "PG-P-1"
- ``vision_ip`` - список получателей (экземпляров Визиона), которым отправляются метрики vmagent'ами по remoteWrite

- в разделе ``all`` ➤ ``children`` ➤ :

- в подразделе ``vision_hosts`` ➤ ``hosts`` ➤ ``vision_host`` (настройки ВМ Визиона и его таргетов согласно инсталляционной карте):

- ``ansible_host`` - ip-адрес целевой ВМ Визиона
- ``snmp_targets`` - список устройств, отслеживаемых системой мониторинга по протоколу SNMP, с параметрами
- ``ipmi_targets`` - список устройств, отслеживаемых системой мониторинга по протоколу IPMI, с параметрами

- в подразделе ``pak_hosts`` (настройки узлов согласно инсталляционной карте) содержится перечень целей для установки агента мониторинга со следующими параметрами каждой цели:

- ``ansible_host`` - ip-адрес целевого узла
- ``mbd_pg_user`` и ``mbd_pg_pass`` - креды для postgres на узлах кластера МБД.П (для МБД.П)
- ``is_ha_cluster`` - признак того, что хост является узлом кластера МБД.П (для МБД.П)
- ``labels`` - перечень дополнительных меток:
- ``module_title`` - имя модуля, к которому относится цель
- ``module_id`` - id модуля, к которому относится цель
- ``cluster`` - имя кластера, к которому относится цель (для МБД.П)
- ``type: 'vm`` - признак того, что цель - виртуальная машина

3.3 Установка

Для установки необходимо из директории с дистрибутивом выполнить скрипт `deploy_and_config.sh`. Например:

```
./deploy_and_config.sh
```

При необходимости, можно выполнить установку последовательно

```
ansible-playbook -i inventory.yml 1_vision_install.yml # Установка пакетов сервера
ansible-playbook -i inventory.yml 2_vision_config.yml # Конфигурирование сервера Визиона
ansible-playbook -i inventory.yml 3_agent_install.yml # Установка пакетов агента
ansible-playbook -i inventory.yml 4_agent_config.yml # Конфигурирование агента Визиона
```

Если установка прошла успешно, то результат будет следующим (пример):

```
PLAY RECAP *****
master                : ok=11  changed=9  unreachable=0  failed=0  skipped=1  rescued=0
  ignored=0
segment1              : ok=11  changed=9  unreachable=0  failed=0  skipped=1  rescued=0
  ignored=0
segment2              : ok=11  changed=9  unreachable=0  failed=0  skipped=1  rescued=0
  ignored=0
segment3              : ok=11  changed=9  unreachable=0  failed=0  skipped=1  rescued=0
  ignored=0
smaster               : ok=11  changed=9  unreachable=0  failed=0  skipped=1  rescued=0
  ignored=0
```

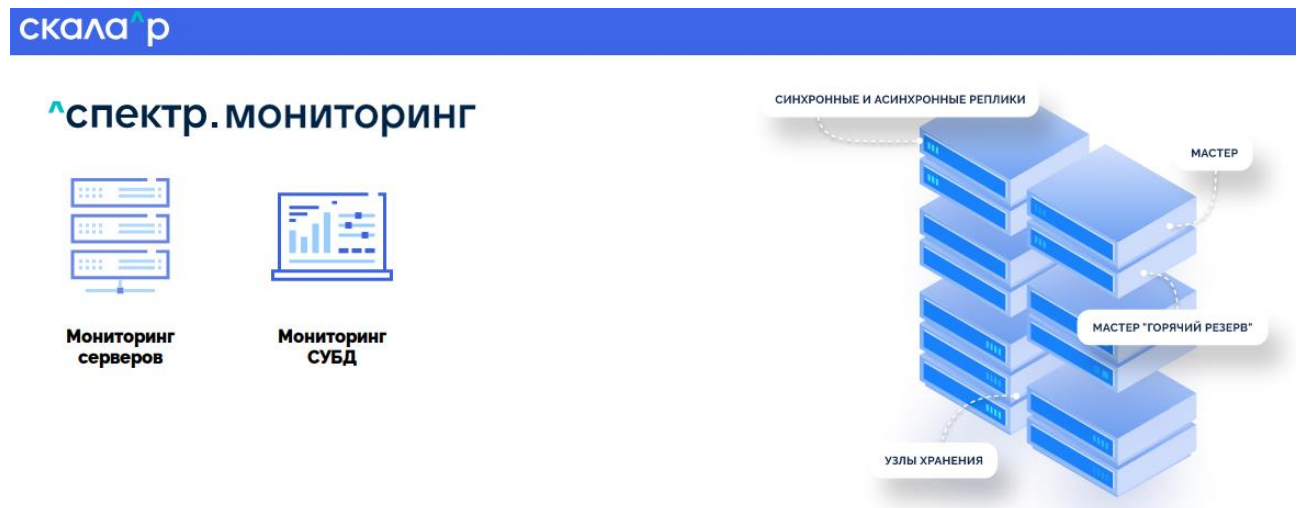
Если в процессе установки возникли ошибки, они будут выделены красным цветом.

3.3 Проверка корректности установки

Далее по тексту предполагается, что сервер Визиона развернут по адресу 192.168.186.130, а агент – по адресу 192.168.186.215.

Зайти на главную страницу и проверить отображение дашбордов в Grafana

<https://192.168.186.130/>



<https://192.168.186.130/vision/grafana/?orgId=1>

General / Визион - Список узлов

Модуль: Модуль баз данных тип 1

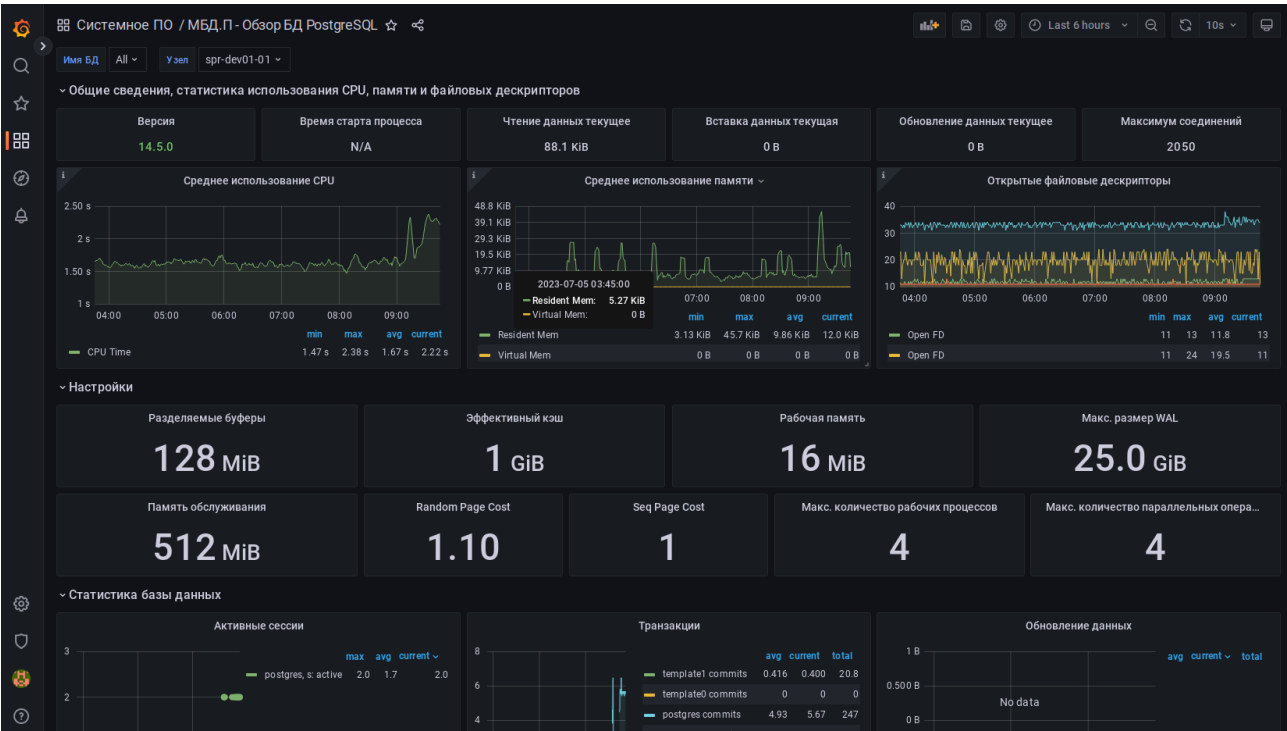
Список узлов [всего: 3]

Узел	Время работы	CPU	Исп. CPU %	RAM	Исп. RAM	Диск	Исп. диск %
spr-mbd-release01	2.4 week	4	2.93%	3.80 GiB	60.34%	59.47 GiB	20.23%
spr-mbd-release02	2.4 week	4	1.72%	3.80 GiB	59.89%	59.47 GiB	20.17%
spr-mbd-release03	2.4 week	4	2.45%	3.80 GiB	60.12%	59.47 GiB	20.48%

Список виртуальных машин [всего: 2]

Узел	Время работы	CPU	Исп. CPU %	RAM	Исп. RAM %	Диск	Исп. диск %
spectrum-136	2.4 week	4	17.00%	3.84 GiB	47.77%	26.16 GiB	68.83%
spectrum-test-stand-0	2.4 week	8	4.01%	3.84 GiB	35.99%	54.72 GiB	94.37%

<https://192.168.186.130/vision/grafana/d/0000000039/mbd-p-obzor-bd-postgresql?orgId=1&refresh=10s> (только для МБД.П)



Зайти в веб-интерфейс VictoriaMetrics и убедиться в наличии зелёных индикаторов в столбце State (пример):
<https://192.168.186.130/vision/victoriametrics/targets>

VictoriaMetrics

Active Targets

AllUnhealthyCollapse allExpand allFilter targets

Active targetsDiscovered targets

alertmanager (1/1 up)

Endpoint	State	Labels	Debug relabeling	Scrapes	Errors	Last Scrape	Duration	Samples	Last error
http://127.0.0.1:9093/vision/alertmanager/metrics (response)	UP	{instance="127.0.0.1:9093", job="alertmanager"}	target metrics	7436	0	10001ms ago	3ms	281	

grafana (1/1 up)

Endpoint	State	Labels	Debug relabeling	Scrapes	Errors	Last Scrape	Duration	Samples	Last error
http://127.0.0.1:3000/vision/grafana/metrics (response)	UP	{instance="127.0.0.1:3000", job="grafana"}	target metrics	7436	0	10414ms ago	5ms	971	

victoriametrics (1/1 up)

Endpoint	State	Labels	Debug relabeling	Scrapes	Errors	Last Scrape	Duration	Samples	Last error
http://127.0.0.1:8428/vision/victoriametrics/metrics (response)	UP	{instance="127.0.0.1:8428", job="victoriametrics"}	target metrics	7437	0	1712ms ago	3ms	1341	

vmaalert (1/1 up)

Endpoint	State	Labels	Debug relabeling	Scrapes	Errors	Last Scrape	Duration	Samples	Last error
http://127.0.0.1:8880/vision/vmaalert/metrics (response)	UP	{instance="127.0.0.1:8880", job="vmaalert"}	target metrics	7436	0	11376ms ago	2ms	379	

Зайти в веб-интерфейс vmaagent на каждом узле и убедиться в наличии зелёных индикаторов в столбце State (пример):
<http://192.168.186.215:8429/targets>

VictoriaMetrics

Active Targets

All
Unhealthy
Collapse all
Expand all
Filter targets

Active targets
Discovered targets

ha_cluster_exporter (1/1 up)
collapse
expand

Endpoint	State	Labels	Debug relabeling	Scrapes	Errors	Last Scrape	Duration	Samples	Last error
http://127.0.0.1:9664/metrics (response)	UP	{cluster="spr-dev", hostname="spr-dev01-01", instance="192.168.186.215:9664", job="ha_cluster_exporter", module_id="MDL-BS-01", module_title="Модуль 1"}	target metrics	95619	0	14770ms ago	9ms	15	

node_exporter (1/1 up)
collapse
expand

Endpoint	State	Labels	Debug relabeling	Scrapes	Errors	Last Scrape	Duration	Samples	Last error
http://127.0.0.1:9101/metrics (response)	UP	{hostname="spr-dev01-01", instance="192.168.186.215:9101", job="node_exporter", module_id="MDL-BS-01", module_title="Модуль 1"}	target metrics	95619	0	14664ms ago	13ms	1068	

postgres_exporter (1/1 up)
collapse
expand

Endpoint	State	Labels	Debug relabeling	Scrapes	Errors	Last Scrape	Duration	Samples	Last error
http://127.0.0.1:9187/metrics (response)	UP	{cluster="spr-dev", hostname="spr-dev01-01", instance="192.168.186.215:9187", job="postgres_exporter", module_id="MDL-BS-01", module_title="Модуль 1"}	target metrics	95619	0	10505ms ago	1002ms	53	

Зайти в веб-интерфейс Визиона и убедиться работоспособности компонентов мониторинга (кнопка Статус Визион в нижнем левом углу экрана)

192.168.186.130:8092

ВИЗИОН

Дашборды
Уведомления
Параметры
Метрики
Правила оповещения
Список получателей
Группы рассылки

Параметры
Метрики

Добавить
Поиск

Имя	Описание	Тип
host_disc_space_used	Используемое дисковое пространство	Пользовательская
hostCpuLoadPerCen	Загрузка CPU	Пользовательская
host_cpu_load_total		Пользовательская
hostDiskUsedPercent		Пользовательская
regress_regress_12345		Пользовательская

Состояние компонентов

- Визион БД ✓
- TaskDaemon ✓
- VictoriaMetrics ✓
- vmalert ✓
- Alertmanager ✓
- Grafana ✓
- Nginx ✓
- Локальные экспортеры ✓

1
Статус Визиона

Настройки

4 КОМПОНЕНТЫ ПО

Состав ВМ Визион

Компонент	Версия	Назначение
vision-api	0.9	Компонент API Визион
vision-core	0.9	Компонент сервера Визион
vision-task-daemon	0.9	Таск демон для выполнения фоновых задач
vision-ui	0.9	Компонент UI
vision_alert_collector	0.9	Сбор и обработка оповещений от AlertManager
victoriametrics	1.89.1	Сбор и хранение метрик
vmalert	1.89.1	Инициирование оповещений согласно правил
grafana	9.2.8	Визуализация метрик
alertmanager	0.25.0	Обработка оповещений
snmp_notifier	1.4.0	Отправка уведомлений по SNMP
ipmi_exporter	1.6.1	Мониторинг IPMI таргетов
snmp_exporter	0.21.0	Мониторинг SNMP таргетов
blackbox_exporter	0.23.0	Мониторинг ICMP таргетов
vision_exporter	0.9	Мониторинг сервисов различных типов ПАК

Состав агентов мониторинга на узлах

Компонент	Версия	Назначение
vmagent	1.89.1	Сбор метрик с экспортеров на узлах и передача их на сервер мониторинга
node_exporter	1.5.0	Мониторинг оборудования и ОС узла
utlz_exporter	0.9	Мониторинг утилизации ресурсов узла. Отчетность об утилизации ПАКа отсылается во ФГИС У
ha_cluster_exporter	1.3.1	Мониторинг кластера МБД.П
postgres_exporter	0.11.1	Мониторинг Postgres МБД.П

5 КОНФИГУРАЦИЯ КОМПОНЕНТОВ

После конфигурации необходимо перезапустить соответствующий компонент, например:

```
systemctl restart prometheus.service
```

VictoriaMetrics

Путь	Файл	Описание	Настраиваемые параметры
/etc/vision/victoriametrics			
	victoriametrics.yml	Общая конфигурация	scrape_interval - глобальный интервал опроса экспортеров scrape_timeout - глобальный интервал ожидания данных от экспортеров Каждый экспортер описывается в виде раздела, например: - job_name: victoriametrics scrape_interval: 15s scrape_timeout: 10s metrics_path: /vision/victoriametrics/metrics static_configs: - targets: - 127.0.0.1:8428
	snmp_targets.yml ipmi_targets.yml blackbox_targets.yml	Файлы со списком таргетов соответствующих экспортеров	Например, для таргетов ipmi_exporter: - targets: - 192.168.186.2 labels: instance: depo job: 'ipmi_exporter'

Путь	Файл	Описание	Настраиваемые параметры
	victoriametrics.env	Параметры запуска	<code>promsrape.config</code> - путь до файла конфигурации <code>storageDataPath</code> - расположение БД <code>retentionPeriod</code> - срок хранения метрик в БД <code>httpListenAddr</code> - прослушиваемый порт

Grafana

Путь	Файл	Описание	Настраиваемые параметры
/etc/vision/grafana/			
	grafana.ini	Общая конфигурация	<code>admin_user</code> - имя администратора <code>admin_password</code> - пароль администратора
	grafana.env	Параметры запуска	<code>config</code> - путь до файла конфигурации

Правка дашбордов осуществляется непосредственно через web-интерфейс.

Alertmanager

Путь	Файл	Описание	Настраиваемые параметры
/etc/vision/alertmanager/			
	alertmanager.yml	Настройка правил оповещения	Структура выглядит следующим образом: <pre># глобальная конфигурация параметров # служит значениями по умолчанию для других разделов конфигурации global: <...></pre>

Путь	Файл	Описание	Настраиваемые параметры
			# файлы, из которых считываются пользовательские шаблоны для уведомлений # можно использовать подстановочные знаки (templates/*.tpl) templates: [- <filepath> ...] # корневой узел дерева маршрутизации уведомлений route: <...> # список получателей уведомлений и настройка отправки receivers: - <receiver> ... # список правил подавления уведомлений inhibit_rules: [- <inhibit_rule> ...] # список временных интервалов для использования в маршрутах time_intervals: [- <time_interval> ...] Настройки по-умолчанию: smtp_smarthost: 'SMTP_хост:порт' smtp_from: 'адрес_электронной_почты_отправителя' smtp_auth_username: 'имя_пользователя' smtp_auth_auth_password: 'пароль' Каждый блок route имеет следующую структуру:

Путь	Файл	Описание	Настраиваемые параметры
			# Значение ключа name из списка receivers. # По факту - ссылка на получателя по умолчанию. receiver: <string> # Список имен меток, по которым группируются уведомления. # Для отключения группировки указать: group_by: ['...'] # Группировка в дочерних узлах переопределяет вышестоящую. group_by: ['<labelname>, ... ']'] # Должно ли оповещение продолжать соответствовать последующим одноуровневым узлам, # если совпадение найдено в текущем. Уведомления будут отправлены всем найденным совпадениям. # По-умолчанию = false continue: <boolean> # Список сопоставителей, которым должно соответствовать оповещение, чтобы соответствовать узлам. # Элемент matcher имеет вид "service=~"mysql cassandra" matchers: - <matcher> ... # Сколько первоначально ждать перед отправкой уведомления для группы после получения # первого предупреждения.Позволяет дождаться поступления запрещающего оповещения или # собрать дополнительные оповещения для группы. group_wait: <duration> # Сколько ждать перед отправкой уведомления о новых оповещениях # для которых уже отправлено первоначальное уведомление group_interval: <duration>

Путь	Файл	Описание	Настраиваемые параметры
			# Сколько ждать перед повторной отправкой уведомления, # если оно уже было успешно отправлено. repeat_interval: <duration> # Времена (из раздела mute_time_intervals), когда маршрут (его оповещения) должен быть отключен. # Корневой узел не может иметь этого параметра!!! mute_time_intervals: - <string> ... # Времена (из раздела time_intervals), когда маршрут (его оповещения) должен быть активен. # Пустое значение = всегда активен. # Корневой узел не может иметь этого параметра!!! active_time_intervals: - <string> ... # Можно далее строить дерево маршрутизации. routes: - <route> ... Каждый receiver (получатель) описывается следующим синтаксисом: # Уникальное имя name: <string> # Конфигурация интеграции с нотификаторами (ниже приведены базовые) email_configs: [- <email_config>, ...] webhook_configs: [- <webhook_config>, ...]

Путь	Файл	Описание	Настраиваемые параметры
	alertmanager.env	Параметры запуска	config.file - путь до файла конфигурации

snmp_notifier

Путь	Файл	Описание	Настраиваемые параметры
/etc/vision/snmp_notifier/			
	description-template.tpl	Шаблон отправляемого сообщения	-
	snmp_notifier.env	Параметры запуска	snmp.destination - адрес отправки уведомления snmp.trap-description-template - путь до шаблона отправляемого сообщения

Конфигурация экспортеров

Таргеты экспортеров описываются в соответствующих файлах /etc/vision/victoriametrics/<имя экспортера>_targets.yml

Параметры запуска описываются в соответствующих файлах /etc/vision/exporters/<имя экспортера>/<имя экспортера>.env

node_exporter

Путь	Файл	Описание	Настраиваемые параметры
/etc/vision/exporters/node_exporter			

Путь	Файл	Описание	Настраиваемые параметры
	node_exporter.env	Параметры запуска	web.listen-address – прослушиваемый порт, по которому возвращаются метрики

ipmi_exporter

Путь	Файл	Описание	Настраиваемые параметры
/etc/vision/exporters/ipmi_exporter/			
	ipmi_remote.yml	Общая конфигурация	Группы настроек (модули) задаются в разделе modules виде: <pre># имя модуля module_name: # Имя пользователя и пароль для доступа user: "admin" pass: "admin" # Используемый драйвер и привелегии для доступа driver: "LAN_2_0" privilege: "admin" # Время ожидания ответа timeout: 10000 # Используемые инструменты для сбора данных # доступны: bmc, ipmi, chassis, dcmi, sel, and sm-lan-mode collectors: - bmc - ipmi - chassis - dcmi - sel</pre>

Путь	Файл	Описание	Настраиваемые параметры
			Соответствующий модуль указывается в <code>/etc/prometheus/prometheus.yml</code> - <code>job_name: devices_ipmi</code> <code>params:</code> <code>module: ['module_name']</code> При необходимости может быть создано несколько разделов с разными модулями
	<code>ipmi_exporter.env</code>	Параметры запуска	web.listen-address - прослушиваемый порт, по которому возвращаются метрики config.file - путь до файла конфигурации freeipmi.path - путь до бинарного файла инструмента FreeIPMI (при необходимости)

snmp_exporter

Путь	Файл	Описание	Настраиваемые параметры
<code>/etc/vision/exporters/snmp_exporter/</code>			
	<code>snmp.yml</code>	Общая конфигурация (описание OID)	-
	<code>snmp_exporter.env</code>	Параметры запуска	web.listen-address - прослушиваемый порт, по которому возвращаются метрики config.file - путь до файла конфигурации

blackbox_exporter

Путь	Файл	Описание	Настраиваемые параметры
/etc/vision/exporters/blackbox_exporter/			
	blackbox_exporter.yml	Общая конфигурация	Группы настроек (модули) задаются в разделе modules в виде: <pre># имя модуля module_name: # используемый инструмент (icmp) prober: icmp # время ожидания ответа timeout: 5s # настройки используемого инструмента icmp: # протокол preferred_ip_protocol: ip4 # исходный адрес source_ip_address: "127.0.0.1"</pre> Соответствующий модуль указывается в /etc/prometheus/prometheus.yml <pre>- job_name: blackbox_exporter_icmp params: module: [module_name]</pre> При необходимости может быть создано несколько разделов с разными модулями
	blackbox_exporter.env	Параметры запуска	web.listen-address - прослушиваемый порт, по которому возвращаются метрики config.file - путь до файла конфигурации

utlz_exporter

Путь	Файл	Описание	Настраиваемые параметры
/etc/vision/exporters/utlz_exporter/			
	utlz_exporter.json	Общая конфигурация	Данные в формате JSON. В файле должно быть 4 главных ключа – `ssd`, `ssd3`, `nvme`, `nlsas` – отображающихся на 4 типа накопителей. Пример: <pre>{ # полное имя блочного устройства, с которого монтируются ФС "dev_name":"/dev/nvme0n1", # смонтированные ФС "mount_points":["/","/boot/efi"], # добавка в байтах к используемому пространству "incr_us": 100000000, # добавка в байтах к доступному пространству "incr_ts": 12000000 }</pre>
	utlz_exporter.env	Параметры запуска	web.listen-address – прослушиваемый порт, по которому возвращаются метрики common.config-fname – путь до файла конфигурации common.use-dmidecode – использовать ли dmidecode вместо lsmem для получения утилизации RAM (yes no) common.disk-only – возврат только метрик дисков (yes no) common.sds-coeff – использовать коэффициент SDS, применяемый к значению утилизации дисков, (yes no)

Собираемые экспортером метрики

`cpu\_num` - количество ядер физических CPU ядер

`cpu\_usage` - общее использование CPU в процентах

`mem\_size` - общий размер RAM в MiB

`mem\_usage` - используемое RAM в MiB

`disk\_size\_nlsas` | `disk\_size\_nvme` | `disk\_size\_ssd` | `disk\_size\_ssd\_dwpd3` - размер дискового пространства каждого типа в MiB

`disk\_usage\_nlsas` | `disk\_usage\_nvme` | `disk\_usage\_ssd` | `disk\_usage\_ssd\_dwpd3` - размер занятого дискового пространства каждого типа в MiB

`utlz\_discovery` - всегда равна 1, метрика для каждого узла, которая будет присутствовать всегда, вне зависимости от того, есть или нет на данном узле тот или иной тип дисков или сведения о CPU и RAM

`utlz\_dev\_status` - статус дискового устройства, который участвует в подсчете утилизации

`utlz\_mount\_point\_status` - статус точки монтирования, которая участвует в подсчете утилизации

ha_cluster_exporter

Путь	Файл	Описание	Настраиваемые параметры
/etc/vision/exporters/ha_cluster_exporter			
	ha_cluster_exporter.env	Параметры запуска	web.listen-address - прослушиваемый порт, по которому возвращаются метрики

postgres_exporter

Путь	Файл	Описание	Настраиваемые параметры
/etc/vision/exporters/postgres_exporter			
	postgres_exporter.env	Параметры запуска	web.listen-address - прослушиваемый порт, по которому возвращаются метрики

Путь	Файл	Описание	Настраиваемые параметры
			DATA_SOURCE_NAME -строка для подключения к локальной БД Postgres вида: "postgresql://USER:PASSWORD@127.0.0.1:5432/postgres?sslmode=disable"

6 УСТРАНЕНИЕ НЕПОЛАДОВ

1) При заходе на страницу Grafana не отображаются никакие данные

- Проверить и, при необходимости, выполнить синхронизацию времени между хостом с Визионом, хостами и системным временем сервера.

2) Не отображаются данные с экспортера (на примере postgres_exporter)

- Проверить доступность и состояние экспортера
- Проверить лог работы экспортера:

```
systemctl status postgres-exporter
```

При недоступности осуществить перезапуск сервиса, либо осуществить манипуляции по ситуации. Так же можно обратиться в тех.поддержку Скала^Р.

3) Ошибка на этапе миграции базы данных на шаге 2 сценария

- Очистить служебную базу данных Визиона 'vision_db' (все данные будут утеряны, применимо только для новой установки)

```
# Подключение к postgres
psql -h localhost -U vision -d postgres

# Закрыть все соединения
SELECT pg_terminate_backend(pid) FROM pg_stat_activity WHERE datname = 'vision_db';

# Удалить таблицу
DROP DATABASE vision_db;
```

- Повторно запустить развертывание с шага 2.