РБДМонитор

Версия 1.3.0

Руководство пользователя

Содержание

1 Общие сведения о программе 4
1.1 Назначение программы . . . 4
1.2 Минимальный состав аппаратных средств . . . 4
1.3 Минимальный состав программных средств . . . 4

2 Установка РБД Монитор 5
2.1 Установка на Linux . . . 5
2.1.1 Установка сервера РБД Монитор . . . 5
2.1.2 Установка агента мониторинга . . . 6
2.2 Установка на Windows . . . 6
2.2.1 Установка сервера РБД Монитор . . . 6
2.2.2 Установка агента мониторинга . . . 7

3 Обновление РБДМонитор 8
3.1 Обновление на Linux . . . 8
3.1.1 Обновление сервера мониторинга . . . 8
3.1.2 Обновление агента мониторинга . . . 8
3.2 Обновление на Windows . . . 9
3.2.1 Обновление сервера мониторинга . . . 9
3.2.2 Обновление агента мониторинга . . . 10

4 Удаление РБДМонитор 11
4.1 Удаление на Linux . . . 11
4.2 Удаление на Windows . . . 11

5 Настройка РБДМонитор 12
5.1 Настройка портов РБД Монитор . . . 13
5.1.1 Наблюдаемая СУБД . . . 13
5.1.2 Настройка порта агента мониторинга . . . 13
5.1.3 Настройка портов сервера мониторинга . . . 14
Порт Prometheus . . . 14
Порт Alertmanager . . . 15
Порт Grafana . . . 15
5.2 Настройки наблюдаемой СУБД . . . 16
5.3 Агент мониторинга . . . 16
5.4 Сервер мониторинга . . . 22
5.4.1 Prometheus . . . 22
5.4.2 Alertmanager . . . 23
5.4.3 Настройка уведомлений на почту . . . 24
5.4.4 Правила уведомлений . . . 25
5.4.5 Grafana . . . 25

6 Управление мониторингом 27
6.1 Наблюдаемые СУБД . . . 27
6.2 Наблюдаемые базы данных . . . 27
6.3 Управление источниками данных . . . 28
6.4 Формирование сравнительного отчёта . . . 29
6.5 Мониторинг при миграции РЕД Базы Данных 3 на РЕД Базу Данных 5 . . . 29

7 Просмотр метрик 30
7.1 Обзорная страница . . . 30

7.2 Информация о выбранном сервере . . . 30
7.2.1 СУБД . . . 31
7.2.2 Базы данных . . . 32
7.2.3 Информация о выбранной базе данных . . . 33
SQL запросы . . . 34
Соединения . . . 36
Таблица блокировок . . . 37
Лог файл . . . 38
7.2.4 Процессы . . . 38
Обзор процесса . . . 39
7.2.5 SQL-запросы . . . 40
7.2.6 Агент мониторинга . . . 41
7.2.7 Диски и рейды . . . 41
Обзор RAID-массива . . . 42
7.3 Мониторинг кластеров . . . 43
7.3.1 Необходимые настройки . . . 43
7.3.2 Обзор кластеров . . . 44
7.3.3 Информация о выбранном кластере . . . 44
Обзор . . . 44
Временные метки . . . 44
7.3.4 Информация об узле кластера . . . 45

Приложение A Собираемые метрики . . . 47
A.1 Метрики таблиц мониторинга . . . 47
A.2 Метрики агрегатного аудита . . . 52
A.3 Метрики утилиты rdb_lock_print . . . 55
A.4 Метрики операционной системы . . . 57
A.5 Метрики RAID-массивов . . . 60
A.6 Метрики агента мониторинга . . . 62
A.7 Другие метрики . . . 63

Глава 1

Общие сведения о программе

1.1 Назначение программы

РБДМонитор - это система для мониторинга СУБД РЕД База Данных и Firebird.

РБДМонитор предназначен для наблюдения за состоянием СУБД. Он предоставляет возможность мониторинга сразу нескольких баз данных. Показывает подробную информацию о пользователях, соединениях, запросах и ошибках. Помогает следить за изменением планов запросов. Показывает, какую нагрузку на сервер и операционную систему оказывает наблюдаемая СУБД.

1.2 Минимальный состав аппаратных средств

Системные требования для агента мониторинга:

- Оперативная память от 512Мб;
- Процессор не менее 1 ядра;
- Запоминающее устройство объёмом не менее 2Гб.

Системные требования для сервера мониторинга:

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

1.3 Минимальный состав программных средств

- СУБД: РЕД База Данных версии не ниже 3 или Firebird версии не ниже 5;
- `Prometheus` версии не ниже 2.37.5;
- `Grafana` версии не ниже 11.6.0. Браузеры, поддерживаемые `Grafana`: `Chrome/Chromium`, `Firefox`, `Safari`, `Microsoft Edge`.

Глава 2

Установка РБД Монитор

РБД Монитор состоит из агента мониторинга и сервера мониторинга. Агент мониторинга необходимо установить на одну машину с наблюдаемой СУБД. Сервер мониторинга можно установить, как на одну машину с агентом, так и на отдельную.

Для установки и запуска РБД Монитора необходим файл лицензии. Одна лицензия позволяет запустить один агент мониторинга. Для каждого наблюдаемого сервера СУБД необходим отдельный файл лицензии.

2.1 Установка на Linux

Рекомендуется установить сначала серверную часть, а затем агент мониторинга.

2.1.1 Установка сервера РБД Монитор

Установка выполняется на сервер, где будут храниться собранные метрики.

Перед началом установки нужно узнать ip-адреса наблюдаемых машин.

1. Скачайте с официального сайта установочный файл `rdbmonitor_server_installer_<версия>.bin`.
2. Назначьте установочному файлу права на выполнение:

   ```
   chmod +x rdbmonitor_server_installer_<версия>.bin
   ```

3. Запустите установку от имени суперпользователя:

   ```
   sudo ./rdbmonitor_server_installer_<версия>.bin
   ```

4. Укажите каталог установки сервера мониторинга. По умолчанию установится по пути `/opt/rdbmonitor/server`.
5. Укажите адреса наблюдаемых машин через запятую.
6. Убедитесь, что сервер мониторинга запущен:

   ```
   systemctl status rdbmonitor-server.target
   ```

Права на каталог РБД Монитор устанавливаются автоматически. Их владельцем становится `rdbmonitor:rdbmonitor`.

Компоненты сервера мониторинга после установки работают на следующих портах:

- `Grafana` - 40300
- `Prometheus` - 40301
- `Alertmanager` - 40302

2.1.2 Установка агента мониторинга

Перед установкой агента мониторинга убедитесь, что сервер мониторинга уже установлен. Необходимо узнать IP-адрес сервера мониторинга.

1. Скачайте с официального сайта установочный файл `rdbmonitor_client_installer_<версия>.bin`.
2. Убедитесь, что наблюдаемая СУБД запущена:

```
systemctl status firebird
```

3. Разместите лицензию и установочный файл в одном каталоге.
4. Назначьте установочному файлу права на выполнение:

```
chmod +x rdbmonitor_client_installer_<версия>.bin
```

5. Запустите установку от имени суперпользователя:

```
sudo ./rdbmonitor_client_installer_<версия>.bin
```

6. Укажите каталог установки агента мониторинга. По умолчанию установится по пути `rdbmonitor/client/exporter/`.
7. Укажите IP-адрес сервера мониторинга.
8. Укажите каталог установки наблюдаемой СУБД. По умолчанию `/opt/RedDatabase/`.
9. Укажите порт наблюдаемой СУБД. По умолчанию используется значение 3050.
10. Укажите пользователя СУБД с правами `RDB$ADMIN`, от имени которого агент мониторинга будет подключаться к наблюдаемой СУБД. По умолчанию используется `SYSDBA`.
11. Укажите через запятую пути к базам данных, которые нужно мониторить, или их алиасы.
12. Перезапустите наблюдаемую СУБД:

```
sudo systemctl restart firebird
```

13. Убедитесь, что агент мониторинга запущен:

```
systemctl status rdbmonitor-exporter
```

14. Откройте в браузере страницу запущенной `Grafana` (по умолчанию http://localhost:40300/). По умолчанию для входа используется логин `admin` и пароль `admin`.

Права на каталог РБД Монитор устанавливаются автоматически. Их владельцем становится `rdbmonitor:rdbmonitor`.

Агент мониторинга после установки работает от имени пользователя root на порту 40304.

2.2 Установка на Windows

2.2.1 Установка сервера РБД Монитор

Установка выполняется на сервер, где будут храниться собранные метрики.

Перед началом установки нужно узнать ip-адреса наблюдаемых машин.

1. Скачайте с официального сайта установочный файл `rdbmonitor_server_installer_win_<версия>.exe`.
2. Запустите установку от имени администратора:
3. Укажите каталог установки сервера мониторинга. По умолчанию установится по пути `C:\Program Files\RedSoft\server`.
4. Укажите адреса наблюдаемых машин через запятую.
5. Убедитесь, что служба `rdbmonitor-server.target` запущена.

Компоненты сервера мониторинга после установки работают на следующих портах:

- `Grafana` - 40300
- `Prometheus` - 40301
- `Alertmanager` - 40302

Изменить порты сервера мониторинга можно по инструкции *Настройка портов сервера мониторинга*.

2.2.2 Установка агента мониторинга

Перед установкой агента мониторинга убедитесь, что сервер мониторинга уже установлен. Необходимо узнать IP-адрес сервера мониторинга.

1. Скачайте с официального сайта установочный файл `rdbmonitor_client_installer_win_<версия>.exe`.
2. Убедитесь, что наблюдаемая СУБД запущена.
3. Разместите лицензию и установочный файл в одном каталоге.
4. Запустите установку от имени администратора.
5. Укажите каталог установки агента мониторинга. По умолчанию установится по пути `C:\Program Files\RedSoft\client\exporter`.
6. Укажите IP-адрес сервера мониторинга.
7. Укажите каталог установки наблюдаемой СУБД. По умолчанию `C:\Program Files\RedDatabase\`.
8. Укажите порт наблюдаемой СУБД. По умолчанию используется значение 3050.
9. Укажите пользователя СУБД с правами `RDB$ADMIN`, от имени которого агент мониторинга будет подключаться к наблюдаемой СУБД. По умолчанию используется `SYSDBA`.
10. Укажите через запятую пути к базам данных, которые нужно мониторить, или их алиасы.
11. Перезапустите наблюдаемую СУБД.
12. Убедитесь, что rdbmonitor-exporter (агент мониторинга) запущен.
13. Откройте в браузере страницу запущенной `Grafana` (по умолчанию `http://localhost:40300/`). По умолчанию для входа используется логин `admin` и пароль `admin`.

Агент мониторинга после установки работает от имени администратора на порту 40304. Изменить порт агента мониторринга можно по инструкции *Настройка порта агента мониторинга*.

Глава 3

Обновление РБДМонитор

3.1 Обновление на Linux

3.1.1 Обновление сервера мониторинга

1. Остановите сервисы мониторинга:

```
sudo systemctl stop grafana-server.service
sudo systemctl stop prometheus.service
sudo systemctl stop loki.service
sudo systemctl stop alertmanager.service
```

2. Удалите каталог установки РБД Монитор:

```
sudo rm -rf /opt/rdbmonitor
```

3. Скачайте с официального сайта установочный файл `rdbmonitor_server_installer_<версия>.bin`.
4. Назначьте установочному файлу права на выполнение:

```
chmod +x rdbmonitor_server_installer_<версия>.bin
```

5. Запустите установку от имени суперпользователя:

```
sudo ./rdbmonitor_server_installer_<версия>.bin
```

6. Укажите каталог установки сервера мониторинга. По умолчанию установится по пути `/opt/rdbmonitor/server`.
7. Укажите адреса наблюдаемых машин через запятую.
8. Компоненты сервера мониторинга после установки работают на следующих портах:
    - `Grafana` - 40300
    - `Prometheus` - 40301
    - `Alertmanager` - 40302

    После завершения установки настройте порты при необходимости по инструкции *Настройка портов сервера мониторинга*.

3.1.2 Обновление агента мониторинга

1. Сохраните лицензионный файл РБД Монитора, расположенный в `/opt/rdbmonitor/client/exporter/`.
2. Остановите наблюдаемую СУБД:

```
sudo sudo systemctl stop firebird
```

3. Остановите сервис агента мониторинга:

```
sudo systemctl stop rdbmonitor-exporter.service
```

4. Удалите каталог установки РБД Монитор:

```
sudo rm -rf /opt/rdbmonitor
```

5. Запустите наблюдаемую СУБД:

```
systemctl status firebird
```

6. Скачайте с официального сайта установочный файл `rdbmonitor_client_installer_<версия>.bin`.
7. Разместите лицензию и установочный файл в одном каталоге.
8. Назначьте установочному файлу права на выполнение:

```
chmod +x rdbmonitor_client_installer_<версия>.bin
```

9. Запустите установку от имени суперпользователя:

```
sudo ./rdbmonitor_client_installer_<версия>.bin
```

10. Укажите каталог установки агента мониторинга. По умолчанию установится по пути `rdbmonitor/client/exporter/`.
11. Укажите IP-адрес сервера мониторинга.
12. Укажите каталог установки наблюдаемой СУБД. По умолчанию `/opt/RedDatabase/`.
13. Укажите порт наблюдаемой СУБД. По умолчанию используется значение 3050.
14. Укажите пользователя СУБД с правами `RDB$ADMIN`, от имени которого агент мониторинга будет подключаться к наблюдаемой СУБД. По умолчанию используется `SYSDBA`.
15. Укажите через запятую пути к базам данных, которые нужно мониторить, или их алиасы.
16. Агент мониторинга по умолчанию работает на порту 40304. После завершения установки настройте порт агента мониторинга при необходимости по инструкции *Настройка порта агента мониторинга*.
17. Перезапустите наблюдаемую СУБД:

```
sudo systemctl restart firebird
```

3.2 Обновление на Windows

3.2.1 Обновление сервера мониторинга

1. Остановите службы loki, prometheus, alertmanager и grafana.
2. Удалите каталог установки РБД Монитор.
3. Скачайте с официального сайта установочный файл `rdbmonitor_server_installer_win_<версия>.exe`.
4. Запустите установку от имени администратора:
5. Укажите каталог установки сервера мониторинга. По умолчанию установится по пути `C:\Program Files\RedSoft\server`.
6. Укажите адреса наблюдаемых машин через запятую.

7. Компоненты сервера мониторинга после установки работают на следующих портах:
    - `Grafana` - 40300
    - `Prometheus` - 40301
    - `Alertmanager` - 40302

    После завершения установки настройте порты при необходимости по инструкции *Настройка портов сервера мониторинга*.

3.2.2 Обновление агента мониторинга

1. Сохраните лицензионный файл РБД Монитора, расположенный в по пути `<каталог установки РБД Монитор>\client\exporter`.
2. Остановите службу агента мониторинга.
3. Удалите каталог установки РБД Монитор.
4. Скачайте с официального сайта установочный файл `rdbmonitor_client_installer_win_<версия>.exe`.
5. Убедитесь, что наблюдаемая СУБД запущена.
6. Разместите лицензию и установочный файл в одном каталоге.
7. Запустите установку от имени администратора.
8. Укажите каталог установки агента мониторинга. По умолчанию установится по пути `C:\Program Files\RedSoft\client\exporter`.
9. Укажите IP-адрес сервера мониторинга.
10. Укажите каталог установки наблюдаемой СУБД. По умолчанию `C:\Program Files\RedDatabase\`.
11. Укажите порт наблюдаемой СУБД. По умолчанию используется значение 3050.
12. Укажите пользователя СУБД с правами `RDB$ADMIN`, от имени которого агент мониторинга будет подключаться к наблюдаемой СУБД. По умолчанию используется `SYSDBA`.
13. Укажите через запятую пути к базам данных, которые нужно мониторить, или их алиасы.
14. Агент мониторинга по умолчанию работает на порту 40304. После завершения установки настройте порт агента мониторинга при необходимости по инструкции *Настройка порта агента мониторинга*.
15. Перезапустите наблюдаемую СУБД.

Глава 4

Удаление РБДМонитор

4.1 Удаление на Linux

Удаление сервера мониторинга

1. Запустите деинсталлятор от имени суперпользователя:

```
sudo ./uninstaller
```

Удаление агента мониторинга

1. Запустите деинсталлятор от имени суперпользователя:

```
sudo ./uninstaller
```

2. Для удаления плагина мониторинга запросов перезапустите наблюдаемую СУБД:

```
sudo systemctl restart firebird
```

4.2 Удаление на Windows

Удаление сервера мониторинга

1. Запустите деинсталлятор uninstaller.exe от имени администратора.

Удаление агента мониторинга

1. Запустите деинсталлятор uninstaller.exe от имени администратора.
2. Для удаления плагина мониторинга запросов перезапустите наблюдаемую СУБД.

Глава 5

Настройка РБДМонитор

РБДМонитор состоит из агента мониторинга и сервера мониторинга. Агент мониторинга необходимо установить на одну машину с наблюдаемой СУБД. Сервер мониторинга можно установить как на одну машину с агентом, так и на отдельную.

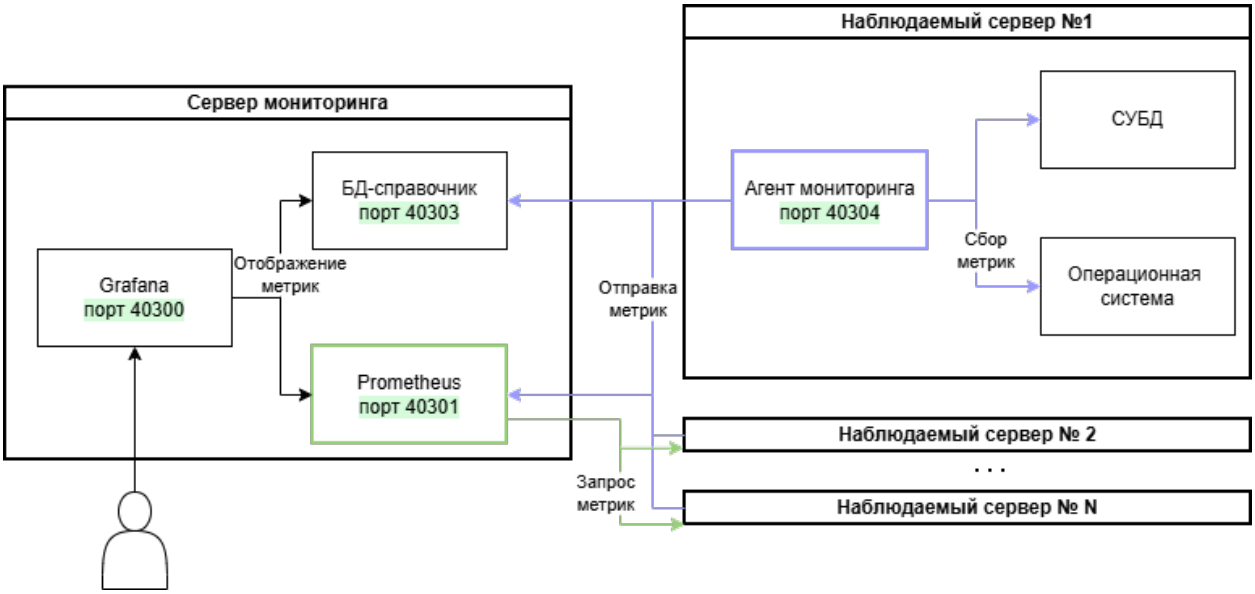


Рисунок 5.1 — Схема взаимодействия компонентов РБДМонитора

Таблица 5.1 — Описание компонентов с параметрами по умолчанию

Название	Порт	Служба
Агент мониторинга	40304	rdbmonitor-exporter
Grafana	40300	rdbmonitor-grafana
Prometheus	40301	rdbmonitor-prometheus
Alertmanager	40302	rdbmonitor-alertmanager
Внутренняя СУБД	40303	rdbmonitor-dbms

Служба сервера мониторинга - `rdbmonitor-server.target`. Все компоненты сервера мониторинга работают от имени пользователя `rdbmonitor_server`. Служба агента мониторинга - `rdbmonitor-exporter`. Агент мониторинга работает от имени суперпользователя.

Расположение конфигурационных файлов по умолчанию:

- `Агент мониторинга`

  На Windows: C:\Program Files\RedSoft\rdbmonitor\client\exporter\exporter_conf.json

  На Linux: /opt/rdbmonitor/client/exporter/exporter_conf.json

- `Grafana`

  На Windows: C:Program FilesRedSoftrdbmonitorservergrafanagrafana.ini

  На Linux: /opt/rdbmonitor/server/grafana/grafana.ini

- `Prometheus`

  На Windows: C:Program FilesRedSoftrdbmonitorserverprometheusprometheus.yml

  На Linux: /opt/rdbmonitor/server/prometheus/prometheus.yml
- `Alertmanager`

  На Windows: C:Program FilesRedSoftrdbmonitorseveralertmanageralertmanager.yml

  На Linux: /opt/rdbmonitor/server/alertmanager/alertmanager.yml

5.1 Настройка портов РБД Монитор

5.1.1 Наблюдаемая СУБД

Агент мониторинга по умолчаниют следит за СУБД с портом 3050. Порт наблюдаемой СУБД настраивается в конфигурационном файле `exporter_conf.json`: Расположение `exporter_conf.json`:

- На Linux - по умолчанию в `/opt/rdbmonitor/client/exporter`
- На Windows - `C:\Program Files\RedSoft\rdbmonitor\client\exporter`

Настройка порта наблюдаемой СУБД

1. Укажите порт наблюдаемой СУБД в в параметре `port` секции `RedDatabase`:

```
...
"RedDatabase": {
  "host": "<хост СУБД>",
  "port": <порт наблюдаемой СУБД>,
  "user": "<логин>",
  "password": "<пароль>",
  "folder_path": "<путь к папке установки RedDatabase>"
},
...
```

2. Перезапустите агент мониторинга.

   На `Linux`:

```
systemctl restart rdbmonitor-exporter
```

   На `Windows` перезапустите службу `rdbmonitor-exporter` в диспетчере задач или в консоли, выполнив от имени администратора следующие команды:

```
net stop rdbmonitor-exporter
net start rdbmonitor-exporter
```

5.1.2 Настройка порта агента мониторинга

По умолчанию у агента мониторинга (`rdbmonitor-exporter`) порт 40304. Порт агента мониторинга настраивается в конфигурационном файле `exporter_conf.json`. Расположение `exporter_conf.json`:

- На Linux - по умолчанию в `/opt/rdbmonitor/client/exporter`
- На Windows - `C:\Program Files\RedSoft\rdbmonitor\client\exporter`

1. Укажите порт агента мониторинга в параметре `port` секции `exporter`:

```
"exporter": {
  "port": <порт агента мониторинга>,
  "rdb_lock_print": {
    "enabled": true
  },
...
```

2. Перезапустите агент мониторинга.

   На `Linux`:

   ```
   systemctl restart rdbmonitor-exporter
   ```

   На `Windows` перезапустите службу `rdbmonitor-exporter` в диспетчере задач или в консоли, выполнив от имени администратора следующие команды:

   ```
   net stop rdbmonitor-exporter
   net start rdbmonitor-exporter
   ```

3. Перейдите на сервер мониторинга.
4. Укажите порта агента мониторинга в настройках `Prometheus``в файле ``prometheus.yml`. Настройте `targets` для группы с именем `RDBMonitor`:

   ```
   scrape_configs:
     - job_name: "RDBMonitor"
       static_configs:
         - targets: ["<хост агента>:<порт агента>", ..., "<хост агента>:<порт
   агента>"]
   ```

5. Перезапустите сервер мониторинга.

   На `Linux`:

   ```
   systemctl restart rdbmonitor-server.target
   ```

   На `Windows` перезапустите службу `rdbmonitor-server` в диспетчере задач.

5.1.3 Настройка портов сервера мониторинга

Порт Prometheus

По умолчанию `Prometheus` из поставки РБД Монитор порт 40301. Порт `Prometheus` настраивается в конфигурационном файле в файле `prometheus.yml`. Расположение `prometheus.yml`:

- На Linux - по умолчанию в `/opt/rdbmonitor/server/prometheus/prometheus.yml`
- На Windows - по умолчанию в `C:\Program Files\RedSoft\rdbmonitor\server\prometheus\prometheus.yml`

Настройка порта Prometheus

1. Укажите порт `Prometheus` в файле `prometheus.yml`. Настройте `targets` для группы с именем `prometheus`:

   ```
   scrape_configs:
     - job_name: "prometheus"
   ```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
    static_configs:
      - targets: ["localhost:40301"]
```

2. Перезапустите сервер мониторинга.

   На Linux:

   ```
   systemctl restart rdbmonitor-server.target
   ```

   На Windows перезапустите службу rdbmonitor-server в диспетчере задач.

3. Укажите новый порт Prometheus в Grafana.

   Откройте в браузере страницу Grafana. Перейдите в Connections → Data sources. Из списка источников выберите Prometheus. Укажите новый порт в URL-адресе для доступа к Prometheus (по умолчанию http://localhost:40301/). Нажмите на кнопку Save & test.

Порт Alertmanager

По умолчанию Alertmanager из поставки РБД Монитор порт 40302. Порт Alertmanager настраивается в конфигурационном файле в файле alertmanager.yml. Расположение alertmanager.yml:

- На Linux - по умолчанию в /opt/rdbmonitor/server/prometheus/alertmanager.yml
- На Windows - по умолчанию в C:\Program Files\RedSoft\rdbmonitor\server\prometheus\alertmanager.yml

Настройка порта Alertmanager

1. Измените порт Alertmanager командой:

   ```
   --web.limits-address=:<новый порт>
   ```

2. Укажите новый порт Alertmanager в файле alertmanager.yml. Настройте targets для секции alerting:

   ```
   alerting:
       alertmanagers:
       - static_configs:
           - targets: ["localhost:порт alertmanager"]
   ```

3. Перезапустите сервер мониторинга.

   На Linux:

   ```
   systemctl restart rdbmonitor-server.target
   ```

   На Windows перезапустите службу rdbmonitor-server в диспетчере задач.

Порт Grafana

По умолчанию Grafana из поставки РБД Монитор порт 40300. Порт Grafana настраивается в конфигурационном файле в файле grafana.ini. Расположение grafana.ini:

- На Linux - по умолчанию в /opt/rdbmonitor/server/grafana/grafana.ini
- На Windows - по умолчанию в C:\Program Files\RedSoft\rdbmonitor\server\grafana\grafana.ini

Настройка порта Grafana

1. Укажите порт Grafana в файле grafana.ini. Настройте параметр http_port в секции server:

```
[server]
http_port = <порт>
```

2. Перезапустите сервер мониторинга.

   На Linux:

   ```
   systemctl restart rdbmonitor-server.target
   ```

   На Windows перезапустите службу rdbmonitor-server в диспетчере задач.

5.2 Настройки наблюдаемой СУБД

В firebird.conf укажите TracePlugin = messagetrace. Для использования нескольких плагинов трассировки необходимо перечислить их через запятую, например, TracePlugin = fbtrace, messagetrace.

В plugins.conf необходимо включить плагин агрегатного аудита:

> Для РЕД Базы Данных нужно указывать библиотеку messagetrace. Для Firebird - messagetrace_fb.

```
Plugin = messagetrace
{
      Module = /opt/rdbmonitor/client/messagetrace_[fb]
      Required = true
      Config = monitor_config
}

Config = monitor_config {
  ConfigFile = /opt/rdbmonitor/client/exporter/exporter_conf.json
}
```

После изменения настроек нужно перезапустить наблюдаемую СУБД:

На Linux:

```
systemctl start firebird
```

5.3 Агент мониторинга

После изменения настроек нужно перезапустить агента мониторинга:

На Linux:

```
systemctl restart rdbmonitor-exporter
```

На Windows перезапустите службу rdbmonitor-exporter в диспетчере задач или в консоли, выполнив от имени администратора следующие команды:

```
net stop rdbmonitor-exporter
net start rdbmonitor-exporter
```

Настройка агента мониторинга осуществляется в `exporter_conf.json`. Расположение конфигурационного файл агента мониторинга:

- На Linux - по умолчанию в `/opt/rdbmonitor/client/exporter/exporter_conf.json`
- На Windows - `C:\Program Files\RedSoft\rdbmonitor\client\exporter\exporter_conf.json`

Конфигурационный файл агента мониторинга:

```
{
 "exporter": {
    "port": <порт агента мониторинга>,
    "rdb_lock_print": {
      "enabled": true
    },
    "scrape_system":{
      "enabled": true
    },
    "scrape_queries": {
      "enabled": true,
      "filters": "<путь к файлу с настройками фильтрации>",
      "default":
        {
          "shmem_size":<размер разделяемой памяти в МиБ>
        }
    },
    "scrape_mon_tables":{
      "enabled": true
    },
    "user_metrics": {
      "enabled": true,
      "timeout": <таймаут сбора метрики в секундах>,
      "max_metric_labels": <максимальное кол-во метрик>,
      "max_user_metrics": <максимальное кол-во пользовательских метрик>,
      "max_metric_data_blocks": <максимальное кол-во блоков метрики>
    }
 },
 "dictserver": {
    "host": "<хост базы данных>",
    "port": <порт базы данных>,
    "database": "/opt/rdbmonitor/server/rdb/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>"
 },
 "databases": {
    "<метка для отображения>": {
      "alias": "<алиас>",
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
      "scrape": {
        "enabled": true,
        "filters":"<путь к файлу с настройками фильтрации>",
        "shmem_size":<размер разделяемой памяти в МиБ>
      }
    }
  }
}
```

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 агент мониторинга не запустится. Параметры секции:

- `port` - Порт, на котором работает агент мониторинга, по умолчанию 40304. 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 информацио об операционной системе. По умолчанию установлено значение `true`.
- `scrape_mon_tables` - Определяет, собирать ли данные из таблиц мониторинга. По умолчанию установлено значение `true`.
- `scrape_queries` - Определяет, собирать ли данные с помощью агрегирующего трейса. По умолчанию установлено значение `true`.

  Необязательная опция `shmem_size` указывает размер разделяемой памяти в МиБ. Допустимы целочисленные значения от 128 до 1024. По умолчанию `shmem_size = 128`.

  Необязательная опция `filters` определяет путь к предварительно созданному файлу с настройками фильтрации `filters.ini`. Собираться будут только запросы, характеристики которых превышают хотя бы одно из указанных значений параметров в `filters.ini`. Например, если `time = 100`, то соберутся только запросы, выполняющиеся дольше 100 миллисекунд. Указанные здесь параметры фильтрации распространяются на все отслеживаемые базы данных. Настроить фильтры для конкретной базы данных можно в секции `databases`.

  Структура `filters.ini`:

  ```
  [limits.statements]
  <параметр>
  ...
  <параметр>
  ```

  Параметры фильтрации:

  - `time` - время выполнения запроса в миллисекундах;
  - `prepare` - количество подготовленных запросов (событий prepare);
  - `prepare failed` - количество запросов, подготовка которых завершилась ошибкой;
  - `start` - количество запущенных запросов (событий start);
  - `start failed` - количество неудачно запустившихся запросов;
  - `finish` - количество завершённых запросов (событий finish);

  - `finish failed` - количество неудачно завершённых запросов;
  - `free` - количество освобождённых запросов (событий free);
  - `free failed` - количество неудачно освобождённых запросов;
  - `reads` - количество страниц, считанных с диска;
  - `marks` - количество отмеченных (mark) страниц;
  - `writes` - количество страниц, записанных на диск;
  - `fetches` - количество страниц, считанных из страничного кэша (fetch);
  - `mem use by sort in total` - количество байт, выделенных под сортировку;
  - `mem use by sort in cache` - количество байт, выделенных под сортировку в кеше;
  - `mem use by sort in disk` - количество байт, выделенных под сортировку на диске.

  Значения всех параметров фильтрации должны быть целым положительным числом.

  Пример настройки `filters.ini`, где будут собираться запросы, которые выполнялись дольше 60 секунд или более 100 раз

```
[limits.statements]
time = 60000
finish = 100
```

- `user_metrics` - определяет, собирать ли пользовательские метрики. По умолчанию выключено (`"enabled": false`). Опция `timeout` указывает таймаут в секундах, по истечении которого метрика перестанет запрашиваться, значение должно быть дробным числом (по умолчанию значение 0.1). Опция `max_metric_labels` определяет максимальное количество собираемых метрик (по умолчанию значение 5). Опция `max_user_metrics` максимальное количество пользовательских метрик (по умолчанию значение 100). Опция `max_metric_data_blocks` максимальное количество блоков метрики (по умолчанию значение 10).

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 40303; параметр имеет целочисленный тип;
- `database` - псевдоним базы данных пользовательских запросов;
- `user` - имя пользователя, от которого агент мониторинга 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; по умолчанию используются плагины `Srp` и `Srp256`.

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 агент мониторинга будет подключаться к СУБД; же-

лательно указать 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`, список плагинов, которые используются для аутентификации, будет взят из `firebird.conf`, расположенному в указанной папке;

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

```
"databases": {
 "<метка для отображения>": {
  "alias": "<алиас>",
  "scrape": {
    "enabled": true,
     "filters":"filters_employee.ini"
  }
 },
 ...
}
```

Где:

- `метка для отображения` - метка базы данных, которая будет отображаться в `Grafana`, можно указать любое название;
- `alias` - псевдоним базы данных, который используется СУБД;
- `enabled` - логический параметр, определяющий отслеживать ли указанную базу данных;
- `filters` - необязательная опция, которая определяет путь к предварительно созданному файлу с настройками фильтрации `filters.ini`. Собираться будут только запросы, характеристики которых превышают указанные значения параметров в `filters.ini`. Например, если `time = 100`, то соберутся только запросы, выполняющиеся дольше 100 миллисекунд. Параметры фильтрации распостраняются на указанную базу данных.

  Пример настроенного файла `exporter_conf.json`:

```
{
    "exporter": {
        "port": 40304,
        "rdb_lock_print": {
           "enabled": true
        },
        "scrape_system": {
           "enabled": true
        },
        "scrape_queries": {
           "enabled": true,
           "filters": "/opt/rdbmonitor/client/exporter/filters.ini"
        },
        "scrape_mon_tables": {
           "enabled": true
        },
        "user_metrics": {
           "enabled": false,
           "timeout": 0.1,
           "max_metric_labels": 5,
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
            "max_user_metrics": 100,
            "max_metric_data_blocks": 10
        }
    },
    "dictserver": {
        "host": "127.0.0.1",
        "port": 40303,
        "database": "/opt/rdbmonitor/server/rdb/statements.fdb",
        "user": "SYSDBA",
        "password": "masterkey",
        "Auth_plugins": ["Srp", "Srp256"]
    },
    "RedDatabase": {
        "host": "127.0.0.1",
        "port": 40303,
        "user": "SYSDBA",
        "password": "masterkey",
        "folder_path": "/opt/RedDatabase"
    },
    "databases": {
        "employee_db": {
           "alias": "employee",
            "scrape": {
              "enabled": true,
              "filters":"filters_employee.ini"
            }
        }
    }
}
```

5.4 Сервер мониторинга

После изменения настроек нужно перезапустить сервер мониторинга:

На `Linux`:

```
systemctl restart rdbmonitor-server.target
```

На `Windows` перезапустите службу `rdbmonitor-server` в диспетчере задач.

5.4.1 Prometheus

`Prometheus` настраивается в конфигурационном файле в файле `prometheus.yml`. Расположение `prometheus.yml`:

- На Linux - по умолчанию в `/opt/rdbmonitor/server/prometheus/prometheus.yml`
- На Windows - по умолчанию в `C:\Program Files\RedSoft\rdbmonitor\server\prometheus\prometheus.yml`

Конфигурационный файл `prometheus.yml`:

```
global:
    scrape_interval: <интервал сбора метрик><единицы измерения s/m/h>
    scrape_timeout: <время ожидания получения в секунду>

scrape_configs:
    - job_name: "RDBMonitor"
      static_configs:
        - targets: ["<хост агента>:<порт агента>", ..., "<хост агента>:<порт агента>"]
    - job_name: "prometheus"
      static_configs:
        - targets: ["localhost:40301"]
    - job_name: "cook" # Секция cook необходима, только при наличии отказоустойчивого
кластера, в других случаях опустите её
      static_configs:
        - targets: ["<хост узла>:<порт узла>", ..., "<хост узла>:<порт узла>"]

rule_files:
    - rules.yml
    - cluster_rules.yml

alerting:
    alertmanagers:
    - static_configs:
        - targets: ["localhost:40302"]
```

Параметр `scrape_interval` определяет, как часто нужно собирать метрики. Допустимые единицы измерения: s - секунды, m - минуты, h - часы.

Параметр `scrape_timeout` устанавливает время ожидания получения метрик.

Параметр `job_name` задаёт имя группы агентов мониторинга. Агенты мониторинга, входящие в группу, определяются в параметре `targets`. Изменять значение параметра `job_name` нельзя.

Параметр `targets` определяет сетевые узлы агентов мониторинга, к которым будет обращаться `Prometheus` для сбора метрик. Портом агента мониторинга по умолчанию является 40304. Портом для сбора метрик отказоустойчивого кластера по умолчанию является 5030.

В секции `rule_files` указывается список файлов с правилами, на соответствие которым нужно проверять собранные метрики. Если нет необходимости в уведомлениях, то эту секцию нужно оставить закомментированной. Базовый набор правил `rules.yml` поставляется с РБДМонитор.

Секция `alerting` определяет настройки уведомлений. Если нет необходимости в уведомлениях, то эту секцию нужно оставить закомментированной. В секции указываются сущности `Alertmanager`, в которые `Prometheus` будет отправлять предупреждения, когда собранные метрики соблюдают правила, указанные в секции `rule_files`.

Подробнее о настройке `Prometheus` см. документацию на официальном сайте.

5.4.2 Alertmanager

`Alertmanager` настраивается в конфигурационном файле в файле `alertmanager.yml`. Расположение `alertmanager.yml`:

- На Linux - по умолчанию в `/opt/rdbmonitor/server/prometheus/alertmanager.yml`
- На Windows - по умолчанию в `C:\Program Files\RedSoft\rdbmonitor\server\prometheus\alertmanager.yml`

Пример настройки:

```
global:
route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 10m
  repeat_interval: 60m
  receiver: 'email'
receivers:
  - name: 'email'
    email_configs:
      - to: 'адрес_электронной_почты_получателя'
        from: 'адрес_электронной_почты_отправителя'
        smarthost: 'SMTP_хост:порт'
        auth_username: 'имя_пользователя'
        auth_identity: 'имя_пользователя'
        auth_password: 'пароль'
```

Более подробно о настройке `Alertmanager` можно узнать в официальной документации.

5.4.3 Настройка уведомлений на почту

1. Перейдите на сервер мониторинга. В `prometheus.yml` настройте секции `alerting` и `rule_files`.

   ```
   rule_files:
       - rules.yml

   alerting:
       alertmanagers:
       - static_configs:
           - targets: ["localhost:40302"]
   ```

   В `rule_files` укажите список файлов, определяющих правила для отправки уведомлений. Можно использовать базовый набор правил `rules.yml`, поставляемый вместе с РБДМонитор.

   В `targets` укажите адрес `Alertmanager`, куда `Prometheus` будет отправлять уведомления.

2. Настройте отправку уведомлений в `alertmanager.yml`:

   ```
   global:
     smtp_smarthost: 'smtp.example.com:587'
     smtp_from: 'monitor@example.com'
     smtp_auth_username: 'monitor@example.com'
     smtp_auth_password: '<пароль>'

   route:
     receiver: email-admin

   receivers:
     - name: email-admin
       email_configs:
         - to: 'admin@example.com'
   ```

Параметры, которые необходимо настроить:

- `smtp_smarthost` - адрес SMTP-сервера и порт
- `smtp_from` - адрес отправителя
- `smtp_auth_username` - логин для SMTP
- `smtp_auth_password` - пароль SMTP
- `to` - адрес получателя уведомлений

3. Перезапустите сервер мониторинга.

   На `Linux`:

```
systemctl restart rdbmonitor-server.target
```

   На `Windows` перезапустите службу `rdbmonitor-server` в диспетчере задач.

5.4.4 Правила уведомлений

Для включения уведомлений необходимо составить правила, по которым `Prometheus` будет проверять метрики и формировать уведомление, если показатели системы выходят за указанные пороги.

Базовый набор правил `rules.yml` поставляется с РБДМонитор. Набор правил уведомлений, поставляемый вместе с РБДМонитор в `rules.yml`:

- Высокое потребление процессора в режиме ядра (СУБД);
- Высокое потребление процессора (СУБД);
- Ухудшение производительности (СУБД);
- Увеличение нагрузки (СУБД);
- Среднее время выполнения запроса увеличилось (СУБД);
- Высокая нагрузка на процессор (Сервер);
- Насыщение процессора (Сервер);
- Мало места в файловой системе (Сервер);
- Высокая утилизация диска по времени (Сервер);
- Высокое потребление памяти (Сервер);
- Высокое использование объёма подкачки (Сервер);
- Наличие процесса подкачки (Сервер);
- Агент мониторинга не ответчает (Агент мониторинга).

5.4.5 Grafana

Откройте в браузере страницу запущенной `Grafana` (по умолчанию http://localhost:40300/). По умолчанию для входа используется логин `admin` и пароль `admin`.

Укажите источники данных:

1. Во вкладке `Connections` → `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:40301/`). Необходимо настроить интервал сбора метрик. Для параметра `Scrape interval` укажите число, совпадающее со значением параметра `scrape_interval` в конфигурационном файле `Prometheus` (`prometheus.yml`). Нажмите на кнопку `Save & test`.
2. Укажите 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 `РЕД База Данных`. Заполните поля для соединения с базой данных и

нажмите на кнопку `Save & test`.

Для отображения собранной информации необходимо включить плагин `РБДМонитор`:

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

После этого в меню `Apps` будет добавлен плагин `РБДМонитор`.

Необходимо указать добавленные источники данных в настройках плагина `РБДМонитор`:

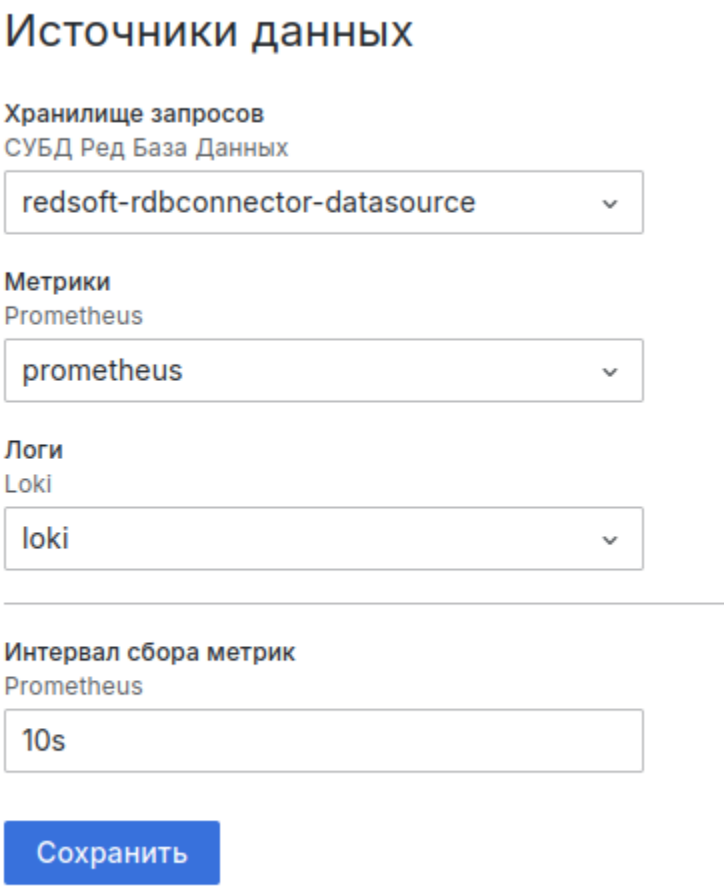


Рисунок 5.2 — Настройка плагина

Глава 6

Управление мониторингом

6.1 Наблюдаемые СУБД

Добавление новой СУБД в мониторинг:

1. Установите агент мониторинга по инструкции.
2. На сервере мониторинга в каталоге установки Prometheus (rdbmonitor/server/prometheus/targets/) создайте файл `N.json` со следующим содержанием:

```
[{"targets": ["<ip-адрес наблюдаемого сервера>:<порт агента мониторинга>"]}]
```

6.2 Наблюдаемые базы данных

Добавление баз данных в мониторинг и исключение их из списка наблюдаемых осуществляется настройкой агента мониторинга на сервере наблюдаемой СУБД.

Расположение конфигурационного файл агента мониторинга:

- На Linux - по умолчанию в `/opt/rdbmonitor/client/exporter/exporter_conf.json`
- На Windows - `C:\Program Files\RedSoft\rdbmonitor\client\exporter\exporter_conf.json`

1. В `exporter_conf.json` укажите список отслеживаемых баз данных в секции `databases`:

```
...

  "databases": {
    "<отображаемая метка бд>": {
      "alias": "<псевдоним бд или путь>",
      "scrape": {
        "enabled": true
      }
    }
    "new_db": {          # Метка наблюдаемой базы для отображения, можно
указать любое название
      "alias": "new_db", # Путь к наблюдаемой базе данных или алиас бд
      "scrape": {
        "enabled": true
      }
  }
...
```

Секция `databases` определяет список баз данных для мониторинга. Параметры секции:

- `метка для отображения` - метка базы данных, которая будет отображаться в `Grafana`, можно указать любое название;
- `alias` - псевдоним базы данных, который используется СУБД;
- `enabled` - логический параметр, определяющий отслеживать ли указанную базу данных;
- `filters` - необязательная опция, которая определяет путь к предварительно со-

зданному файлу с настройками фильтрации `filters.ini`. Собираться будут только запросы, характеристики которых превышают указанные значения параметров в `filters.ini`. Например, если `time = 100`, то соберутся только запросы, выполняющиеся дольше 100 миллисекунд. Параметры фильтрации распостраняются на указанную базу данных.

2. Перезапустите агент мониторинга.

   На `Linux`:

   ```
   systemctl restart rdbmonitor-exporter
   ```

   На `Windows` перезапустите службу `rdbmonitor-exporter` в диспетчере задач или в консоли, выполнив от имени администратора следующие команды:

   ```
   net stop rdbmonitor-exporter
   net start rdbmonitor-exporter
   ```

6.3 Управление источниками данных

Включение и выключение сбора метрик с источника данных осуществляется настройкой агента мониторинга на сервере наблюдаемой СУБД.

Расположение конфигурационного файл агента мониторинга (`exporter_conf.json`):

- На Linux - по умолчанию в `/opt/rdbmonitor/client/exporter`
- На Windows - `C:\Program Files\RedSoft\rdbmonitor\client\exporter`

Источники данных для мониторинга:

- Вывод утилиты rdb_lock_print
- Операционная система
- Выполнение запросов

Настройка сбора метрик с источников данных:

1. В `exporter_conf.json` укажите значение `"enabled": true` для источников, с которых нужно собирать метрики и `false` в обратном случае:

   ```
   ...
   "rdb_lock_print": { # Сбор метрик вывода rdb_lock_print
     "enabled": true
   },
   "scrape_system":{ # Сбор метрик операционной системы
     "enabled": true
   },
   "scrape_queries": { # Сбор метрик выполнений запросов
     "enabled": true,
     "filters": "<путь к файлу с настройками фильтрации>"
   },
   ...
   ```

2. Перезапустите агент мониторинга.

   На `Linux`:

   ```
   systemctl restart rdbmonitor-exporter
   ```

На `Windows` перезапустите службу `rdbmonitor-exporter` в диспетчере задач или в консоли, выполнив от имени администратора следующие команды:

```
net stop rdbmonitor-exporter
net start rdbmonitor-exporter
```

6.4 Формирование сравнительного отчёта

РБДМонитор позволяет выполнять сравнение собранных метрик за два указанных периода времени. Созданные отчёты можно просматривать на дашборде Grafana и скачать в формате HTML.

Инструкция по генерации отчёта:

1. Выберите меню `Apps` → `РБДМонитор` → `Отчёты`.
2. Перейдите во вкладку `Генерация отчтёта` и заполните все поля:
   - `Разделяющая метка` - в отчёте будут сравниваться периоды до и после указаной даты и времениж; дата и время в формате <гггг-мм-дд> <чч-мм-сс> (например, 2026-10-19 12:16:43)
   - `Продолжительность` - определяет периоды до и после разделяющей метки в часах (ч) или минутах (м) для сравнения; например, со значением 24ч будут сравниваться значения за сутки до разделяющей метки с сутками после неё.
   - `Допустимая доля изменений в %` - если хотя бы одна метрика запроса превышает указанный процент, то запрос попадёт в отчёт.
3. Нажмите `Сгенерировать отчёт`.

> Не покидайте и не обновляйте страницу до конца формирования отчёта.

После формирования отчёта появится возможность его экспортировать. Сформированный отчёт будет добавлен на вкладку `Отчёты`.

6.5 Мониторинг при миграции РЕД Базы Данных 3 на РЕД Базу Данных 5

Для сохранения данных мониторинга РЕД Базы Данных 3 и продолжения наблюдения на РЕД Базе Данных 5 выполните следующее:

1. Остановите агент мониторинга:

   ```
   systemctl stop rdbmonitor-exporter
   ```

2. Выполните миграцию наблюдаемой СУБД на РЕД Базу Данных 5, следуя инструкции в Руководстве администратора.

   > Если агент мониторинга и сервер мониторинга установлены на одну машину, то необходимо мигрировать `statements.fdb` на РЕД Базу Данных 5.

3. Запустите агент мониторинга:

   ```
   systemctl start rdbmonitor-exporter
   ```

Глава 7

Просмотр метрик

Собираемые метрики описаны в приложении *Собираемые метрики*.

7.1 Обзорная страница

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

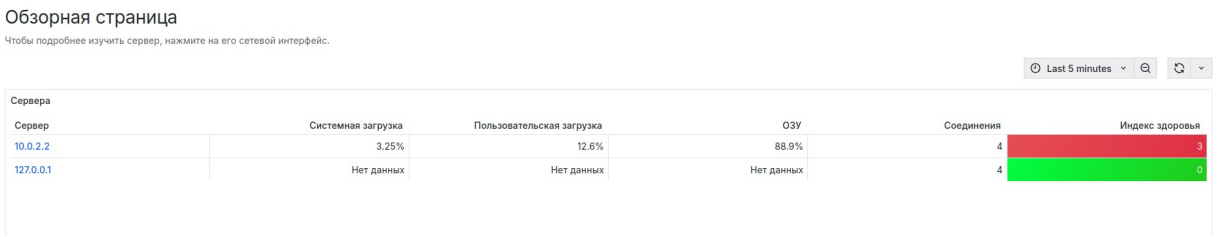


Рисунок 7.1 — Обзорная страница

7.2 Информация о выбранном сервере

На этой странице находится подробная информация о выбранном сервере.

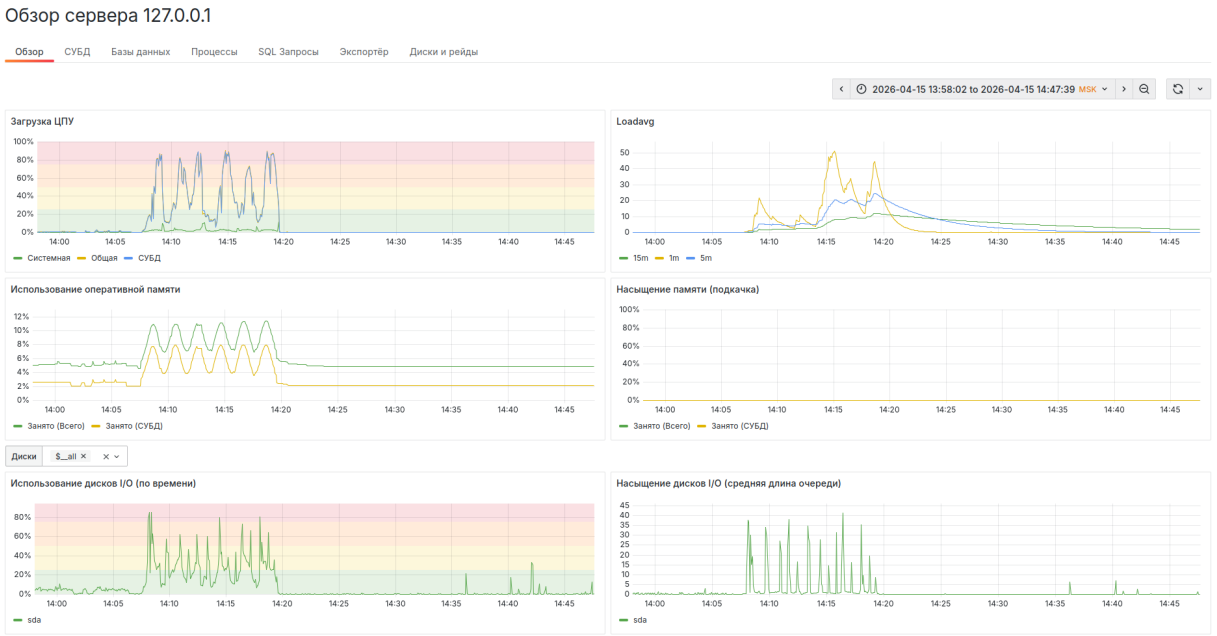


Рисунок 7.2 — Обзор выбранного сервера - Часть 1

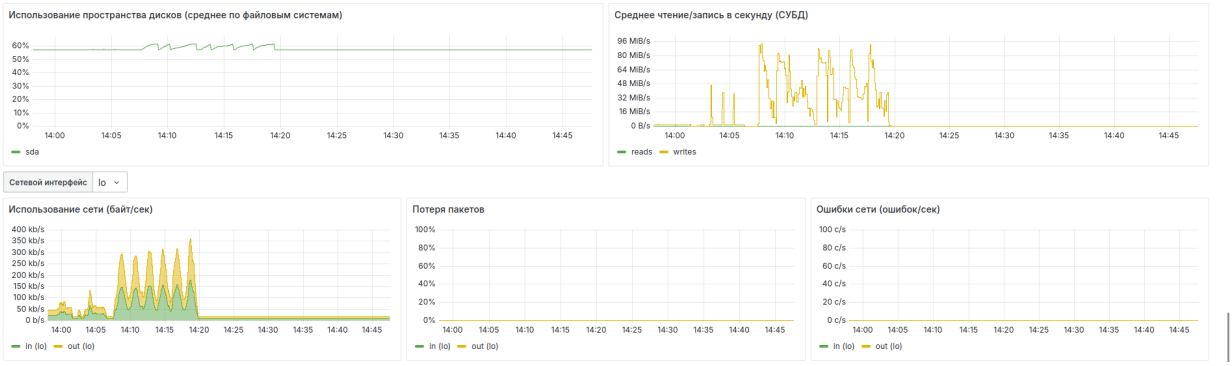


Рисунок 7.3 — Обзор выбранного сервера - Часть 2

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

График `Использование оперативной 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.

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 `Потеря пакетов` показывает процент потери пакетов в зависимости от времени.

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

7.2.1 СУБД

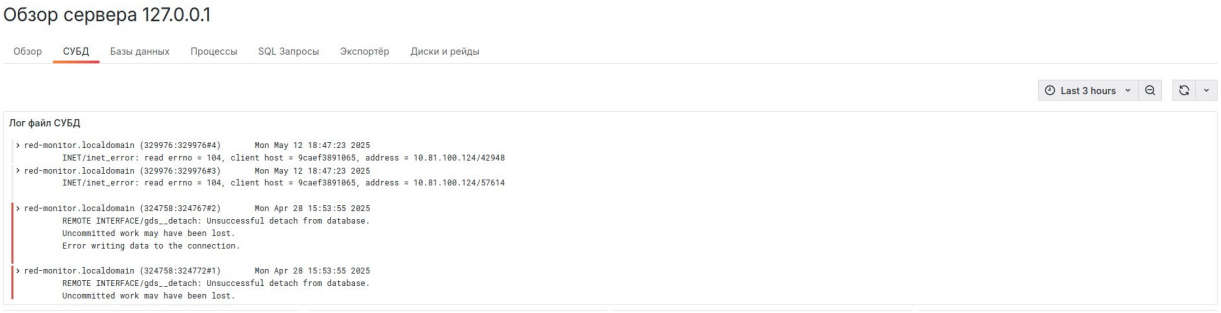


Рисунок 7.4 — Лог СУБД

Панель `Лог файл СУБД` показывает содержание файла `firebird.log`.



Рисунок 7.5 — 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.

7.2.2 Базы данных

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

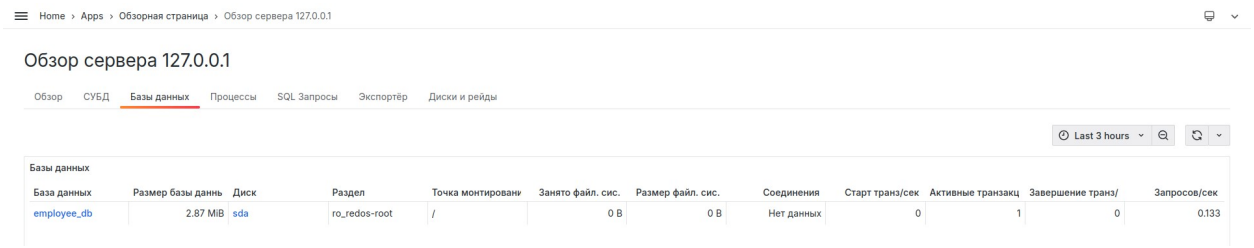


Рисунок 7.6 — Обзор баз данных

7.2.3 Информация о выбранной базе данных

На этой странице находится подробная информация о базе данных.

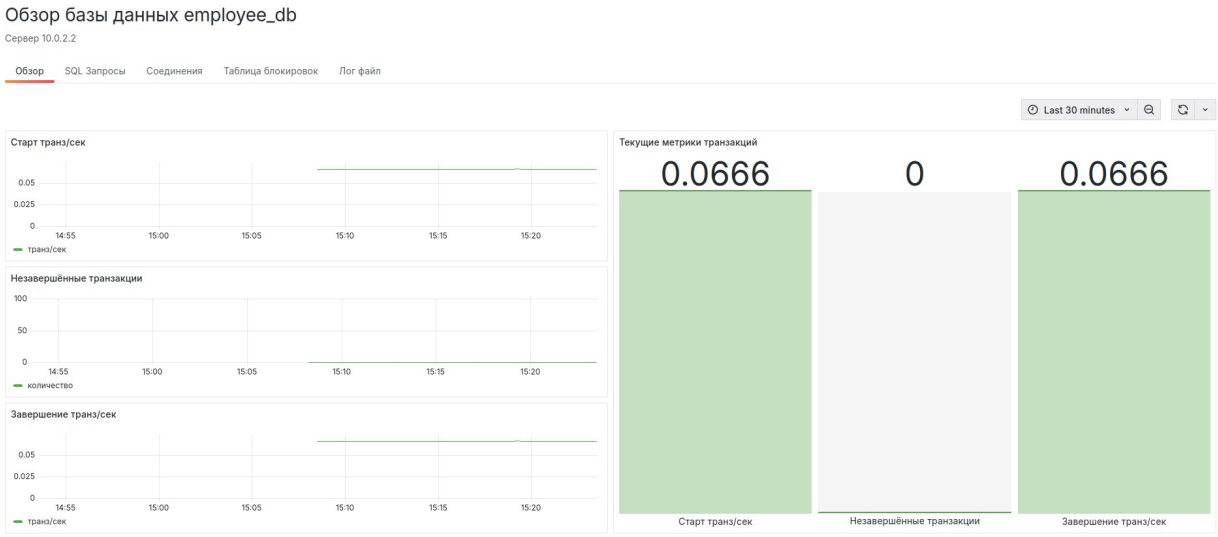


Рисунок 7.7 — Обзор выбранной базы данных - Часть 1

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 7.8 — 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 `Запросов/сек` показывает скорость возникновения запросов (количество запросов в секунду) в зависимости от времени.

График `Маркеры транзакций` отражает значения маркеров транзакций.

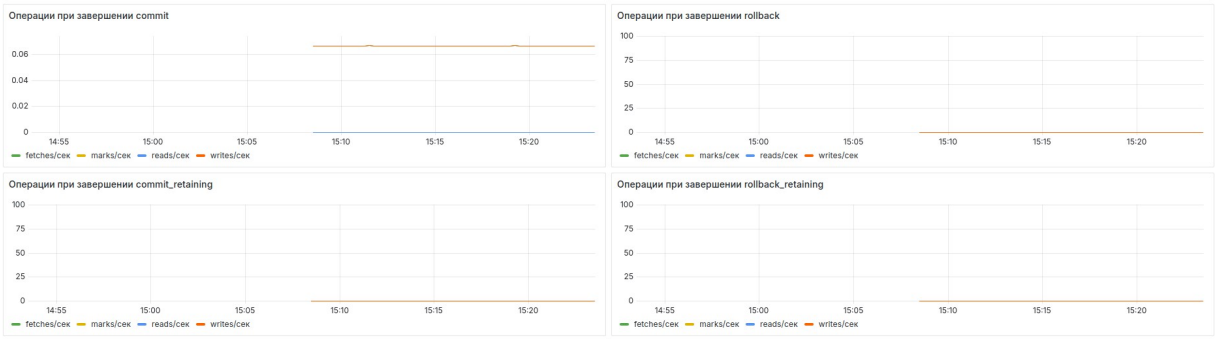


Рисунок 7.9 — Обзор выбранной базы данных - Часть 3

График `Операции при завершении commit` показывает количество операций `reads`, `fetches`, `marks`, `writes`, выполненных за секунду, при завершении транзакции `commit`.

График `Операции при завершении rollback` показывает количество операций `reads`, `fetches`, `marks`, `writes`, выполненных за секунду, при завершении транзакции `rollback`.

График `Операции при завершении commit_retaining` показывает количество операций `reads`, `fetches`, `marks`, `writes`, выполненных за секунду, при завершении транзакции `commit_retaining`.

График `Операции при завершении rollback_retaining` показывает количество операций `reads`, `fetches`, `marks`, `writes`, выполненных 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.

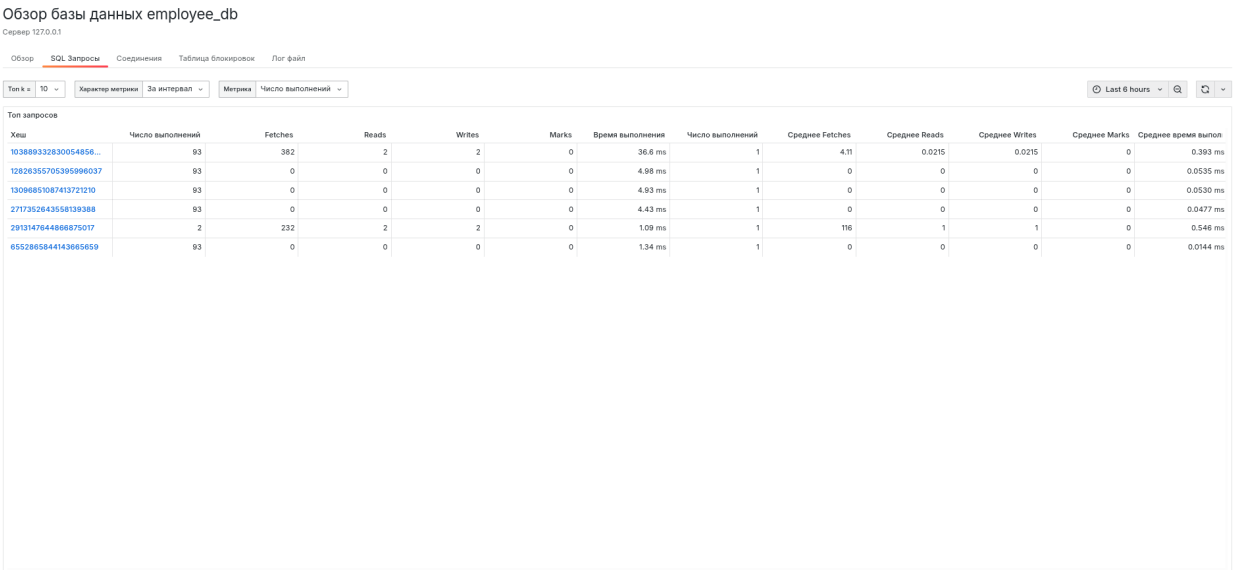


Рисунок 7.10 — SQL запросы выбранной базы данных

Обзор запроса

На данной странице находится подробная информация о выбранном запросе.

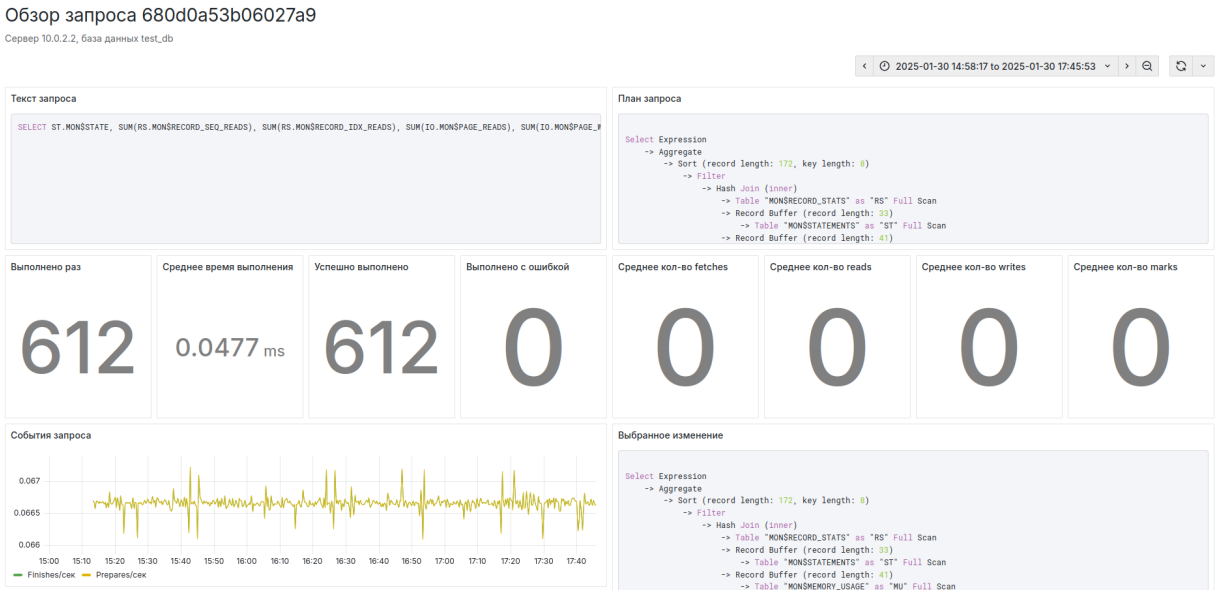


Рисунок 7.11 — Обзор запроса - Часть 1

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

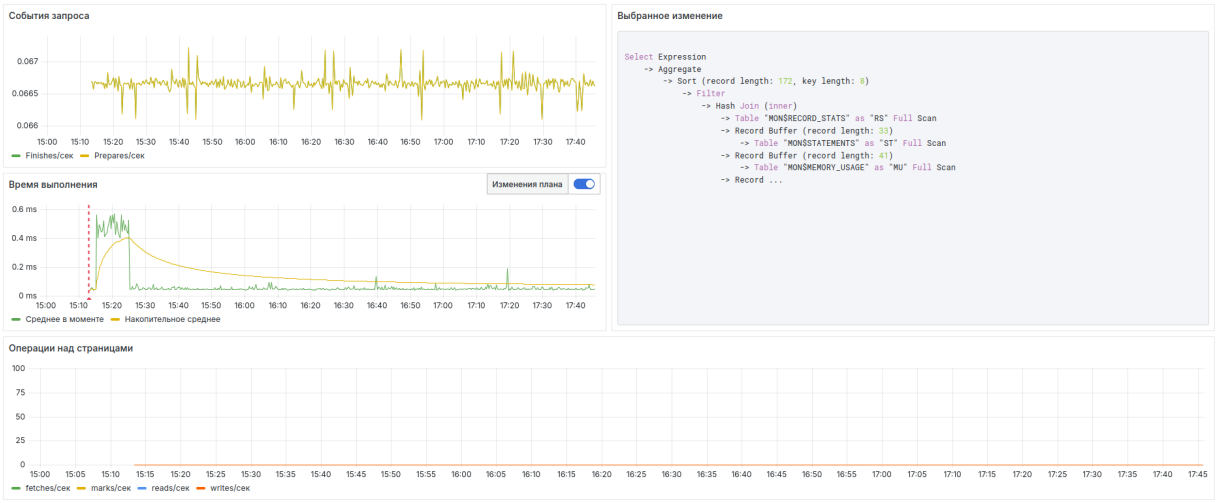


Рисунок 7.12 — Обзор запроса - Часть 2

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 результатам двух последних сборов 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`.

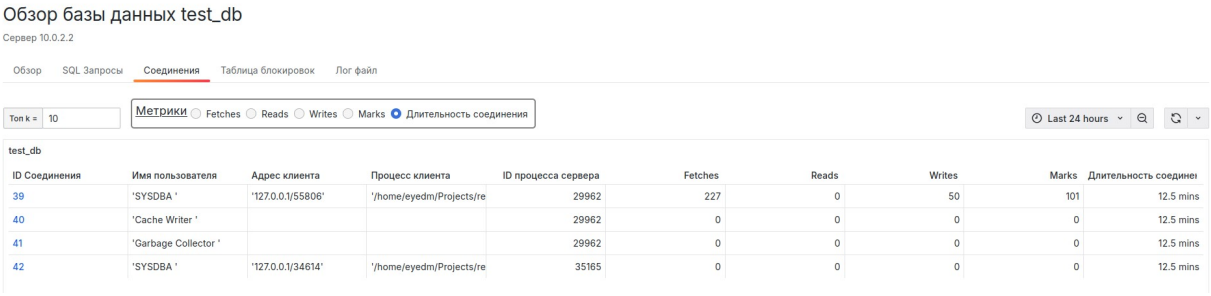


Рисунок 7.13 — Соединения с базой данных

Обзор соединения

На данной странице находится подробная информация о выбранном соединении.

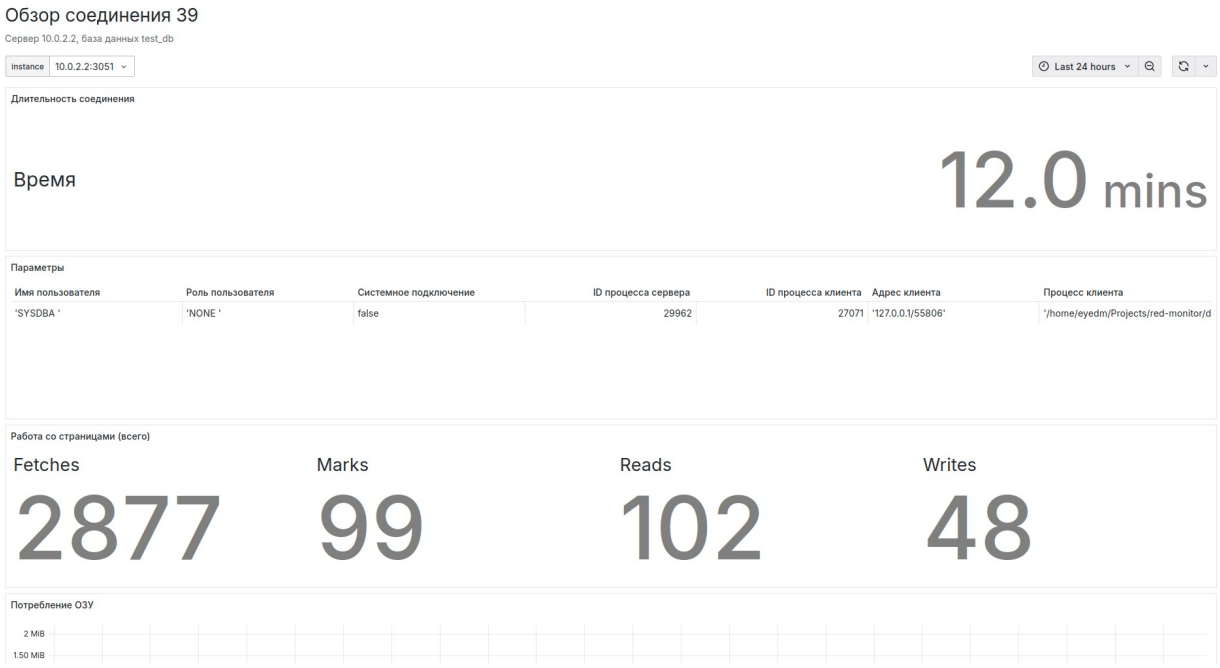


Рисунок 7.14 — Обзор соединения

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 7.15 — Использование памяти таблицей блокировок

Лог файл

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

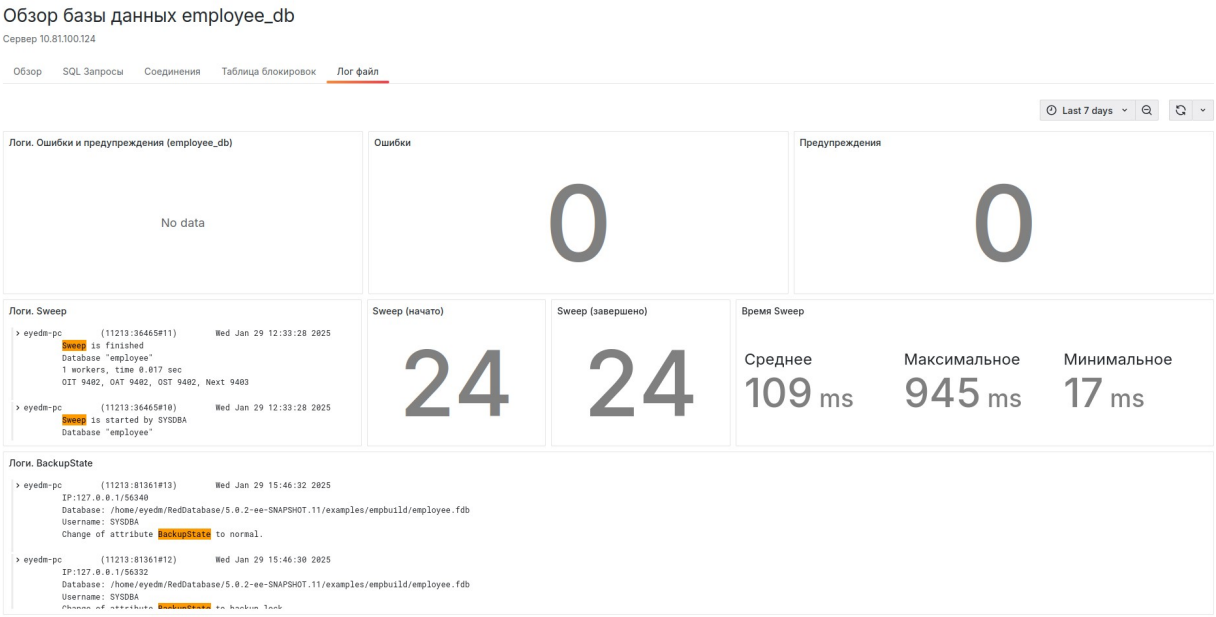


Рисунок 7.16 — Анализ лог файла

7.2.4 Процессы

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

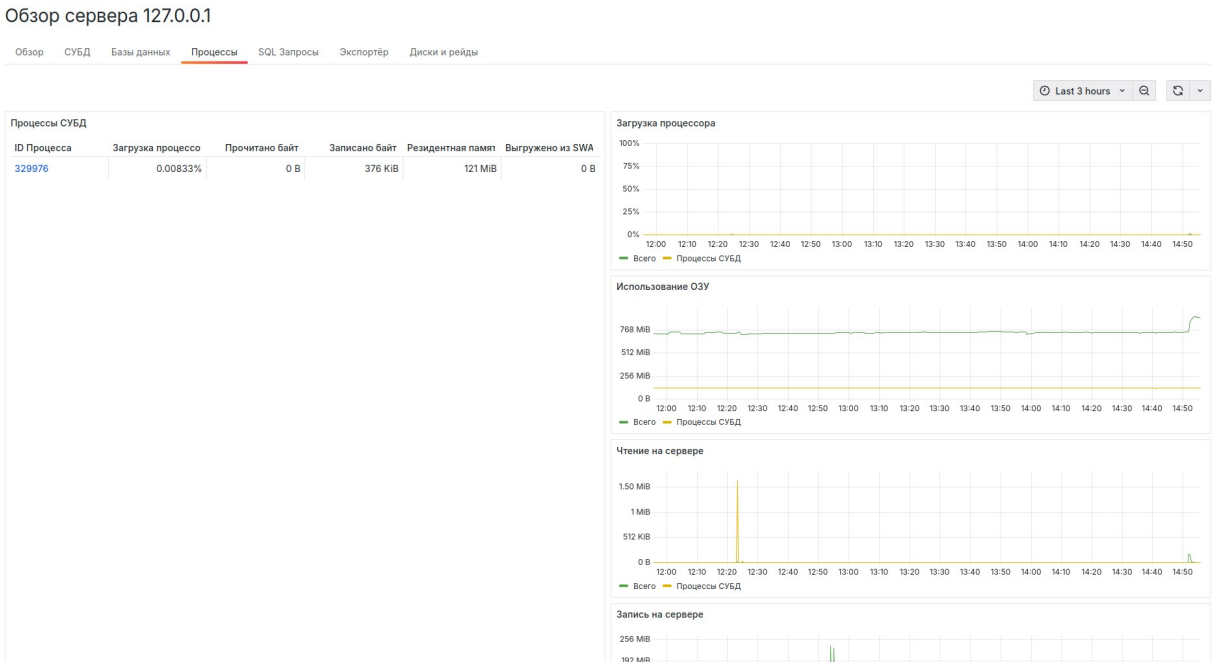


Рисунок 7.17 — Процессы СУБД

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

Обзор процесса

На данной странице находится подробная информация о выбранном процессе.



Рисунок 7.18 — 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.

7.2.5 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*.

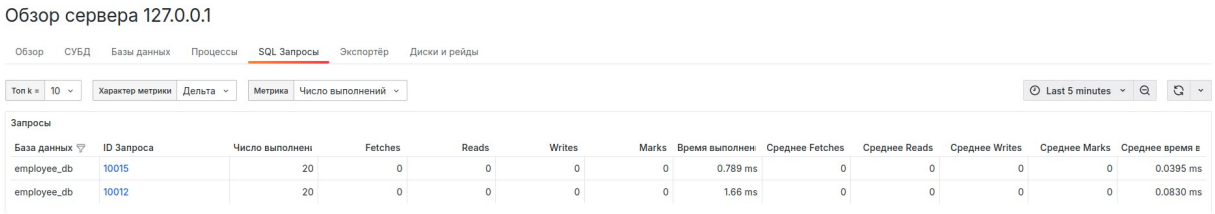


Рисунок 7.19 — Запросы

7.2.6 Агент мониторинга

На странице находится информация о сборе метрик агентом мониторинга.

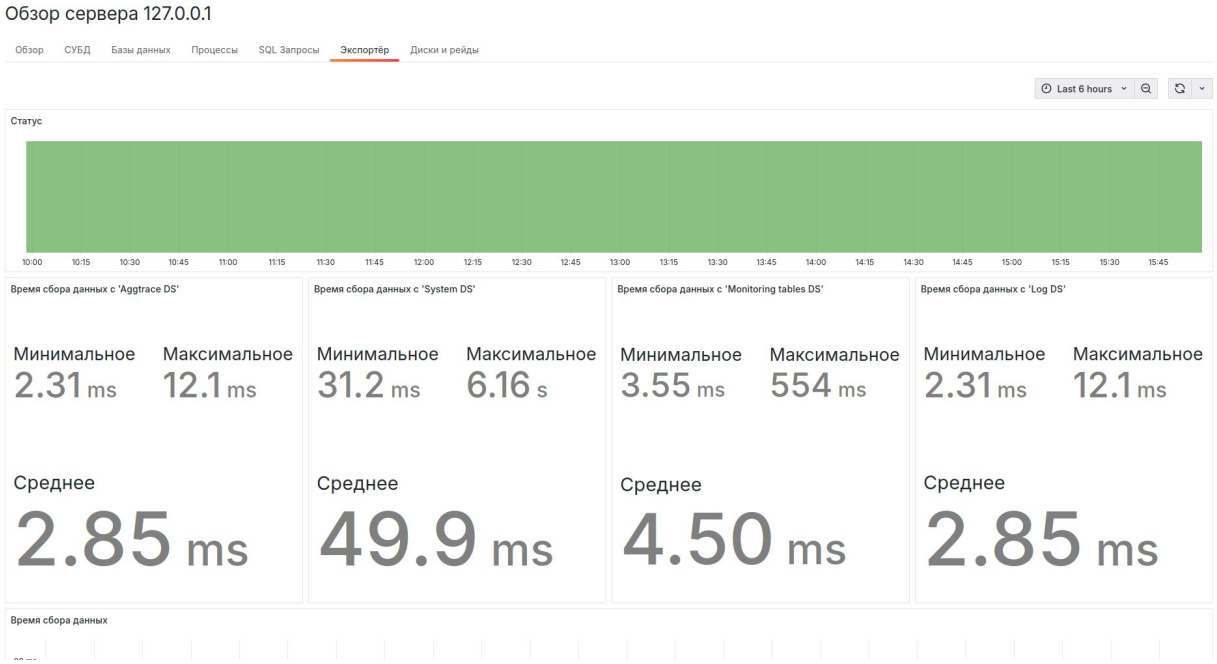


Рисунок 7.20 — Агент мониторинга

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

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

7.2.7 Диски и рейды

На странице отображается информация о выбранных дисках.

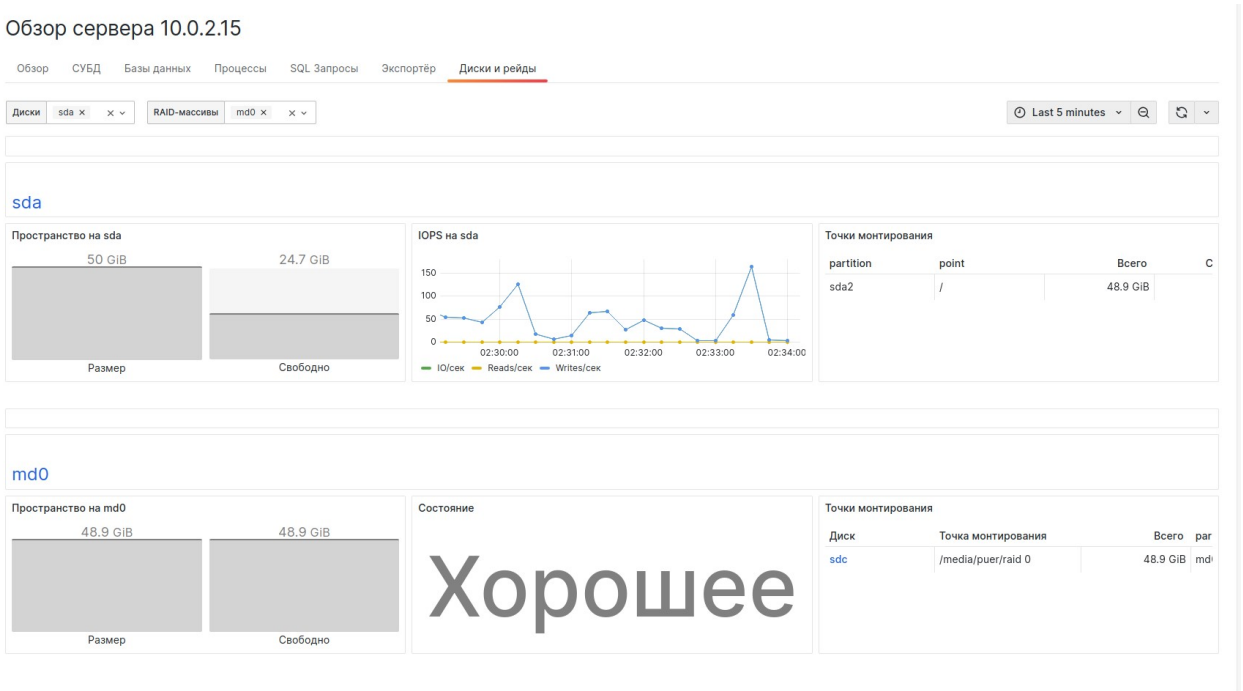


Рисунок 7.21 — Информация о диске

Панель `Пространство` показывает общий и свободный объём памяти на диске и RAID-массиве.

График `IOPS` отображает количество операций ввода/вывода в секунду.

График `Состояние` показывает состояние RAID-массива:

- Хорошее (`clean`)
- Повреждён (`degraded`)
- Синхронизация (`resyncing`)
- Восстановление (`recovering`)
- Неисправен (`faulty`)

Обзор RAID-массива

На странице отображается подробная информация о выбранном RAID-массиве.

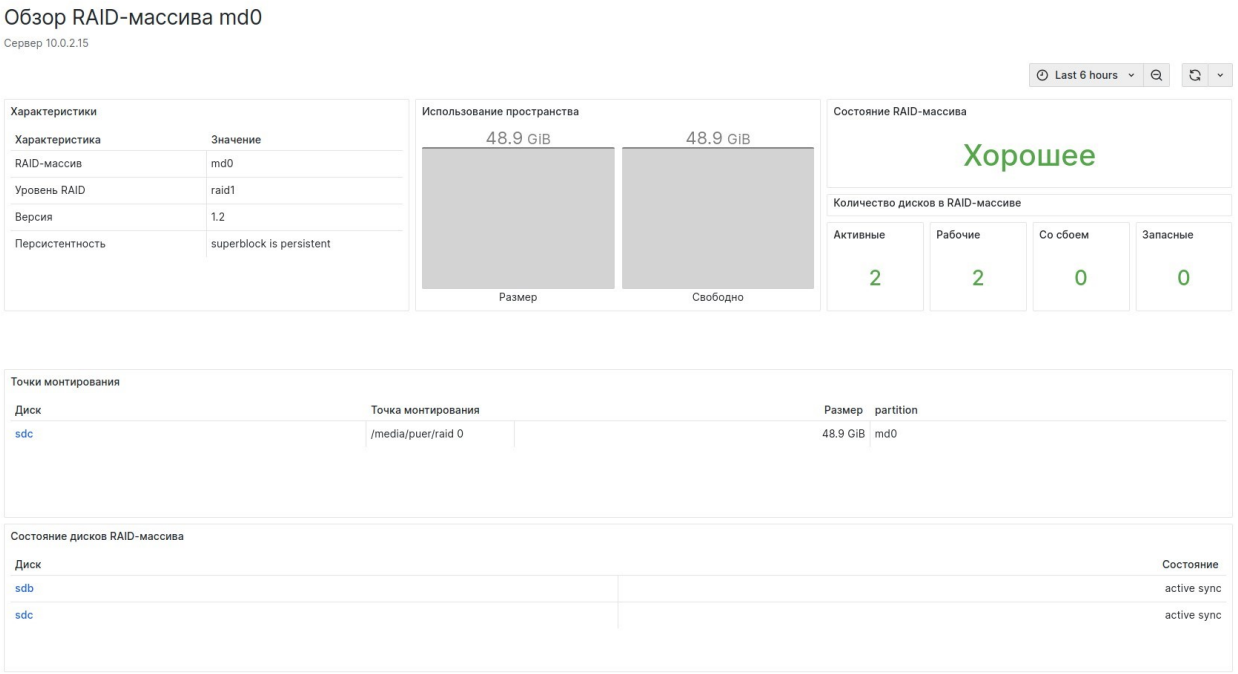


Рисунок 7.22 — Информация о RAID-массиве

Панель `Количество дисков в RAID-массиве` показывает число дисков, находящихся в конкретном состоянии.

Панель `Состояние дисков RAID-массива` показывает статус каждого диска:

- `unknown` (неизвестное состояние диска)
- `absent` (диск отсутствует)
- `active` (диск активен)
- `sync` (диск синхронизирован)
- `spare` (запасной диск)
- `write-mostly` (диск используется преимущественно для записи)
- `blocked` (диск заблокирован)
- `faulty` (диск неисправен)

Диск может находиться сразу в нескольких состояниях, например `"active sync"`.

7.3 Мониторинг кластеров

7.3.1 Необходимые настройки

Необходимо установить `Сервер СУБД` на каждый узел кластера, следуя инструкции в Руководстве администратора.

В конфигурационном файле `prometheus.yml` в параметре `targets` нужно указать сокет каждого узла. Подробнее о настройке `Prometheus` см. документацию на официальном сайте.

7.3.2 Обзор кластеров

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

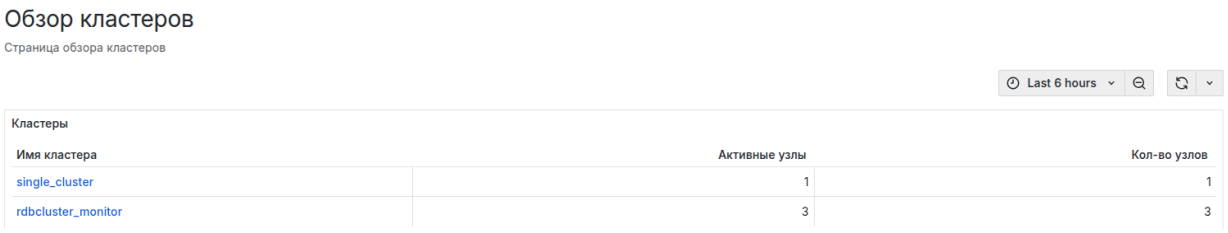


Рисунок 7.23 — Обзор кластеров

7.3.3 Информация о выбранном кластере

Обзор

На данной странице представлена краткая информация о выбранном кластере. Подробнее о режиме обслуживания и настройках отказоустойчивого кластера см. в Документации по настройке отказоустойчивого кластера.

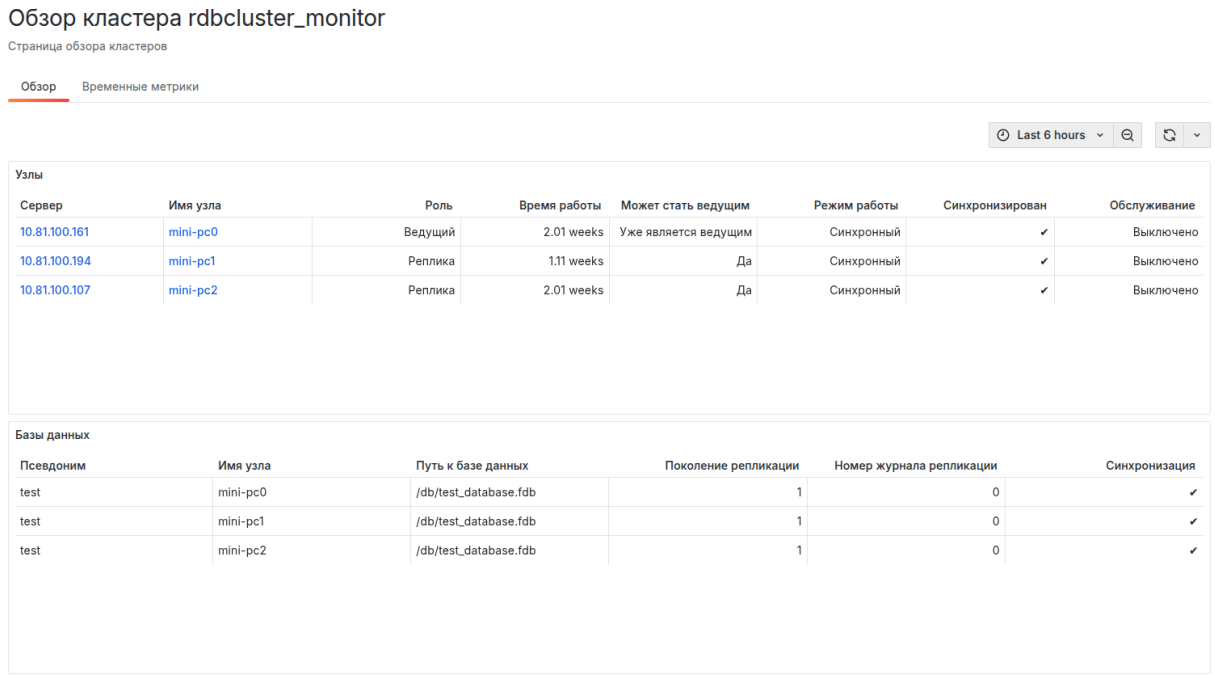


Рисунок 7.24 — Информация о выбранном кластере

Временные метки

Временные метки отображают время, затраченное на опрос состояния узла и кластера и публикацию собранной информации.

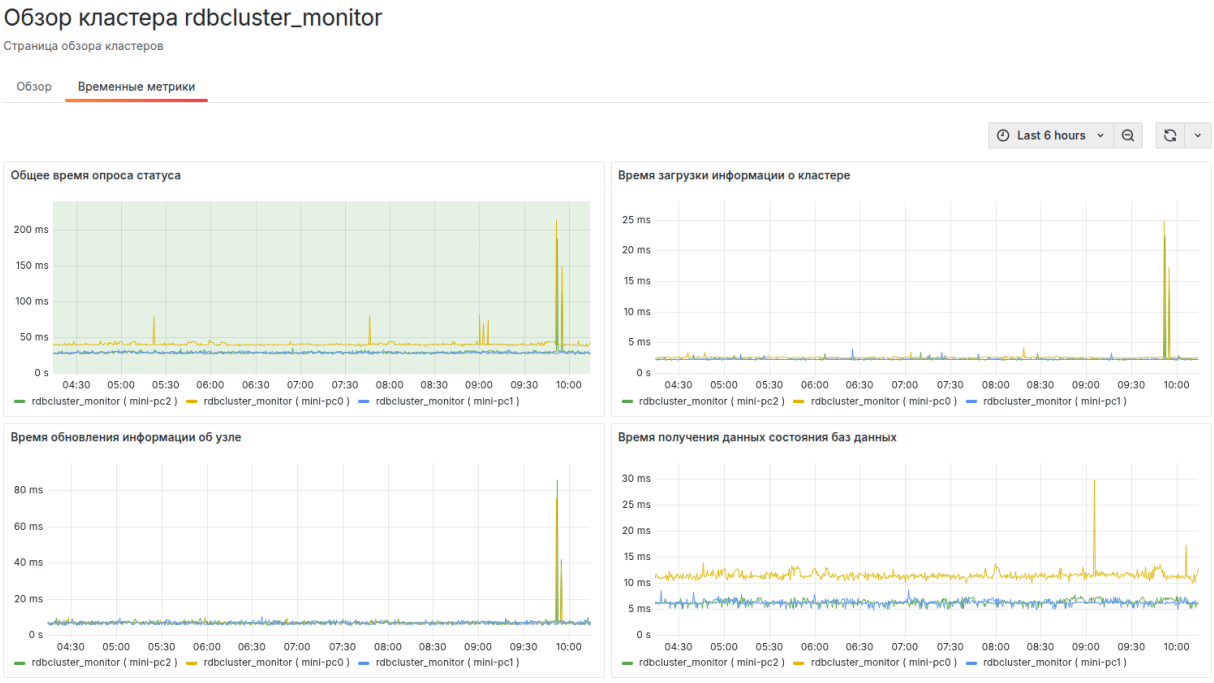


Рисунок 7.25 — Временные метки

7.3.4 Информация об узле кластера

На этой странице отображается информация о состоянии и временных метках конкретного узла кластера.

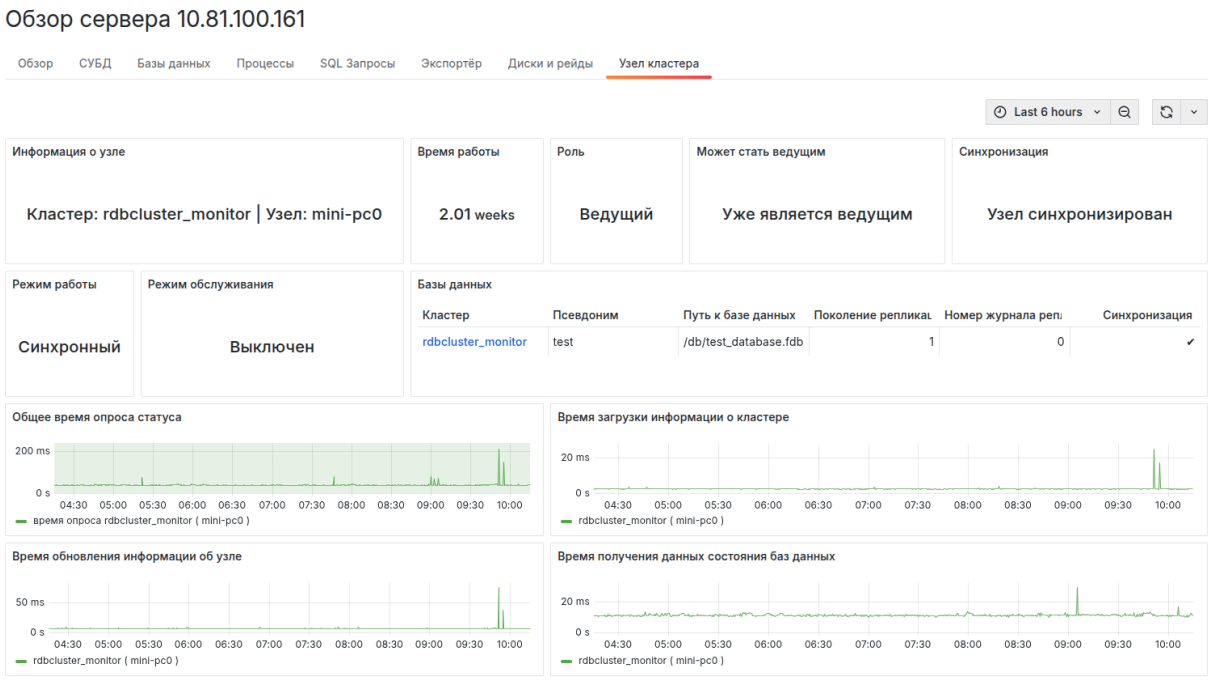


Рисунок 7.26 — Информация об узле кластера

Приложение А Собираемые метрики

А.1 Метрики таблиц мониторинга

Таблица А.1 — Метрики таблиц мониторинга

Название метрики	Метки	Описание метрики
rdb_call_stack_pages	database - база данных. object_type - тип объекта базы данных: • procedure; • trigger; • function. operation - тип операции: • reads; • fetches; • marks; • writes.	Количество операций указанного типа, выполненных вызовами со страницами базы данных в момент сбора информации.
rdb_call_stack_memory	database - база данных. object_type - тип объекта базы данных: • procedure; • trigger; • function; usage - использование памяти: • 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	database - база данных. object_type - тип объекта базы данных: • procedure; • trigger; • function;	Количество вызовов указанного типа, выполняемых в в момент сбора информации.
rdb_attachments_pages	database - база данных. att_id - идентификатор соединения. operation - тип операции: • 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: • 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: • 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: • 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: • 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: • 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: • 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	database - база данных. state - состояние запроса: • idle — бездействующий; • active — активный; • stalled — приостановленный, то есть запрос или курсор "живой" , но в данный момент не выполняется. operation - тип операции: • reads; • fetches; • marks; • writes.	Количесво запросов, которые выполняются над страницами базы данных в момент сбора информации.
rdb_statements_memory	database - база данных. state - состояние запроса: • idle — бездействующий; • active — активный; • stalled — приостановленный, то есть запрос или курсор "живой" , но в данный момент не выполняется. usage - использование памяти: • used - объём используемой памяти. • allocated - количество выделенной памяти.	Объём памяти (в байтах), используемой запросами, выполняющимися в момент сбора информации.
rdb_statements_count	database - база данных. state - состояние запроса: • 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	database - база данных. table - таблица MON$DATABASE.	Время выполнения запроса к таблице MON$DATABASE в секундах.
rdb_database_sql _dialect	database - база данных.	SQL диалект.
rdb_gstat_database _page_size	database - база данных.	Размер страницы файлов базы данных в байтах.
rdb_gstat_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: • 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: • 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 агент мониторинга с базой данных: • 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_db_stmt _prepare_count	database - база данных;	Количество подготовки запросов (событий PREPARE STATEMENT) в указанной базе данных.
rdb_db_stmt _prepare_count_failed	database - база данных;	Количество событий PREPARE STATEMENT, завершившихся ошибкой в указанной базе данных.
rdb_db_stmt_start _count	database - база данных;	Количество запусков запросов (событий START STATEMENT) в указанной базе данных.
rdb_db_stmt _start_count_failed	database - база данных;	Количество запусков запросов, завершившихся ошибкой в указанной базе данных.
rdb_db_stmt _finish_count	database - база данных;	Количество завершённых запросов (событий FINISH STATEMENT) в указанной базе данных.
rdb_db_stmt _finish_count_failed	database - база данных;	Количество неудачно завершённых запросов в указанной базе данных.
rdb_db_stmt_free _count	database - база данных;	Количество освобождений запросов (событий FREE STATEMENT) в указанной базе данных.
rdb_db_stmt _free_count_failed	database - база данных;	Количество освобождений запросов, завершившихся ошибкой в указанной базе данных.
rdb_db_stmt_times	database - база данных;	Время выполнения запроса в миллисекундах в указанной базе данных.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_db_stmt_pages _reads	database - база данных;	Количество страниц, считанных с диска в указанной базе данных.
rdb_db_stmt_pages _writes	database - база данных;	Количество страниц, записанных из кэша на диск в указанной базе данных.
rdb_db_stmt_pages _fetches	database - база данных;	Количество страниц, считанных из страничного кэша в указанной базе данных.
rdb_db_stmt_page _marks	database - база данных;	Количество страниц, изменённых в страничном кэше в указанной базе данных.
rdb_db_stmt _sort_cache	database - база данных;	Количество памяти, выделенной в кеше под сортировку запроса в указанной базе данных.
rdb_db_stmt _sort_disk	database - база данных;	Количество памяти, выделенной на диске под сортировку запроса в указанной базе данных.
rdb_db_meta _shmem_overflows	database - база данных;	Количество запросов, которые не отслеживаются мониторингом из-за переполнения разделяемой памяти.
rdb_db_meta _bloom_rejections	database - база данных;	Количество запросов, которые не отслеживаются мониторингом из-за предполагаемого дублирования.
rdb_aggtrace_stmt _finish_count	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество завершённых запросов (событий FINISH STATEMENT).
rdb_aggtrace_stmt _finish_count_failed	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество неудачно завершённых запросов.
rdb_aggtrace_stmt_start _count	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество запусков запросов (событий START STATEMENT).
rdb_aggtrace_stmt _start_count_failed	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество запусков запросов, завершившихся ошибкой.
rdb_aggtrace_stmt_free _count	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество освобождений запросов (событий FREE STATEMENT).
rdb_aggtrace_stmt _free_count_failed	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество освобождений запросов, завершившихся ошибкой.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_aggtrace_stmt _prepare_count	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество подготовки запросов (событий PREPARE STATEMENT).
rdb_aggtrace_stmt _prepare_count_failed	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество событий PREPARE STATEMENT, завершившихся ошибкой.
rdb_aggtrace_stmt_times	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Время выполнения запроса в миллисекундах.
rdb_aggtrace_stmt_pages _reads	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество страниц, считанных с диска.
rdb_aggtrace_stmt_pages _writes	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество страниц, записанных из кэша на диск.
rdb_aggtrace_stmt_pages _fetches	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество страниц, считанных из страничного кэша.
rdb_aggtrace_stmt_page _marks	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество страниц, изменённых в страничном кэше.
rdb_aggtrace_stmt _sort_cache	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество памяти, выделенной в кеше под сортировку запроса.
rdb_aggtrace_stmt _sort_disk	database - база данных; hash - хэш запроса; planHash - хэш плана запроса;	Количество памяти, выделенной на диске под сортировку запроса.
rdb_aggtrace_trns _start_count	database - база данных	Количество событий START TRANSACTION.
rdb_aggtrace_trns _start_count_failed	database - база данных	Количество событий START TRANSACTION, завершившихся ошибкой.
rdb_aggtrace_trns _rollback_count	database - база данных	Количество событий ROLLBACK TRANSACTION.
rdb_aggtrace_trns _rollback_failed_count	database - база данных	Количество откатов транзакций, завершившихся ошибкой.
rdb_aggtrace_trns _rollback_retaining_count	database - база данных	Количество событий ROLLBACK RETAINING.
rdb_aggtrace_trns _rollback_retaining _failed_count	database - база данных	Количество событий ROLLBACK RETAINING, завершившихся ошибкой.
rdb_aggtrace_trns _commit_count	database - база данных	Количество событий COMMIT.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_aggtrace_trns _commit_failed_count	database - база данных	Количество событий COMMIT, завершившихся ошибкой.
rdb_aggtrace_trns _commit_retaining_count	database - база данных	Количество событий COMMIT RETAINING.
rdb_aggtrace_trns _commit_retaining _failed_count	database - база данных	Количество событий COMMIT RETAINING, завершившихся ошибкой.
rdb_aggtrace_trns _commit_writes	database - база данных	Количество чтений в транзакции commit.
rdb_aggtrace_trns _commit_time	database - база данных	Время выполнения транзакции commit.
rdb_aggtrace_trns _commit_retaining_writes	database - база данных	Количество чтений в транзакции commit retaining.
rdb_aggtrace_trns _commit_retaining_time	database - база данных	Время выполнения транзакции commit retaining.
rdb_aggtrace _proc_finish_count	database - база данных	Количество событий FINISH PROCEDURE.
rdb_aggtrace_proc_finish_count_failed	database - база данных	Количество событий FINISH PROCEDURE, завершившихся ошибкой.
rdb_aggtrace_func_finish_count	database - база данных	Количество событий FINISH FUNCTION.
rdb_aggtrace_func_finish_count_failed	database - база данных	Количество событий FINISH FUNCTION, завершившихся ошибкой.
rdb_aggtrace_trigger_finish_count	database - база данных	Количество событий FINISH TRIGGER.
rdb_aggtrace_trigger_finish_count_failed	database - база данных	Количество событий FINISH TRIGGER, завершившихся ошибкой.

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

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_lck_table_hash _slots	database - база данных	Число слотов кэширования блокировок
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	

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_lck_table_scans	database - база данных	Показывает число просмотров менеджером блокировок цепочки блокировок и владельцев для поиска взаимных блокировок
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: • reads • writes	Количество операций чтения и записи на диск.
rdb_disks_io_bytes	disk - диск. operation - тип операции: • 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: • normal - время, затраченное на фактический ввод/выводж; • 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: • used • free • total	Использование точки монтирования.
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: • 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: • 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: • 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: • 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: • reads • writes	Количество операций ввода/вывода, выполняемых процессом СУБД.
rdb_procs_io_bytes	pid - идентификатор процесса СУБД operation - тип операции: • 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: • free • used • total	Использование оперативной памяти.
rdb_swap_usage	usage: • used • free • total	Использование памяти, выделенной для swap.
rdb_swapped	direction: • in • out	Объём данных (в байтах), затронутый swap.
rdb_cpu_freq	marker: • current	Частота работы процессора.
rdb_cpu_times	kind: • system • user • iowait • irq • soft_irq	Время, потраченное процессором.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_cpu_ctx		Число изменений контекста процессора.
rdb_cpu_cores	kind: • logical • physical	Количество ядер процессора.
rdb_cpu_interrupts	marker: • soft • casual	Число прерываний процессора.
rdb_tmp_mount_point_usage	point usage: • 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.

A.5 Метрики RAID-массивов

Таблица А.5 — Метрики RAID-массивов

Название метрики	**Метки**	**Описание метрики**
`rdb_raid_disks_count`	`raid` - имя массива; `state` - состояние диска: • `active` - количество активных дисков • `working` - количество рабочих дисков • `failed` - количество неисправных дисков • `spare` - количество запасных дисков	Количество дисков, находящихся в конкретном состоянии.
`rdb_raid_state`	`raid` - имя массива.	Текущее состояние RAID-массива: • 1 - `clean` • 2 - `degraded` • 3 - `resyncing` • 4 - `recovering` • 5 - `faulty`
`rdb_raid_info`	raid - имя RAID-массива `raid_level` - уровень `RAID`; `version` - версия `RAID`; `consistency_policy` - политика согласованности; `persistence` - информация о персистентности.	Общая информация о RAID-массиве.
`rdb_link_raid_disk`	`raid` - имя массива; `disk` - диск	Сопоставление RAID-массива и диска.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_raid_device_state	raid - имя массива; disk - диск	Состояние каждого диска в RAID-массиве: • 0 - unknown (неизвестное состояние диска) • 1 - absent (диск отсутствует) • 2 - active (диск активен) • 3 - sync (диск синхронизирован) • 4 - spare (запасной диск) • 5 - write-mostly (диск используется преимущественно для записи) • 6 - blocked (диск заблокирован) • 7 - faulty (диск неисправен) Если диск находится в нескольких состояниях, например "active sync", код формируется как комбинация чисел ("23" - 2 для active, 3 для sync).
rdb_raid_size	raid - имя массива.	Размер RAID-массива в байтах.

А.6 Метрики агента мониторинга

Таблица А.6 — Метрики агента мониторинга

Название метрики	Метки	Описание метрики
rdb_registry_blocks_count	ds - источник данных.	Количество блоков в регистре.
rdb_registry_free_blocks_count	ds - источник данных.	Количество пустых блоков в регистре.
rdb_registry_state	ds - источник данных.	Состояние регистра: 0 - готов к отправке; 1 - в процессе формирования метрик/не готов к отправке.
rdb_exporter_mem_rss		Количество резидентной памяти в байтах.
rdb_exporter_mem_vms		Количество виртуальной памяти в байтах.
rdb_exporter_mem_uss		Количество USS памяти в байтах.
rdb_exporter_mem_pss		Количество PSS памяти в байтах.
rdb_exporter_mem_data		Количество памяти в байтах, выделенной под секцию data.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_exporter_mem_lib		Количество памяти в байтах, занимаемое библиотеками (*.so).
rdb_exporter_mem_shared		Количество памяти в байтах, разделяемой между процессами.
rdb_exporter_mem_swap		Количество памяти в байтах, записанной в swap.
rdb_exporter_mem_text		Количество памяти в байтах, выделенной под исполняемый код.
rdb_exporter_cpu_user		Время работы процессора (в секундах) в режиме пользователя.
rdb_exporter_cpu_sys		Время работы процессора (в секундах) в режиме ядра.
rdb_exporter_threads_count		Количество потоков процесса агента мониторинга.

А.7 Другие метрики

Таблица А.7 — Другие метрики

Название метрики	Метки	Описание метрики
rdb_gstat_database _oit	database - база данных;	Значение идентификатора старейшей заинтересованной транзакции в базе данных (Oldest Interesting Transaction или OIT).
rdb_gstat_database _oat	database - база данных;	Значение идентификатора старейшей транзакции, которую пользователь стартовал, и до сих пор не завершил по commit или rollback (Oldest Active Transaction или OAT).
rdb_gstat_database _ost	database - база данных;	Значение идентификатора последней транзакции , которая влияет на процесс сборки мусора (Oldest Snapshot Transaction или OST).
rdb_gstat_database _nt	database - база данных;	Значение идентификатора, который будет присвоен следующей транзакции при ее старте.
rdb_gstat_database _next_att_id	database - база данных;	Значение идентификатора, который будет присвоен следующему подключению.
rdb_gstat_database_attributes	database - база данных;	Атрибуты и флаги файла базы данных.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
`rdb_log_errors`	`kind` - тип ошибки: • `fatal` • `critical` • `normal` • `event`	Количество ошибок указанного типа.
`rdb_server_version`	`kind`: `major`, `minor`, `fix`	Версия СУБД.