
Ред Монитор
Версия 0.7
Руководство пользователя

Содержание

1	Общие сведения о программе	4
1.1	Назначение программы	4
1.2	Минимальный состав аппаратных средств	5
1.3	Минимальный состав программных средств	5
2	Установка и настройка Ред Монитор	6
2.1	Сервер СУБД	6
2.1.1	Ред База Данных	6
2.1.2	Экспортёр	6
	Установка экспортёра	6
	Настройка экспортёра	7
2.1.3	Promtail	12
2.2	Сервер мониторинга	12
2.2.1	Prometheus	12
	Установка Prometheus	12
	Основные настройки Prometheus	13
	Секция global	14
	Секция rule_files	14
	Секция scrape_configs	15
	Секция alerting	15
	Запуск Prometheus	15
2.2.2	Alertmanager	16
	Установка Alertmanager	16
	Основные настройки Alertmanager	17
	Секция global	17
	Секция route	18
	Секция receivers	18
	Секция inhibit_rules	18
	Включение уведомлений	19
	Запуск Alertmanager	19
2.2.3	Grafana	19
2.2.4	Ред База Данных	20
2.2.5	Loki	21
2.3	Установка на один сервер	21
3	Просмотр метрик	22
3.1	Просмотр метрик с помощью Prometheus	22
3.2	Просмотр метрик с помощью Grafana	23
3.3	Обзорная страница	24
3.4	Информация о выбранном сервере	24
3.4.1	СУБД	26
3.4.2	Базы данных	27
3.4.3	Информация о выбранной базе данных	27
	SQL запросы	29
	Соединения	31
	Таблица блокировок	33
	Лог файл	33
3.4.4	Процессы	34
	Обзор процесса	34
3.4.5	SQL-запросы	35

3.4.6 Экспортёр	36
3.4.7 Диски	37
Приложение А Собираемые метрики	38
А.1 Метрики таблиц мониторинга	38
А.2 Метрики агрегатного аудита	43
А.3 Метрики утилиты rdb_lock_print	46
А.4 Метрики операционной системы	48
А.5 Другие метрики	52

Глава 1

Общие сведения о программе

1.1 Назначение программы

Ред Монитор - это система для мониторинга СУБД Ред База Данных.

Ред Монитор предназначен для наблюдения за состоянием СУБД. Он предоставляет возможность мониторинга сразу нескольких баз и серверов. Показывает подробную информацию о пользователях, соединениях, запросах, ошибках и отображает топ соединений и запросов сервера по указанной характеристике за определённое время. Например, самые долгие запросы, выполненные за последний час. Помогает следить за изменением планов запросов. Показывает, какую нагрузку на сервер и операционную систему оказывает наблюдаемая СУБД.

Экспортёр, Ред База Данных, Prometheus, Alertmanager, Grafana, Loki и Promtail вместе образуют продукт Ред Монитор. Экспортёр используется для сбора показателей состояния (метрик) СУБД и отправки их в систему мониторинга Prometheus. Для управления уведомлениями используется Alertmanager. Для визуализации собранных метрик используется Grafana.

Подробное описание собираемых метрик см. в приложении *Собираемые метрики*.

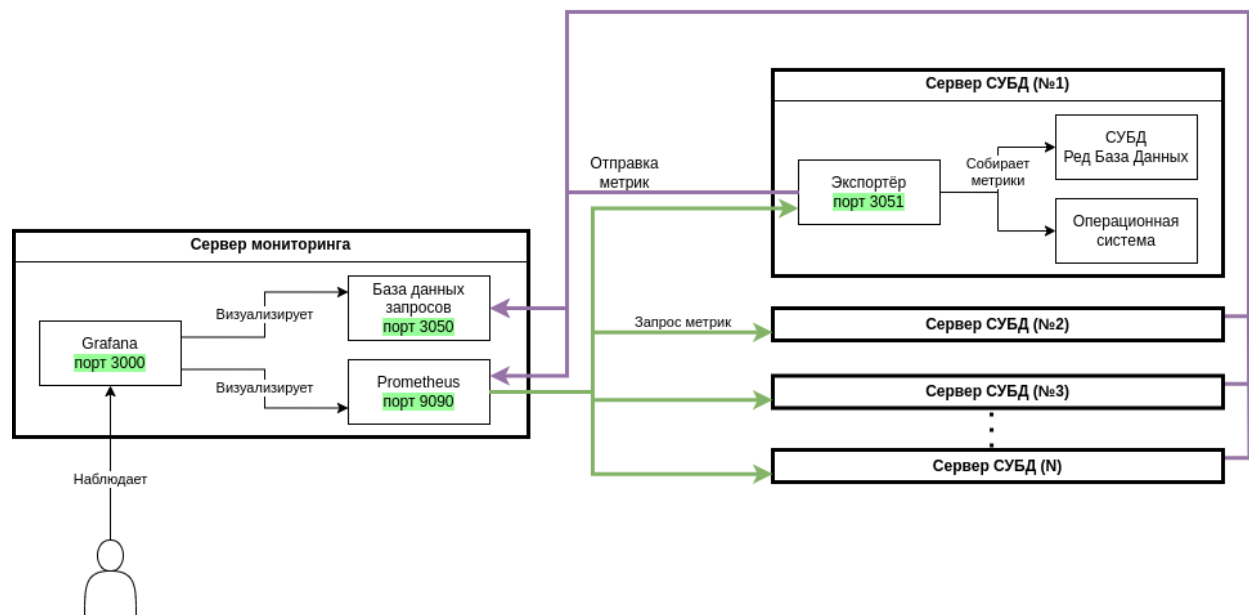


Рисунок 1.1 — Схема взаимодействия компонентов Ред Монитора

- Сервер мониторинга - компьютер, на котором работают программы Prometheus, СУБД Ред База Данных и Grafana.
- Сервер СУБД - компьютер, на котором работает Наблюдаемая СУБД и Экспортёр.

Сервер мониторинга должен иметь возможность выполнения запроса к серверам СУБД. Сервер мониторинга выполняет функцию хранения и демонстрации данных, собранных с серверов СУБД. Prometheus отправляет запросы на экспортёры серверов СУБД. Экспортёры, получив запрос, собирают метрики с Наблюдаемой СУБД и отправляют данные обратно в Prometheus. Некоторые данные (например, текст SQL-запроса) отправляются на хранение в базу данных пользовательских запросов.

Для отображения данных сервер мониторинга использует Grafana. Она получает данные из

Prometheus и Базы данных пользовательских запросов.

Порты, используемые по умолчанию:

- Сервер мониторинга:
 - Grafana - порт 3000;
 - Prometheus - порт 9090;
 - Loki - порт 3100;
 - СУБД с базой данных пользовательских запросов - порт 3050.
- Сервер СУБД:
 - Экспортёр - порт 3051.

Сервер мониторинга и сервер СУБД можно установить как на одну машину, так и на разные.

1.2 Минимальный состав аппаратных средств

- Оперативная память от 16Гб;
- Процессор не менее 4х ядер;
- Запоминающее устройство объёмом не менее 64Гб.

1.3 Минимальный состав программных средств

Перед началом работы с Ред Монитор необходимо:

- Установить СУБД Ред База Данных версии не ниже 3.0. Узнать об этом подробнее можно в [Руководстве администратора](#);
- Установить и настроить Prometheus версии не ниже 2.37.5;
- Установить и настроить Grafana версии не ниже 9.3.2;
- Установить Loki версии не ниже 3.3.1;
- Установить Promtail версии не ниже 3.3.1.

Глава 2

Установка и настройка Ред Монитор

2.1 Сервер СУБД

Все элементы сервера СУБД, а именно Ред База Данных (наблюдаемая СУБД), Экспортёр и Promtail должны быть установлены на одну машину.

2.1.1 Ред База Данных

На Наблюдаемой СУБД для сбора метрик пользовательских запросов необходимо:

1. В `firebird.conf` настроить параметры `TracePlugin` и `AuditTraceConfigFiles`:

```
TracePlugin = aggtrace
AuditTraceConfigFiles = fbtrace.conf
```

2. Включить агрегатный аудит, настроив `fbtrace.conf`:

```
database
{
  enabled = true
  format = 3
  reset_counters = false
}
```

Если в `firebird.conf` уже включен какой-либо трейс, то для агрегатного аудита нужно создать отдельный файл конфигурации (например, `aggtrace.conf`) в каталоге установки Ред База Данных и указать его в `AuditTraceConfigFiles`:

```
TracePlugin = fbtrace, aggtrace
AuditTraceConfigFiles = fbtrace.conf; aggtrace.conf
```

2.1.2 Экспортёр

Установка экспортёра

Экспортёр необходимо скачать по [ссылке](#). После скачивания распакуйте архив, выполнив команду:

```
tar -xzf <архив>
```

Перейдите в распакованную папку:

```
cd <папка>
```

Установите Экспортёр, выполнив `install.sh` с правами администратора:

```
sudo ./install.sh -o <install | update> -p <client | grafana_plugins [-d] | loki>
```

- `-o` указывает, какую операцию нужно выполнить: установку или обновление.
- `-p` определяет, что нужно установить/обновить. Опция `client` используется для установки/обновления всего, что необходимо для сервера СУБД (экспортёр и `Promtail`). На сервер мониторинга необходимо установить плагины для `Grafana` и `Loki`, используя соответствующие опции. При установке/обновлении плагинов для `Grafana` нужно будет указать путь до конфигурационного файла (`grafana.ini`) и путь до папки с плагинами `Grafana`. Если используются пути по умолчанию, то можно использовать опцию `-d`.
- `-d` указывает использовать пути по умолчанию при установке/обновлении плагинов для `Grafana`. Конфигурационный файл `Grafana` по умолчанию располагается по пути `/etc/grafana/grafana.ini`, а папка с плагинами `Grafana` по пути `/var/lib/grafana/plugins/`.
- `-h` выводит справку о доступных опциях.

Экспортёр будет установлен на сервер СУБД по пути `/opt/RedMonitor`.

Для удаления Ред Монитора выполните команду:

```
sudo ./uninstall.sh [<опции>]
```

```
<опции>:=  
  client  
  | grafana_plugins  
  | loki
```

Настройка экспортёра

Экспортёр должен находиться на том же устройстве, где и наблюдаемая СУБД. Для настройки параметров экспортёра используется файл `exporter_conf.json`, расположенный в `/opt/RedMonitor/exporter`.

Структура файла `exporter_conf.json`:

```
{  
  "exporter": {  
    "host": "<хост экспортёра>",  
    "port": <порт экспортёра>,  
    "rdb_lock_print": {  
      "enabled": true  
    },  
    "scrape_system": {  
      "enabled": false  
    },  
    "scrape_mon_tables": {  
      "enabled": true  
    },  
    "scrape_log": {  
      "enabled": true,  
      "start_from_end": true  
    },  
    "scrape_atrace": {  
      "enabled": true  
    }  
  }  
}
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
},
  "dictserver": {
    "host": "<хост базы данных>",
    "port": <порт базы данных>,
    "database": "/db/statements.fdb",
    "user": "<пользователь>",
    "password": "<пароль>",
    "Auth_plugins": ["<плагин>", ..., "<плагин>"]
  },
  "RedDatabase": {
    "host": "<хост СУБД>",
    "port": <порт СУБД>,
    "user": "<логин>",
    "password": "<пароль>",
    "folder_path": "<путь к папке установки RedDatabase>",
    "Auth_plugins": ["<плагин>", ..., "<плагин>"],
    "loki": {
      "port": <порт loki>
    }
  },
  "databases": {
    "<псевдоним>": {
      "alias": "<алиас>",
      "path": "<путь к базе данных>"
    }
  }
}
```

Для каждого параметра должно быть указано значение. Значения строковых параметров должны заключаться в двойные кавычки. Значения для целочисленных параметров нужно указывать без кавычек.

Параметры конфигурации:

В секции `exporter` обязательно должны быть указаны все параметры, иначе экспортёр не запустится. Параметры секции:

- `host` - определяет ip-адрес устройства, на котором работает экспортёр; параметр имеет строковый тип;
- `port` - порт, на котором работает экспортёр, по умолчанию 3051; параметр имеет целочисленный тип;
- `rdb_lock_print` - определяет, собирать ли данные файла блокировок. Аналогично `rdb_lock_print -n -l -o -c`. По умолчанию установлено значение `true`;
- `scrape_system` - определяет, собирать ли информацию об операционной системе. По умолчанию установлено значение `true`;
- `scrape_mon_tables` - определяет, собирать ли данные из таблиц мониторинга. По умолчанию установлено значение `true`;
- `scrape_log` - определяет, собирать ли данные из лог-файла. Опция `start_from_end` определяет порядок чтения файла. Если установлено значение `true`, то чтение будет начато с конца лог-файла, то есть будут прочитаны только новые записи. Если установлено значение `false`, то с начала лог-файла будут прочитаны все записи;
- `scrape_atrace` - определяет, собирать ли данные с помощью агрегирующего трейса. По умолчанию установлено значение `false`.

Секция `dictserver`:

- `host` - определяет ip-адрес устройства, на котором работает база данных для пользовательских запросов; параметр имеет строковый тип;
- `port` - порт, на котором работает база данных для пользовательских запросов, по умолчанию 3050; параметр имеет целочисленный тип;
- `database` - псевдоним базы данных пользовательских запросов;
- `user` - имя пользователя, от которого экспортёр будет подключаться к базе данных пользовательских запросов; параметр имеет строковый тип;
- `password` - пароль пользователя для подключения к базе данных пользовательских запросов; параметр имеет строковый тип;
- `Auth_plugins` - определяет список плагинов, которые используются для аутентификации при подключении к базе данных пользовательских запросов, в качестве разделителя используется запятая, каждый плагин должен заключаться в двойные кавычки; список заключается в квадратные скобки;
- `loki` - определяет порт, на котором работает Loki. По умолчанию 3100.

Секция `RedDatabase`:

- `host` - определяет ip-адрес устройства, на котором работает наблюдаемая СУБД; параметр имеет строковый тип;
- `port` - порт, на котором работает наблюдаемая СУБД; параметр имеет целочисленный тип;
- `login` - имя пользователя, от которого экспортёр будет подключаться к СУБД. Желательно использовать пользователя с административными привилегиями для мониторинга всех подключений; параметр имеет строковый тип;
- `password` - пароль пользователя; параметр имеет строковый тип;
- `folder_path` - путь к установочной папке `RedDatabase`;
- `Auth_plugins` - определяет список плагинов, которые экспортёр использует для аутентификации, в качестве разделителя используется запятая, каждый плагин должен заключаться в двойные кавычки; список заключается в квадратные скобки;

Секция `databases` определяет список баз данных для мониторинга, указывающийся следующим образом:

```
"databases": {
  "<псевдоним>": {
    "alias": "<алиас>",
    "path": "<путь к базе данных>"
  },
  ...
}
```

Где:

- `псевдоним` - уникальный псевдоним базы данных;
- `path` - путь до базы данных;
- `alias` - псевдоним базы данных, который используется СУБД. Если псевдонима у базы данных нет, то нужно указать значение `null`.

Пример настроенного файла `exporter_conf.json`:

```
{  
  "exporter": {  
    "host": "127.0.0.1",  
    "port": 3051,  
  }  
}
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
"rdb_lock_print": {
  "enabled": true
},
"scrape_system":{
  "enabled": false
},
"scrape_mon_tables":{
  "enabled": true
},
"scrape_log": {
  "enabled": true,
  "start_from_end": true
},
"scrape_atrace": {
  "enabled": false
}
},
"dictserver": {
  "host": "172.17.0.2",
  "port": 3050,
  "database": "/db/statements.fdb",
  "user": "SYSDBA",
  "password": "masterkey",
  "Auth_plugins": ["Srp"],
  "loki": {
    "port": 3100
  }
},
"RedDatabase": {
  "host": "127.0.0.1",
  "port": 3050,
  "user": "SYSDBA",
  "password": "masterkey",
  "folder_path": "/opt/RedDatabase",
  "Auth_plugins": ["Srp"]
},
"databases": {
  "employee_db": {
    "alias": "employee",
    "path": "/opt/RedDatabase/examples/empbuild/employee.fdb"
  }
}
}
```

Перед запуском экспортёра убедитесь, что сервер СУБД запущен.

Запуск и остановка утилиты осуществляется с помощью `systemd`:

```
systemctl start redmonitor-exporter.service
systemctl stop redmonitor-exporter.service
```

При запуске через `systemd` используется файл конфигурации по умолчанию, расположенный по

следующему пути: `/opt/RedMonitor/exporter/exporter_conf.json`. Чтобы использовать другой файл конфигурации (созданный самостоятельно), нужно изменить путь до файла конфигурации экспортёра, отредактировав файл `redmonitor.sh`, расположенный в `/usr/local/bin`. В этом файле нужно изменить значение переменной `$configure` на другой путь до файла конфигурации.

Также можно запустить бинарный файл. Для этого нужно выполнить в терминале:

```
./exporter <полный путь до файла конфигурации>
```

2.1.3 Promtail

Promtail - это инструмент, необходимый для сбора логов СУБД и операционной системы.

Promtail устанавливается на сервер СУБД вместе с экспортёром командой:

```
sudo ./install.sh -o install -p client
```

Запустите Promtail:

```
sudo systemctl start redmonitor-promtail.service
```

Убедитесь, что Экспортёр запущен. Важно запустить Promtail после Экспортёра.

2.2 Сервер мониторинга

В этом разделе находится описание настроек программ, которые должны быть установлены на сервере мониторинга.

Все элементы сервера мониторинга, а именно Prometheus, Alertmanager (опционально), Grafana, Ред База Данных и Loki должны быть установлены на одну машину или иметь между собой связь по сети.

2.2.1 Prometheus

Установка Prometheus

Установить Prometheus в РЕД ОС можно с помощью пакетного менеджера операционной системы:

Для установки Prometheus выполните команду:

```
dnf install golang-github-prometheus
```

Другой вариант установки - скачать Prometheus с официального сайта (prometheus.io) и выполнить следующее:

1. Распаковать скачанный архив:

```
tar -xf <путь_к_архиву>
```

2. Переместить распакованный архив в предварительно созданную папку:

```
sudo mv -Z ./<распакованный архив> /opt/prometheus
```

3. Создать пользователя и группу `prometheus`:

```
sudo useradd -M -U prometheus
```

4. Назначить права на папку `/opt/prometheus`:

```
sudo chown prometheus:prometheus -R /opt/prometheus
```

5. Создать `unit`-файл для запуска через службу:

```
[Unit]
Description=Prometheus Server
Documentation=https://prometheus.io/docs/introduction/overview/
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Restart=on-failure
ExecStart=/opt/prometheus/prometheus \
--config.file=/opt/prometheus/prometheus.yml \
--storage.tsdb.path=/opt/prometheus/data \
--storage.tsdb.retention.time=30d

[Install]
WantedBy=multi-user.target
```

После создания `unit`-файла выполнить:

```
sudo systemctl daemon-reload
```

Основные настройки Prometheus

`Prometheus` собирает и хранит метрики СУБД. Но обращаться к наблюдаемой базе напрямую он не может, поэтому для передачи данных от СУБД к `Prometheus` используется Экспортёр.

Для получения уведомлений о состоянии наблюдаемой базы нужно настроить правила проверки значений собранных метрик в секции `rule_files`. `Prometheus` проверяет собранные метрики на соблюдение указанным правилам с заданной периодичностью. Если правило соблюдается (например, правило для проверки загрузки процессора `rate(rdb_system_cpu_times[1m]) * 100 > 20`), то `Prometheus` сообщит об этом `Alertmanager`, отправив предупреждение. `Alertmanager` сформирует из предупреждений уведомления и отправит их на указанный адрес.

Для настройки `Prometheus` используется файл `prometheus.yml`.

```
global:
  scrape_interval: 20s
  scrape_timeout: 15s
  evaluation_interval: 30s
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
scrape_configs:
  - job_name: "RedDatabase"
    static_configs:
      - targets: ["192.168.0.100:3051", "192.168.0.100:8100", "192.168.0.150:8000"]

rule_files:
  - rules.yml

# alerting:
# alertmanagers:
#   - static_configs:
#     - targets: ["localhost:9093"]
```

Файл `prometheus.yml` разделён на четыре основные секции: `global`, `rule_files`, `scrape_configs`, `alerting`.

Секция `global`

Настройки, заданные в секции `global` по умолчанию распространяются на весь файл.

```
global:
  scrape_interval: 10s
  scrape_timeout: 10s
  evaluation_interval: 30s
```

Параметр `scrape_interval` определяет, как часто нужно собирать метрики.

Параметр `scrape_timeout` устанавливает время ожидания получения метрик.

Параметр `evaluation_interval` устанавливает интервал, с которым собранные метрики будут проверяться на соответствие правилам, указанным в секции `rule_files`.

Секция `rule_files`

Файл `rules.yml` необходимо предварительно скопировать из корневого каталога `RedMonitor` в корневой каталог `Prometheus`.

В секции `rule_files` указывается список файлов с правилами, на соответствие которым нужно проверить собранные метрики:

```
rule_files:
  - rules.yml
  - "side/*_rules.yml"
```

Секция `scrape_configs`

В секции `scrape_configs` задаются настройки для сбора метрик:

```
scrape_configs:
- job_name: "RedMonitor"
  scrape_interval: 15s
  scrape_timeout: 10s
  static_configs:
    - targets: ["192.168.0.100:3051", "192.168.0.100:8100", "192.168.0.150:8000"]
```

Параметр `job_name` задаёт уникальное имя экспортёра. Можно указать несколько экспортёров.

Параметр `scrape_interval` определяет, как часто должны собираться метрики. По умолчанию принимает значение, установленное в секции `global`.

Параметр `scrape_timeout` устанавливает время ожидания получения метрик. По умолчанию принимает значение, установленное в секции `global`.

Параметр `targets` определяет сетевые узлы экспортёров, к которым будет обращаться Prometheus, чтобы получить метрики. Портом экспортёра по умолчанию является 3051.

Секция `alerting`

Секция `alerting` определяет сущности `Alertmanager`, в которые Prometheus будет отправлять предупреждения, когда собранные метрики соблюдают правила, указанные в секции `rule_files`.

Если нет необходимости в уведомлениях, то эту секцию нужно оставить закомментированной.

```
alerting:
  alertmanagers:
    - static_configs:
      - targets: ["localhost:9093"]
```

Запуск Prometheus

Запуск Prometheus осуществляется следующей командой:

```
sudo -u prometheus ./prometheus --config.file="prometheus.yml"
```

Или можно запустить службу:

```
sudo systemctl start prometheus
```

Более подробно о настройке Prometheus можно узнать на официальном сайте – prometheus.io.

2.2.2 Alertmanager

Если уведомления не нужны, то этот пункт можно пропустить.

Установка Alertmanager

Скачать Alertmanager можно с официального сайта -- prometheus.io.

Для установки нужно выполнить следующие действия:

1. Распаковать скачанный архив:

```
tar -xf <путь_к_архиву>
```

2. Переместить распакованный архив в предварительно созданную папку:

```
sudo mv -Z ./<распакованный архив> /opt/alertmanager
```

3. Создать пользователя и группу alertmanager:

```
sudo useradd -M -U alertmanager
```

4. Назначить права на папку /opt/alertmanager:

```
sudo chown alertmanager:alertmanager -R /opt/alertmanager
```

5. Создать unit-файл для запуска через службу:

```
[Unit]
Description=Prometheus Alertmanager
Documentation=https://prometheus.io/docs/alerting/latest/overview/
After=network-online.target

[Service]
User=alertmanager
Group=alertmanager
Restart=on-failure
ExecStart=/opt/alertmanager/alertmanager \
--config.file=/opt/alertmanager/alertmanager.yml \
--storage.path=/opt/alertmanager/data

[Install]
WantedBy=multi-user.target
```

После создания unit-файла выполнить:

```
sudo systemctl daemon-reload
```


Основные настройки Alertmanager

Чтобы получать уведомления, необходимо настроить Alertmanager. Для этого используется файл `alertmanager.yml`.

Пример конфигурации Alertmanager:

```
global:
  smtp_require_tls: true
  smtp_from: 'sender@example.com'
  smtp_smarthost: 'smtp.example.com:587'
  smtp_auth_username: 'user_name'
  smtp_auth_password: 'user_password'

route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 1h
  receiver: 'receiver_name'

receivers:
- name: 'receiver_name'
  email_configs:
  - to: 'receiver@example.com'

inhibit_rules:
- source_match:
  severity: 'critical'
  target_match:
  severity: 'warning'
```

Файл делится на следующие основные секции: `global`, `route`, `receivers`, `inhibit_rules`.

Секция `global`

В секции `global` указывается, куда отправлять данные. Пример настройки секции для отправки уведомлений на почту:

```
global:
  smtp_require_tls: true
  smtp_from: 'sender@example.com'
  smtp_smarthost: 'smtp.example.com:587'
  smtp_auth_username: 'user_name'
  smtp_auth_password: 'user_password'
```

Параметр `smtp_require_tls` указывает, использовать ли протокол TLS.

Параметр `smtp_smarthost` устанавливает адрес сервера почты.

Параметр `smtp_from` указывает почту отправителя.

Параметр `smtp_auth_username` определяет имя пользователя для аутентификации на сервере почты.

Параметр `smtp_auth_password` определяет пароль пользователя для аутентификации на сервере почты.

Секция `route`

Секция `route` определяет маршруты уведомлений в виде структуры дерева. Маршрут - это список проверок, через которые проходит уведомление для нахождения получателя.

```
route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 1h
  receiver: 'receiver_name'
```

Параметр `group_by` определяет список меток, по которым происходит группировка входящих уведомлений.

Параметр `group_wait` устанавливает время ожидания перед отправкой новой группы уведомлений.

Параметр `group_interval` задаёт время между отправкой уведомлений по группам.

Параметр `repeat_interval` задаёт время между повторной отправкой уведомлений.

Параметр `receiver` определяет получателя.

Секция `receivers`

В секции `receivers` указывается список получателей уведомлений.

```
receivers:
- name: 'receiver_name'
  email_configs:
  - to: 'receiver@example.com'
```

Параметр `name` указывает имя получателя.

Параметр `to` определяет почту получателя.

Секция `inhibit_rules`

Секция `inhibit_rules` устанавливает правила, по которым уведомления будут отключаться.

```
inhibit_rules:
- source_match:
  severity: 'critical'
  target_match:
  severity: 'warning'
```

Параметры `source_matchers` и `target_matchers` это списки меток со значениями. Те предупреждения, значения меток которых совпадают с метками из параметра `source_matchers`, блокируют отправку уведомления для тех предупреждений, значения меток которых совпадают с метками из параметра `target_matchers`.

Включение уведомлений

После настройки `alertmanager.yml` нужно указать в настройках Prometheus адрес, по которому доступен Alertmanager:

```
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        - localhost:9093
```

Далее нужно добавить правила, по которым будут приходить уведомления. Можно импортировать готовый набор правил уведомлений, поставляемый вместе с Ред Монитор. Для этого нужно указать в настройках Prometheus путь к файлу с готовыми правилами:

```
rule_files:
  - rules.yml
```

Файл `rules.yml` необходимо предварительно скопировать из корневого каталога RedMonitor в корневой каталог Prometheus. Более подробно о настройке Alertmanager можно узнать в [официальной документации](#).

Запуск Alertmanager

Запуск Alertmanager осуществляется следующей командой:

```
sudo -u alertmanager ./alertmanager --config.file="alertmanager.yml"
```

Или можно запустить службу Alertmanager:

```
sudo systemctl start alertmanager
```

2.2.3 Grafana

На Ред ОС Grafana можно установить через пакетный менеджер:

```
sudo dnf install grafana
```

На других операционных системах нужно скачать Grafana с [официального сайта](#) и установить по [инструкции](#)

Перенесите ранее скаченный архив Экспортёра на сервер мониторинга и установите плагины командой:

```
sudo ./install.sh -o install -p grafana_plugins
```

При установке плагинов для Grafana нужно внести `redsoft-redmonitor-app` и `redsoft-rdbconnector-datasource` в список доверенных, разрешив редактировать файл `grafana.ini`:

```
Установка плагинов для grafana.
Введите путь до папки плагинов: /var/lib/grafana/plugins
Копирование файлов...
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

Успешно!

Введите путь до файла конфигурации Grafana: `/etc/grafana/grafana.ini`
Включение плагинов в список доверенных...
Успешно!

Плагины Grafana установлены!

Копирование скрипта удаления...
Успешно!

В случае успешной установки в файле конфигурации Grafana (по умолчанию `grafana.ini`) будут указаны плагины `redsoft-redmonitor-app` и `redsoft-rdbconnector-datasource`:

```
allow_loading_unsigned_plugins = redsoft-redmonitor-app,redsoft-rdbconnector-  
datasource, <другие плагины>
```

Плагины для Grafana будут установлены на сервер мониторинга.
Запуск Grafana выполняется командой:

```
sudo systemctl start grafana-server
```

Для настройки откройте в браузере страницу запущенной Grafana (по умолчанию <http://localhost:3000/>). По умолчанию для входа используется логин `admin` и пароль `admin`. Чтобы установить источник данных, перейдите в настройки. Во вкладке `Data sources` нажмите на кнопку `Add data source`. В открывшемся окне из списка источников выберите `Prometheus`. Укажите URL-адрес для доступа к `Prometheus` (по умолчанию `http://localhost:9090/`) и нажмите на кнопку `Save & test`.

Также в качестве источника нужно установить базу данных, в которой хранятся пользовательские запросы. Для этого во вкладке `Data sources` нажмите на кнопку `Add new data source`. В открывшемся окне из списка источников выберите `Rdbconnector`. Заполните поля для соединения с базой данных и нажмите на кнопку `Save & test`.

Для хранения логов операционной системы и СУБД также добавьте `Loki` в качестве источника данных. Во вкладке `Data sources` нажмите на кнопку `Add data source`. В открывшемся окне из списка источников выберите `Loki`. Укажите URL-адрес для доступа к `Loki` (по умолчанию `http://127.0.0.1:3100`) и нажмите на кнопку `Save & test`.

2.2.4 Ред База Данных

Базу данных для хранения пользовательских запросов нужно создать самостоятельно на Сервере мониторинга, то есть там, где запущены `Prometheus`, `Grafana` и СУБД. Для этого нужно выполнить скрипт `create_tables.sql`, указав в нём пользователя, от имени которого будет создана база данных и каталог, где она будет расположена. По умолчанию база будет создана по пути `/db/statements.fdb`, каталог `db` необходимо самостоятельно создать заранее.

Пример настройки скрипта:

```
CREATE DATABASE 'localhost:<путь_до_бд>'  
  USER '<имя пользователя>' PASSWORD '<пароль>';  
...
```

Запуск скрипта:

```
<каталог Ред Базы Данных>/bin/isql -i ./dists/dictserver/create_tables.sql
```

По умолчанию доступ к СУБД с базой данных пользовательских запросов осуществляется по порту 3050.

2.2.5 Loki

Loki - это инструмент, необходимый для хранения логов, собираемых с помощью экспортёра логов (Promtail).

Loki необходимо установить на сервер мониторинга, выполнив команду:

```
sudo ./install.sh -o install -p loki
```

Запустите Loki:

```
sudo systemctl start redmonitor-loki.service
```

2.3 Установка на один сервер

При установке всех компонентов Ред Монитора на один сервер необходимо указать в конфигурационном файле аудита наблюдаемой СУБД следующие настройки(`fbtrace.conf`):

```
database = <путь или alias к базе данных>
{
    enabled = true
    format = 3
    reset_counters = true
}
```

Такие настройки нужно произвести для каждой базы данных, которая указана в секции `databases` файла конфигурации экспортёра.

Глава 3

Просмотр метрик

Собираемые метрики описаны в приложении *Собираемые метрики*.

3.1 Просмотр метрик с помощью Prometheus

Для просмотра метрик с помощью Prometheus нужно открыть в браузере страницу запущенного Prometheus (по умолчанию <http://localhost:9090/>). Можете запустить Prometheus с другим портом:

```
sudo ./prometheus --web.listen-address=":8080"
```

По умолчанию для просмотра метрик используется нулевой часовой пояс, чтобы использовать локальное время укажите флаг `Use local time`:

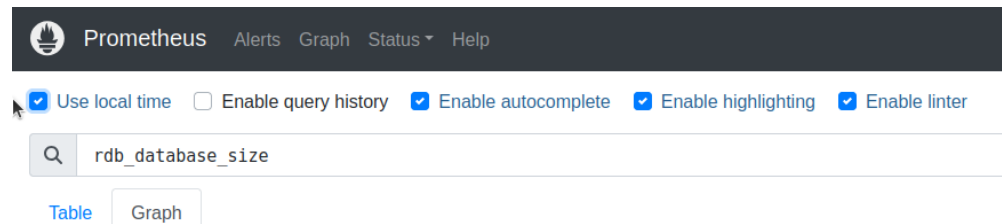


Рисунок 3.1 — Использование локального времени

Чтобы узнать значение конкретной метрики, нужно ввести PromQL запрос.

Примеры PromQL запросов:

1. Вывод объёма базы данных:

```
rdb_database_size{database="employee_db", instance="127.0.0.1:3051"}
```

2. Вывод скорости записи на диск в секунду:

```
irate(rdb_disks_io_bytes{instance="127.0.0.1:3051", database="employee_db", disk="sda1"}[1m])
```

3. Вывод разницы между Next transaction и Oldest interesting transaction:

```
rdb_transactions_markers{instance="127.0.0.1:3051", database="employee_db", marker="NT"} - ignoring(marker)
rdb_transactions_markers{instance="127.0.0.1:3051", database="employee_db", marker="OIT"}
```

Средства визуализации Prometheus довольно скудны, поэтому для просмотра метрик лучше использовать Grafana.

3.2 Просмотр метрик с помощью Grafana

Для просмотра метрик с помощью Grafana откройте в браузере страницу запущенной Grafana (по умолчанию <http://localhost:3000/>).

Для отображения собранной информации необходимо включить плагин RedMonitor:

1. Выберите Home → Administration → Plugins;
2. В открывшейся вкладке перейдите в Type → Applications → RedMonitor;
3. Нажмите на кнопку Enable.

После этого в меню Apps будет добавлен плагин RedMonitor:

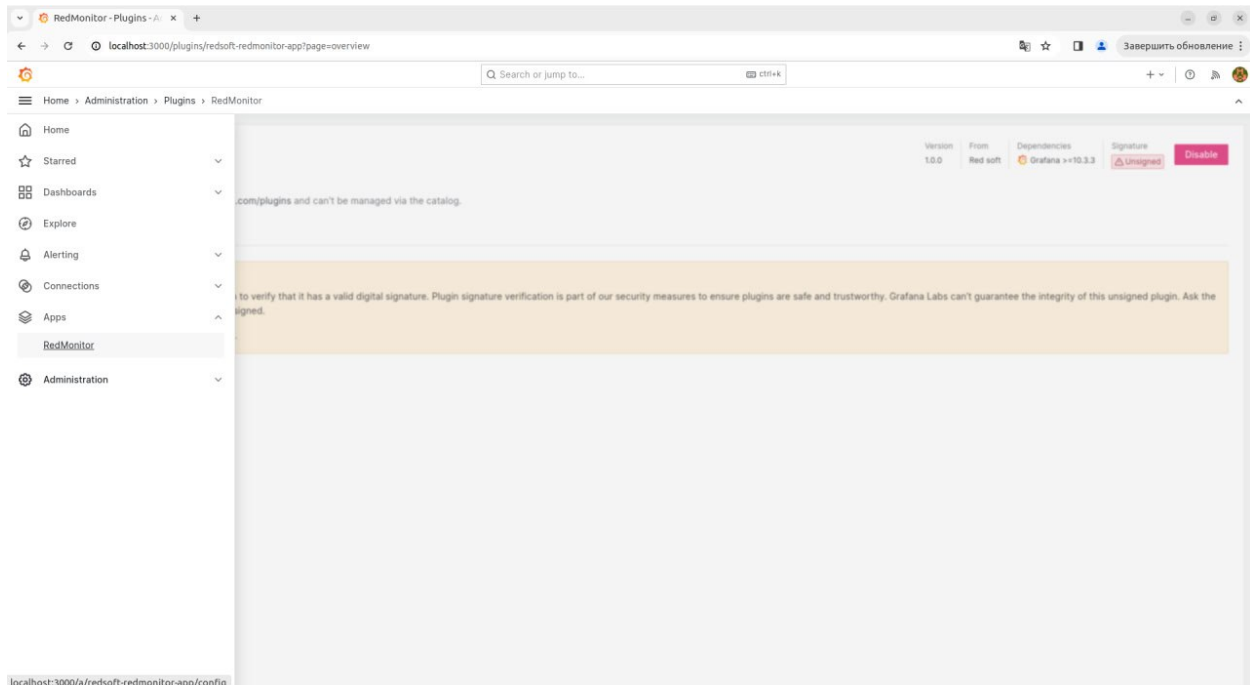


Рисунок 3.2 — Плагин Ред Монитор

Для настройки плагина нужно перейти на его страницу и выбрать источники данных для Prometheus и Базы данных пользовательских запросов.

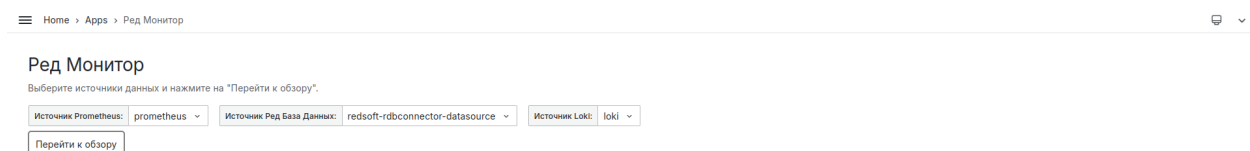


Рисунок 3.3 — Настройка плагина Ред Монитор

После нажатия на кнопку **Перейти к обзору** откроется Обзорная страница.

3.3 Обзорная страница

На этой странице отображается краткая информация о каждом сервере. Для получения подробной информации о конкретном сервере нужно нажать на его адрес.

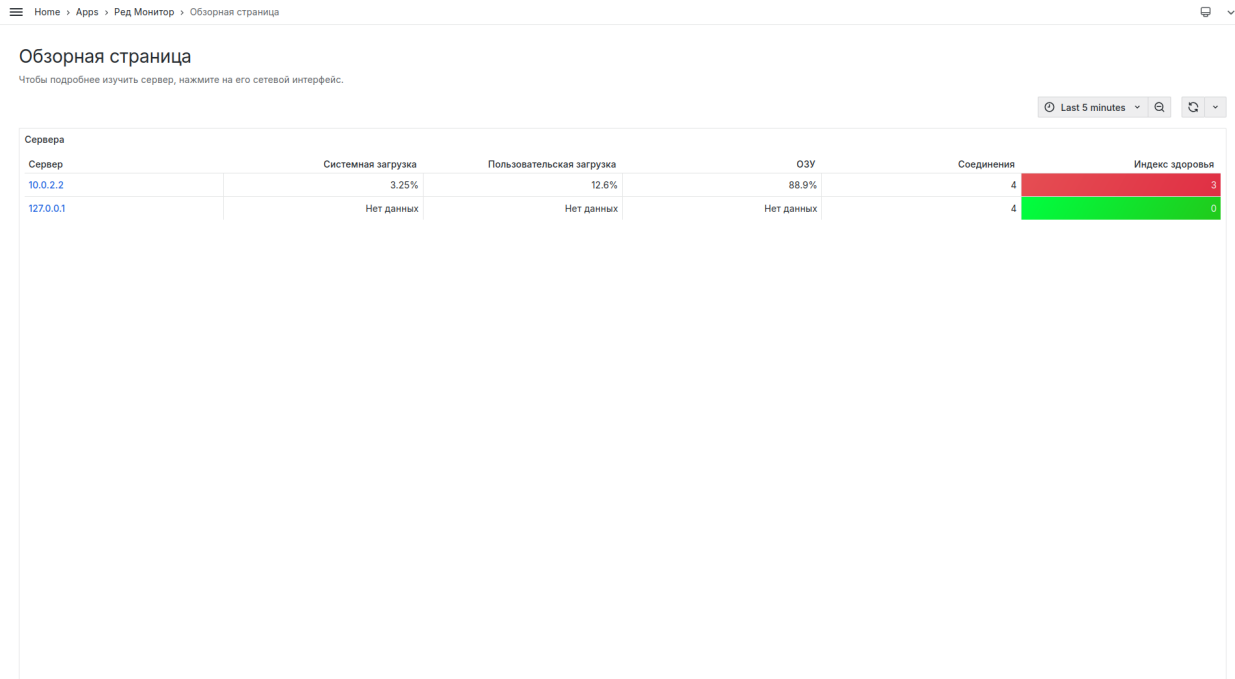


Рисунок 3.4 — Обзорная страница

3.4 Информация о выбранном сервере

На этой странице находится подробная информация о выбранном сервере.



Рисунок 3.5 — Обзор выбранного сервера - Часть 1



Рисунок 3.6 — Обзор выбранного сервера - Часть 2

График Загрузка ЦПУ показывает пользовательскую и системную нагрузку на процессор в зависимости от времени.

График Использование памяти показывает использование оперативной памяти в зависимости от времени.

График Loadavg отображает среднее количество готовых к выполнению процессов в системе за определённое время: 1 минута, 5 минут и 15 минут.

График Насыщение памяти показывает использование SWAP в зависимости от времени.

Панель MCC и Machine Check Exceptions отображает количество ошибок в логе системы с сообщением, содержащим MCC или Machine Check Exceptions.

Панель Error-correction code отображает количество ошибок в логе системы с сообщением, содержащим ECC или Error-correction code.

График Использование дисков показывает какой процент времени занимают операции чтения с дисков и записи на них в зависимости от времени.

График Насыщение дисков показывает среднюю длину очереди для операций чтения с дисков и записи на них в зависимости от времени.

График Использование пространства дисков отображает среднюю заполненность файловых систем, находящихся на диске, в зависимости от времени.

График Использование сети показывает скорость отправки/получения пакетов (количество пакетов в секунду) в зависимости от времени.

График Насыщение сети показывает скорость потери пакетов (количество пакетов в секунду) в зависимости от времени.

График Ошибки сети показывает скорость возникновения ошибок (количество ошибок в секунду) отправки/получения пакетов в зависимости от времени.

3.4.1 СУБД

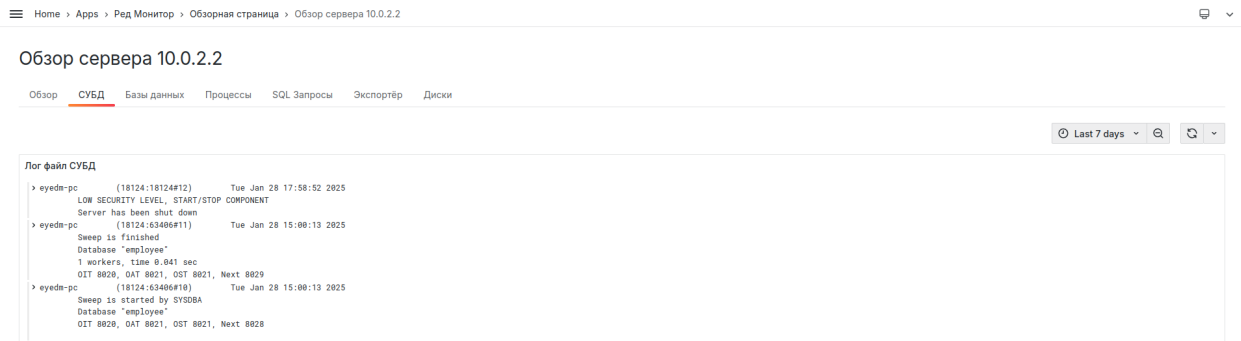


Рисунок 3.7 — Лог СУБД

Панель Лог файл СУБД показывает содержание файла firebird.log.



Рисунок 3.8 — Общая информация о СУБД

График **Скорость подготовки запросов** показывает скорость возникновения событий PREPARE_STATEMENT (количество событий в секунду) в зависимости от времени.

График **Время обработки запросов** отображает общее время выполнения запросов.

График **Средняя скорость возникновения ошибок выполнения запросов** показывает скорость возникновения ошибок при выполнении запроса.

График **Средняя скорость завершения транзакций** показывает скорость возникновения событий, связанных с завершением транзакции (COMMIT, ROLLBACK, COMMIT_RETAINING, ROLLBACK_RETAINING), в зависимости от времени.

График **Время обработки событий завершения транзакций** отображает время, потраченное на выполнение событий, связанных с завершением транзакции.

График **Средняя скорость возникновения ошибок выполнения транзакций** показывает скорость возникновения ошибок при завершении транзакции.

3.4.2 Базы данных

На этой странице отображается краткая информация о каждой базе данных. Для получения подробной информации о конкретной базе нужно нажать на её имя.

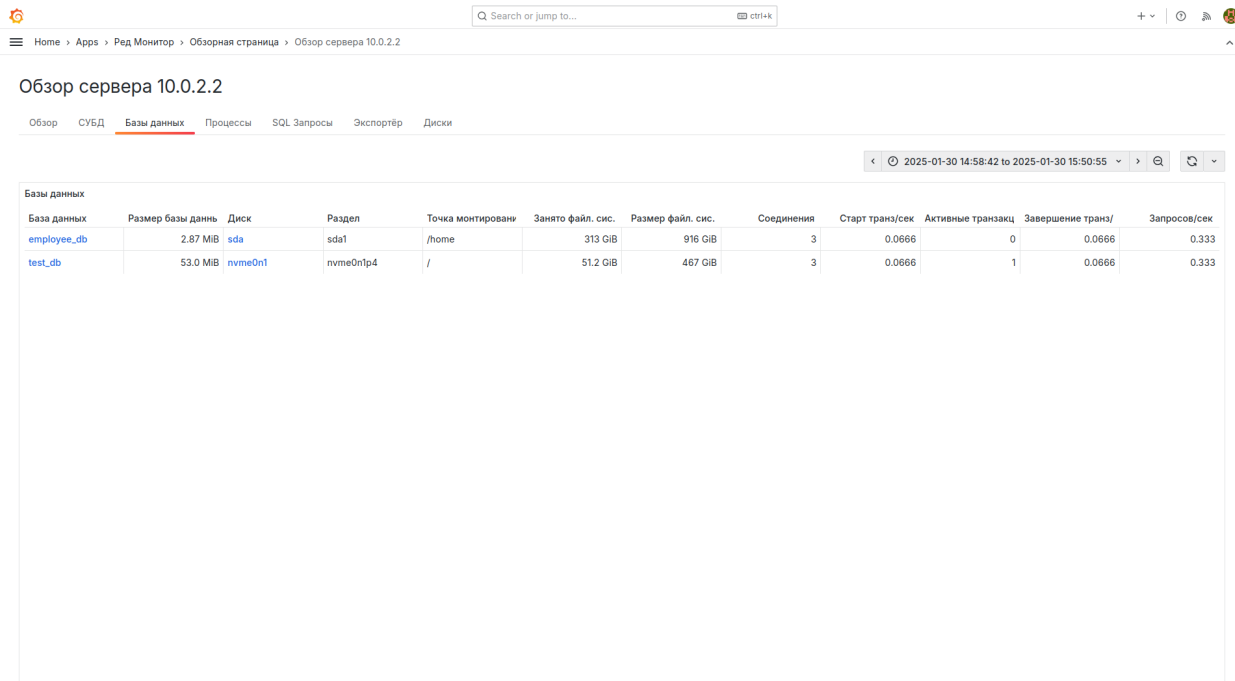


Рисунок 3.9 — Обзор баз данных

3.4.3 Информация о выбранной базе данных

На этой странице находится подробная информация о базе данных.

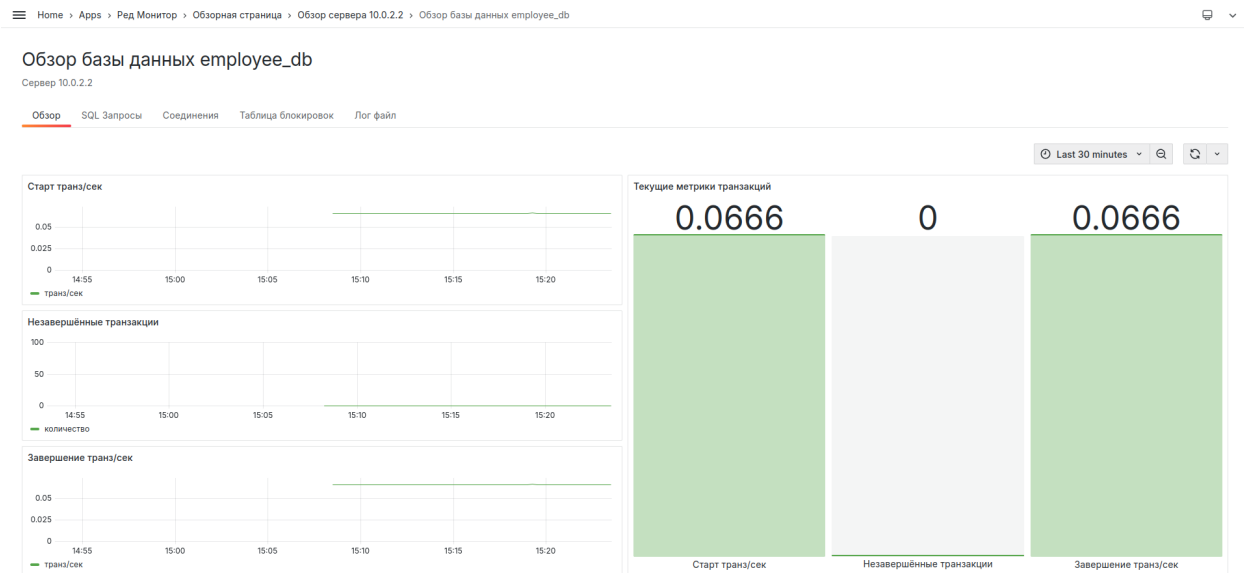


Рисунок 3.10 — Обзор выбранной базы данных - Часть 1

График Частота стартов транзакций показывает количество транзакций, запущенных за секунду, в зависимости от времени.

График Незавершённые транзакции показывает количество незавершённых транзакций (активных и бездействующих).

График Частота завершений транзакций показывает скорость возникновения транзакций завершения в зависимости от времени.



Рисунок 3.11 — Обзор выбранной базы данных - Часть 2

График Объём базы данных показывает размер базы данных в зависимости от времени.

График Использование ОЗУ соединениями показывает использование оперативной памяти соединениями с выбранной базой данных.

График Количество соединений показывает количество активных соединений с выбранной базой данных.

График **Изменение маркеров транзакций** показывает изменение значения маркеров транзакций, произошедшее с прошлого запроса информации.

График **Запросы/сек** показывает скорость возникновения запросов (количество запросов в секунду) в зависимости от времени.

График **Маркеры транзакций** отражает значения маркеров транзакций.

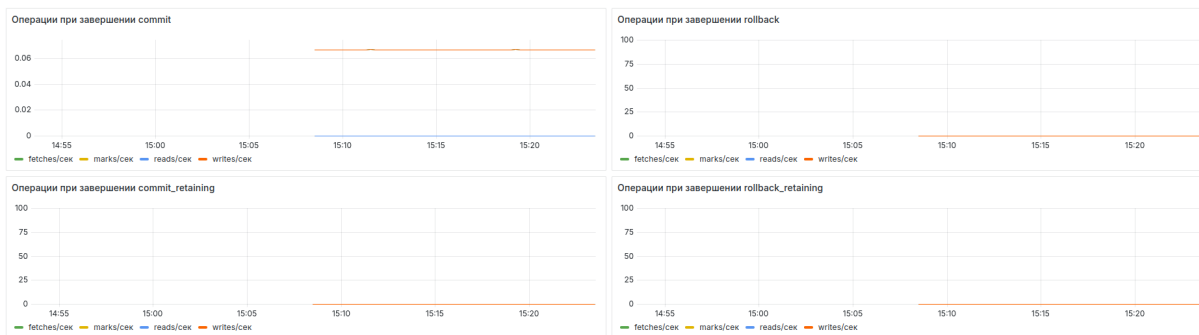


Рисунок 3.12 — Обзор выбранной базы данных - Часть 3

График **Операции при завершении commit** показывают количество операций **reads**, **fetches**, **marks**, **writes**, выполнившихся за секунду, при завершении транзакции **commit**.

График **Операции при завершении rollback** показывают количество операций **reads**, **fetches**, **marks**, **writes**, выполнившихся за секунду, при завершении транзакции **rollback**.

График **Операции при завершении commit_retaining** показывают количество операций **reads**, **fetches**, **marks**, **writes**, выполнившихся за секунду, при завершении транзакции **commit_retaining**.

График **Операции при завершении rollback_retaining** показывают количество операций **reads**, **fetches**, **marks**, **writes**, выполнившихся за секунду, при завершении транзакции **rollback_retaining**.

SQL запросы

На этой странице отображается краткая информация о выполненных запросах к выбранной базе данных. Для получения подробной информации о запросе нужно нажать на его хэш.

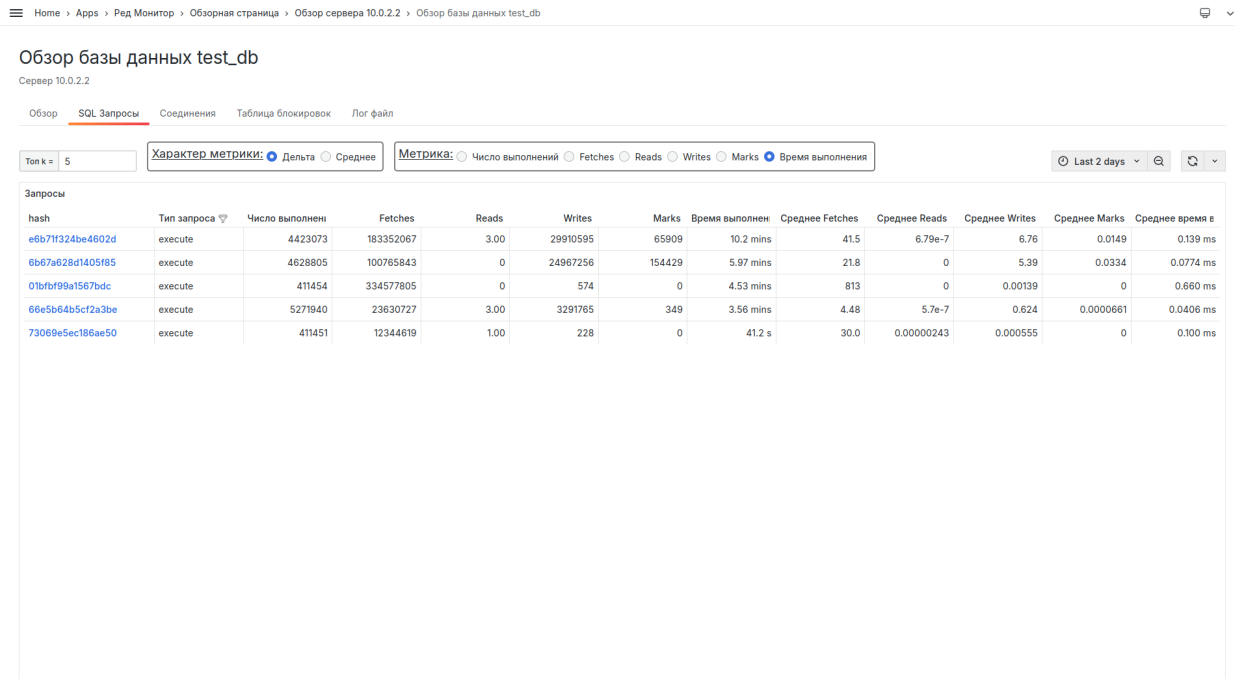


Рисунок 3.13 — SQL запросы выбранной базы данных

Обзор запроса

На данной странице находится подробная информация о выбранном запросе.

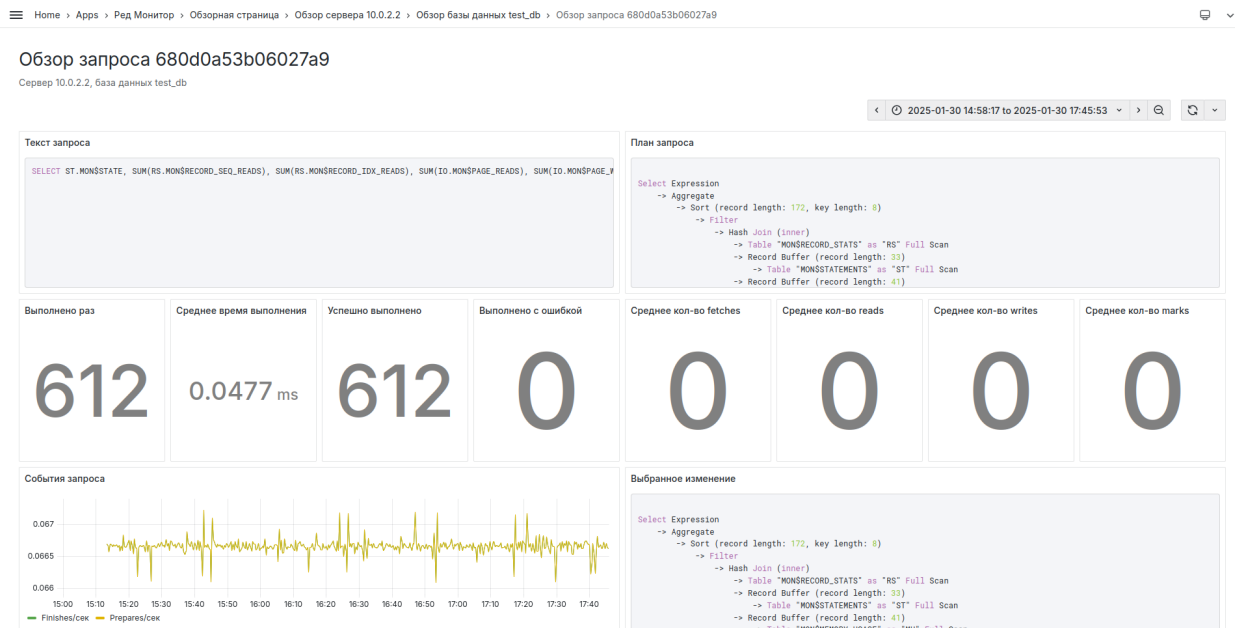


Рисунок 3.14 — Обзор запроса - Часть 1

Страница содержит текст запроса, план запроса и его изменения, информацию о количестве и времени выполнений.

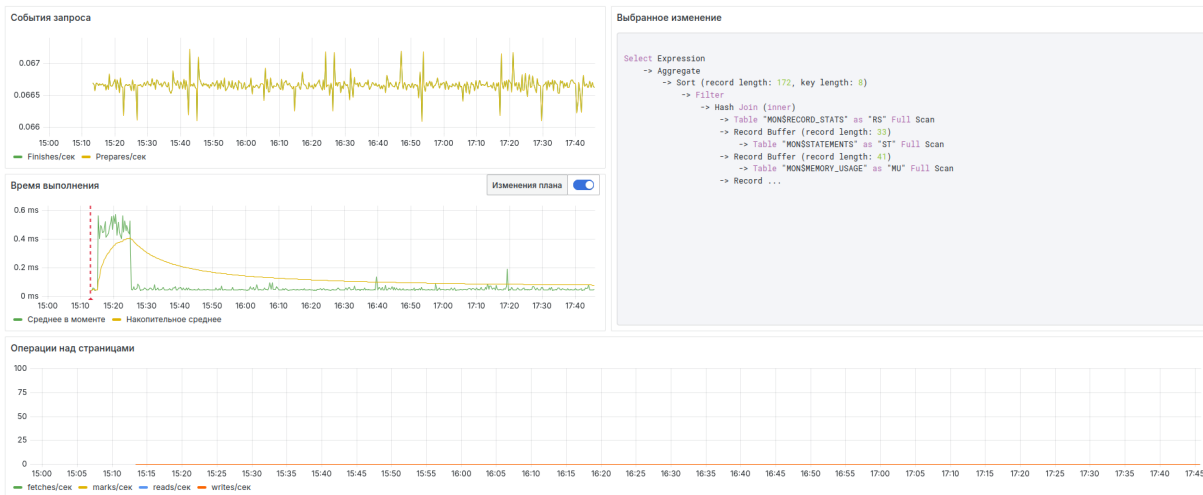


Рисунок 3.15 — Обзор запроса - Часть 2

График **События запроса** показывает частоту возникновения событий подготовки и завершения запроса за выбранный промежуток времени.

График **Операции над страницами** показывает количество страниц считанных из страничного кэша, считанных с диска, записанных на него и изменённых в страничном кэше в зависимости от времени.

График **Время выполнения** показывает время выполнения запроса. **Среднее в моменте** - среднее время выполнения по двум последним сборам метрик. **Накопительное среднее** - среднее время выполнения за указанный промежуток времени. **Изменение плана** включает/отключает отображение моментов времени, когда план запроса изменялся.

Соединения

На этой странице отображается краткая информация об активных соединениях с выбранной базой данных. Для получения подробной информации о подключении нужно нажать на его ID.

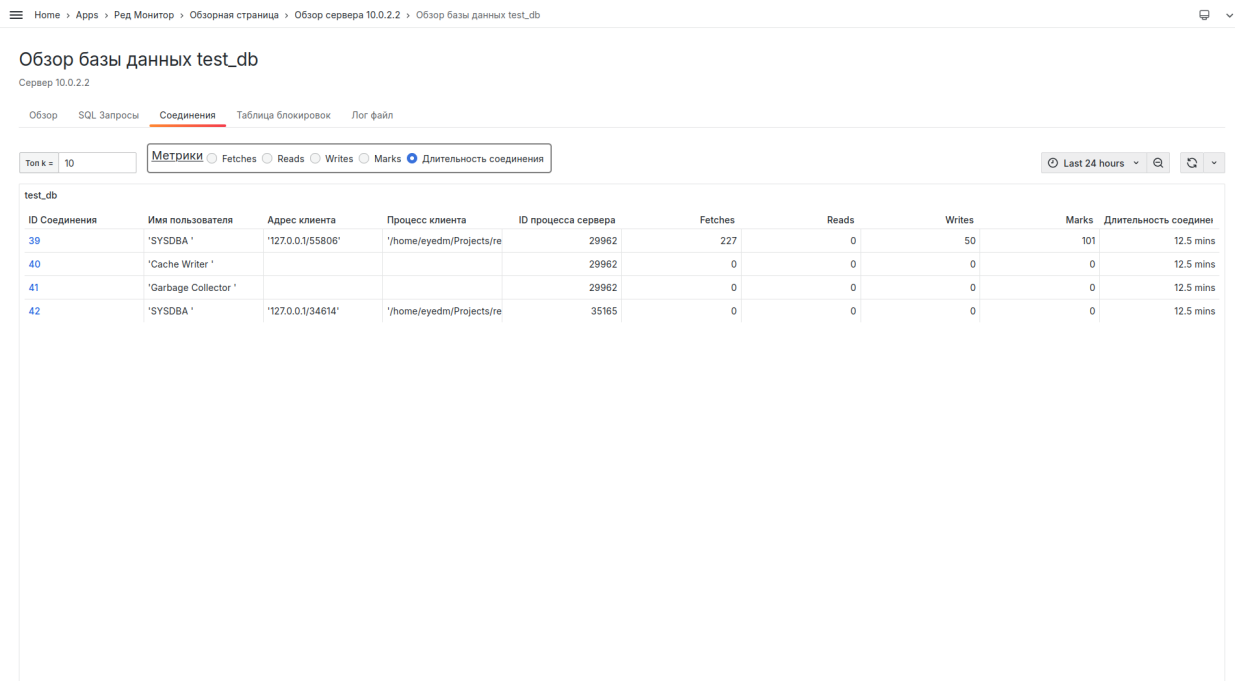


Рисунок 3.16 — Соединения с базой данных

Обзор соединения

На данной странице находится подробная информация о выбранном соединении.

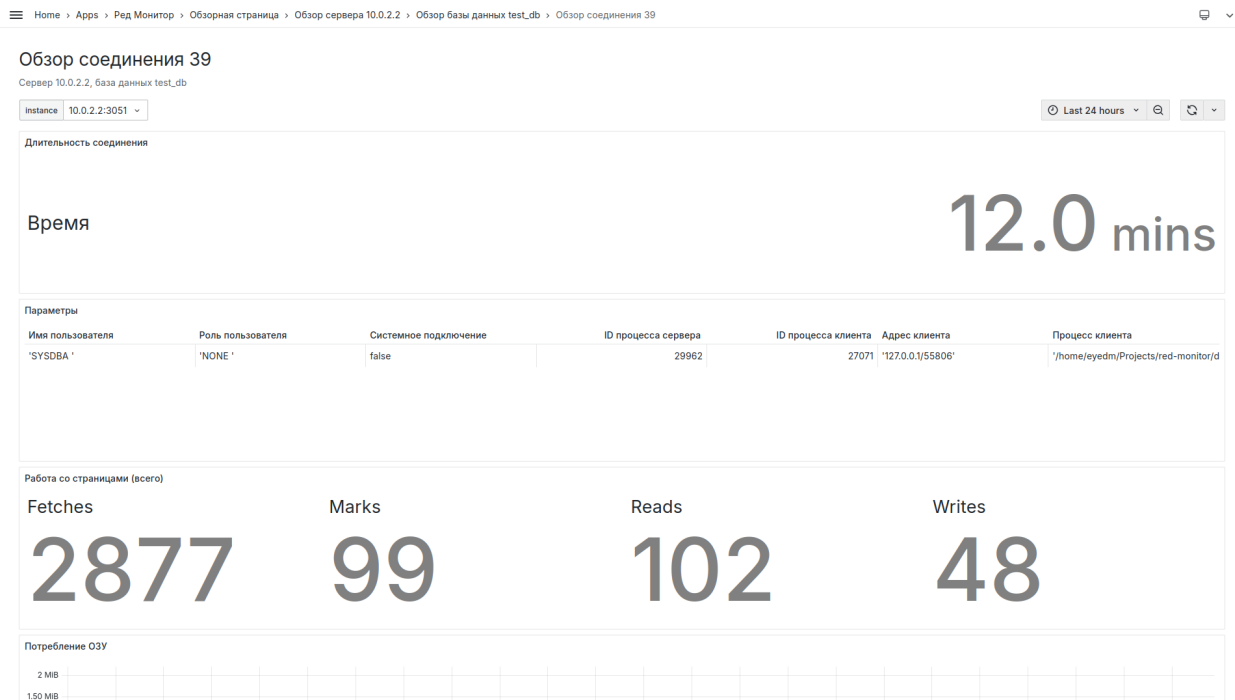


Рисунок 3.17 — Обзор соединения

Страница содержит информацию о длительности соединения, параметрах подключения, количестве обработанных страниц и объёме оперативной памяти, которая используется соединением.

Таблица блокировок

Содержание данной вкладки соответствует выводу утилиты `rdb_lock_print` в интерактивном режиме. Также показано использование памяти таблицей блокировок.



Рисунок 3.18 — Использование памяти таблицей блокировок

Лог файл

Вкладка содержит результат анализа файла `firebird.log`. Учитываются только события, относящиеся к выбранной базе данных.

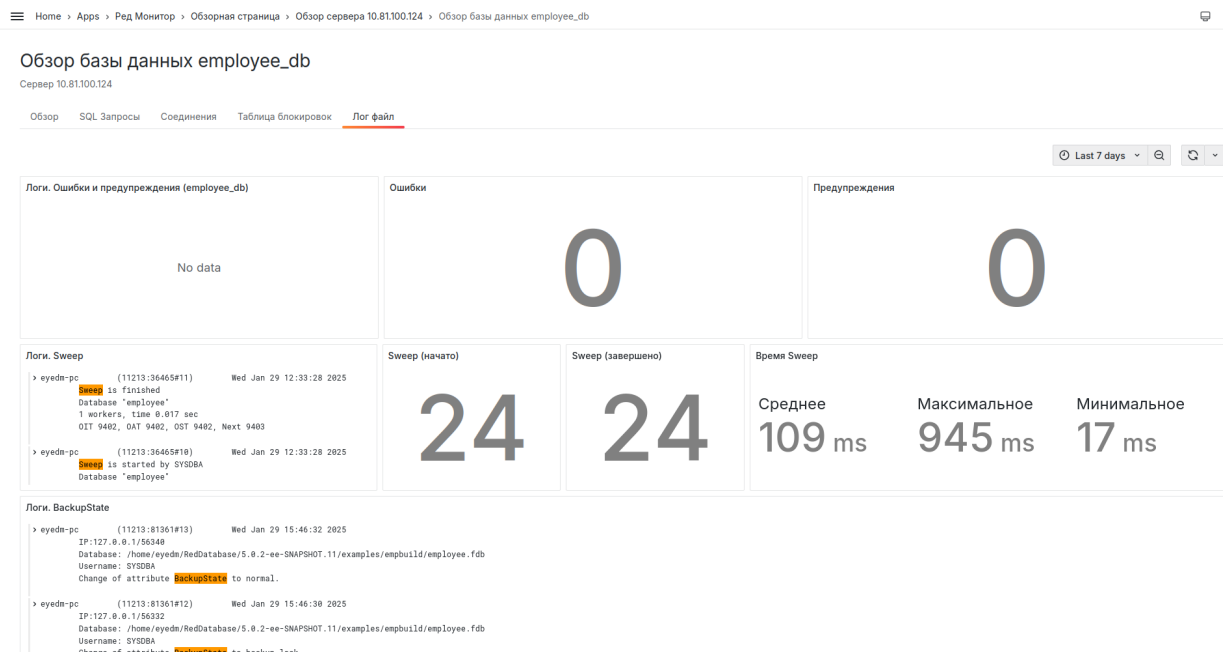


Рисунок 3.19 — Анализ лог файла

3.4.4 Процессы

На этой странице отображается краткая информация о процессах СУБД. Для получения подробной информации о процессе нужно нажать на его ID.

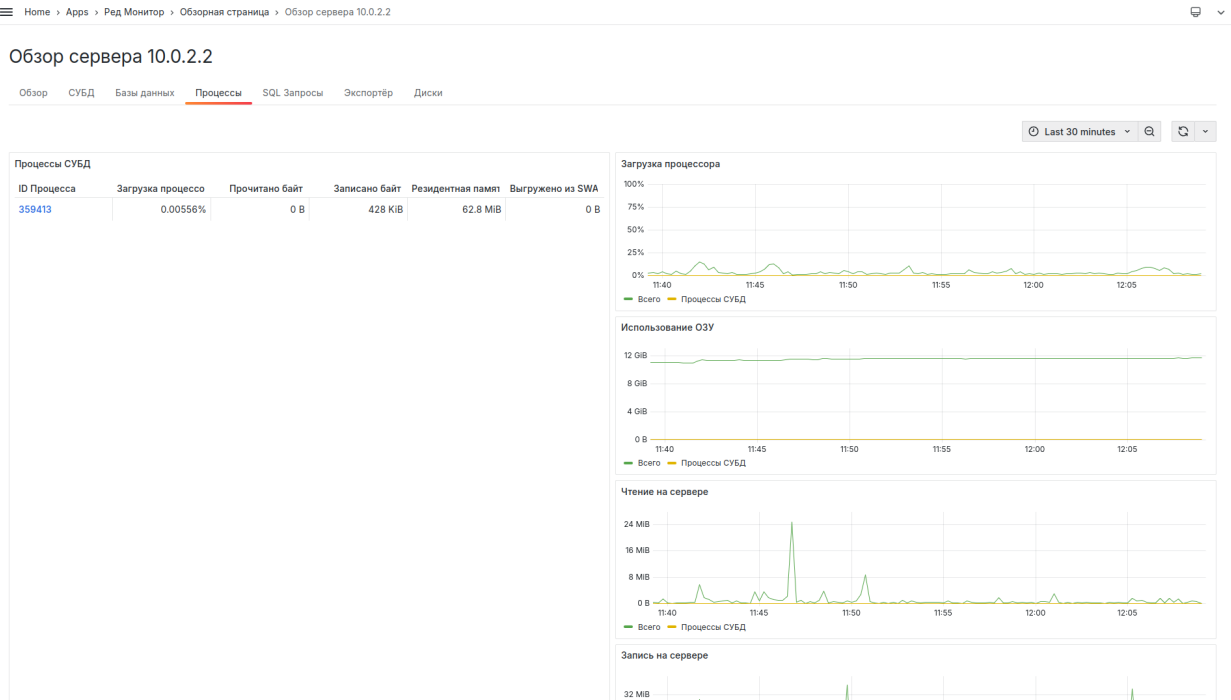


Рисунок 3.20 — Процессы СУБД

Страница показывает текущую нагрузку на процессор, использование памяти, объём записи и чтения диска, объём памяти, выгруженной в SWAP.

Обзор процесса

На данной странице находится подробная информация о выбранном процессе.



Рисунок 3.21 — Обзор процесса

График **Загрузка процессора** показывает нагрузку на процессор всеми процессами в системе и выбранным процессом СУБД.

График **Использование ОЗУ** показывает использование оперативной памяти всеми процессами в системе и выбранным процессом СУБД.

График **Потоки процесса** показывает количество потоков, открытых для выполнения выбранного процесса.

График **Использование процессора по типу нагрузки** показывает системную и пользовательскую нагрузку при выполнении выбранного процесса.

3.4.5 SQL-запросы

На странице находится краткая информация о выполненных запросах, которые соответствуют заданному фильтру. По умолчанию отображаются 10 запросов, которые выполнялись чаще всего.

Для получения подробной информации о запросе нужно нажать на его хэш. Подробнее см. [Обзор запроса](#).

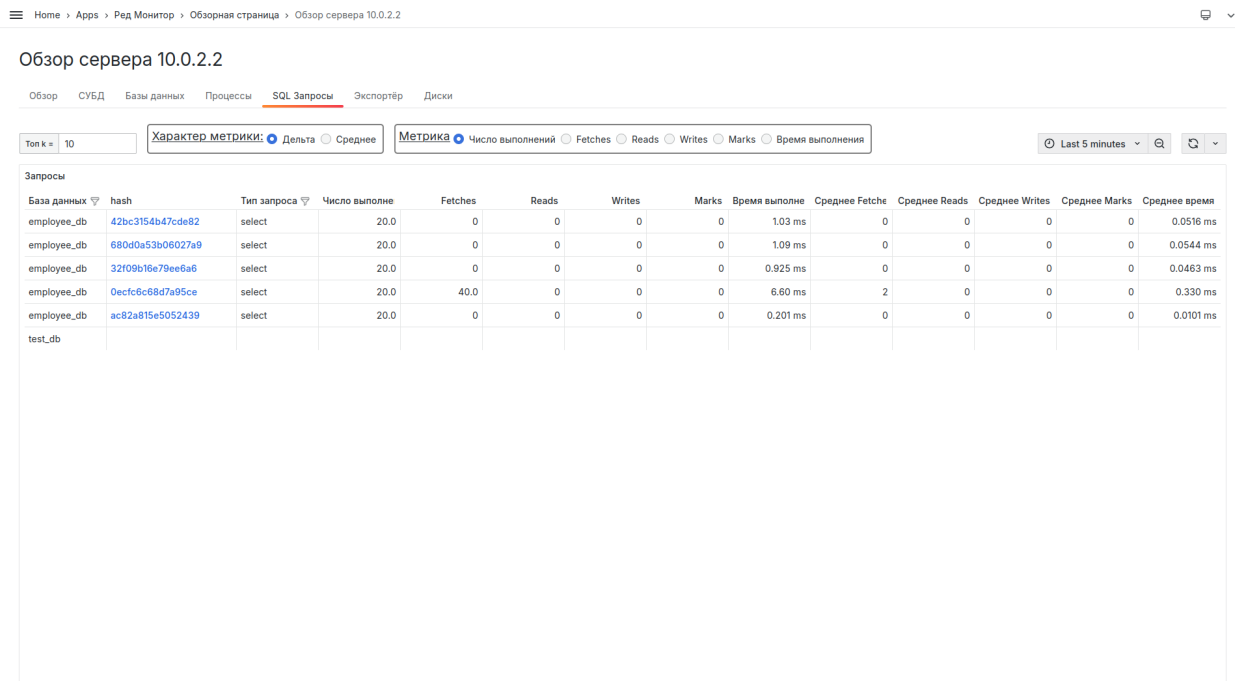


Рисунок 3.22 — Запросы

3.4.6 Экспортёр

На странице находится информация о сборе метрик экспортёром.

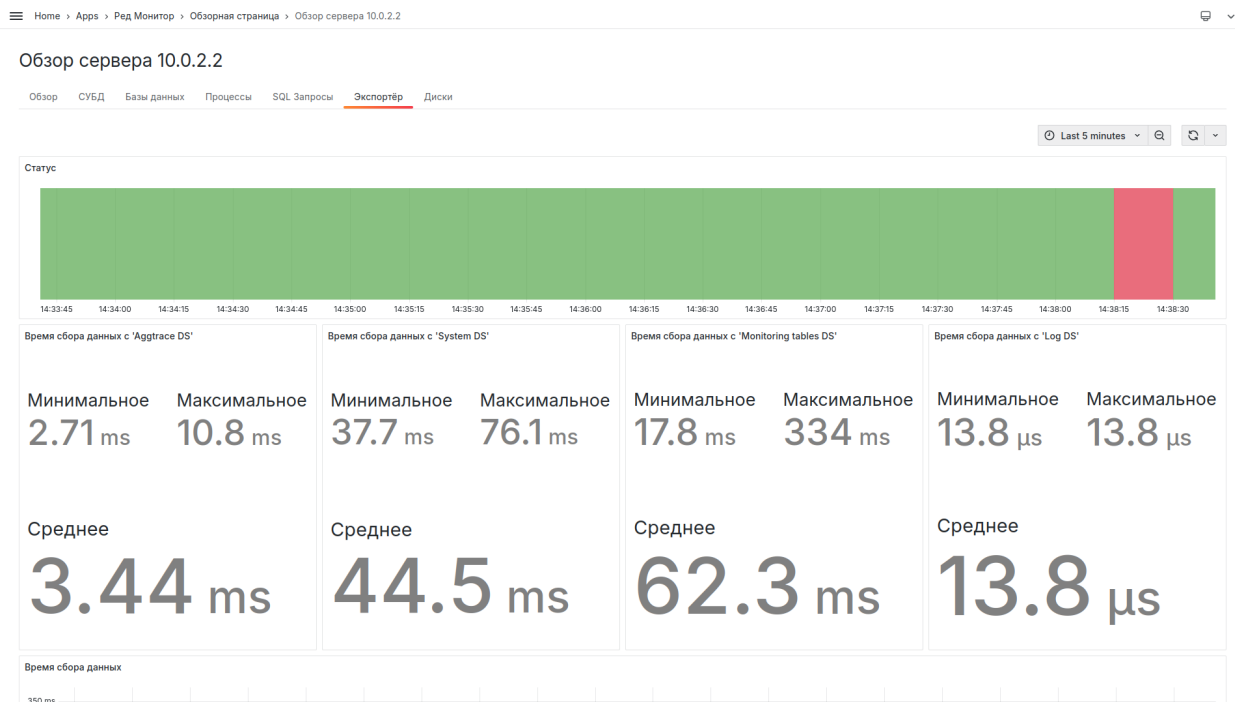


Рисунок 3.23 — Экспортёр

Панель статус показывает интервалы работы экспортёра: зелёным цветом показано время, когда экспортёр работает и собирает метрики, а красным, когда он не запущен.

На других панелях находится информация о продолжительности сбора метрик с каждого источника данных.

3.4.7 Диски

На странице отображается информация о выбранных дисках.



Рисунок 3.24 — Информация о диске

Панель Пространство показывает общий и свободный объём памяти на диске.

График IOPS отображает количество операций ввода/вывода в секунду.

Приложение А Собираемые метрики

А.1 Метрики таблиц мониторинга

Таблица А.1 — Метрики таблиц мониторинга

Название метрики	Метки	Описание метрики
rdb_call_stack_pages	<p>database - база данных.</p> <p>object_type - тип объекта базы данных:</p> <ul style="list-style-type: none"> • procedure; • trigger; • function. <p>operation - тип операции:</p> <ul style="list-style-type: none"> • reads; • fetches; • marks; • writes. 	Количество операций указанного типа, выполненных вызовами со страницами базы данных в момент сбора информации.
rdb_call_stack_memory	<p>database - база данных.</p> <p>object_type - тип объекта базы данных:</p> <ul style="list-style-type: none"> • procedure; • trigger; • function; <p>usage - использование памяти:</p> <ul style="list-style-type: none"> • used - объём используемой памяти; • allocated - количество выделенной памяти. 	Объём оперативной памяти в байтах, используемой для указанного типа вызова в момент сбора информации.
rdb_call_stack_count	<p>database - база данных.</p> <p>object_type - тип объекта базы данных:</p> <ul style="list-style-type: none"> • procedure; • trigger; • function; 	Количество вызовов указанного типа, выполняемых в момент сбора информации.
rdb_attachments_pages	<p>database - база данных.</p> <p>att_id - идентификатор соединения.</p> <p>operation - тип операции:</p> <ul style="list-style-type: none"> • reads; • fetches; • marks; • writes. 	Количество операций, выполненных соединением со страницами базы данных в момент сбора информации.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_attachments_memory	database - база данных. att_id - идентификатор соединения. usage - использование памяти: <ul style="list-style-type: none"> • used - объём используемой памяти; • allocated - количество выделенной памяти. 	Объём оперативной памяти в байтах, используемой соединением в момент сбора информации.
rdb_attachments_connection_time	database - база данных. att_id - идентификатор соединения.	Продолжительность соединения в наносекундах.
rdb_attachments_count	database - база данных.	Количество соединений с базой данных в момент сбора информации.
rdb_transactions_pages	database - база данных. is_active - активна ли транзакция; isolation_mode - уровень изоляции: <ul style="list-style-type: none"> • consistency • concurrency • read_committed_rec_ver • read_committed_no_rec_ver • read_committed_read_consistency read_only - выполняется ли транзакция в режиме "read_only"; auto_commit - используется ли режим автоматической фиксации; auto_undo - используется ли автоматическая отмена транзакции; operation - тип операции: <ul style="list-style-type: none"> • reads; • fetches; • marks; • writes. 	Количество операций, выполненных транзакциями.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_transactions_memory	database - база данных. is_active - активна ли транзакция. isolation_mode - уровень изоляции: <ul style="list-style-type: none"> • consistency • concurrency • read_committed_rec_ver • read_committed_no_rec_ver • read_committed_read_consistency read_only - выполняется ли транзакция в режиме read_only. auto_commit - используется ли режим автоматической фиксации. auto_undo - используется ли автоматическая отмена транзакции. usage - использование памяти: <ul style="list-style-type: none"> • used - объём используемой памяти. • allocated - количество выделенной памяти. 	Объём памяти (в байтах), используемой транзакциями, выполняющимися в момент сбора информации.
rdb_transactions_count	database - база данных. is_active - активна ли транзакция. isolation_mode - уровень изоляции: <ul style="list-style-type: none"> • consistency • concurrency • read_committed_rec_ver • read_committed_no_rec_ver • read_committed_read_consistency read_only - выполняется ли транзакция в режиме read_only. auto_commit - используется ли режим автоматической фиксации. auto_undo - используется ли автоматическая отмена транзакции.	Количество транзакций, выполняющихся в момент сбора информации.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_statements_pages	<p>database - база данных.</p> <p>state - состояние запроса:</p> <ul style="list-style-type: none"> • idle — бездействующий; • active — активный; • stalled — приостановленный, то есть запрос или курсор "живой", но в данный момент не выполняется. <p>operation - тип операции:</p> <ul style="list-style-type: none"> • reads; • fetches; • marks; • writes. 	Количество запросов, которые выполняются над страницами базы данных в момент сбора информации.
rdb_statements_memory	<p>database - база данных.</p> <p>state - состояние запроса:</p> <ul style="list-style-type: none"> • idle — бездействующий; • active — активный; • stalled — приостановленный, то есть запрос или курсор "живой", но в данный момент не выполняется. <p>usage - использование памяти:</p> <ul style="list-style-type: none"> • used - объём используемой памяти. • allocated - количество выделенной памяти. 	Объём памяти (в байтах), используемой запросами, выполняющимися в момент сбора информации.
rdb_statements_count	<p>database - база данных.</p> <p>state - состояние запроса:</p> <ul style="list-style-type: none"> • idle — бездействующий; • active — активный; • stalled — приостановленный, то есть запрос или курсор "живой", но в данный момент не выполняется. 	Количество запросов, выполняющихся в момент сбора информации.
rdb_response_times	<p>database - база данных.</p> <p>table - таблица MON\$DATABASE.</p>	Время выполнения запроса к таблице MON\$DATABASE в секундах.
rdb_database_sql_dialect	database - база данных.	SQL диалект.
rdb_database_page_size	database - база данных.	Размер страницы файлов базы данных в байтах.
rdb_database_page_buffers	database - база данных.	Количество страниц, выделенных в оперативной памяти для кэша;

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_database_sweep_interval	database - база данных.	Интервал автоматической сборки мусора;
rdb_database_read_only	database - база данных.	Является ли база данных доступной только для чтения. Read-only - значение 1, read-write значение 0.
rdb_database_forced_writes	database - база данных.	Указывает, установлен для базы режим синхронного вывода (forced writes, значение 1) или режим асинхронного вывода (значение 0).
rdb_database_backup_state	database - база данных.	Указывает состояние бэкапа базы данных: <ul style="list-style-type: none"> • 0 — база не затронута бэкапом, • 1 — база заблокирована для резервирования, • 2 — объединение временного файла дельты и основного файла базы данных.
rdb_database_pages	database - база данных.	Количество страниц, выделенных для базы данных на внешнем устройстве.
rdb_database_shutdown_mode	database - база данных.	текущее состояние остановки (shutdown) базы данных: <ul style="list-style-type: none"> • 0 — база данных активна (online); • 1 — остановлена для нескольких пользователей (multi-user shutdown); • 2 — остановлена для одного пользователя (single-user shutdown); • 3 — полностью остановка (full shutdown).
rdb_database_size	database - база данных.	Объём памяти в байтах, занимаемый базой данных на диске. Это произведение количества выделенных страниц и размера одной страницы базы данных.
rdb_database_cache	database - база данных.	Размер выделенной оперативной памяти под кеш СУБД в байтах.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_database_up	database - база данных.	Указывает на наличие соединения экспортёра с базой данных: <ul style="list-style-type: none"> • 0 - соединение отсутствует; • 1 - соединение установлено.

А.2 Метрики агрегатного аудита

Эти метрики показывают на сколько изменилось значение с предыдущего сбора информации.

Таблица А.2 — Метрики агрегатного аудита

Название метрики	Метки	Описание метрики
rdb_aggtrace_stmt_pages	database - база данных; hash - хэш запроса; q_operation - тип запроса: <ul style="list-style-type: none"> • select • insert • update • create • delete • drop operation - тип операции: <ul style="list-style-type: none"> • reads; • fetches; • marks; • writes. marker - количество операций: <ul style="list-style-type: none"> • min; • max; • avg. 	Количество операций, выполненных запросом.
rdb_aggtrace_stmt_times	database - база данных; hash - хэш запроса; q_operation - тип запроса: <ul style="list-style-type: none"> • select • insert • update • create • delete • drop marker - затраченное время: <ul style="list-style-type: none"> • min; • max; • avg. 	Время, затраченное на выполнение запросов.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_aggtrace_stmt_count	database - база данных; hash - хэш запроса; q_operation - тип запроса: <ul style="list-style-type: none"> • select • insert • update • create • delete • drop marker - результат выполнения: <ul style="list-style-type: none"> • failed • succeed • total 	Количество выполненных запросов выбранного типа.
rdb_aggtrace_sort_mu_total	database - база данных; hash - хэш запроса; q_operation - тип запроса: <ul style="list-style-type: none"> • select • insert • update • create • delete • drop place - место выполнения сортировки: <ul style="list-style-type: none"> • cache - в кэше • disk - на диске • total - все сортировки (и в кэше и на диске) marker - объём сортировки: <ul style="list-style-type: none"> • avg • min • max • total 	Объём памяти (в байтах), выделенной под сортировку.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_aggtrace_stmt_prepare_count	database - база данных; hash - хэш запроса; q_operation - тип запроса: <ul style="list-style-type: none"> • select • insert • update • create • delete • drop marker - результат подготовки запроса: <ul style="list-style-type: none"> • failed - количество выполнений, завершившихся ошибкой • succeed - количество успешных выполнений • total - количество событий PREPARE STATEMENT 	Количество событий PREPARE STATEMENT
rdb_aggtrace_trns_start_count	database - база данных marker - результат выполнения: <ul style="list-style-type: none"> • failed - количество выполнений, завершившихся ошибкой • succeed - количество успешных выполнений • total - количество событий START TRANSACTION 	Количество событий START TRANSACTION
rdb_aggtrace_trns_finish_count	database - база данных finish - тип транзакции: <ul style="list-style-type: none"> • commit • rollback • commit_retaining • rollback_retaining marker - результат выполнения: <ul style="list-style-type: none"> • failed - количество выполнений, завершившихся ошибкой • succeed - количество успешных выполнений 	Количество событий FINISH TRANSACTION

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_aggtrace_trns_finish_pages	database - база данных finish - тип транзакции: <ul style="list-style-type: none"> • commit • rollback • commit_retaining • rollback_retaining operation - тип операции: <ul style="list-style-type: none"> • reads • fetches • marks • writes marker - время выполнения выполнения: <ul style="list-style-type: none"> • avg • min • max • total 	Операции, выполненные над страницами базы данных в рамках завершения транзакции
rdb_aggtrace_trns_finish_times	database - база данных finish - тип транзакции: <ul style="list-style-type: none"> • commit • rollback • commit_retaining • rollback_retaining marker - время выполнения выполнения: <ul style="list-style-type: none"> • avg • min • max • total 	Время, потраченное на завершение транзакции

A.3 Метрики утилиты rdb_lock_print

Таблица A.3 — Метрики утилиты rdb_lock_print

Название метрики	Метки	Описание метрики
rdb_lck_table_length	database - база данных	Общий объем памяти, выделенный таблице блокировок (в байтах)
rdb_lck_table_used	database - база данных	Наибольшая величина смещения в таблице блокировок, которая используется в настоящий момент
rdb_lck_table_hash_slots	database - база данных	Число слотов кэширования блокировок

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_lck_table_scan_interval	database - база данных	Время (в секундах), которое ожидает менеджер блокировок до того как запустить к поиску взаимных блокировок
rdb_lck_table_acquire_spins	database - база данных	Режим ожидания взаимной блокировки, когда повторяется запрос к таблице блокировок
rdb_lck_table_acquires	database - база данных	Сколько раз владелец запрашивает исключительное управление таблицей блокировок, чтобы выполнить изменения
rdb_lck_table_acquire_blocks	database - база данных	Сколько раз владелец находился в состоянии ожидания при запросе исключительного управления таблицей блокировок.
rdb_lck_table_acquire_retries	database - база данных	
rdb_lck_table_retry_success	database - база данных	
rdb_lck_table_enqs	database - база данных	Число запросов, полученных на блокировку (не включает запросы, которые пришли и ушли)
rdb_lck_table_converts	database - база данных	Запросы на повышение уровня блокировки
rdb_lck_table_downgrades	database - база данных	
rdb_lck_table_deqs	database - база данных	
rdb_lck_read_data	database - база данных	
rdb_lck_table_write_data	database - база данных	
rdb_lck_table_query_data	database - база данных	
rdb_lck_table_waits	database - база данных	
rdb_lck_table_denies	database - база данных	
rdb_lck_table_timeouts	database - база данных	
rdb_lck_table_blocks	database - база данных	Запросы, которые не могут быть удовлетворены немедленно
rdb_lck_table_wakeups	database - база данных	
rdb_lck_table_scans	database - база данных	Показывает число просмотров менеджером блокировок цепочки блокировок и владельцев для поиска взаимных блокировок

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_lck_table_deadlocks	database - база данных	Число найденных взаимных блокировок
rdb_lck_table_lops	database - база данных lop - операция над блокировкой	

А.4 Метрики операционной системы

Таблица А.4 — Метрики операционной системы

Название метрики	Метки	Описание метрики
rdb_disks_io_counters	disk - диск. operation - тип операции: <ul style="list-style-type: none"> • reads • writes 	Количество операций чтения и записи на диск.
rdb_disks_io_bytes	disk - диск. operation - тип операции: <ul style="list-style-type: none"> • reads • writes 	Объём прочитанной/записанной на диск информации в байтах.
rdb_disks_io_wtime	database - база данных. disk - диск.	Взвешенное время, потраченное на операции ввода-вывода.
rdb_disks_busy_time	disk - диск. kind - тип времени, потраченного на ввод/вывод: <ul style="list-style-type: none"> • normal - время, затраченное на фактический ввод/вывод; • weighted - взвешенное время, затраченное на фактический ввод/вывод. 	Время, потраченное на фактический ввод-вывод.
rdb_disk_size	disk - диск.	Объём диска
rdb_disks_info	disk - диск. vendor - поставщик. revision - ревизия. model - модель. serial - серийный номер. scheduler - планировщик диска.	Общая информация о диске.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_mount_point_usage	disk - диск. partition - раздел диска. point - точка монтирования. usage: <ul style="list-style-type: none"> • used • free • total 	Использование точки мониторинга.
rdb_link_mp_db	disk - диск partition - раздел диска point - точка монтирования database - база данных.	Устанавливает связь с другими метриками.
rdb_procs_open_files_size	pid - идентификатор процесса СУБД kind - тип временного объекта: <ul style="list-style-type: none"> • table • blob • undo • redbuf • merge • sort • tpc • snap 	Размер временных файлов, созданных процессами СУБД (в байтах).
rdb_procs_open_files_count	pid - идентификатор процесса СУБД kind - тип временного объекта: <ul style="list-style-type: none"> • table • blob • undo • redbuf • merge • sort • tpc • snap 	Количество временных файлов, созданных процессами СУБД.
rdb_procs_cpu_times	pid- идентификатор процесса СУБД kind - тип времени, потраченного процессором: <ul style="list-style-type: none"> • system • user • iowait 	Время, потраченное процессором на выполнение процесса.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_procs_memory_usage	pid - идентификатор процесса СУБД usage: <ul style="list-style-type: none"> • resident - резидентная память • virtual - виртуальная память • swapped_out - занимаемая память в пространстве swap • shared - разделяемая память: text, lib, data, dirty, uss, pss. 	Объём резидентной, виртуальной и выгруженной в swap памяти в байтах.
rdb_procs_io_counters	pid - идентификатор процесса СУБД operation - тип операции: <ul style="list-style-type: none"> • reads • writes 	Количество операций ввода/вывода, выполняемых процессом СУБД.
rdb_procs_io_bytes	pid - идентификатор процесса СУБД operation - тип операции: <ul style="list-style-type: none"> • reads • writes 	Размер прочитанной и записанной информации в байтах процессами СУБД.
rdb_procs_threads_count	pid - идентификатор процесса СУБД	Количество открытых потоков для процесса
rdb_memory_usage	field: <ul style="list-style-type: none"> • free • used • total 	Использование оперативной памяти.
rdb_swap_usage	usage: <ul style="list-style-type: none"> • used • free • total 	Использование памяти, выделенной для swap.
rdb_swapped	direction: <ul style="list-style-type: none"> • in • out 	Объём данных (в байтах), затронутый swap.
rdb_cpu_freq	marker: <ul style="list-style-type: none"> • max • min • current 	Частота работы процессора.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_cpu_times	kind: <ul style="list-style-type: none"> • system • user • iowait • irq • soft_irq 	Время, потраченное процессором.
rdb_cpu_ctx		Число изменений контекста процессора.
rdb_cpu_cores	kind: <ul style="list-style-type: none"> • logical • physical 	Количество ядер процессора.
rdb_cpu_interrupts	marker: <ul style="list-style-type: none"> • soft • casual 	Число прерываний процессора.
rdb_tmp_mount_point_usage	point usage: <ul style="list-style-type: none"> • total • used • free 	Использование точки монтирования.
rdb_sys_loadavg	period: 1 минута, 5 минут, 15 минут.	Количество готовых к выполнению процессов.
rdb_net_io_bytes	interface - сетевой интерфейс direction - направление трафика (in, out)	Количество байт, переданных по сети.
rdb_net_io_packets	interface - сетевой интерфейс direction - направление трафика (in, out)	Количество пакетов, переданных по сети.
rdb_net_io_errs	interface - сетевой интерфейс direction - направление трафика (in, out)	Количество ошибок, возникших при передаче по сети.
rdb_net_io_drops	interface - сетевой интерфейс direction - направление трафика (in, out)	Количество потерянных пакетов.

А.5 Другие метрики

Таблица А.5 — Другие метрики

Название метрики	Метки	Описание метрики
<code>rdb_transactions</code> <code>_markers</code>	<code>database</code> - база данных; <code>marker</code> : <ul style="list-style-type: none">• NT• OST• OAT• OIT	Значения маркеров транзакций в момент сбора информации.
<code>rdb_log_errors</code>	<code>kind</code> - тип ошибки: <ul style="list-style-type: none">• fatal• critical• normal• event	Количество ошибок указанного типа.
<code>rdb_server_version</code>	<code>kind</code> : major, minor, fix	Версия СУБД.