

Скала^р Визион

v.0.3.2

Руководство пользователя

СОДЕРЖАНИЕ

ГЛОССАРИЙ	4
1 ОБЩИЕ СВЕДЕНИЯ.....	7
1.1 Наименование программы.....	7
1.2 Область применения	7
1.3 Лингвистическое обеспечение	7
1.4 Обеспечение защиты информации	7
2 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ.....	8
2.1 Виды деятельности, функции и классы решаемых задач.....	8
2.2 Характеристики и конфигурация технических средств	8
2.3 Операционная среда и общесистемные программные средства.....	8
3 ОПИСАНИЕ ОБЪЕКТОВ.....	9
Программно-аппаратный комплекс (ПАК)	9
МХД.О.....	9
МБД.П	9
МБД8	9
МВ.С.....	10
4 ГРАФИЧЕСКИЙ WEB-ИНТЕРФЕЙС.....	11
4.1 Компоненты мониторинга.....	11
4.1.1 Раздел приложения (Стартовая страница)	11
4.1.2 Раздел «Дашборды»	11
4.1.3 Раздел «Уведомления».....	11
4.1.4 Раздел «Параметры».....	12
4.1.5 Раздел «Статус Визиона»	18
4.1.6 Раздел «Настройки»	18
5 РЕГИСТРАЦИЯ И ВЕДЕНИЕ СПИСКА ОБЪЕКТОВ МОНИТОРИНГА	20
5.1 Регистрация объекта мониторинга	20
6 УПРАВЛЕНИЕ МОНИТОРИНГОМ	21
6.1 Работа с аналитическими панелями	21
6.2 Работа с метриками.....	21

6.2.1	Добавление метрики	21
6.2.2	Работа с контекстным меню.....	23
6.2.3	Обзор метрики.....	25
6.3	Работа с правилами оповещения.....	25
6.3.1	Добавление правила оповещения	25
6.3.2	Работа с контекстным меню списка правил оповещения	26
6.3.3	Обзор правила оповещения.....	27
6.4	Работа с получателями.....	28
6.4.1	Добавление получателя	28
6.4.2	Работа с контекстным меню списка получателей	28
6.4.3	Обзор получателя.....	29
6.5	Работа с группами рассылки.....	30
6.5.1	Добавление группы	30
6.5.2	Работа с контекстным меню списка групп рассылки	31
6.5.3	Обзор группы рассылки	32
6.6	Настройки	32
7	РЕШЕНИЕ ПРОБЛЕМ	33
7.1	Логирование	33
7.2	Рекомендованный способ обработки ошибок	33
8	ПРИЛОЖЕНИЕ 1.....	34

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 модуля 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, правил оповещений, исполняемый файлы агентов сбора данных
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, агент сбора данных	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 визуальном 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
Уведомление	Текстовое сообщение о предупреждении
Канал уведомления	Информация о получателях и методах отправки оповещений
Правило оповещения	Информация, определяющая действие по отношению к конкретному экземпляру предупреждения при его получении
Предупреждение	Индикатор состояния соответствующей метрики, определяющий факт срабатывания правила предупреждения

Правило предупреждения	Условие, применяемое к значению метрики и необходимое для срабатывания предупреждения
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 взаимосвязанного ПО, обеспечивающего сбор, обработку, хранение и визуализацию информации, полученной от агентов, а также 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	Система Управления Базами Данных . 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
VictoriaMetrics	Стороннее ПО, отвечающее за сбор метрик от различных агентов и хранение их в БДВР
PromQL	Язык запросов Prometheus

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 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.

1.4 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, VictoriaMetrics, 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 Альт Сервер версии 8.4.

3 ОПИСАНИЕ ОБЪЕКТОВ

Программно-аппаратный комплекс (ПАК)

ПАК – Совокупность аппаратных компонентов и программного обеспечения, решающих прикладные задачи.

Существует 4 основным вида ПАКа:

МХД.О

МХД.О – Машина хранения данных Скала^р МХД.О предназначена для создания горизонтально масштабируемого объектного хранилища, совместимого с Amazon S3.

В состав продукта входят:

- Базовый модуль – Предназначен для реализации базового набора сервисов, необходимых для организации и функционирования хранилища S3.
- Модули хранения – Устанавливаются в составе Базового модуля и обеспечивают физическое хранение объектов.
- Модули коммутации и хранения – Подключаются к Базовому модулю для расширения емкости S3 хранилища.
- Программное обеспечение – Развертывания, обновления и управления; Балансировки нагрузки; Распределенного дискового массива; S3 хранилища.

МБД.П

МБД.П – Машина баз данных Скала^р МБД.П предназначена для обработки хранения данных, специально предназначенный для работы СУБД PostgreSQL в высоконагруженных системах.

В состав продукта входят:

- Вычислительный блок – состоит из Вычислительные узлы и СУБД Postgres Pro Enterprise. Необходим для обработки данных.
- Блок сетевого взаимодействия – Для обеспечения доставки данных между устройствами.
- Блок мониторинга и управления – Мониторинг и визуализация работы сети и оборудования, входящего в состав комплекса (Скала^р Мониторинг).
- Блок резервного копирования – Предназначен создания копии данных на носителе, предназначенном для восстановления информации в оригинальном месте их расположения в случае их повреждения или разрушения.

МБД8

МБД8 – Машина баз данных Скала^р МБД8 предназначена для хранения и аналитической обработки данных. На сегодняшний день существует 5 типов машин:

- хранилище слабоструктурированных и неструктурированных данных, модель Скала^р МБД.Х (ранее МБД8.АДХ)
- аналитическая распределённая СУБД массивно-параллельной обработки, модель Скала^р МБД.Г (ранее МБД8.АДБ)
- сверхбыстрая система распределённых вычислений в оперативной памяти, модель Скала^р МБД.Т (ранее МБД8.АДГ)
- колоночная аналитическая СУБД, модель Скала^р МБД.КХ (ранее МБД8.АДКМ)
- система потоковой обработки данных, модель Скала^р МБД.С (ранее МБД8.АДС)

В состав продукта входят:

- ПО развертывания, обновления, управления и мониторинга
- ПО для работы с большими данными
- Сетевая инфраструктура

МВ.С

МВ.С – Машина баз данных Скала^р МВ.С предназначена для создания горизонтально масштабируемой и отказоустойчивой среды виртуализации и инфраструктуры виртуальных рабочих мест пользователей.

В состав продукта входят:

- Серверы
- Сеть
- ПО виртуализации
- ПО управления
- Средства ИБ

4 ГРАФИЧЕСКИЙ WEB-ИНТЕРФЕЙС

4.1 Компоненты мониторинга.

Интерфейс Визион состоит из следующих компонентов:

4.1.1 Раздел приложения (Стартовая страница)

Интерфейс стартовой страницы отображает состояние компонентов ПАК и уведомления, в зависимости от вида ПАК разделы могут меняться.

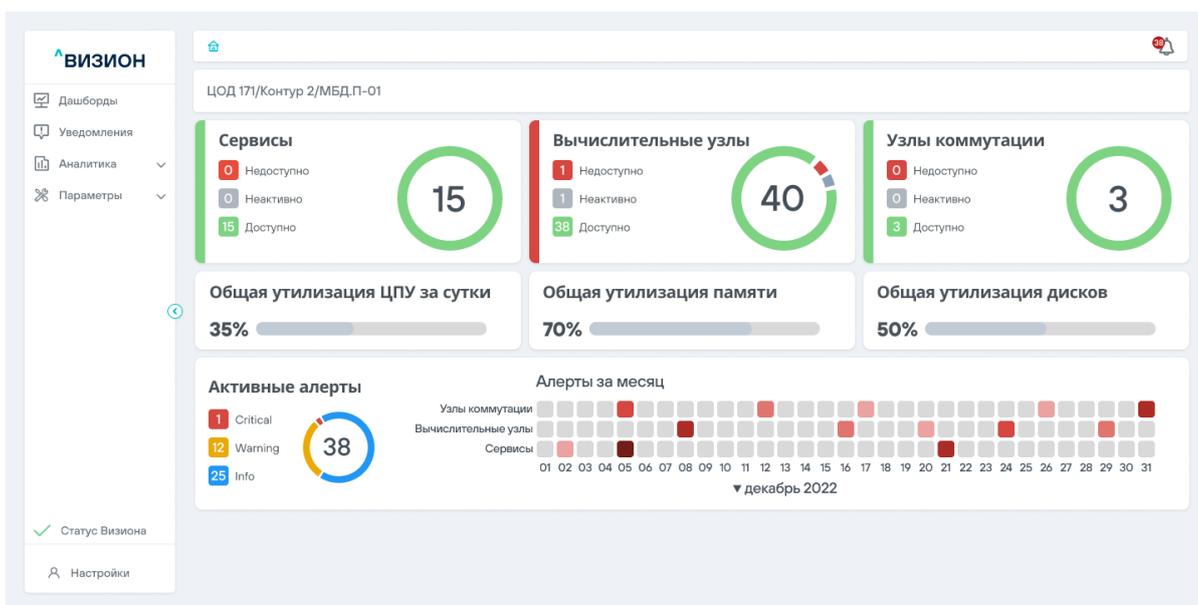


Рисунок 1 – Пример интерфейса стартовой страницы

4.1.2 Раздел «Дашборды»

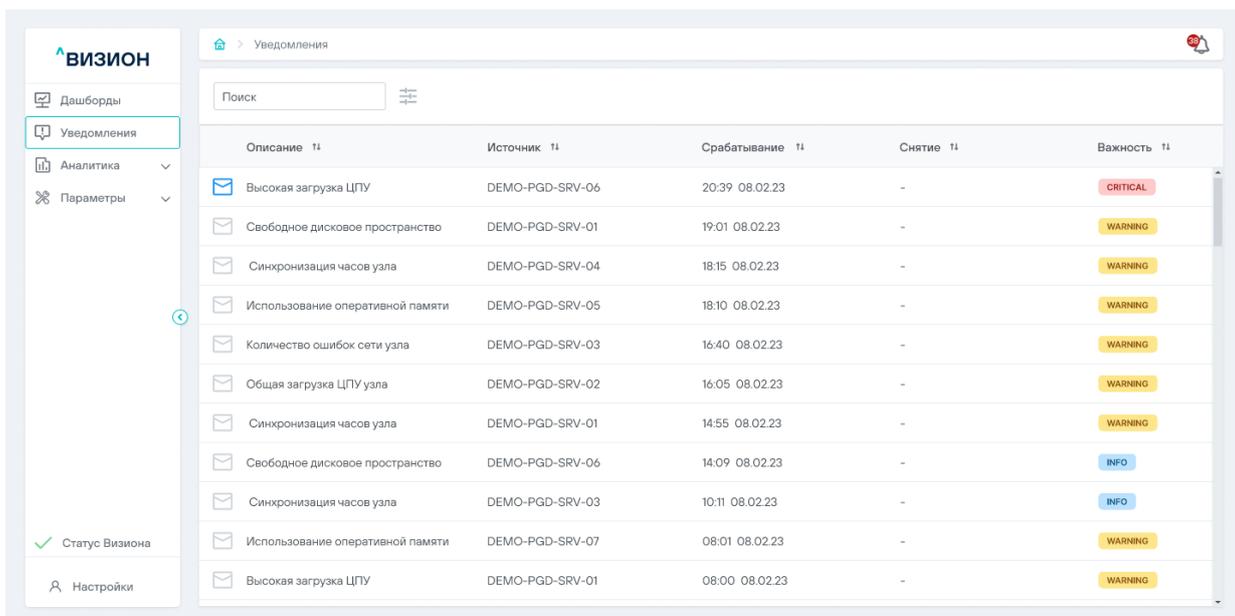
В разделе Дашбордов происходит переход в стороннее ПО «Grafana».

Для работы с этим разделом используйте документацию с официального сайта <https://grafana.com/>

4.1.3 Раздел «Уведомления»

Раздел «Уведомления» содержит информацию о сработавших уведомлениях

Содержит список уведомлений



The screenshot shows the 'Визион' (Vision) notification interface. On the left is a sidebar with navigation options: 'Дашборды', 'Уведомления' (selected), 'Аналитика', 'Параметры', 'Статус Визиона', and 'Настройки'. The main area displays a list of notifications with columns: 'Описание', 'Источник', 'Срабатывание', 'Снятие', and 'Важность'. The notifications include alerts for CPU load, free disk space, node clock synchronization, and memory usage, with severity levels ranging from 'CRITICAL' to 'INFO'.

Описание	Источник	Срабатывание	Снятие	Важность
Высокая загрузка ЦПУ	DEMO-PGD-SRV-06	20:39 08.02.23	-	CRITICAL
Свободное дисковое пространство	DEMO-PGD-SRV-01	19:01 08.02.23	-	WARNING
Синхронизация часов узла	DEMO-PGD-SRV-04	18:15 08.02.23	-	WARNING
Использование оперативной памяти	DEMO-PGD-SRV-05	18:10 08.02.23	-	WARNING
Количество ошибок сети узла	DEMO-PGD-SRV-03	16:40 08.02.23	-	WARNING
Общая загрузка ЦПУ узла	DEMO-PGD-SRV-02	16:05 08.02.23	-	WARNING
Синхронизация часов узла	DEMO-PGD-SRV-01	14:55 08.02.23	-	WARNING
Свободное дисковое пространство	DEMO-PGD-SRV-06	14:09 08.02.23	-	INFO
Синхронизация часов узла	DEMO-PGD-SRV-03	10:11 08.02.23	-	INFO
Использование оперативной памяти	DEMO-PGD-SRV-07	08:01 08.02.23	-	WARNING
Высокая загрузка ЦПУ	DEMO-PGD-SRV-01	08:00 08.02.23	-	WARNING

Рисунок 2 – Список уведомлений

Список содержит следующую информацию о уведомлении:

- Название;
- Источник;
- Срабатывание;
- Снятие;
- Важность;

Список можно отфильтровать и сортировать.

4.1.4 Раздел «Параметры»

4.1.4.1 Подраздел «Метрики»

Раздел «Метрики» содержит информацию о добавленных пользователем и системных метриках (дефолтных) и инструменты для работы с ними.

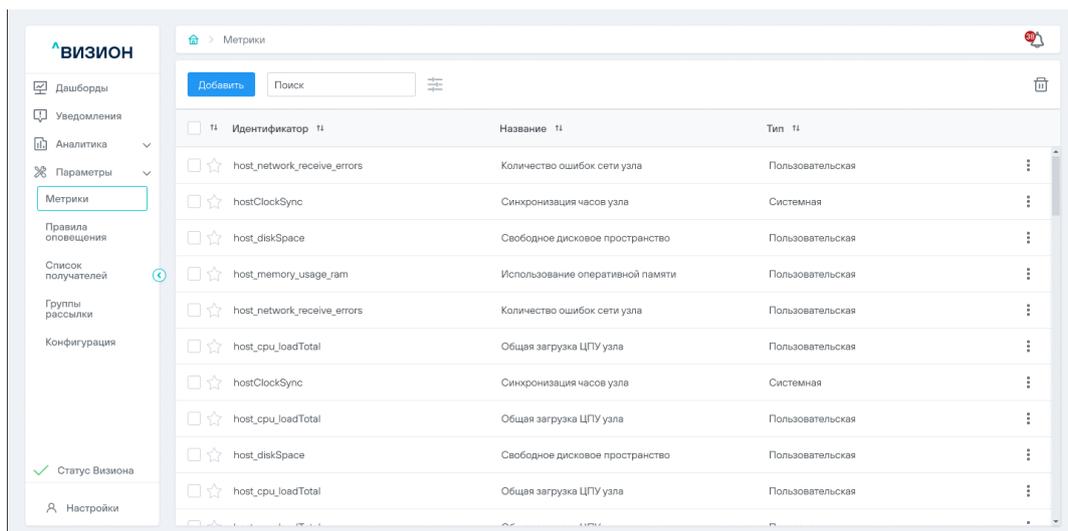


Рисунок 3 – Список метрик

Список содержит следующую информацию о метрике:

- Идентификатор
- Название
- Тип

Также с помощью кнопки «Добавить» открывается форма «Добавление метрики» с помощью которой можно добавить новую метрику.

Идентификатор:

Название:

Описание:

Выражение:

Рисунок 4 – Форма «Добавление метрики»

Форма «Добавление метрики» содержит следующие поля:

- Идентификатор;
- Название;
- Описание;
- Выражение.

А также кнопки проверки выражения и перехода в браузер метрики.

4.1.4.2 Подраздел «Правила оповещения»

В разделе «Правила оповещения» содержит информацию о добавленных пользователем и системных (дефолтных) правилах оповещения и инструменты для работы с ними.

Содержит список оповещений

<input type="checkbox"/>	Идентификатор	Название	Правило	Последнее срабатывание	Важность	Активно
<input type="checkbox"/>	HighHostCpuLoadTotal	Высокая загрузка ЦПУ	host_cpuLoadTotal >70	13:39 08.02.23	5	WARNING
<input type="checkbox"/>	Host_diskSpace	Свободное дисковое...	host_diskUsage >80	13:39 08.02.23	13	WARNING
<input type="checkbox"/>	Host_memory_usage	Использование операт...	host_memoryUsage >80	13:39 08.02.23	2	WARNING
<input type="checkbox"/>	Host_memory_usage	Использование операт...	host_memoryUsage >80	13:39 08.02.23	2	WARNING
<input type="checkbox"/>	host_cpu_loadTotal	Общая загрузка ЦПУ...	host_cpuLoadTotal >50	13:39 08.02.23	7	WARNING
<input type="checkbox"/>	hostClockSync	Синхронизация часов...	host_clockSync=0	13:39 08.02.23	5	INFO
<input type="checkbox"/>	HighHostCpuLoadTotal	Высокая загрузка ЦПУ	host_cpuLoadTotal >90	13:39 08.02.23	5	CRITICAL
<input type="checkbox"/>	HighHostCpuLoadTotal	Высокая загрузка ЦПУ	host_cpuLoadTotal >90	13:39 08.02.23	5	WARNING
<input type="checkbox"/>	HighHostCpuLoadTotal	Высокая загрузка ЦПУ	host_cpuLoadTotal >90	13:39 08.02.23	5	WARNING
<input type="checkbox"/>	HighHostCpuLoadTotal	Высокая загрузка ЦПУ	host_cpuLoadTotal >90	13:39 08.02.23	5	WARNING
<input type="checkbox"/>	HighHostCpuLoadTotal	Высокая загрузка ЦПУ	host_cpuLoadTotal >90	13:39 08.02.23	5	WARNING

Рисунок 5 – Список правил оповещения

Список содержит следующую информацию о правилах оповещения:

- Идентификатор;
- Название;
- Правило;
- Последнее срабатывание;
- Важность;
- Активность;

С помощью кнопки «Добавить» открывается форма «Добавление правила оповещения» с помощью которой можно добавить новое правило оповещения.

Добавление правила оповещения

Идентификатор ⓘ

Название ⓘ

Метрика ⓘ Оператор ⓘ Порог ⓘ

Период активности ⓘ

Важность ⓘ
 info warning critical

Шаблон сообщения
Text

0 / 100

Добавить еще

Рисунок 6 – Форма "Добавление правила оповещения"

Форма «Добавление правила оповещения» содержит следующие поля:

- Идентификатор;
- Название;
- Метрика;
- Оператор;
- Порог;
- Период активности;
- Важность;
- Шаблон сообщения;

4.1.4.3 Подраздел «Правила оповещения»

Раздел «Список получателей» содержит информацию о получателях и инструменты для работы с ними.

Содержит список получателей

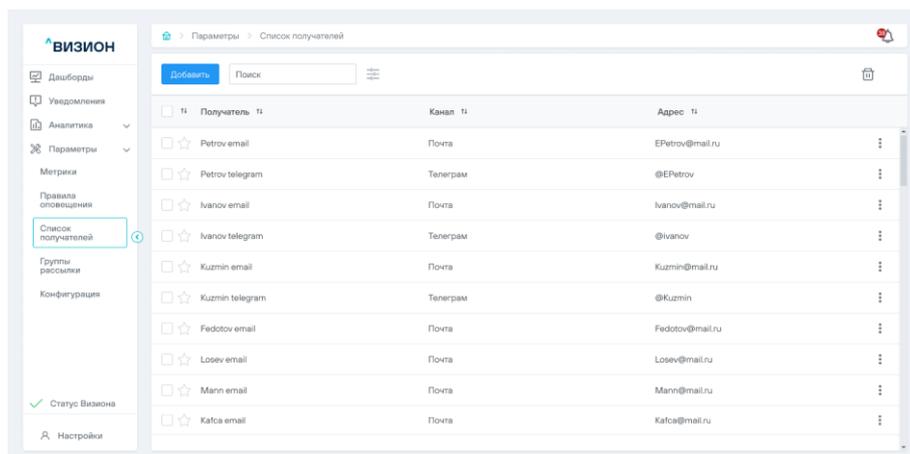


Рисунок 7 – Список получателей

Список содержит следующую информацию о получателях:

- Получатель
- Канал
- Адрес

С помощью кнопки «Добавить» открывается форма «Добавление получателя» с помощью которой можно добавить нового получателя.

Добавление получателя

Получатель

Канал

Адрес

Группа

+ Добавить в группу

Добавить еще

Рисунок 8 – Форма "Добавление получателя"

Форма «Добавление правила оповещения» содержит следующие поля:

- Получатель
- Канал
- Адрес

- Группа
- Возможность добавить еще группу

4.1.4.4 Подраздел «Группы рассылки»

Раздел «Группы рассылки» содержит информацию о группах рассылки и инструменты для работы с ними.

Содержит список групп рассылки

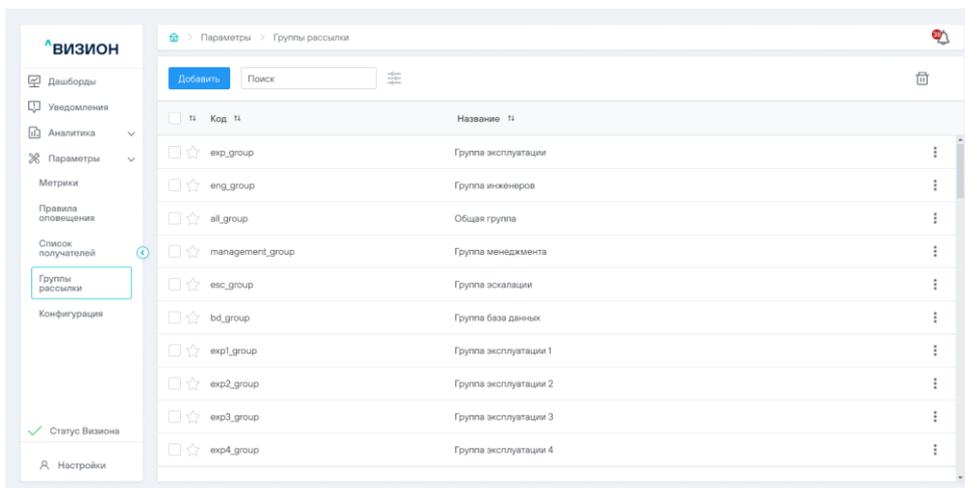


Рисунок 9 – Список групп рассылки

Список содержит следующую информацию о группах рассылки:

- Идентификатор
- Название

С помощью кнопки «Добавить» открывается форма «Добавление группы» с помощью которой можно добавить новую группу.

Рисунок 10 – Форма "Добавление группы"

Форма «Добавление правила оповещения» содержит следующие поля:

- Идентификатор;
- Название;
- Описание;
- Метка;
- Оператор;
- Значение;
- Получатель.

А также возможность «Добавить получателя» и «Добавить правило».

4.1.5 Раздел «Статус Визиона»

Статус Визиона отображает состояние Визиона, при нажатии открывает форму «Состояние компонентов».

Форма «Состояние компонентов» отображает состояние компонентов Визиона.

Состояние компонентов	×
Визион БД	✓
TaskDaemon	✓
VictoriaMetrics	✓
vmaalert	✓
Alertmanager	✓
Grafana	✓
Nginx	✓
Локальные экспортеры	✓

Рисунок 11 – Форма «Состояние компонентов»

4.1.6 Раздел «Настройки»

В разделе «Настройки» находятся настройки интерфейса, позволяющие пользователю настроить интерфейс под себя.

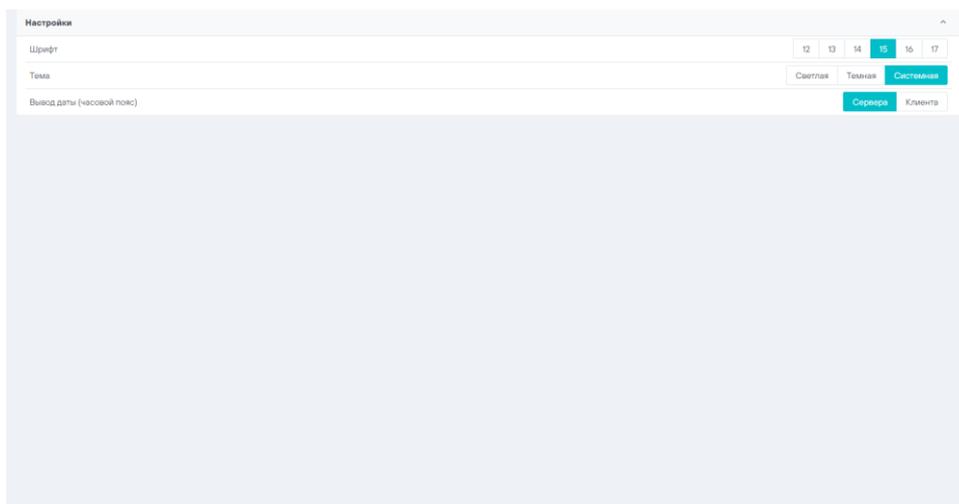


Рисунок 12 – Раздел «Настройки»

5 РЕГИСТРАЦИЯ И ВЕДЕНИЕ СПИСКА ОБЪЕКТОВ МОНИТОРИНГА

5.1 Регистрация объекта мониторинга

Для регистрации объекта мониторинга смотреть руководство администратора (п.2)

6 УПРАВЛЕНИЕ МОНИТОРИНГОМ

6.1 Работа с аналитическими панелями

Для работы с данным разделом воспользуйтесь документацией с сайта <https://grafana.com/>

6.2 Работа с метриками

6.2.1 Добавление метрики

Для добавления новой метрики необходимо открыть раздел «Параметры» и открыть подраздел «Метрики».

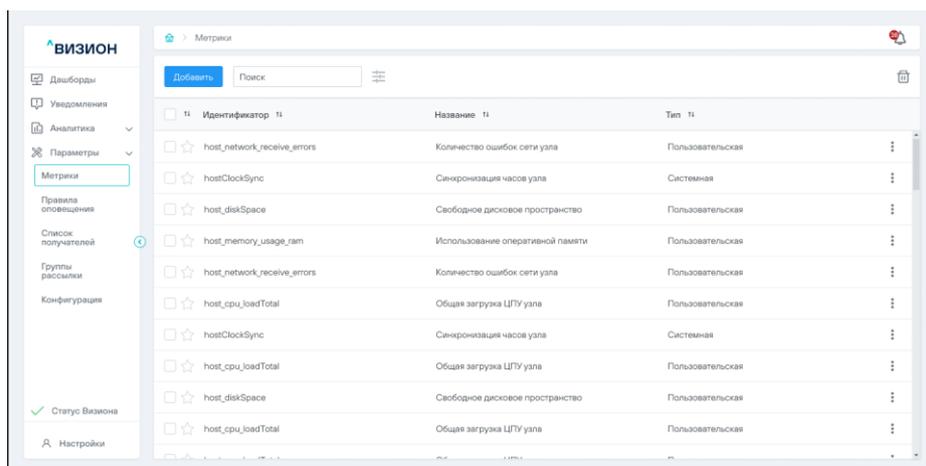


Рисунок 13 – Подраздел «Метрики»

После этого нажать кнопку «Добавить» откроется форма «Добавление метрики»

Идентификатор

Название

Описание

Выражение

Добавить в избранное Отмена Сохранить

Рисунок 14 – Форма «Добавление метрики»

Далее необходимо заполнить все поля и нажать кнопку  для проверки выражения. С помощью кнопки  можно открыть браузер метрики в котором можно посмотреть график временных рядов.

Если выражение заполнено неверно появится сообщение об ошибке

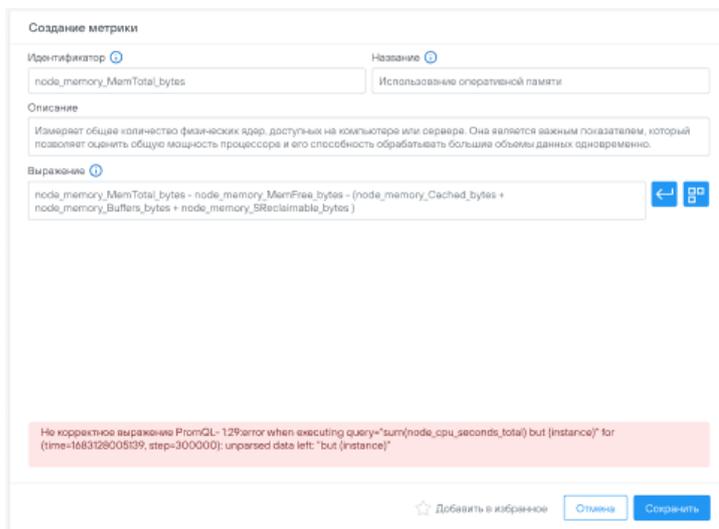


Рисунок 15 – Пример сообщения с ошибкой

Если выражение верно появится строка с временем запроса и количеством временных рядов. А также будет активна кнопка «Сохранить».

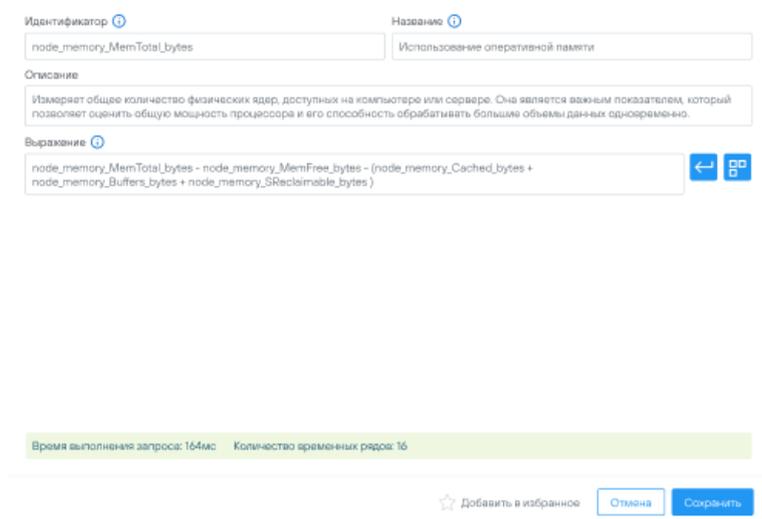


Рисунок 16 – Пример верного выражения

Также здесь можно добавить метрику в список избранного нажав на звёздочку. После нажатия кнопки «Сохранить» открывается окно с предложением дальнейших действий.

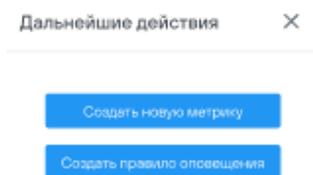


Рисунок 17 – Окно «Дальнейшие действия»

6.2.2 Работа с контекстным меню

Для вызова контекстного меню нужно нажать на кнопку  после этого откроется контекстное меню.

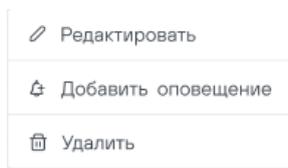


Рисунок 18 – Контекстное меню списка метрик

6.2.2.2 Удаление метрики

Для удаления метрики нужно нажать на пункт «Удалить» далее в открывшейся форме подтвердить удаление.

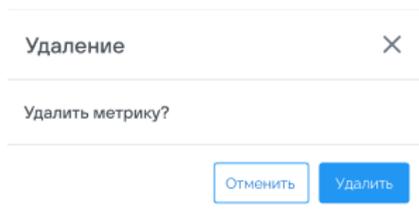


Рисунок 19 – Форма «Удаление»

6.2.2.3 Добавление оповещения

После нажатия кнопки «Добавить оповещение» открывается форма «Добавление правила оповещения».

Добавление правила оповещения

Идентификатор

Название

Метрика Оператор Порог

Период активности

Важность

info warning critical

Шаблон сообщения

Text

0 / 100

Добавить еще

Рисунок 20 – Форма «Добавление правила оповещения»

6.2.2.4 Редактирование метрики

При нажатии на пункт «Редактировать» открывается форма «Редактирование метрики»

Редактирование метрики

Идентификатор Название

Описание

Измеряет общее количество физических ядер, доступных на компьютере или сервере. Она является важным показателем, который позволяет оценить общую мощность процессора и его способность обрабатывать большие объемы данных одновременно.

Выражение

← 🗑

Время выполнения запроса: 164мс Количество временных рядов: 16

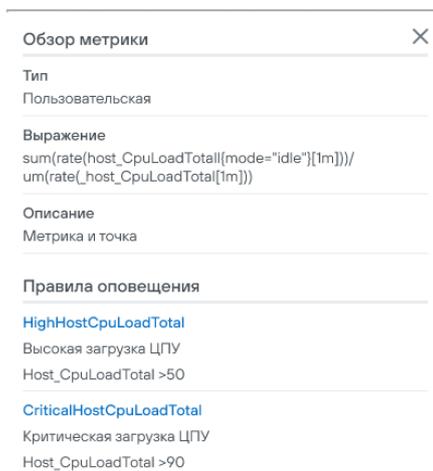
☆ Добавить в избранное

Рисунок 21 – Форма «Редактирование метрики»

Вносим необходимые изменения и нажимаем кнопку «Сохранить».

6.2.3 Обзор метрики

При нажатии на метрику из списка открывается форма «Обзор метрики»



Обзор метрики	✕
Тип	Пользовательская
Выражение	sum(rate(host_CpuLoadTotal{mode='idle'}[1m]))/um(rate(_host_CpuLoadTotal[1m]))
Описание	Метрика и точка
Правила оповещения	
HighHostCpuLoadTotal	Высокая нагрузка ЦПУ Host_CpuLoadTotal >50
CriticalHostCpuLoadTotal	Критическая нагрузка ЦПУ Host_CpuLoadTotal >90

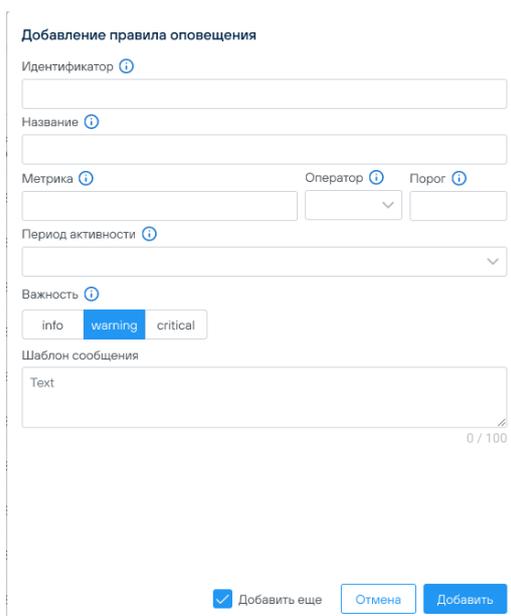
Рисунок 22 – Форма «Обзор метрики»

В обзоре метрики можно узнать информацию о метрике и правилах оповещения созданных на основе этой метрике. А также перейти к выбранному оповещению.

6.3 Работа с правилами оповещения

6.3.1 Добавление правила оповещения

Для добавления нового правила оповещения можно использовать несколько вариантов правило оповещения можно добавить сразу при создании метрики (п.6.2.1) или добавить на существующую метрику с помощью контекстного меню (п.6.2.2.3). Также можно зайти в подраздел «Правила оповещения» и нажать кнопку «Добавить» при любом варианте открывается форма «Добавление правила оповещения»



Добавление правила оповещения

Идентификатор ⓘ

Название ⓘ

Метрика ⓘ Оператор ⓘ Порог ⓘ

Период активности ⓘ

Важность ⓘ

info warning critical

Шаблон сообщения

Text

0 / 100

Добавить еще Отмена Добавить

Рисунок 23 – Форма «Добавление правила оповещения»

Необходимо заполнить все поля. Если необходимо добавить несколько правил, то необходимо чекбоксом отметить «Добавить еще» тогда при сохранении правила форма откроется повторно.

6.3.2 Работа с контекстным меню списка правил оповещения

Для вызова контекстного меню нужно нажать на кнопку  после этого откроется контекстное меню.

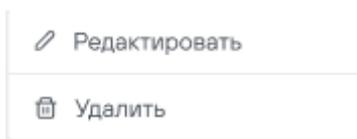


Рисунок 24 – Контекстное меню списка правил оповещения

6.3.2.2 Удаление правила оповещения

Для удаления правил оповещения нужно нажать на пункт «Удалить» далее в открывшейся форме подтвердить удаление.

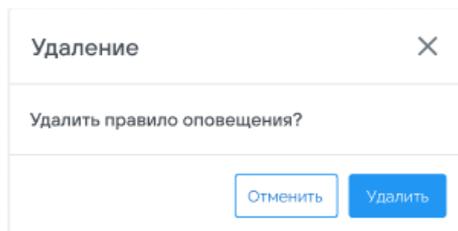


Рисунок 25 – Форма «Удаление»

6.3.2.3 Редактирование правила оповещения

При нажатии на пункт «Редактировать» открывается форма «Редактирование правила оповещения»

The screenshot shows a form titled "Редактирование правила оповещения" (Edit notification rule). It contains the following fields and options:

- Идентификатор** (Identifier): HighHostCpuLoadTotal
- Название** (Name): Высокая загрузка ЦПУ (High CPU load)
- Метрика** (Metric): host_cpuLoadTotal
- Оператор** (Operator): A dropdown menu with a downward arrow.
- Порог** (Threshold): 70
- Период активности** (Activity period): A dropdown menu.
- Важность** (Severity): Three radio buttons labeled "info", "warning" (which is selected), and "critical".
- Шаблон сообщения** (Message template): A text area containing the word "Text" and a character count "0 / 100".
- Buttons: "Отмена" (Cancel) and "Сохранить" (Save).

Рисунок 26 – Форма «Редактирование правила оповещения»

Вносим необходимые изменения и нажимаем кнопку «Сохранить».

6.3.3 Обзор правила оповещения

При нажатии на правило оповещения из списка открывается форма «Обзор правила оповещения»

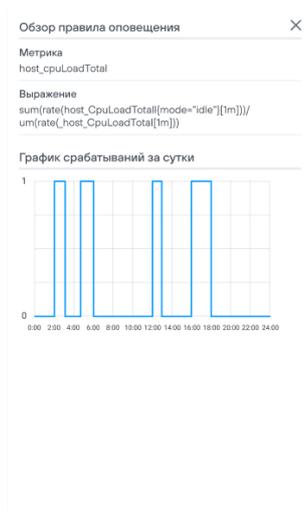


Рисунок 27 – Форма «Обзор правила оповещения»

В обзоре правила оповещения можно узнать информацию о метрике и график срабатываний за сутки. А также перейти к исходной метрике.

6.4 Работа с получателями

6.4.1 Добавление получателя

Для добавления нового получателя нужно зайти в подраздел «Получатели» и нажать кнопку «Добавить» при любом варианте открывается форма «Добавление получателя»

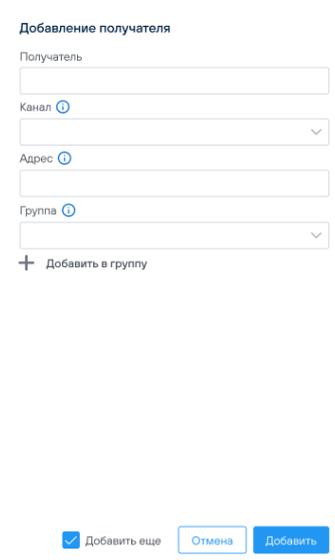


Рисунок 28 – Форма «Добавление получателя»

Необходимо заполнить все поля. Если необходимо добавить нескольких получателей, то необходимо чекбоксом отметить «Добавить еще» тогда при сохранении получателя форма откроется повторно.

6.4.2 Работа с контекстным меню списка получателей

Для вызова контекстного меню нужно нажать на кнопку  после этого откроется контекстное меню.

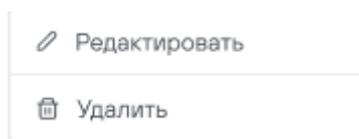


Рисунок 29 – Контекстное меню списка получателей

6.4.2.2 Удаление получателя

Для удаления получателя нужно нажать на пункт «Удалить» далее в открывшейся форме подтвердить удаление.

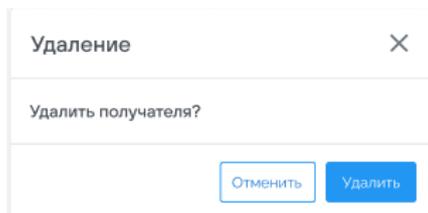


Рисунок 30 – Форма «Удаление»

6.4.2.3 Редактирование получателя

При нажатии на пункт «Редактировать» открывается форма «Редактирование получателя»

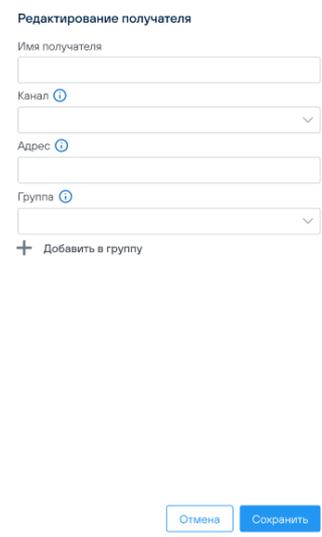


Рисунок 31 – Форма «Редактирование получателя»

Вносим необходимые изменения и нажимаем кнопку «Сохранить».

6.4.3 Обзор получателя

При нажатии на получателя из списка открывается форма «Обзор получателя»

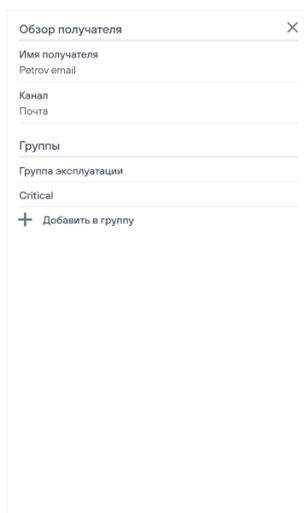


Рисунок 32 – Форма «Обзор получателя»

В обзоре получателя можно узнать информацию о получателе и группах рассылки в которые входит данный получатель. А также перейти к выбранной группе.

6.5 Работа с группами рассылки

6.5.1 Добавление группы

Для добавления новой группы нужно зайти в подраздел «Группы рассылки» и нажать кнопку «Добавить» при любом варианте открывается форма «Добавление группы»

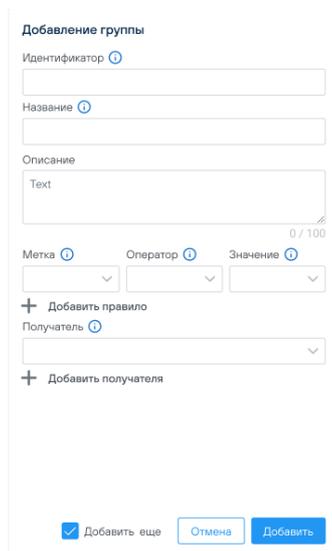


Рисунок 33 – Форма «Добавление группы»

Необходимо заполнить все поля. Если необходимо добавить несколько групп, то необходимо чекбоксом отметить «Добавить еще» тогда при сохранении получателя форма откроется повторно.

6.5.2 Работа с контекстным меню списка групп рассылки

Для вызова контекстного меню нужно нажать на кнопку  после этого откроется контекстное меню.

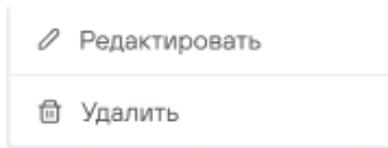


Рисунок 34 – Контекстное меню групп рассылки

6.5.2.2 Удаление группы рассылки

Для удаления группы рассылки нужно нажать на пункт «Удалить» далее в открывшейся форме подтвердить удаление.

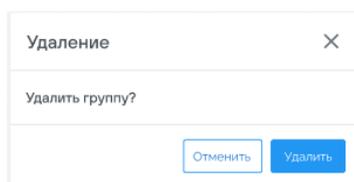


Рисунок 35 – Форма «Удаление»

6.5.2.3 Редактирование группы

При нажатии на пункт «Редактировать» открывается форма «Редактирование группы»

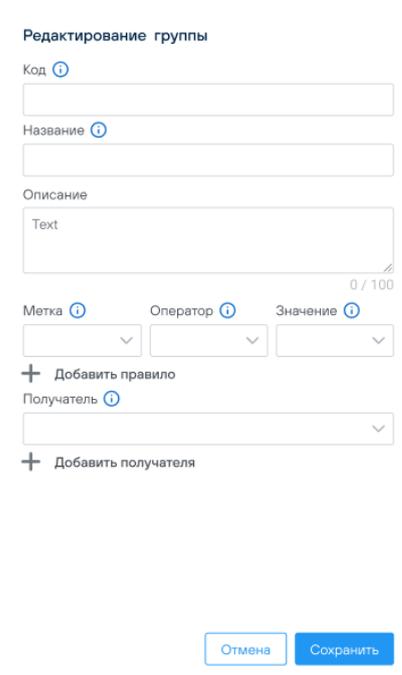
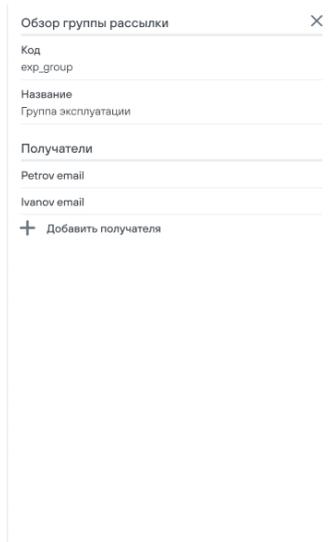


Рисунок 36 – Форма «Редактирование группы»

Вносим необходимые изменения и нажимаем кнопку «Сохранить».

6.5.3 Обзор группы рассылки

При нажатии на получателя из списка открывается форма «Обзор группы рассылки»



Обзор группы рассылки

Код
exp_group

Название
Группа эксплуатации

Получатели
Petrov email
Ivanov email

+ Добавить получателя

Рисунок 37 – Форма «Обзор группы рассылки»

В обзоре группы рассылки можно узнать информацию о группе рассылки и получателях в которые входят данную группу. А также перейти к выбранному получателю.

6.6 Настройки

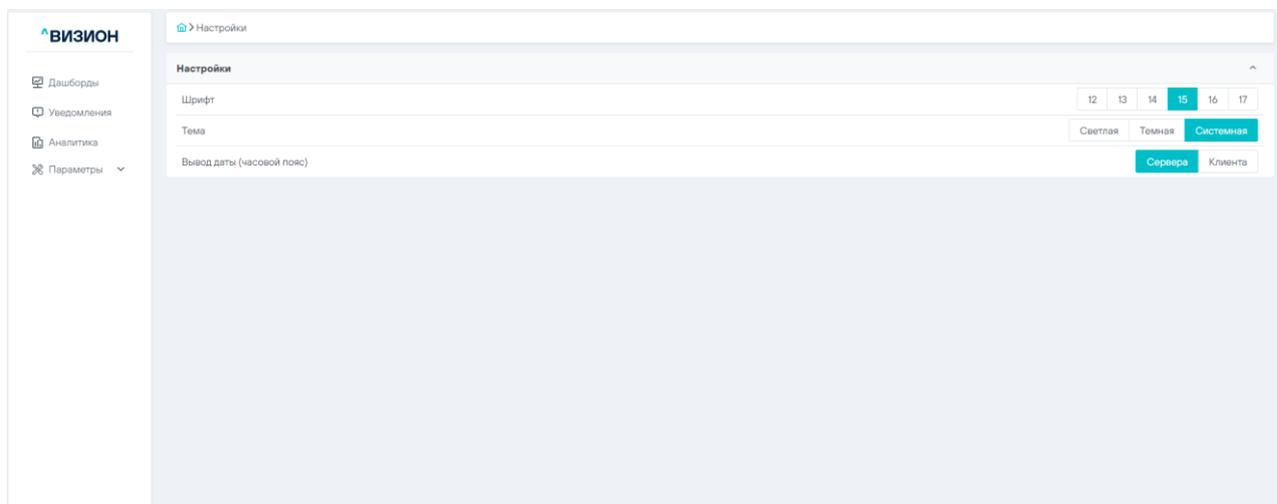


Рисунок 38 – Раздел «Настройки»

В разделе настройки можно изменить:

- Шрифт
- Тему
- Часовой пояс

7 РЕШЕНИЕ ПРОБЛЕМ

7.1 Логирование

Компоненты «Скала^р Визион» выводят отладочную информацию, содержащую сведения о своей работе, в системный журнал в бинарные файлы компонента journal и могут быть просмотрены с помощью команды:

```
journalctl -fu <название_компонента>
```

Узнать объём имеющихся на текущий момент логов можно с помощью команды:

```
journalctl --disk-usage
```

7.2 Рекомендованный способ обработки ошибок

При возникновении ошибки, необходимо проанализировать причину.

В случае аварийной остановки компонентов, они перезапускаются автоматически. Если компонент не перезапускается автоматически необходимо просмотреть содержимое лога компонента (раздел логирование). При невозможности разрешить проблему самостоятельно необходимо обратиться к разработчику.

8 ПРИЛОЖЕНИЕ 1.

Описание графических панелей и метрик, реализованных в Визион v 0.1

Название панели	Формула	Пояснение
Информация об узле (None)		
Ядра процессора (CPU)	<code>count(count(node_cpu_seconds_total{instance="\$node",job="\$job"}) by (cpu))</code>	
Оперативная память сервера (RAM)	<code>sum(mem_size{instance="\$node"})/1024</code>	
Корневая файловая система	<code>node_filesystem_size_bytes{instance="\$node",job="\$job",mountpoint="/",fstype!="rootfs"}</code>	
Время работы	<code>node_time_seconds{instance="\$node",job="\$job"} - node_boot_time_seconds{instance="\$node",job="\$job"}</code>	
Общий размер дисков SSD	<code>sum by (instance) (disk_size_ssd{module_id!="",instance=~"\$node"})</code>	
Общий размер дисков SSD DWPD3	<code>sum by (instance) (disk_size_ssd_dwpd3{module_id!="",instance=~"\$node"})</code>	
Общий размер дисков NVME	<code>sum by (instance) (disk_size_nvme{module_id!="",instance=~"\$node"})</code>	
Общий размер дисков NL SAS	<code>sum by (instance) (disk_size_nlsas{module_id!="",instance=~"\$node"})</code>	

Обзор CPU / Память / Диски		
Загрузка CPU	$\frac{(((\text{count}(\text{count}(\text{node_cpu_seconds_total}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\}) \text{ by } (\text{cpu}))) - \text{avg}(\text{sum} \text{ by } (\text{mode})(\text{irate}(\text{node_cpu_seconds_total}\{\text{mode}=\text{'idle'},\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\}[\text{\$_rate_interval}]))) * 100) / \text{count}(\text{count}(\text{node_cpu_seconds_total}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\}) \text{ by } (\text{cpu}))$	
Средняя загрузка CPU (5 мин)	$\frac{\text{avg}(\text{node_load5}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\})}{\text{count}(\text{count}(\text{node_cpu_seconds_total}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\}) \text{ by } (\text{cpu}))} * 100$	
Средняя загрузка CPU (15 мин)	$\frac{\text{avg}(\text{node_load15}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\})}{\text{count}(\text{count}(\text{node_cpu_seconds_total}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\}) \text{ by } (\text{cpu}))} * 100$	
Использование памяти	$\frac{((\text{node_memory_MemTotal_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\}) - \text{node_memory_MemFree_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\})}{(\text{node_memory_MemTotal_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\})} * 100$	
	$100 - \frac{(\text{node_memory_MemAvailable_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\}) * 100}{\text{node_memory_MemTotal_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\}}$	

Использование SWAP	$\frac{((\text{node_memory_SwapTotal_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\} - \text{node_memory_SwapFree_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\}))}{(\text{node_memory_SwapTotal_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\})} * 100$	
Исп. корневой ФС	$100 - ((\text{node_filesystem_avail_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"},\text{mountpoint}=\text{"/"},\text{fstype}!\text{="rootfs"}\} * 100) / \text{node_filesystem_size_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"},\text{mountpoint}=\text{"/"},\text{fstype}!\text{="rootfs"}\})$	
Базовая информация о CPU / Памяти / Сети / Дисках		
Использование CPU	$\text{avg without (mode,cpu) (1 - irate(node_cpu_seconds_total}\{\text{mode}=\text{"idle"},\text{instance}=\text{"\$node"}\}\{\text{\$_rate_interval}\}) * 100$	Cpu Usage
Memory Basic	$\text{node_memory_MemTotal_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\}$	RAM Total
	$\text{node_memory_MemTotal_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\} - \text{node_memory_MemFree_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\} - (\text{node_memory_Cached_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\} + \text{node_memory_Buffers_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\} + \text{node_memory_SReclaimable_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\})$	RAM Used
	$\text{node_memory_Cached_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\} + \text{node_memory_Buffers_bytes}\{\text{instance}=\text{"\$node"},\text{job}=\text{"\$job"}\} +$	RAM Cache + Buffer

	node_memory_SReclaimable_bytes{instance="\$node",job="\$job"}	
	node_memory_MemFree_bytes{instance="\$node",job="\$job"}	RAM Free
	(node_memory_SwapTotal_bytes{instance="\$node",job="\$job"} - node_memory_SwapFree_bytes{instance="\$node",job="\$job"})	SWAP Used
Network Traffic Basic	rate(node_network_receive_bytes_total{instance="\$node",job="\$job"}[\$_rate_interval])*8	Входящий трафик интерфейсов, по типу интерфейса
	rate(node_network_transmit_bytes_total{instance="\$node",job="\$job"}[\$_rate_interval])*8	Исходящий трафик интерфейсов, по типу интерфейса
Disk Space Used Basic	100 - ((node_filesystem_avail_bytes{instance="\$node",job="\$job",device!~'rootfs'} * 100) / node_filesystem_size_bytes{instance="\$node",job="\$job",device!~'rootfs'})	/ /var/lib/systemd/coredump /run /run/user/0
CPU / Memory / Disk / Net		
CPU Usage by Mode	avg by(mode) (rate(node_cpu_seconds_total{instance="\$node"}[1m]) *100)	Информация по использованным процентах Процессора в каждом режиме iowait irq nice softirq steal system

		user
Memory Stack	node_memory_MemTotal_bytes{instance="\$node",job="\$job"} - node_memory_MemFree_bytes{instance="\$node",job="\$job"} - node_memory_Buffers_bytes{instance="\$node",job="\$job"} - node_memory_Cached_bytes{instance="\$node",job="\$job"} - node_memory_Slab_bytes{instance="\$node",job="\$job"} - node_memory_PageTables_bytes{instance="\$node",job="\$job"} - node_memory_SwapCached_bytes{instance="\$node",job="\$job"}	min max avg current Apps - Memory used by user-space applications PageTables - Memory used to map between virtual and physical memory addresses SwapCache - Memory that keeps track of pages that have been fetched from swap but not yet been modified Slab - Memory used by the kernel to cache data structures for its own use (caches like inode, dentry, etc) (file content) cache Buffers - Block device (e.g. harddisk) cache Unused - Free memory unassigned Swap - Swap space used Hardware Corrupted - Amount of RAM that the kernel identified as corrupted / not working
	node_memory_PageTables_bytes{instance="\$node",job="\$job"}	
	node_memory_SwapCached_bytes{instance="\$node",job="\$job"}	
	node_memory_Slab_bytes{instance="\$node",job="\$job"}	
	node_memory_Cached_bytes{instance="\$node",job="\$job"}	
	node_memory_Buffers_bytes{instance="\$node",job="\$job"}	
	node_memory_MemFree_bytes{instance="\$node",job="\$job"}	
	(node_memory_SwapTotal_bytes{instance="\$node",job="\$job"} - node_memory_SwapFree_bytes{instance="\$node",job="\$job"})	

	node_memory_HardwareCorrupted_bytes{instance="\$node",job="\$job"}	
Disk Space Used	node_filesystem_size_bytes{instance="\$node",job="\$job",device!~'rootfs'} - node_filesystem_avail_bytes{instance="\$node",job="\$job",device!~'rootfs'}	<pre> mi ma av cur- n x g rent /run/user/0 /run / /var/lib/systemd/coredump </pre>
Disk IOps	rate(node_disk_reads_completed_total{instance="\$node",job="\$job",device=~"\$disk-devices"}[\$_rate_interval])	Успешно прочитанные и записанные байты в разрезе каждого диска (min,max,avg,current)
	rate(node_disk_writes_completed_total{instance="\$node",job="\$job",device=~"\$disk-devices"}[\$_rate_interval])	
I/O Usage Read / Write	rate(node_disk_read_bytes_total{instance="\$node",job="\$job",device=~"\$disk-devices"}[\$_rate_interval])	

	<code>rate(node_disk_written_bytes_total{instance="\$node",job="\$job",device=~"\$disk-devices"}[\$_rate_interval])</code>	
Network Traffic	<code>rate(node_network_receive_bytes_total{instance="\$node",job="\$job"}[\$_rate_interval])*8</code>	Входящий трафик интерфейсов, по типу интерфейса
	<code>rate(node_network_transmit_bytes_total{instance="\$node",job="\$job"}[\$_rate_interval])*8</code>	Исходящий трафик интерфейсов, по типу интерфейса
I/O Utilization (min,max,avg,current)	<code>rate(node_disk_io_time_seconds_total{instance="\$node",job="\$job",device=~"\$diskdevices"}[\$_rate_interval])</code>	Информация о утилизации в разрезе каждого диска
Memory Meminfo		
Memory Active / Inactive (min,max,avg,current)	<code>node_memory_Inactive_bytes{instance="\$node",job="\$job"}</code>	Inactive - Memory which has been less recently used. It is more eligible to be reclaimed for other purposes
	<code>node_memory_Active_bytes{instance="\$node",job="\$job"}</code>	Active - Memory that has been used more recently and usually not reclaimed unless absolutely necessary
Memory Committed (min,max,avg,current)	<code>node_memory_Committed_AS_bytes{instance="\$node",job="\$job"}</code>	Committed_AS - Amount of memory presently allocated on the system

	node_memory_CommitLimit_bytes{instance="\$node",job="\$job"}	CommitLimit - Amount of memory currently available to be allocated on the system
Memory Active / Inactive Detail (min,max,avg,current)	node_memory_Inactive_file_bytes{instance="\$node",job="\$job"}	Inactive_file - File-backed memory on inactive LRU list
	node_memory_Inactive_anon_bytes{instance="\$node",job="\$job"}	Inactive_anon - Anonymous and swap cache on inactive LRU list, including tmpfs (shmem)
	node_memory_Active_file_bytes{instance="\$node",job="\$job"}	Active_file - File-backed memory on active LRU list
	node_memory_Active_anon_bytes{instance="\$node",job="\$job"}	Active_anon - Anonymous and swap cache on active least-recently-used (LRU) list, including tmpfs
Memory Writeback and Dirty (min,max,avg,current)	node_memory_Writeback_bytes{instance="\$node",job="\$job"}	Writeback - Memory which is actively being written back to disk
	node_memory_WritebackTmp_bytes{instance="\$node",job="\$job"}	WritebackTmp - Memory used by FUSE for temporary writeback buffers
	node_memory_Dirty_bytes{instance="\$node",job="\$job"}	Dirty - Memory which is waiting to get written back to the disk
Memory Shared and Mapped (min,max,avg,current)	node_memory_Mapped_bytes{instance="\$node",job="\$job"}	Mapped - Used memory in mapped pages files which have been mmaped, such as libraries
	node_memory_Shmem_bytes{instance="\$node",job="\$job"}	Shmem - Used shared memory (shared between several processes, thus including RAM disks)
	node_memory_ShmemHugePages_bytes{instance="\$node",job="\$job"}	ShmemHugePages - Memory used by shared memory (shmem) and tmpfs allocated with huge pages

	node_memory_ShmemPmdMapped_bytes{instance="\$node",job="\$job"}	ShmemPmdMapped - Ammount of shared (shmem/tmpfs) memory backed by huge pages
Memory Slab (min,max,avg,current)	node_memory_SUnreclaim_bytes{instance="\$node",job="\$job"}	SUnreclaim - Part of Slab, that cannot be reclaimed on memory pressure
	node_memory_SReclaimable_bytes{instance="\$node",job="\$job"}	SReclaimable - Part of Slab, that might be reclaimed, such as caches
Memory Vmalloc (min,max,avg,current)	node_memory_VmallocChunk_bytes{instance="\$node",job="\$job"}	VmallocChunk - Largest contiguous block of vmalloc area which is free
	node_memory_VmallocTotal_bytes{instance="\$node",job="\$job"}	VmallocTotal - Total size of vmalloc memory area
	node_memory_VmallocUsed_bytes{instance="\$node",job="\$job"}	VmallocTotal - Total size of vmalloc memory area
Memory Bounce (min,max,avg,current)	node_memory_Bounce_bytes{instance="\$node",job="\$job"}	Bounce - Memory used for block device bounce buffers
Memory Anonymous (min,max,avg,current)	node_memory_AnonHugePages_bytes{instance="\$node",job="\$job"}	AnonHugePages - Memory in anonymous huge pages
	node_memory_AnonPages_bytes{instance="\$node",job="\$job"}	AnonPages - Memory in user pages not backed by files
Memory Kernel / CPU (min,max,avg,current)	node_memory_KernelStack_bytes{instance="\$node",job="\$job"}	KernelStack - Kernel memory stack. This is not reclaimable

	<code>node_memory_Percpu_bytes{instance="\$node",job="\$job"}</code>	PerCPU - Per CPU memory allocated dynamically by loadable modules
Memory HugePages Counter (min,max,avg,current)	<code>node_memory_HugePages_Free{instance="\$node",job="\$job"}</code>	HugePages_Free - Huge pages in the pool that are not yet allocated
	<code>node_memory_HugePages_Rsvd{instance="\$node",job="\$job"}</code>	HugePages_Rsvd - Huge pages for which a commitment to allocate from the pool has been made, but no allocation has yet been
	<code>node_memory_HugePages_Surp{instance="\$node",job="\$job"}</code>	HugePages_Surp - Huge pages in the pool above the value in <code>/proc/sys/vm/nr_hugepages</code>
Memory HugePages Size (min,max,avg,current)	<code>node_memory_HugePages_Total{instance="\$node",job="\$job"}</code>	HugePages - Total size of the pool of huge pages
	<code>node_memory_Hugepagesize_bytes{instance="\$node",job="\$job"}</code>	Hugepagesize - Huge Page size
Memory DirectMap (min,max,avg,current)	<code>node_memory_DirectMap1G_bytes{instance="\$node",job="\$job"}</code>	DirectMap1G - Amount of pages mapped as this size
	<code>node_memory_DirectMap2M_bytes{instance="\$node",job="\$job"}</code>	DirectMap2M - Amount of pages mapped as this size
	<code>node_memory_DirectMap4k_bytes{instance="\$node",job="\$job"}</code>	DirectMap4K - Amount of pages mapped as this size
Memory Unevictable and MLocked (min,max,avg,current)	<code>node_memory_Unevictable_bytes{instance="\$node",job="\$job"}</code>	Unevictable - Amount of unevictable memory that can't be swapped out for a variety of reasons
	<code>node_memory_Mlocked_bytes{instance="\$node",job="\$job"}</code>	MLocked - Size of pages locked to memory using the <code>mlock()</code> system call
Memory NFS (min,max,avg,current)	<code>node_memory_NFS_Unstable_bytes{instance="\$node",job="\$job"}</code>	NFS Unstable - Memory in NFS pages sent to the server, but not yet committed to the storage

Memory Vmstat		
Memory Pages In / Out (min,max,avg,current)	rate(node_vmstat_pgpgin{instance="\$node",job="\$job"}[\$_rate_interval])	Pagesin - Page in operations
	rate(node_vmstat_pgpgout{instance="\$node",job="\$job"}[\$_rate_interval])	Pagesout - Page out operations
Memory Pages Swap In / Out (min,max,avg,current)	rate(node_vmstat_pswpin{instance="\$node",job="\$job"}[\$_rate_interval])	Pswpin - Pages swapped in
	rate(node_vmstat_pswpout{instance="\$node",job="\$job"}[\$_rate_interval])	Pswpout - Pages swapped out
Memory Page Faults (min,max,avg,current)	rate(node_vmstat_pgfault{instance="\$node",job="\$job"}[\$_rate_interval])	Pgfault - Page major and minor fault operations
	rate(node_vmstat_pgmajfault{instance="\$node",job="\$job"}[\$_rate_interval])	Pgmajfault - Major page fault operations
	rate(node_vmstat_pgfault{instance="\$node",job="\$job"}[\$_rate_interval]) - rate(node_vmstat_pgmajfault{instance="\$node",job="\$job"}[\$_rate_interval])	Pgminfault - Minor page fault operations
OOM Killer (min,max,avg,current)	rate(node_vmstat_oom_kill{instance="\$node",job="\$job"}[\$_rate_interval])	oom killer invocations
System Timesync		

Time Synchronized Drift (min,max,avg,current)	node_timex_estimated_error_seconds{instance="\$node",job="\$job"}	Estimated error in seconds
	node_timex_offset_seconds{instance="\$node",job="\$job"}	Time offset in between local system and reference clock
	node_timex_maxerror_seconds{instance="\$node",job="\$job"}	Maximum error in seconds
Time PLL Adjust (min,max,avg,current)	node_timex_loop_time_constant{instance="\$node",job="\$job"}	Phase-locked loop time adjust
Time Synchronized Status (min,max,avg,current)	node_timex_sync_status{instance="\$node",job="\$job"}	Is clock synchronized to a reliable server (1 = yes, 0 = no)
	node_timex_frequency_adjustment_ratio{instance="\$node",job="\$job"}	Local clock frequency adjustment
Time Misc (min,max,avg,current)	node_timex_tick_seconds{instance="\$node",job="\$job"}	Seconds between clock ticks
	node_timex_tai_offset_seconds{instance="\$node",job="\$job"}	International Atomic Time (TAI) offset
System Processes		
Processes Status (min,max,avg,current)	node_procs_blocked{instance="\$node",job="\$job"}	Processes blocked waiting for I/O to complete
	node_procs_running{instance="\$node",job="\$job"}	Processes in runnable state
Processes Forks (min,max,avg,current)	rate(node_forks_total{instance="\$node",job="\$job"}[\$_rate_interval])	Processes forks second
Process schedule stats Running /	rate(node_schedstat_running_seconds_total{instance="\$node",job="\$job"}[\$_rate_interval])	seconds spent running a process в разрезе CPU

Waiting (min,max,avg,current)	rate(node_schedstat_waiting_seconds_total{instance="\$node",job="\$job"}[\$_rate_interval])	seconds spent by processing waiting for this в разрезе CPU
System Misc		
Context Switches / Interrupts (min,max,avg,current)	rate(node_context_switches_total{instance="\$node",job="\$job"}[\$_rate_interval])	Context switches
	rate(node_intr_total{instance="\$node",job="\$job"}[\$_rate_interval])	Interrupts
System Load (min,max,avg,current)	node_load1{instance="\$node",job="\$job"}	Load 1m
	node_load5{instance="\$node",job="\$job"}	Load 5m
	node_load15{instance="\$node",job="\$job"}	Load 15m
Entropy (min,max,avg,current)	node_entropy_available_bits{instance="\$node",job="\$job"}	Entropy available to random number generators
Schedule timeslices executed by each cpu (min,max,avg,current)	rate(node_schedstat_timeslices_total{instance="\$node",job="\$job"}[\$_rate_interval])	в разрезе CPU
File Descriptors (min,max,avg,current)	process_max_fds{instance="\$node",job="\$job"}	Maximum open file descriptors
	process_open_fds{instance="\$node",job="\$job"}	Open file descriptors
CPU time spent in user and system contexts (min,max,avg,current)	rate(process_cpu_seconds_total{instance="\$node",job="\$job"}[\$_rate_interval])	Time spent
Hardware Misc		
	node_hwmon_temp_celsius{instance="\$node",job="\$job"}	

Hardware temperature monitor (min,max,avg,current)	node_hwmon_temp_crit_alarm_celsius{instance="\$node",job="\$job"}	
	node_hwmon_temp_crit_celsius{instance="\$node",job="\$job"}	
	node_hwmon_temp_crit_hyst_celsius{instance="\$node",job="\$job"}	
	node_hwmon_temp_max_celsius{instance="\$node",job="\$job"}	
Throttle cooling device (min,max,avg,current)	node_cooling_device_cur_state{instance="\$node",job="\$job"}	Current 0 in Processor
	node_cooling_device_max_state{instance="\$node",job="\$job"}	Max 0 in Processor
Storage Disk		
Disk IOps Completed (min,max,avg,current)	rate(node_disk_reads_completed_total{instance="\$node",job="\$job"}[\$_rate_interval])	Reads completed по девайсам
	rate(node_disk_writes_completed_total{instance="\$node",job="\$job"}[\$_rate_interval])	Writes completed по девайсам
Disk R/W Data (min,max,avg,current)	rate(node_disk_read_bytes_total{instance="\$node",job="\$job"}[\$_rate_interval])	Read bytes
	rate(node_disk_written_bytes_total{instance="\$node",job="\$job"}[\$_rate_interval])	Written bytes
	rate(node_disk_read_time_seconds_total{instance="\$node",job="\$job"}[\$_rate_interval]) /	Read wait time avg по девайсам

Disk Average Wait Time (min,max,avg,current)	rate(node_disk_reads_completed_total{instance="\$node",job="\$job"}[\$_rate_interval])	
	rate(node_disk_write_time_seconds_total{instance="\$node",job="\$job"}[\$_rate_interval]) / rate(node_disk_writes_completed_total{instance="\$node",job="\$job"}[\$_rate_interval])	Write wait time avg по девайсам
Average Queue Size (min,max,avg,current)	rate(node_disk_io_time_weighted_seconds_total{instance="\$node",job="\$job"}[\$_rate_interval])	По устройствам
Disk R/W Merged (min,max,avg,current)	rate(node_disk_reads_merged_total{instance="\$node",job="\$job"}[\$_rate_interval])	Read merged
	rate(node_disk_writes_merged_total{instance="\$node",job="\$job"}[\$_rate_interval])	Write merged
Time Spent Doing I/Os (min,max,avg,current)	rate(node_disk_io_time_seconds_total{instance="\$node",job="\$job"}[\$_rate_interval])	По устройствам
	rate(node_disk_discard_time_seconds_total{instance="\$node",job="\$job"}[\$_rate_interval])	
Instantaneous Queue Size (min,max,avg,current)	node_disk_io_now{instance="\$node",job="\$job"}	По устройствам
Disk IOps Discards completed / merged (min,max,avg,current)	rate(node_disk_discards_completed_total{instance="\$node",job="\$job"}[\$_rate_interval])	
	rate(node_disk_discards_merged_total{instance="\$node",job="\$job"}[\$_rate_interval])	

Storage Filesystem		
Filesystem space available (min,max,avg,current)	node_filesystem_avail_bytes{instance="\$node",job="\$job",device!~'rootfs'}	/ - Available /var/lib/systemd/coredump - Available /run - Available
	node_filesystem_free_bytes{instance="\$node",job="\$job",device!~'rootfs'}	/run/user/0 - Available
	node_filesystem_size_bytes{instance="\$node",job="\$job",device!~'rootfs'}	
File Nodes Free (min,max,avg,current)	node_filesystem_files_free{instance="\$node",job="\$job",device!~'rootfs'}	/ - Free file nodes /var/lib/systemd/coredump - Free file nodes /run - Free file nodes /run/user/0 - Free file nodes
File Descriptor (min,max,avg,current)	node_filefd_maximum{instance="\$node",job="\$job"}	Max open files Open files
	node_filefd_allocated{instance="\$node",job="\$job"}	
File Nodes Size (min,max,avg,current)	node_filesystem_files{instance="\$node",job="\$job",device!~'rootfs'}	/ - File nodes total /var/lib/systemd/coredump - File nodes total /run - File nodes total /run/user/0 - File nodes total
Storage Filesystem		
File Descriptor (min,max,avg,current)	node_filefd_maximum{instance="\$node",job="\$job"}	Max open files Open files
	node_filefd_allocated{instance="\$node",job="\$job"}	
File Nodes Size (min,max,avg,current)	node_filesystem_files{instance="\$node",job="\$job",device!~'rootfs'}	/ - File nodes total /var/lib/systemd/coredump - File nodes total /run - File nodes total

		/run/user/0 - File nodes total
Network Traffic by Packets (min,max,avg,current)	rate(node_network_receive_packets_total{instance="\$node",job="\$job"}[\$_rate_interval])	Входящий трафик интерфейсов, по типу интерфейса по пакетам
	rate(node_network_transmit_packets_total{instance="\$node",job="\$job"}[\$_rate_interval])	Исходящий трафик интерфейсов, по типу интерфейса по пакетам
Network Traffic Errors (min,max,avg,current)	rate(node_network_receive_errs_total{instance="\$node",job="\$job"}[\$_rate_interval])	Receive errors по устройствам
	rate(node_network_transmit_errs_total{instance="\$node",job="\$job"}[\$_rate_interval])	Transmit errors по устройствам
Network Traffic Drop (min,max,avg,current)	rate(node_network_receive_drop_total{instance="\$node",job="\$job"}[\$_rate_interval])	Receive drop по устройствам
	rate(node_network_transmit_drop_total{instance="\$node",job="\$job"}[\$_rate_interval])	Transmit drop по устройствам
Network Traffic Compressed (min,max,avg,current)	rate(node_network_receive_compressed_total{instance="\$node",job="\$job"}[\$_rate_interval])	Receive compressed по устройствам
	rate(node_network_transmit_compressed_total{instance="\$node",job="\$job"}[\$_rate_interval])	Transmit compressed по устройствам
Network Traffic Multicast (min,max,avg,current)	rate(node_network_receive_multicast_total{instance="\$node",job="\$job"}[\$_rate_interval])	Receive multicast по устройствам
Network Traffic Fifo (min,max,avg,current)	rate(node_network_receive_fifo_total{instance="\$node",job="\$job"}[\$_rate_interval])	Receive fifo по устройствам

	rate(node_network_transmit_fifo_total{instance="\$node",job="\$job"}[\$_rate_interval])	Transmit fifo по устройствам
Network Traffic Frame (min,max,avg,current)	rate(node_network_receive_frame_total{instance="\$node",job="\$job"}[\$_rate_interval])	Receive frame по устройствам
Network Traffic Carrier (min,max,avg,current)	rate(node_network_transmit_carrier_total{instance="\$node",job="\$job"}[\$_rate_interval])	Statistic transmit_carrier по устройствам
Network Traffic Colls (min,max,avg,current)	rate(node_network_transmit_colls_total{instance="\$node",job="\$job"}[\$_rate_interval])	Transmit colls по устройствам
NF Contrack (min,max,avg,current)	node_nf_contrack_entries{instance="\$node",job="\$job"}	
	node_nf_contrack_entries_limit{instance="\$node",job="\$job"}	
ARP Entries (min,max,avg,current)	node_arp_entries{instance="\$node",job="\$job"}	ARP entries по устройствам
MTU (min,max,avg,current)	node_network_mtu_bytes{instance="\$node",job="\$job"}	Bytes по устройствам
Speed (min,max,avg,current)	node_network_speed_bytes{instance="\$node",job="\$job"}	Speed по устройствам
Queue Length (min,max,avg,current)	node_network_transmit_queue_length{instance="\$node",job="\$job"}	Interface transmit queue length по устройствам
Softnet Packets (min,max,avg,current)	rate(node_softnet_processed_total{instance="\$node",job="\$job"}[\$_rate_interval])	Processed по CPU
	rate(node_softnet_dropped_total{instance="\$node",job="\$job"}[\$_rate_interval])	Dropped по CPU

Softnet Out of Quota (min,max,avg,current)	rate(node_softnet_times_squeezed_total{instance="\$node",job="\$job"}[\$_rate_interval])	Squeezed по устройствам
Network Operational Status (min,max,avg,current)	node_network_up{operstate="up",instance="\$node",job="\$job"}	Physical link state по устройствам
	node_network_carrier{instance="\$node",job="\$job"}	
Network Sockstat		
Sockstat TCP (min,max,avg,current)	node_sockstat_TCP_alloc{instance="\$node",job="\$job"}	TCP_alloc - Allocated sockets
	node_sockstat_TCP_inuse{instance="\$node",job="\$job"}	TCP_inuse - Tcp sockets currently in use
	node_sockstat_TCP_mem{instance="\$node",job="\$job"}	TCP_orphan - Orphan sockets
	node_sockstat_TCP_orphan{instance="\$node",job="\$job"}	TCP_tw - Sockets wating close
	node_sockstat_TCP_tw{instance="\$node",job="\$job"}	
Sockstat UDP (min,max,avg,current)	node_sockstat_UDPLITE_inuse{instance="\$node",job="\$job"}	UDPLITE_inuse - Udplite sockets currently in use
	node_sockstat_UDP_inuse{instance="\$node",job="\$job"}	UDP_inuse - Udp sockets currently in use
	node_sockstat_UDP_mem{instance="\$node",job="\$job"}	UDP_mem - Used memory for udp
Sockstat FRAG / RAW (min,max,avg,current)	node_sockstat_FRAG_inuse{instance="\$node",job="\$job"}	FRAG_inuse - Frag sockets currently in use
	node_sockstat_RAW_inuse{instance="\$node",job="\$job"}	RAW_inuse - Raw sockets currently in use
Sockstat Used (min,max,avg,current)	node_sockstat_sockets_used{instance="\$node",job="\$job"}	Sockets_used - Sockets currently in use

Network Netstat		
Netstat IP In / Out Octets	rate(node_netstat_IpExt_InOctets{instance="\$node",job="\$job"}[\$_rate_interval])	OutOctets - Sent octets
	rate(node_netstat_IpExt_OutOctets{instance="\$node",job="\$job"}[\$_rate_interval])	InOctets - Received octets
Netstat IP Forwarding	rate(node_netstat_Ip_Forwarding{instance="\$node",job="\$job"}[\$_rate_interval])	Forwarding - IP forwarding
ICMP In / Out	rate(node_netstat_Icmp_InMsgs{instance="\$node",job="\$job"}[\$_rate_interval])	InMsgs - Messages which the entity received. Note that this counter includes all those counted by icmpInErrors
	rate(node_netstat_Icmp_OutMsgs{instance="\$node",job="\$job"}[\$_rate_interval])	OutMsgs - Messages which this entity attempted to send. Note that this counter includes all those counted by icmpOutErrors
ICMP Errors	rate(node_netstat_Icmp_InErrors{instance="\$node",job="\$job"}[\$_rate_interval])	InErrors - Messages which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length,
UDP In / Out	rate(node_netstat_Udp_InDatagrams{instance="\$node",job="\$job"}[\$_rate_interval])	InDatagrams - Datagrams received
	rate(node_netstat_Udp_OutDatagrams{instance="\$node",job="\$job"}[\$_rate_interval])	OutDatagrams - Datagrams sent
UDP Errors	rate(node_netstat_Udp_InErrors{instance="\$node",job="\$job"}[\$_rate_interval])	InErrors - UDP Datagrams that could not be delivered to an application
	rate(node_netstat_Udp_NoPorts{instance="\$node",job="\$job"}[\$_rate_interval])	NoPorts - UDP Datagrams received on a port with no listener

	rate(node_netstat_UdpLite_InErrors{instance="\$node",job="\$job"}[\$_rate_interval])	InErrors Lite - UDPLite Datagrams that could not be delivered to an application
	rate(node_netstat_Udp_RcvbufErrors{instance="\$node",job="\$job"}[\$_rate_interval])	RcvbufErrors - UDP buffer errors received
	rate(node_netstat_Udp_SndbufErrors{instance="\$node",job="\$job"}[\$_rate_interval])	SndbufErrors - UDP buffer errors send
TCP In / Out	rate(node_netstat_Tcp_InSegs{instance="\$node",job="\$job"}[\$_rate_interval])	InSegs - Segments received, including those received in error. This count includes segments received on currently establi
	rate(node_netstat_Tcp_OutSegs{instance="\$node",job="\$job"}[\$_rate_interval])	OutSegs - Segments sent, including those on current connections but excluding those containing only retransmitted octets
TCP Errors	rate(node_netstat_TcpExt_ListenOverflows{instance="\$node",job="\$job"}[\$_rate_interval])	ListenOverflows - Times the listen queue of a socket overflowed
	rate(node_netstat_TcpExt_TCPSynRetrans{instance="\$node",job="\$job"}[\$_rate_interval])	TCPSynRetrans - SYN-SYN/ACK retransmits to break down retransmissions in SYN, fast/timeout retransmits
	rate(node_netstat_Tcp_RetransSegs{instance="\$node",job="\$job"}[\$_rate_interval])	RetransSegs - Segments retransmitted - that is, the number of TCP segments transmitted containing one or more previously
	rate(node_netstat_Tcp_InErrs{instance="\$node",job="\$job"}[\$_rate_interval])	InErrs - Segments received in error (e.g., bad TCP checksums)
	rate(node_netstat_Tcp_OutRsts{instance="\$node",job="\$job"}[\$_rate_interval])	OutRsts - Segments sent with RST flag
	rate(node_netstat_TcpExt_ListenDrops{instance="\$node",job="\$job"}[\$_rate_interval])	ListenDrops - SYNs to LISTEN sockets ignored
TCP Connections	node_netstat_Tcp_CurrEstab{instance="\$node",job="\$job"}	

	node_netstat_Tcp_MaxConn{instance="\$node",job="\$job"}	CurrEstab - TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT
TCP SynCookie	rate(node_netstat_TcpExt_SyncookiesFailed{instance="\$node",job="\$job"}[\$_rate_interval])	SyncookiesFailed - Invalid SYN cookies received
	rate(node_netstat_TcpExt_SyncookiesRecv{instance="\$node",job="\$job"}[\$_rate_interval])	SyncookiesRecv - SYN cookies received
	rate(node_netstat_TcpExt_SyncookiesSent{instance="\$node",job="\$job"}[\$_rate_interval])	SyncookiesSent - SYN cookies sent
TCP Direct Transition	rate(node_netstat_Tcp_ActiveOpens{instance="\$node",job="\$job"}[\$_rate_interval])	ActiveOpens - TCP connections that have made a direct transition to the SYN-SENT state from the CLOSED state

Метрики утилизации экспортера

Название панели	Формула	Пояснение
Количество ядер CPU по модулям	sum by (module_id) (cpu_num{module_id!=""})	
Оперативная память по модулям, MiB	sum by (module_id) (mem_size{module_id!=""})	
Размер SSD по модулям, MiB	sum by (module_id) (disk_size_ssd{module_id!=""})	
Размер SSD DWPD3 по модулям, MiB	sum by (module_id) (disk_size_ssd_dwprd3{module_id!=""})	
Размер NL SAS по модулям, MiB	sum by (module_id) (disk_size_nlsas{module_id!=""})	
Размер NVME по модулям MiB	sum by (module_id) (disk_size_nvme{module_id!=""})	

Использование CPU по хостам	sum(cpu_usage) by (instance)	ПО хостам
Использование памяти по хостам, MiB	sum(mem_usage) by (instance)	
Использование SSD по хостам, MiB	sum(disk_usage_ssd) by (instance)	
Использование NVME по хостам, MiB	sum(disk_usage_nvme) by (instance)	
Использование SSD DWPD3 по хостам, MiB	sum(disk_usage_ssd_dwprd3) by (instance)	
Использование NLSAS по хостам, MiB	sum(disk_usage_nlsas) by (instance)	

Отчет о ресурсах

Переменные:

- Host- label_values({_name_ =~ "utlz_discovery", module_title="\$module"}, instance)
- module label_values({_name_ =~ "up"}, module_title)

Название панели	Формула	Пояснение
CPU	cpu_num{instance="\$host"}	
RAM	mem_size{instance="\$host"}	
SSD	disk_size_ssd{instance="\$host"}	
NVME	disk_size_nvme{instance="\$host"}	
DWPD3	disk_size_ssd_dwprd3{instance="\$host"}	
NL-SAS	disk_size_nlsas{instance="\$host"}	

Визион - Отчет об утилизации ресурсов

Переменные

- Host -label_values({_name_ =~ "utlz_discovery"}, instance)
- module -label_values({_name_ =~ "utlz_discovery"}, module_id)

Название панели	Формула	Пояснение
\${host}	max_over_time(avg(cpu_usage{instance="\$host"})[\$_range:1m])	Информация по каждому хосту о метриках: Процессор\ядра, % Память, Мб Дисковые ресурсы (SSD), Мб
	sum(max_over_time(mem_usage{instance=~"\$host"}[\$_range]))	
	sum(max_over_time(disk_usage_nlsas{instance=~"\$host"}[\$_range]))	
	sum(max_over_time(disk_usage_nvme{instance=~"\$host"}[\$_range]))	
	sum(max_over_time(disk_usage_ssd_dwpd3{instance=~"\$host"}[\$_range]))	
	sum(max_over_time(disk_usage_ssd{instance=~"\$host"}[\$_range]))	

Визион – Предупреждения

Переменные:

- Datasource - prometheus
- instance - label_values(ALERTS_FOR_STATE,instance)
- alertname - label_values(ALERTS_FOR_STATE,alertname)
- alertstate - label_values(ALERTS, alertstate)
- Filters

Название панели	Формула	Пояснение
-----------------	---------	-----------

История предупреждений	ALERTS{instance=~"\$instance", alertname=~"\$alertname", alert-state=~"\$alertstate"}	
	1	
Общее число предупреждений		Показывает, как много предупреждений было в указанный период. Предупреждение может быть активным или ожидающим.
Предупреждения		Список предупреждений

Визион - Список узлов

Переменные:

- Interval - 1m,2m,3m,5m,10m,30m
- total_srv - query_result(count(node_uname_info{instance_type!="VM", module_title="\$module_title"}))
- total_vm - query_result(count(node_uname_info{instance_type="VM"}))
- module_title - label_values({_name_ =~ "up"}, module_title)

Название панели	Формула	Пояснение
Список узлов [всего: \$total_srv]	node_uname_info{instance_type!="VM", module_title="\$module_title"} - 0	Таблица метрик по хостам (Узел, Исп. CPU%, Время работы, RAM, CPU, Исп. RAM %, Исп. диск %, Disk read, Disk write, Receive , Transmit)
	(1 - avg(irate(node_cpu_seconds_total{instance_type!="VM", module_title="\$module_title", mode="idle"}[\$interval])) by (instance)) * 100	
	sum(time() - node_boot_time_seconds{instance_type!="VM", module_title="\$module_title"})by(instance) node_memory_MemTotal_bytes{instance_type!="VM", module_title="\$module_title"} - 0	
	count(node_cpu_seconds_total{instance_type!="VM", module_title="\$module_title", mode='system'}) by (instance)	

	node_load5{instance_type!="VM", module_title="\$module_title"}	
	(node_memory_Buffers_bytes{instance_type!="VM", module_title="\$module_title"}+ node_memory_Cached_bytes{instance_type!="VM", module_title="\$module_title"}) / node_memory_MemTotal_bytes{instance_type!="VM", module_title="\$module_title"} * 100	
	max((node_filesystem_size_bytes{instance_type!="VM", module_title="\$module_title", fstype=~"ext.? xfs"}-node_filesystem_free_bytes{instance_type!="VM", module_title="\$module_title", fstype=~"ext.? xfs"}) *100/(node_filesystem_avail_bytes {instance_type!="VM", module_title="\$module_title", fstype=~"ext.? xfs"}+(node_filesystem_size_bytes{instance_type!="VM", module_title="\$module_title", fstype=~"ext.? xfs"}- node_filesystem_free_bytes{instance_type!="VM", module_title="\$module_title", fstype=~"ext.? xfs"})))by(instance)	
	max(rate(node_disk_read_bytes_total{instance_type!="VM", module_title="\$module_title"}[\$interval])) by (instance)	
	max(rate(node_disk_written_bytes_total{instance_type!="VM", module_title="\$module_title"}[\$interval])) by (instance)	
	node_netstat_Tcp_CurrEstab{instance_type!="VM", module_title="\$module_title"} - 0	
	node_sockstat_TCP_tw{instance_type!="VM", module_title="\$module_title"} - 0	
	max(rate(node_network_receive_bytes_total{instance_type!="VM", module_title="\$module_title"}[\$interval])*8) by (instance)	

	$\text{max}(\text{rate}(\text{node_network_transmit_bytes_total}\{\text{instance_type}=\text{"VM"}, \text{module_title}=\text{"\$module_title"}\}[\text{\$interval}]) * 8) \text{ by } (\text{instance})$	
Список виртуальных машин [всего: $\text{\$total_vm}$]	$\text{node_uname_info}\{\text{instance_type}=\text{"VM"}\} - 0$	Таблица метрик по VM (Узел, Исп. CPU%, Время работы, RAM, CPU, Исп. RAM %, Исп. диск %, Disk read, Disk write, Receive, Transmit)
	$(1 - \text{avg}(\text{irate}(\text{node_cpu_seconds_total}\{\text{instance_type}=\text{"VM"}, \text{mode}=\text{"idle"}\}[\text{\$interval}])) \text{ by } (\text{instance})) * 100$	
	$\text{sum}(\text{time}() - \text{node_boot_time_seconds}\{\text{instance_type}=\text{"VM"}\}) \text{ by } (\text{instance})$	
	$\text{node_memory_MemTotal_bytes}\{\text{instance_type}=\text{"VM"}\} - 0$	
	$\text{count}(\text{node_cpu_seconds_total}\{\text{instance_type}=\text{"VM"}, \text{mode}=\text{'system'}\}) \text{ by } (\text{instance})$	
	$\text{node_load5}\{\text{instance_type}=\text{"VM"}\}$	
	$(\text{node_memory_Buffers_bytes}\{\text{instance_type}=\text{"VM"}\} + \text{node_memory_Cached_bytes}\{\text{instance_type}=\text{"VM"}\}) / \text{node_memory_MemTotal_bytes}\{\text{instance_type}=\text{"VM"}\} * 100$	
	$\text{max}((\text{node_filesystem_size_bytes}\{\text{instance_type}=\text{"VM"}, \text{fstype}=\sim\text{"ext.?\{xfs\}} - \text{node_filesystem_free_bytes}\{\text{instance_type}=\text{"VM"}, \text{fstype}=\sim\text{"ext.?\{xfs\}}\}) * 100 / (\text{node_filesystem_avail_bytes}\{\text{instance_type}=\text{"VM"}, \text{fstype}=\sim\text{"ext.?\{xfs\}}\} + (\text{node_filesystem_size_bytes}\{\text{instance_type}=\text{"VM"}, \text{fstype}=\sim\text{"ext.?\{xfs\}} - \text{node_filesystem_free_bytes}\{\text{instance_type}=\text{"VM"}, \text{fstype}=\sim\text{"ext.?\{xfs\}}\}))) \text{ by } (\text{instance})$	
$\text{max}(\text{rate}(\text{node_disk_read_bytes_total}\{\text{instance_type}=\text{"VM"}\}[\text{\$interval}])) \text{ by } (\text{instance})$		

	$\max(\text{rate}(\text{node_disk_written_bytes_total}\{\text{instance_type}=\text{"VM"}\}[\$\text{interval}]))$ $\text{by } (\text{instance})$	
	$\text{node_netstat_Tcp_CurrEstab}\{\text{instance_type}=\text{"VM"}\} - 0$	
	$\text{node_sockstat_TCP_tw}\{\text{instance_type}=\text{"VM"}\} - 0$	
	$\max(\text{rate}(\text{node_network_receive_bytes_total}\{\text{instance_type}=\text{"VM"}\}[\$\text{interval}])) * 8$ $\text{by } (\text{instance})$	
	$\max(\text{rate}(\text{node_network_transmit_bytes_total}\{\text{instance_type}=\text{"VM"}\}[\$\text{interval}])) * 8$ $\text{by } (\text{instance})$	
Предупреждения		Список предупреждений

Визион - Статус Blackbox Exporter

Переменные:

- Interval - 5s,10s,30s,1m,10m,30m,1h,6h,12h,1d,7d,14d,30d
- target

Название панели	Формула	Пояснение
Обзор датчиков	$\text{probe_success}\{\text{instance}=\sim\text{"\$target"}\}$	Состояние по инстансам
	$\text{avg_over_time}(\text{probe_duration_seconds}\{\text{instance}=\sim\text{"\$target"}\}[\$\text{interval}])) * 1000$	

	avg_over_time(probe_dns_lookup_time_seconds{instance=~"\$target"}[\$interval])*1000	
Продолжительность опросов	sum(probe_duration_seconds{instance=~"\$target"}) by (instance)	(Min, max, avg) по инстансам

Визион - Статус Grok Exporter

Название панели	Формула	Пояснение
Журнал предупреждений	grok_journalctl_alert	

Визион - Статус IPMI Exporter

- Instance - label_values(ipmi_bmc_info, instance)

Название панели	Формула	Пояснение
Счетчик событий журнала IPMI	ipmi_sel_logs_count	
Состояние питания	ipmi_chassis_power_state	
Информация об узле	ipmi_bmc_info	
Состояние кулера	ipmi_fan_speed_state{instance="\$instance"}	
Скорость кулера (обороты в мин.)	ipmi_fan_speed_rpm{instance="\$instance"}	
Потребляемая мощность (Вт)	ipmi_dcmi_power_consumption_watts{instance="\$instance"}	
Состояние питания	ipmi_power_state{instance="\$instance"}	
Мощность (Вт)	ipmi_power_watts{instance="\$instance"}	

Состояние датчиков IPMI	<code>ipmi_sensor_state{instance=~"\$instance"}</code>	
Состояние температуры	<code>ipmi_temperature_state{instance=~"\$instance"}</code>	
Температура	<code>ipmi_temperature_celsius{instance=~"\$instance"}</code>	
Состояние датчика напряжения	<code>ipmi_voltage_state{instance=~"\$instance"}</code>	
Напряжение (Вольт)	<code>ipmi_voltage_volts{instance=~"\$instance"}</code>	

Визион - Статус Prometheus

Переменные:

- job
- instance
- interval - 1h, 3h, 6h, 12h, 1d, 2d, 7d, 30d, 90d, 180d

Название панели	Формула	Пояснение
Информация об экземпляре Prometheus	<code>max(time() - process_start_time_seconds{instance=~"\$instance"}) by (instance)</code>	Таблица метрик по каждому экземпляру Prometheus (Время работы, Версия, Целей опроса, Сохраненный период, Объем данных)
	<code>max(prometheus_build_info{instance=~"\$instance"}) by (instance, version)</code>	
	<code>max(sum(prometheus_sd_discovered_targets{instance=~"\$instance", name="scrape"}) by (instance)) by (instance)</code>	
	<code>time() - max(prometheus_tsdb_lowest_timestamp_seconds{instance=~"\$instance"}) by (instance)</code>	

	<pre>min(prometheus_tsdb_storage_blocks_bytes{instance=~"\$instance"}) by (instance)</pre>	
Состояние работы	<pre>up{instance=~"\$instance",job=~"\$job"}</pre>	
Недоступные цели	<pre>up < 1</pre>	Таблица доступности хостов
Ошибки Prometheus за \$interval	<pre>sum(increase(label_replace({_name_ =~ "(?:prometheus net_contrack)_*(?:fail duplicate out_of error not_found corrupt dropped missed exceeded).*_total", instance=~"\$instance",job="prometheus"}, "metric_name", "\$1", "_name_", "(.*)")[\$interval:20s])) by (instance,job,metric_name,alertmanager,name,mechanism,queue) > 0</pre>	<pre>{instance="127.0.0.1:9090", job="prometheus", metric_name="net_contrack_dialer_conn_failed_total"}</pre>
Количество серий	<pre>sum(prometheus_tsdb_head_series{job=~"\$job",instance=~"\$instance"})</pre>	<pre>sum(prometheus_tsdb_head_series{job=~"prometheus",instance=~"127\\.0\\.0\\.1:9090"})</pre>
Использование CPU	<pre>sum(rate(process_cpu_seconds_total{instance=~"\$instance"}[\$interval])) by (instance)</pre>	
Использование памяти	<pre>sum(process_resident_memory_bytes{instance=~"\$instance"}) by (instance)</pre>	Resident memory of 127.0.0.1:9090/
	<pre>sum(go_memstats_alloc_bytes{instance=~"\$instance"}) by (instance)</pre>	Total allocated bytes of 127.0.0.1:9090/

Самые старые данные	<code>time() - min(prometheus_tsdb_lowest_timestamp_seconds{instance=~"\$instance"}) by (instance)</code>	Oldest data
Размер хранилища	<code>max(prometheus_tsdb_storage_blocks_bytes{instance=~"\$instance"}) by (instance)</code>	Oldest data
Количество отосланных уведомлений за $\$interval$	<code>sum(increase(prometheus_notifications_sent_total{instance=~"\$instance"}[\$interval])) by (instance, alertmanager)</code>	

Визион - Статус SNMP

Переменные:

- Job - label_values(ifHCInOctets, job)
- Filters

Название панели	Формула	Пояснение
Время безотказной работы	<code>sysUpTime</code>	
Макс. исходящий (Сейчас)	<code>max(irate(ifHCOutOctets{job="\$Job"}[5m]))</code>	
Макс. входящий (Сейчас)	<code>max(irate(ifHCInOctets{job="\$Job"}[5m]))</code>	
Общий исходящий	<code>max(delta(ifHCOutOctets{job="\$Job"}[\$_range]))</code>	
Общий входящий	<code>max(delta(ifHCInOctets{job="\$Job"}[\$_range]))</code>	
Состояние	<code>delta(ifHCInOctets{job="\$Job"}[\$_range])</code>	
	<code>ifHighSpeed{job="\$Job"}</code>	

	delta(ifHCOctets{job="\$Job"}[\$_range])	
	delta(ifHCOctets{job="\$Job"}[\$_range])	
	irate(ifHCInOctets{job="\$Job"}[5m])	
	irate(ifHCOctets{job="\$Job"}[5m])	
Исходящий (Сейчас)	irate(ifHCOctets{job="\$Job"}[5m])	
Входящий (Сейчас)	irate(ifHCOctets{job="\$Job"}[5m])	
Исходящий / Входящий	irate(ifHCOctets{job="\$Job"}[5m])	
	-irate(ifHCInOctets{job="\$Job"}[5m])	

Визион - Статус коммутатора

Переменные:

DS_PROMETHEUS - prometheus

Job - label_values(ifHCInOctets, job)

instance - query_result(ifInOctets{job="snmp-exporter"})

index - label_values(ifHighSpeed{instance='\$instance'},ifIndex)

Interface - query_result(sort(ifIndex{ifName=~^(ether).+',instance='\$instance'}) or sort(ifIndex{ifName=~^(sfp|wlan).+',instance='\$instance'}) or sort(ifIndex{ifName=~^bridge.+',instance='\$instance'}) or sort(ifIndex{ifName!~^(ether|bridge).+',instance='\$instance'})))

desc - query_result(ifHCInOctets{ifName="\$Interface",instance="\$instance"})

Название панели	Формула	Пояснение
Время безотказной работы	sysUpTime{instance="\$instance",job="\$Job"}	

Макс. исходящий (Сейчас)	<code>max(irate(ifHCOutOctets{instance="\$instance",job="\$Job"}[5m]))</code>	
Макс. входящий (Сейчас)	<code>max(irate(ifHCInOctets{instance="\$instance",job="\$Job"}[5m]))</code>	
Общий исходящий	<code>max(delta(ifHCOutOctets{instance="\$instance",job="\$Job"}[\$_range]))</code>	
Общий входящий	<code>max(delta(ifHCInOctets{instance="\$instance",job="\$Job"}[\$_range]))</code>	
Статус интерфейсов		
<code>\$Interface</code>	<code>ifOperStatus{instance='\$instance',ifName='\$Interface'}</code>	
Трафик на интерфейсах		
Общий Входящий/Исходящий	<code>sum by(job,instance)(irate(ifHCOutOctets{job='snmp',instance='\$instance'}[5m]))</code>	
	<code>sum by(job,instance)(irate(ifHCInOctets{job='snmp',instance='\$instance'}[5m]))</code>	
Трафик "\$Interface"	<code>irate(ifHCInOctets{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])*8</code>	
	<code>-irate(ifHCOutOctets{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])*8</code>	
Статистика пакетов		
Интерфейс "\$Interface"	<code>irate(ifHCInMulticastPkts{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])</code>	

-irate(ifHCOutMulticastPkts{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	
irate(ifHCInBroadcastPkts{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	
-irate(ifHCOutBroadcastPkts{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	
irate(ifHCInUcastPkts{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	
irate(ifHCInUcastPkts{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	
-irate(ifHCOutUcastPkts{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	
irate(ifInDiscards{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	
irate(ifOutDiscards{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	
irate(ifInErrors{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	
irate(ifOutErrors{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	
irate(ifInUnknownProtos{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])	

	<code>-irate(ifOutQLen{ifName=~'\$Interface',instance='\$instance'}[\$_rate_interval])</code>	

Утилизации дисков по модулям

Название панели	Формула	Пояснение
Утилизация SSD по модулям	<code>sum by (module_id) (disk_usage_ssd{module_id!=""})/sum by (module_id) (disk_size_ssd{module_id!=""})</code>	
Утилизация SSD DWPD3 по модулям	<code>sum by (module_id) (disk_usage_ssd_dwprd3{module_id!=""})/sum by (module_id) (disk_size_ssd_dwprd3{module_id!=""})</code>	
Утилизация NL SAS по модулям	<code>sum by (module_id) (disk_usage_nlsas{module_id!=""})/sum by (module_id) (disk_size_nlsas{module_id!=""})</code>	
Утилизация NVME по модулям	<code>sum by (module_id) (disk_usage_nvme{module_id!=""})/sum by (module_id) (disk_size_nvme{module_id!=""})</code>	

Визион - Утилизация дисков по узлам

Переменные:

- Host - `label_values({_name_ =~ "utlz_discovery"}, instance)`
- module - `label_values({_name_ =~ "utlz_discovery"}, module_id)`

Название панели	Формула	Пояснение
SSD	<code>disk_usage_ssd{instance="\$host"}/disk_size_ssd{instance="\$host"}</code>	
NVME	<code>disk_usage_nvme{instance="\$host"}/disk_size_nvme{instance="\$host"}</code>	

DWPD3	disk_usage_ssd_dwpd3{instance="\$host"}/disk_size_ssd_dwpd3{instance="\$host"}	
NL-SAS	disk_usage_nlsas{instance="\$host"}/disk_size_nlsas{instance="\$host"}	

МБД.П - Информация о БД PostgreSQL

Переменные:

- Server - query_result(pg_settings_server_version_num)
- interval - 1s,5s,1m,5m,1h,6h,1d
- Instance - query_result(pg_up)
- datname - label_values(pg_stat_activity_count{server="\$server"}, datname)
- mode - label_values({mode=~"accessexclusive|accessshare|exclusive|rowexclusive|rowshare|sharelock|sharerowexclusive|shareupdateexclusive"}, mode)

Название панели	Формула	По-яснение
Обзор		
Время безотказной работы	pg_postmaster_start_time_seconds{server="\$server"} * 1000	
откат/фиксация	(sum(pg_stat_database_xact_rollback{server="\$server"})/sum(pg_stat_database_xact_commit{server="\$server"}))*100	
Подсчет простоя в транзакции	sum(pg_stat_activity_count{server="\$server",state="idle in transaction"})	
Сумма Exclusive-Lock на экземпляре	sum(pg_locks_count{datname=~"\$datname", datname!="postgres", server=~"\$server", mode=~"\$mode", mode!="accesssharelock", mode!="rowsharelock", mode!="sharelock"} > 1)	
PG частота попаданий в кэш	round(100*sum(pg_stat_database_blks_hit{server="\$server"}) / (sum(pg_stat_database_blks_hit{server="\$server"}) + sum(pg_stat_database_blks_read{server="\$server"})),0.1)	

% текущих/всего подключений	$(\text{sum}(\text{pg_stat_activity_count}\{\text{server}=\text{"\$server"}\}) / \text{sum}(\text{pg_settings_max_connections})) * 100$	
Транзакций в секунду	$(\text{sum}(\text{rate}(\text{pg_stat_database_xact_rollback}\{\text{server}=\text{"\$server"}\}[1\text{m}])) + \text{sum}(\text{rate}(\text{pg_stat_database_xact_commit}\{\text{server}=\text{"\$server"}\}[1\text{m}])))/60$	
Сумма ShareLock на экземпляре	$\text{sum}(\text{pg_locks_count}\{\text{datname}=\sim\text{"\$datname"}\}, \text{datname}!\sim\text{"postgres"}, \text{server}=\sim\text{"\$server"}, \text{mode}=\sim\text{"\$mode"}\}, \text{mode}!\sim\text{"accessexclusive\lock"}, \text{mode}!\sim\text{"exclusive\lock"}, \text{mode}!\sim\text{"rowexclusive\lock"}, \text{mode}!\sim\text{"sharerowexclusive\lock"}, \text{mode}!\sim\text{"shareupdateexclusive\lock"}\})$	
Запросов в секунду	$\text{sum}(\text{rate}(\text{pg_stat_statements_calls}\{\text{server}=\sim\text{"\$server"}\}[1\text{m}]))/60$	
Среднее время запроса за 1 мин. по экземпляру	$\text{sum}(\text{rate}(\text{pg_stat_statements_mean_time_seconds}\{\text{datname}!\sim\text{"template.*"}\}, \text{datname}!\sim\text{"postgres"}, \text{server}=\sim\text{"\$server"}\}[1\text{m}]))$	
Активные сессии	$\text{sum}(\text{pg_stat_activity_count}\{\text{state}=\text{"active"}\})$	
ТОП-5 самых долго выполняющихся запросов по среднему времени		
ТОП-5 самых долго выполняющихся запросов по среднему времени (сек.)	$\text{topk}(5, \text{pg_stat_statements_mean_time_seconds}\{\text{datname}!\sim\text{"template.*"}\}, \text{datname}!\sim\text{"postgres"}, \text{server}=\sim\text{"\$server"}, \text{datname}=\sim\text{"\$datname"}\})$	
ТОП-5 Частых запросов		
ТОП-5 Частых запросов	$\text{topk}(5, \text{pg_stat_statements_calls}\{\text{datname}!\sim\text{"template.*"}\}, \text{datname}!\sim\text{"postgres"}, \text{server}=\sim\text{"\$server"}, \text{datname}=\sim\text{"\$datname"}\})$	
Время синтетического запроса		
Среднее время запроса за 1 мин. по экземпляру	$\text{sum}(\text{rate}(\text{pg_stat_statements_mean_time_seconds}\{\text{datname}!\sim\text{"template.*"}\}, \text{datname}!\sim\text{"postgres"}, \text{server}=\sim\text{"\$server"}\}[1\text{m}]))$	

Среднее время запроса по базе данных	sum(increase(pg_stat_statements_calls{datname!~"template.*", datname!~"postgres", server=~"\$server", datname=~"\$datname"}[1m])) by (datname) > 100	
Количество синтетических запросов		
Общее количество запросов (за 1 мин.) по всему экземпляру	sum(increase(pg_stat_statements_calls{datname!~"template.*", datname!~"postgres", server=~"\$server"}[1m])) by (server)	
Общее количество запросов (за 1 мин.) по базе данных	sum(increase(pg_stat_statements_calls{datname!~"template.*", datname!~"postgres", server=~"\$server", datname=~"\$datname"}[1m])) by (datname) > 100	
Размер базы данных		
Размер базы данных	topk(5, (pg_database_size{server="\$server"}))	
Контрольная точка		
Статус контрольных точек	rate(pg_stat_bgwriter_checkpoint_write_time{server="\$server"}[\$interval]) or irate(pg_stat_bgwriter_checkpoint_write_time{server="\$server"}[5m])	
	rate(pg_stat_bgwriter_checkpoint_sync_time{server="\$server"}[\$interval]) or irate(pg_stat_bgwriter_checkpoint_sync_time{server="\$server"}[5m])	
Статистика контрольных точек	rate(pg_stat_bgwriter_checkpoint_write_time{server="\$server"}[5m])	
	rate(pg_stat_bgwriter_checkpoint_sync_time{server="\$server"}[5m])	
Буферные и блочные операции		
Операции с блоками	rate(pg_stat_database_blk_read_time{datname=~"\$datname", server=~"\$server", datname!~"template.* postgres"}[5m])	

	rate(pg_stat_database_blk_write_time{datname=~"\$datname",server=~"\$server",datname!~"template.* postgres"}[5m])	
Буферы (bgwriter)	rate(pg_stat_bgwriter_buffers_alloc{server="\$server"}[\$interval]) or irate(pg_stat_bgwriter_buffers_alloc{server="\$server"}[5m])	
	rate(pg_stat_bgwriter_buffers_backend_fsync{server="\$server"}[\$interval]) or irate(pg_stat_bgwriter_buffers_backend_fsync{server="\$server"}[5m])	
	rate(pg_stat_bgwriter_buffers_backend{server="\$server"}[\$interval]) or irate(pg_stat_bgwriter_buffers_backend{server="\$server"}[5m])	
	rate(pg_stat_bgwriter_buffers_clean{server="\$server"}[\$interval]) or irate(pg_stat_bgwriter_buffers_clean{server="\$server"}[5m])	
	rate(pg_stat_bgwriter_buffers_checkpoint{server="\$server"}[5m]) or irate(pg_stat_bgwriter_buffers_checkpoint{server="\$server"}[5m])	
Соединения		
Количество общий соединений	pg_settings_max_connections	
	sum(pg_stat_activity_count{server="\$server"})	
	sum(pg_stat_activity_count{server="\$server",state="idle"})	
	sum(pg_stat_activity_count{server="\$server",state="active"})	
	sum(pg_stat_activity_count{server="\$server",state="idle in transaction"})	
	sum(pg_stat_activity_count{server="\$server",state="idle in transaction (aborted)"})	
Активные сессии	pg_stat_activity_count{datname=~"\$datname", server=~"\$server", state="active"} !=0	

Соединения	topk(5, (pg_stat_activity_count{server="\$server"}))	
Общие счётчики		
Текущее чтение данных	SUM(pg_stat_database_tup_fetched{datname=~"\$datname", server=~"\$server"})	
Эффективный кэш	pg_settings_effective_cache_size_bytes{server="\$server"}	
Рабочая память	pg_settings_work_mem_bytes{server="\$server"}	
Макс. размер WAL	pg_settings_max_wal_size_bytes{server="\$server"}	
Макс. количество рабочих процессов	pg_settings_max_worker_processes{server="\$server"}	
Seq Page Cost	pg_settings_seq_page_cost{server="\$server"}	
Буфер Disk-Page	pg_settings_wal_buffers_bytes{server=~"\$server"}	
Текущая вставка данных	SUM(pg_stat_database_tup_inserted{datname=~"\$datname", server=~"\$server"})	
Текущее обновление данных	SUM(pg_stat_database_tup_updated{datname=~"\$datname", server=~"\$server"})	
Память (обслуживание)	pg_settings_maintenance_work_mem_bytes{server="\$server"}	
Random Page Cost	pg_settings_random_page_cost{server="\$server"}	
Разделяемый буфер	pg_settings_shared_buffers_bytes{server="\$server"}	
Версия	pg_static{server="\$server"}	

Макс. кол-во параллельных операций	<code>pg_settings_max_parallel_workers{server="\$server"}</code>	
Транзакции		
Фиксация транзакций	<code>rate(pg_stat_database_xact_commit{server="\$server"}[1m]) > 1</code>	
Откаты транзакций	<code>rate(pg_stat_database_xact_rollback{server="\$server"}[1m]) > 1</code>	
Продолжительная транзакция		
Продолжительная транзакция	<code>pg_stat_activity_max_tx_duration{server="\$server"} > 0</code>	
Выбрать, вставить, обновить		
Чтение	<code>rate(pg_stat_database_tup_fetched{server="\$server"}[5m]) > 1000</code>	
	<code>rate(pg_stat_database_tup_returned{server="\$server"}[5m]) > 1000</code>	
Вставка	<code>rate(pg_stat_database_tup_inserted{server="\$server"}[5m]) > 1</code>	
	<code>rate(pg_stat_database_tup_updated{server="\$server"}[5m]) > 1</code>	
	<code>rate(pg_stat_database_tup_deleted{server="\$server"}[5m]) > 1</code>	
Задержка репликации		
Задержка репликации	<code>rate(pg_replication_lag{server="\$server"}[5m])</code>	
Конфликты и блокировки		
Табличная блокировка	<code>pg_locks_count{datname=~"\$datname", datname!="postgres", server=~"\$server", mode=~"\$mode"} > 1</code>	

Отмененные запросы	$\text{sum}(\text{rate}(\text{pg_stat_database_conflicts_confl_bufferpin}\{\text{server}=\sim\$\text{server}\}\{\$\text{interval}\}))$ or $\text{sum}(\text{irate}(\text{pg_stat_database_conflicts_confl_bufferpin}\{\text{server}=\sim\$\text{server}\}\{5\text{m}\}))$	
	$\text{sum}(\text{rate}(\text{pg_stat_database_conflicts_confl_deadlock}\{\text{server}=\sim\$\text{server}\}\{\$\text{interval}\}))$ or $\text{sum}(\text{irate}(\text{pg_stat_database_conflicts_confl_deadlock}\{\text{server}=\sim\$\text{server}\}\{5\text{m}\}))$	
	$\text{sum}(\text{rate}(\text{pg_stat_database_conflicts_confl_lock}\{\text{server}=\sim\$\text{server}\}\{\$\text{interval}\}))$ or $\text{sum}(\text{irate}(\text{pg_stat_database_conflicts_confl_lock}\{\text{server}=\sim\$\text{server}\}\{5\text{m}\}))$	
	$\text{sum}(\text{rate}(\text{pg_stat_database_conflicts_confl_snapshot}\{\text{server}=\sim\$\text{server}\}\{\$\text{interval}\}))$ or $\text{sum}(\text{irate}(\text{pg_stat_database_conflicts_confl_snapshot}\{\text{server}=\sim\$\text{server}\}\{5\text{m}\}))$	
	$\text{sum}(\text{rate}(\text{pg_stat_database_conflicts_confl_tablespace}\{\text{server}=\sim\$\text{server}\}\{\$\text{interval}\}))$ or $\text{sum}(\text{irate}(\text{pg_stat_database_conflicts_confl_tablespace}\{\text{server}=\sim\$\text{server}\}\{5\text{m}\}))$	
Конфликты/Deadlocks	$\text{rate}(\text{pg_stat_database_conflicts}\{\text{server}=\$\text{server}\}\{5\text{m}\})$	
	$\text{rate}(\text{pg_stat_database_deadlocks}\{\text{server}=\$\text{server}\}\{5\text{m}\})$	
Репл. Конфликты/Deadlocks	$\text{sum}(\text{rate}(\text{pg_stat_database_deadlocks}\{\text{server}=\$\text{server}\}\{5\text{m}\}))$	
	$\text{sum}(\text{rate}(\text{pg_stat_database_conflicts}\{\text{server}=\$\text{server}\}\{5\text{m}\}))$	
Кэш и временные файлы		
PG частота попаданий в кэш	$\text{round}(100 * \text{sum}(\text{pg_stat_database_blks_hit}\{\text{server}=\$\text{server}\}) / (\text{sum}(\text{pg_stat_database_blks_hit}\{\text{server}=\$\text{server}\}) + \text{sum}(\text{pg_stat_database_blks_read}\{\text{server}=\$\text{server}\})), 0.1)$	
Временные файлы	$\text{irate}(\text{pg_stat_database_temp_bytes}\{\text{server}=\$\text{server}\}\{5\text{m}\})$	
Cache Hit Rate	$\text{pg_stat_database_blks_hit}\{\text{server}=\$\text{server}\} / (\text{pg_stat_database_blks_read}\{\text{server}=\$\text{server}\} + \text{pg_stat_database_blks_hit}\{\text{server}=\$\text{server}\})$	

МБД.П - Информация о кластере

Название панели	Формула	Пояснение
Работающие узлы	<code>count(ha_cluster_pacemaker_nodes{instance="\$dc_instance", status="online"} == 1) / count(count(ha_cluster_pacemaker_nodes{instance="\$dc_instance"}) by (node))</code>	
Узлы	<code>sum(ha_cluster_pacemaker_nodes{status="online", instance="\$dc_instance"}) by (node)</code>	
	<code>sum(ha_cluster_pacemaker_nodes{status="unclean", instance="\$dc_instance"}) by (node)</code>	
	<code>sum(ha_cluster_pacemaker_nodes{instance="\$dc_instance"}) by (type)</code>	
Активные ресурсы	<code>count(ha_cluster_pacemaker_resources{instance="\$dc_instance", status="active"} == 1) / count(count(ha_cluster_pacemaker_resources{instance="\$dc_instance"}) by (resource,node))</code>	
Кворум	<code>ha_cluster_corosync_quorum_votes{instance="\$dc_instance",type="expected_votes"}</code>	
	<code>ha_cluster_corosync_quorum_votes{instance="\$dc_instance",type="highest_expected"}</code>	
	<code>ha_cluster_corosync_quorum_votes{instance="\$dc_instance", type="total_votes"}</code>	
	<code>ha_cluster_corosync_quorum_votes{instance="\$dc_instance", type="quorum"}</code>	

Дефектные кольца Corosync	ha_cluster_corosync_ring_errors{instance="\$dc_instance"}	
Кворум кластера	ha_cluster_corosync_quorate{instance="\$dc_instance"}	
Последнее изменение CIB	ha_cluster_pacemaker_config_last_change{instance="\$dc_instance"} * 1000	
Статусы узлов по времени	sum(ha_cluster_pacemaker_nodes{instance="\$dc_instance"}) by (status)	
Статусы ресурсов по времени	sum(ha_cluster_pacemaker_resources{instance="\$dc_instance"}) by (status)	
Ресурсы кластера	sum(ha_cluster_pacemaker_resources{instance="\$dc_instance"} == 1) by(resource, node, status, role, agent)	
	sum(ha_cluster_pacemaker_migration_threshold{instance="\$dc_instance"}) by(resource, node)	
	sum(ha_cluster_pacemaker_fail_count{instance="\$dc_instance"}) by(resource, node)	
Свойства узлов	ha_cluster_pacemaker_node_attributes{instance="\$dc_instance"}	
Сервисы systemd	node_systemd_unit_state{name=~"((corosync pacemaker sbd prometheus hawk).*)", state="active"} + on(instance) group_left(nodename) 0 * node_uname_info{job="\$cluster"}	
Ограничения размещения ресурсов	clamp_min(clamp_max(ha_cluster_pacemaker_location_constraints{instance="\$dc_instance"}, 1000000), -1000000)	
Устройства SBD	count(ha_cluster_sbd_devices{instance="\$dc_instance"}) by (device,status)	

	count(ha_cluster_sbd_devices{instance="\$dc_instance",status="healthy"}) by (device,status)	

МБД.П - Обзор БД PostgreSQL

- Server - query_result(pg_settings_server_version_num)
- instance
- datname - label_values(pg_stat_activity_count{server=~"\$server"}, datname)
- mode

Название панели	Формула	Пояснение
Общие сведения, статистика использования CPU, памяти и файловых дескрипторов		
Версия	pg_static{server="\$server"}	
Время старта процесса	pg_postmaster_start_time_seconds{server="\$server"} * 1000	
Чтение данных текущее	SUM(pg_stat_database_tup_fetched{datname=~"\$datname", server=~"\$server"})	
Вставка данных текущая	SUM(pg_stat_database_tup_inserted{datname=~"\$datname", server=~"\$server"})	
Обновление данных текущее	SUM(pg_stat_database_tup_updated{datname=~"\$datname", server=~"\$server"})	
Максимум соединений	pg_settings_max_connections{server="\$server"}	

Среднее использование CPU (min, max, avg, current)	avg(rate(process_cpu_seconds_total{server="\$server"}[5m]) * 1000)	
Среднее использование памяти (min, max, avg, current)	avg(rate(process_resident_memory_bytes{server="\$server"}[5m]))	Resident Mem
	avg(rate(process_virtual_memory_bytes{server="\$server"}[5m]))	Virtual Mem
Открытые файловые дескрипторы (min, max, avg, current)	process_open_fds{server="\$server"}	
Настройки		
Разделяемые буферы	pg_settings_shared_buffers_bytes{server="\$server"}	
Эффективный кэш	pg_settings_effective_cache_size_bytes{server="\$server"}	
Рабочая память	pg_settings_work_mem_bytes{server="\$server"}	
Макс. размер WAL	pg_settings_max_wal_size_bytes{server="\$server"}	
Память обслуживания	pg_settings_maintenance_work_mem_bytes{server="\$server"}	
Random Page Cost	pg_settings_random_page_cost{server="\$server"}	
Seq Page Cost	pg_settings_seq_page_cost{server="\$server"}	
Макс. количество рабочих процессов	pg_settings_max_worker_processes{server="\$server"}	

Макс. количество параллельных операций	<code>pg_settings_max_parallel_workers{server="\$server"}</code>	
Статистика базы данных		
Статистика базы данных (avg, current, total)	<code>pg_stat_activity_count{datname=~"\$datname", server=~"\$server", state="active"} != 0</code>	
Транзакции (avg, current, total)	<code>irate(pg_stat_database_xact_commit{server="\$server", datname=~"\$datname"}[5m])</code>	
	<code>irate(pg_stat_database_xact_rollback{server="\$server", datname=~"\$datname"}[5m])</code>	
Обновление данных (avg, current, total)	<code>pg_stat_database_tup_updated{datname=~"\$datname", server=~"\$server"} != 0</code>	
Чтение данных (SELECT) (avg, current, total)	<code>pg_stat_database_tup_fetched{datname=~"\$datname", server=~"\$server"} != 0</code>	
Вставка данных (avg, current, total)	<code>pg_stat_database_tup_inserted{datname=~"\$datname", server=~"\$server"} != 0</code>	
Табличная блокировка (avg, current, total)	<code>pg_locks_count{datname=~"\$datname", server=~"\$server", mode=~"\$mode"} != 0</code>	
Возвращаемые данные (avg, current, total)	<code>pg_stat_database_tup_returned{datname=~"\$datname", server=~"\$server"} != 0</code>	
Бездействующие сеансы (max, current)	<code>pg_stat_activity_count{datname=~"\$datname", server=~"\$server", state=~"idle idle in transaction idle in transaction (aborted)"}</code>	

Удаление данных (avg, current, total)	<code>pg_stat_database_tup_deleted{datname=~"\$datname", server=~"\$server"} != 0</code>	
Cache Hit Rate (avg, current)	<code>pg_stat_database_blks_hit{server="\$server", datname=~"\$datname"} / (pg_stat_database_blks_read{server="\$server", datname=~"\$datname"} + pg_stat_database_blks_hit{server="\$server", datname=~"\$datname"})</code>	
Буферы (bgwriter) (min, max, avg, current)	<code>irate(pg_stat_bgwriter_buffers_backend{server="\$server"}[5m])</code>	
	<code>irate(pg_stat_bgwriter_buffers_alloc{server="\$server"}[5m])</code>	
	<code>irate(pg_stat_bgwriter_buffers_backend_fsync{server="\$server"}[5m])</code>	
	<code>irate(pg_stat_bgwriter_buffers_checkpoint{server="\$server"}[5m])</code>	
	<code>irate(pg_stat_bgwriter_buffers_clean{server="\$server"}[5m])</code>	
Конфликты/dead- lock (avg, current, total)	<code>irate(pg_stat_database_conflicts{server="\$server", datname=~"\$dat- name"}[5m])</code>	
	<code>irate(pg_stat_database_deadlocks{server="\$server", datname=~"\$dat- name"}[5m])</code>	
Временные файлы (Bytes) (avg, cur- rent, total)	<code>irate(pg_stat_database_temp_bytes{server="\$server", datname=~"\$dat- name"}[5m])</code>	
Статистика кон- трольных точек (min, max, avg, cur- rent)	<code>irate(pg_stat_bgwriter_checkpoint_write_time{server="\$server"}[5m])</code>	
	<code>irate(pg_stat_bgwriter_checkpoint_sync_time{server="\$server"}[5m])</code>	