

---

РБДМонитор  
Версия 1.1.0  
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 РБДМонитор	5
2.1	Установка на Linux	5
2.1.1	RPM-пакет	5
2.1.2	Установка из архива	5
2.2	Установка и настройка на Windows	6
3	Обновление и удаление РБДМонитор	11
3.1	Обновление на Linux	11
3.1.1	Обновление через RPM-пакет	11
3.1.2	Обновление через архив	12
3.2	Удаление РБДМонитор	12
3.2.1	Удаление через RPM-пакет	12
3.2.2	Удаление через архив	13
4	Настройка РБДМонитор	14
4.1	Настройка серверной части	14
4.2	Настройка клиентской части	17
5	Мониторинг при миграции РЕД Базы Данных 3 на версию 5	23
6	Просмотр метрик	24
6.1	Пользовательские метрики	24
6.2	Просмотр метрик с помощью Prometheus	25
6.3	Просмотр метрик с помощью Grafana	26
6.4	Анализ метрик	27
6.5	Обзорная страница	28
6.6	Информация о выбранном сервере	28
6.6.1	СУБД	29
6.6.2	Базы данных	30
6.6.3	Информация о выбранной базе данных	31
	SQL запросы	32
	Соединения	34
	Таблица блокировок	35
	Лог файл	36
6.6.4	Процессы	36
	Обзор процесса	37
6.6.5	SQL-запросы	38
6.6.6	Агент мониторинга	39
6.6.7	Диски и рейды	39
	Обзор RAID-массива	40
6.7	Мониторинг кластеров	41
6.7.1	Необходимые настройки	41
6.7.2	Обзор кластеров	42
6.7.3	Информация о выбранном кластере	42
	Обзор	42
	Временные метки	42

6.7.4	Информация об узле кластера	43
Приложение А	Собираемые метрики	45
А.1	Метрики таблиц мониторинга	45
А.2	Метрики агрегатного аудита	50
А.3	Метрики утилиты <code>rdb_lock_print</code>	53
А.4	Метрики операционной системы	55
А.5	Метрики RAID-массивов	58
А.6	Метрики агента мониторинга	60
А.7	Другие метрики	61

## 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.

РБДМонитор состоит из Агента мониторинга (клиентской части) и Сервера мониторинга (серверной части). Агент мониторинга собирает информацию о состоянии СУБД и передаёт её серверной части. Сервер мониторинга хранит и визуализирует собранные метрики СУБД. Подробное описание собираемых метрик см. в приложении [Собираемые метрики](#).

## 1.2 Минимальный состав аппаратных средств

**Системные требования для клиентской части:**

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

**Системные требования для серверной части:**

- 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.

## Глава 2

# Установка РБДМонитор

## 2.1 Установка на Linux

Доступно несколько способов установки:

- [Установка через RPM-пакет](#);
- [Установка из архива](#).

### 2.1.1 RPM-пакет

1. Скачайте с [официального сайта](#) rpm-пакеты серверной и клиентской части РБДМонитор.
2. Установите серверную часть РБДМонитор:

```
dnf install <путь_до_rpm_пакета_rdbmonitor-server>
```

3. Установите СУБД для серверной части: РЕД Базу Данных, следуя инструкции в [Руководстве администратора](#) или Firebird.
4. Установите клиентскую часть РБДМонитор:

```
dnf install <путь_до_rpm_пакета_rdbmonitor-client>
```

Для мониторинга отказоустойчивого кластера клиентскую часть необходимо установить на каждый узел.

5. Настройте РБДМонитор по инструкции [Настройка РБДМонитор](#).

### 2.1.2 Установка из архива

1. Установите СУБД для серверной части: РЕД Базу Данных, следуя инструкции в [Руководстве администратора](#) или Firebird.
2. Скачайте установочный архив для серверной части с [официального сайта](#). После скачивания распакуйте архив, выполнив команду:

```
tar -xzf <архив>
```

3. Установите плагины `redsoft-rdbmonitor-app` и `redsoft-reddatabase-datasource` командой:

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

При установке плагинов нужно внести `redsoft-rdbmonitor-app` и `redsoft-reddatabase-datasource` в список доверенных, разрешив редактировать файл `/etc/grafana/grafana.ini`.

4. Установите Loki:

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

5. Скачайте установочный архив для клиентской части с [официального сайта](#). После скачивания распакуйте архив, выполнив команду:

```
tar -xzf <архив>
```

- Установите клиентскую часть, выполнив `install.sh` с правами администратора:

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

- Скачайте установочный архив для клиентской части с [официального сайта](#). После скачивания распакуйте архив, выполнив команду:

```
tar -xzf <архив>
```

- Установите клиентскую часть, выполнив `install.sh` с правами администратора:

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

- Настройте РБДМонитор по инструкции [Настройка РБДМонитор](#).

## 2.2 Установка и настройка на Windows

- Скачайте с [официального сайта](#) zip-архив РБДМонитор.
- Распакуйте скачанный архив в любой каталог.
- Установите службу агента мониторинга:

```
<путь до каталога установки>\RdbMonitor\install_service.bat
```

- Остановите наблюдаемую СУБД.
- Для сервера, который необходимо мониторить, в `firebird.conf` укажите следующие настройки:

```
TracePlugin = messagetrace
```

- Для сервера, который необходимо мониторить, в `plugins.conf` укажите следующие настройки:

```
Plugin = messagetrace {  
    Module = <путь к корневому каталогу РБДМонитор>\RdbMonitor\plugins\  
    messagetrace\messagetrace  
    Required = true  
    Config = monitor_config  
}  
  
Config = monitor_config {  
    ConfigFile = <путь к корневому каталогу РБДМонитор>\RdbMonitor\exporter\  
    exporter_conf.json  
}
```

- Запустите наблюдаемую СУБД.
- Установите СУБД для серверной части: РЕД Базу Данных, следуя инструкции в [Руководстве администратора](#) или Firebird.
- Откройте `\RdbMonitor\external\promtail\promtail.yml` и укажите там следующее:

```
server:
http_listen_port: 9080
grpc_listen_port: 0

positions:
filename: "<путь до каталога установки РБДМонитор>\\RdbMonitor\\external\\
promtail\\positions.yaml"

clients:
- url: http://localhost:3100/loki/api/v1/push # IP-адрес Loki

scrape_configs:
- job_name: RedDatabase
  pipeline_stages:
    - multiline:
        firstline: '^[A-Za-z0-9\._%+~]+\.s+\([0-9]+\:[0-9]+\#[0-9]+\)\.s+[A-Za-
z]+\s+[A-Za-z]+\s+[0-9]+\s+[0-9]+\:[0-9]+\:[0-9]+\s+[0-9]+'
        max_wait_time: 3s
    static_configs:
    - targets:
        - localhost
      labels:
        job: RedDatabase
        host: 127.0.0.1 # IP-адрес машины в сети
        __path__: "<корневой каталог СУБД>\\firebird.log" # Путь до firebird.log
```

10. Распакуйте архив promtail-windows-amd64.exe.zip в ту же папку.
11. Запустите Promtail:

```
Start-Process -FilePath ".\promtail-windows-amd64.exe\promtail-windows-amd64.
exe" -ArgumentList '--config.file="\RdbMonitor\external\promtail\promtail.yaml
"' -WindowStyle Hidden
```

12. Откройте файл \RdbMonitor\external\loki\loki.yaml и укажите там следующее:

```
auth_enabled: false

server:
http_listen_port: 3100

common:
ring:
  kvstore:
    store: inmemory
replication_factor: 1
path_prefix: "<путь до каталога установки РБДМонитор>\\RdbMonitor\\external\\
loki"

schema_config:
configs:
- from: 2020-05-15
  store: tsdb
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
object_store: filesystem
schema: v13
index:
  prefix: index_
  period: 24h

storage_config:
filesystem:
  directory: "<путь до каталога установки РБДМонитор>\RdbMonitor\external\loki"
```

13. Запустите Loki:

```
Start-Process -FilePath ".\loki-windows-amd64.exe\loki-windows-amd64.exe" -
ArgumentList '--config.file="\RdbMonitor\external\loki\loki.yaml"' -
WindowStyle Hidden
```

14. Установите Prometheus.

15. Установите Grafana.

16. Переместите из \RdbMonitor\plugins\grafana\ папки redsoft-rdbconnector-datasource и redsoft-rdbmonitor-app в \grafana-12.4.2\data\plugins\. Папку data необходимо предварительно создать.

17. Откройте файл \grafana-12.4.2\conf\defaults.ini и укажите там следующее:

```
allow_loading_unsigned_plugins = redsoft-rdbconnector-datasource,redsoft-
rdbmonitor-app
```

18. Запустите СУБД в серверной части.

19. Скорректируйте скрипт create.sql, расположенный по пути \RdbMonitor\scripts\create.sql.

Выполните скрипт:

```
isql -user <имя пользователя> -password <пароль> -i \RdbMonitor\scripts\
create.sql
```

20. Установите асинхронный режим записи на диск:

```
gfix -write async <база данных statements.fdb>
```

21. Укажите настройки Prometheus в файле \prometheus\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>:<порт>", ..., "<хост>:<порт>"]

rule_files:
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
- rules.yml

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

22. Запустите Prometheus.
23. Запустите Grafana.
24. Укажите настройки агента мониторинга в файле `\RDBMonitor\exporter\exporter_conf.json`.

```
{
  "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>
    }
  },
  "promtail": {
    "host": "<хост наблюдаемого сервера>"
  },
  "dictserver": {
    "host": "<хост базы данных>",
    "port": <порт базы данных>,
    "database": "/opt/rdbmonitor/server/rdb/statements.fdb",
    "user": "<пользователь>",
    "password": "<пароль>",
    "Auth_plugins": ["<плагин>", ..., "<плагин>"],
    "loki": {
      "port": <порт loki>
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
    }  
  },  
  "RedDatabase": {  
    "host": "<хост СУБД>",  
    "port": <порт СУБД>,  
    "user": "<логин>",  
    "password": "<пароль>",  
    "folder_path": "<путь к папке установки RedDatabase>"  
  },  
  "databases": {  
    "<псевдоним>": {  
      "alias": "<алиас>",  
      "scrape": {  
        "enabled": true,  
        "filters": "<путь к файлу с настройками фильтрации>",  
        "endpoint": "<сокет>"  
      }  
    }  
  }  
}
```

25. Запустите агент мониторинга от имени администратора:

```
Start-Process -FilePath ".\exporter.exe" -ArgumentList ".\exporter_conf.json"  
-WindowStyle Hidden
```

Или командой:

```
"<путь до каталога установки>\RDBMonitor\start.bat"
```

26. После настройки необходим перезапуск сервиса агента мониторинга:

```
Restart-Service -Name ExporterService
```

27. Откройте в браузере страницу запущенной Grafana (по умолчанию <http://localhost:3000/>). По умолчанию для входа используется логин `admin` и пароль `admin`.

Укажите источники данных:

- 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:9090/>). 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**.
- Укажите источником 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 **РЭД База Данных**. Заполните поля для соединения с базой данных и нажмите на кнопку **Save & test**.
- Во вкладке **Data sources** нажмите на кнопку **Add data source**. В открывшемся окне из списка источников выберите **Loki**, для сбора логов операционной системы и СУБД. Укажите URL-адрес для доступа к **Loki** (по умолчанию <http://127.0.0.1:3100>) и нажмите на кнопку **Save & test**.

## Глава 3

# Обновление и удаление РБДМонитор

## 3.1 Обновление на Linux

### 3.1.1 Обновление через RPM-пакет

1. Скачайте с [официального сайта](#) rpm-пакеты серверной и клиентской части РБДМонитор.
2. Сохраните конфигурационный файл `exporter_conf.json`.
3. Обновите серверную часть РБДМонитор:

```
dnf update <путь_до_rpm_пакета_rdbmonitor-server>
```

4. Обновите клиентскую часть РБДМонитор:

```
dnf update <путь_до_rpm_пакета_rdbmonitor-client>
```

5. Перезапустите наблюдаемую СУБД:

```
systemctl restart firebird
```

6. В серверной части убедитесь, что к `statements.fdb` нет подключений:

```
ps-ef|grep rdbserver  
sudo lsof /path/to/db.fdb
```

Если есть подключения, то необходимо дождаться их завершения или принудительно отключить их.

7. В серверной части обновите справочник запросов. Выполните скрипт, расположенный по пути `\RdbMonitor\scripts\update.sql`:

```
isql <путь к statements.fdb> -user <имя пользователя> -password <пароль> -i \  
RdbMonitor\scripts\update.sql
```

8. Перезапустите Loki:

```
sudo systemctl restart loki.service
```

9. Перезапустите Grafana:

```
sudo systemctl restart grafana-server
```

10. Настройте `exporter_conf.json` или используйте сохранённый файл.

11. Перезапустите клиентскую часть РБДМонитор:

```
systemctl restart rdbmonitor-client.target
```

## 3.1.2 Обновление через архив

1. Сохраните конфигурационный файл `exporter_conf.json`.
2. Скачайте установочный архив для клиентской части с [официального сайта](#). После скачивания распакуйте архив, выполнив команду:

```
tar -xzf <архив>
```

3. Обновите клиентскую часть, выполнив `install.sh` с правами администратора:

```
sudo ./install.sh -o update -p full
sudo ./install.sh -o update -p grafana_plugins
sudo ./install.sh -o update -p loki
```

4. Перезапустите наблюдаемую СУБД:

```
systemctl restart firebird
```

5. В серверной части убедитесь, что к `statements.fdb` нет подключений:

```
ps -ef | grep rdbserver
sudo lsof /path/to/db.fdb
```

Если есть подключения, то необходимо дождаться их завершения или принудительно отключить их.

6. В серверной части обновите справочник запросов. Выполните скрипт, расположенный по пути `\RdbMonitor\scripts\update.sql`:

```
isql <путь к statements.fdb> -user <имя пользователя> -password <пароль> -i \
RdbMonitor\scripts\update.sql
```

7. Перезапустите Loki:

```
sudo systemctl restart loki.service
```

8. Перезапустите Grafana:

```
sudo systemctl restart grafana-server
```

9. Настройте `exporter_conf.json` или используйте сохранённый файл.
10. Перезапустите клиентскую часть РБДМонитор:

```
systemctl restart rdbmonitor-client.target
```

## 3.2 Удаление РБДМонитор

### 3.2.1 Удаление через RPM-пакет

1. Удалите серверную часть РБДМонитор:

```
dnf remove <путь_до_rpm_пакета_rdbmonitor-server> --setopt=clean_requirements_
on_remove=1
```

2. Удалите клиентскую часть РБДМонитор:

```
dnf remove <путь_до_rpm_пакета_rdbmonitor-client>
```

### 3.2.2 Удаление через архив

1. Удалите клиентскую часть, выполнив `uninstall.sh` с правами администратора:

```
sudo ./uninstall.sh client
```

2. Удалите серверную часть, выполнив `uninstall.sh` с правами администратора:

```
sudo ./uninstall.sh grafana_plugins  
sudo ./uninstall.sh loki
```

## Глава 4

# Настройка РБДМонитор

## 4.1 Настройка серверной части

1. Запустите СУБД (РЕД База Данных или Firebird):

```
systemctl start firebird
```

2. Скорректируйте скрипт `create_tables.sql`, расположенный по пути `/opt/rdbmonitor/server/rdb/create_tables.sql`. По умолчанию база данных пользовательских запросов (`statements.fdb`) будет создана по пути `/db/statemens.fdb`, каталог `db` при этом необходимо предварительно создать самостоятельно. По умолчанию доступ к СУБД с базой данных пользовательских запросов осуществляется по порту 3050.

Выполните скрипт:

```
/opt/RedDatabase/bin/isql -user <имя пользователя> -password <пароль> -i /opt/rdbmonitor/server/rdb/create_tables.sql
```

3. Установите асинхронный режим записи на диск:

```
gfix -write async <база данных statements.fdb>
```

4. Запустите Loki:

```
sudo systemctl start loki.service
```

5. Укажите настройки Prometheus в файле `/etc/prometheus/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>:<порт>", ..., "<хост>:<порт>"]
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
  - 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:
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

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

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 3051. 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 поставляется с РБДМонитор, для его использования файл `rules.yml` необходимо скопировать из `/opt/RedMonitor/server/prometheus/` в корневой каталог `Prometheus` (`/etc/prometheus/`).

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 отвечает (Агент мониторинга).

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](#).

#### 6. Запустите `Prometheus`:

```
sudo systemctl start prometheus
```

#### 7. Настройте уведомления:

Если уведомления не нужны, то этот пункт можно пропустить.

Для получения уведомлений укажите настройки Alertmanager в файле `/etc/alertmanager/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'
```

После настройки запустите Alertmanager:

```
sudo systemctl start alertmanager
```

Более подробно о настройке Alertmanager можно узнать в [официальной документации](#).

#### 8. Запустите Grafana:

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

#### 9. Откройте в браузере страницу запущенной Grafana (по умолчанию <http://localhost:3000/>). По умолчанию для входа используется логин `admin` и пароль `admin`.

Укажите источники данных:

- 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:9090/`). 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**.
- Укажите источником 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**.
- Во вкладке **Data sources** нажмите на кнопку **Add data source**. В открывшемся окне из списка источников выберите **Loki**, для сбора логов операционной системы и СУБД. Укажите URL-адрес для доступа к Loki (по умолчанию `http://127.0.0.1:3100`) и нажмите на кнопку **Save & test**.

## 4.2 Настройка клиентской части

1. Остановите наблюдаемую СУБД (РЕД Базу Данных или Firebird):

```
systemctl stop firebird
```

2. Для сервера, который необходимо мониторить, в `firebird.conf` укажите следующие настройки:

```
TracePlugin = messagetrace
```

3. У наблюдаемой СУБД включите плагин агрегатного аудита в `plugins.conf`:

```
Plugin = messagetrace
{
    Module = /opt/rdbmonitor/client/messagetrace
    Required = true
    Config = monitor_config
}

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

4. Запустите наблюдаемую СУБД (РЕД Базу Данных или Firebird):

```
systemctl start firebird
```

5. Укажите настройки агента мониторинга в файле `/opt/rdbmonitor/client/exporter/exporter_conf.json`.

```
{
  "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,

```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
"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>
}
},
"promtail": {
  "host": "<хост наблюдаемого сервера>"
},
"dictserver": {
  "host": "<хост базы данных>",
  "port": <порт базы данных>,
  "database": "/opt/rdbmonitor/server/rdb/statements.fdb",
  "user": "<пользователь>",
  "password": "<пароль>",
  "Auth_plugins": ["<плагин>", ..., "<плагин>"],
  "loki": {
    "port": <порт loki>
  }
},
"RedDatabase": {
  "host": "<хост СУБД>",
  "port": <порт СУБД>,
  "user": "<логин>",
  "password": "<пароль>",
  "folder_path": "<путь к папке установки RedDatabase>"
},
"databases": {
  "<псевдоним>": {
    "alias": "<алиас>",
    "scrape": {
      "enabled": true,
      "filters": "<путь к файлу с настройками фильтрации>",
      "shmem_size": <размер разделяемой памяти в МиБ>,
      "endpoint": "tcp://<127.0.0.1>:<порт>"
    }
  }
}
}
}
```

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 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_queries` - Определяет, собирать ли данные с помощью агрегирующего трейса. По умолчанию установлено значение `false`. Агрегатный аудит необходимо предварительно настроить *по инструкции*.

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

Опция `endpoint` определяет сокет для каждой базы данных на Windows. Опция обязательна при настройке на Windows, на Linux не используется.

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 `promtail`:

- **host** - определяет ip-адрес устройства наблюдаемого сервера, указанное значение должно совпадать со значением в конфигурационном файле Prometheus. параметр имеет строковый тип.

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 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;
- **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 агент мониторинга 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`, 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": {
  "<псевдоним>": {
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

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

Где:

- псевдоним - уникальный псевдоним базы данных;
- `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`. Собираются 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 распространяются на указанную базу данных.

Пример настроенного файла `exporter_conf.json`:

```
{
  "exporter": {
    "port": 3051,
    "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,
      "max_user_metrics": 100,
      "max_metric_data_blocks": 10
    }
  },
  "promtail": {
    "host": "127.0.0.1"
  },
  "dictserver": {
    "host": "127.0.0.1",
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
"port": 3050,  
"database": "/opt/rdbmonitor/server/rdb/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"  
},  
"databases": {  
  "employee_db": {  
    "alias": "employee",  
    "scrape": {  
      "enabled": true,  
      "filters": "filters_employee.ini"  
    }  
  }  
}  
}
```

6. Перед запуском агента мониторинга убедитесь, что СУБД запущена:

```
systemctl status firebird
```

В конфигурационном файле агент мониторинга в секции `scrape_queries` должно быть установлено значение `"enabled": true`.

Запустите клиентскую часть РБДМонитор:

```
systemctl start rdbmonitor-client.target
```

## Глава 5

# Мониторинг при миграции РЕД Базы Данных 3 на версию 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-client.target
```

2. Удалите файлы `/dev/shm/rdb_exporter_*` и каталог `/tmp/reddatabase`.
3. Выполните миграцию наблюдаемой СУБД на РЕД Базу Данных 5, следуя инструкции в [Руководстве администратора](#).

Если клиент и сервер РБДМонитор установлены на одну машину, то необходимо мигрировать `statemens.fdb` на РЕД Базу Данных 5.

4. Запустите агент мониторинга:

```
systemctl start rdbmonitor-client.target
```

## Глава 6

# Просмотр метрик

Собираемые метрики описаны в приложении *Собираемые метрики*.

## 6.1 Пользовательские метрики

Для добавления пользовательской метрики необходимо создать описывающий её файл с расширением `.sh` по пути `/opt/rdbmonitor/exporter/UserMetrics`.

Структура описания метрики:

```
#!/bin/bash

# Название метрики
echo "имя_метрики"

echo "#" # Разделительный символ

# Опциональный список меток
echo "имя_метки" # Название метки 1
echo "имя_метки" # Название метки 2

echo "#" # Разделительный символ

# Опциональный список характеристик меток
# Характеристика метки 1
echo "" # Значение метки
echo "lable_var1" # Название метки 1
echo "lable_var2" # Название метки 2

echo "#" # Разделительный символ

# Характеристика метки 2
...
# Характеристика метки n
```

Пример описания метрики:

```
#!/bin/bash

# Чтение значений памяти (в килобайтах)
read TOTAL_MEMORY FREE_MEMORY <<< $(awk '/MemTotal/ {t=$2} /MemFree/ {f=$2} END {print t, f}' /proc/meminfo)

# Расчет использованной памяти
USED_MEMORY=$((TOTAL_MEMORY - FREE_MEMORY))

# Блок с названием метрики
echo "meminfo"
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
# Разделительный символ
echo "#"

# Блок с названиями меток
echo "type" # Название метки

# Разделительный символ
echo "#"

# Блок со значением метрики (total)
echo "$TOTAL_MEMORY" # Вывод значения метрики
echo "total"          # Значение метки

# Разделительный символ
echo "#"

# Блок со значением метрики (free)
echo "$FREE_MEMORY"  # Вывод значения метрики
echo "free"          # Значение метки

# Разделительный символ
echo "#"

# Блок со значением метрики (used)
echo "$USED_MEMORY"  # Вывод значения метрики
echo "used"          # Значение метки
```

Далее нужно включить отображение пользовательских метрик в `exporter_conf.json`, настроив секцию `"user_metrics"`:

```
"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>
}
```

Пользовательские метрики просматриваются только с помощью Prometheus. Для просмотра пользовательских метрик в Grafana нужно самостоятельно создать панель.

## 6.2 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`:

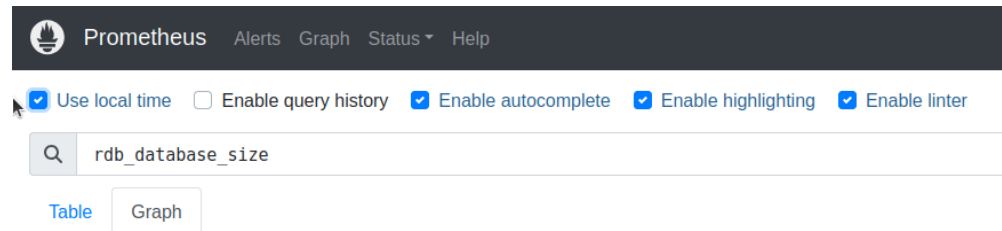


Рисунок 6.1 — Использование локального времени

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.

## 6.3 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 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.

После добавления источников (подробнее см. grafana) необходимо указать их в настройках плагина РБДМонитор:

## Источники данных

### Хранилище запросов

СУБД Ред База Данных

redsoft-rdbconnector-datasource

### Метрики

Prometheus

prometheus

### Логи

Loki

loki

### Интервал сбора метрик

Prometheus

10s

Сохранить

Рисунок 6.2 — Настройка плагина

После нажатия на РБДМонитор в списке приложений (Apps) откроется Обзорная страница.

## 6.4 Анализ метрик

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 отчёта и заполните все поля:
  - Разделяющая метка
  - Продолжительность - определяет периоды времени до и после разделяющей метки в часах (ч) или минутах (м)
  - Допустимая доля изменений в %
3. Нажмите Сгенерировать отчёт.

Не покидайте и не обновляйте страницу до конца формирования отчёта.

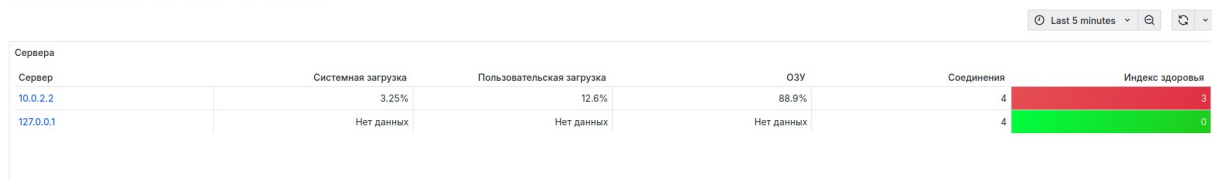
После формирования отчёта появится возможность его экспортировать. Сформированный отчёт будет добавлен на вкладку **Отчёты**.

## 6.5 Обзорная страница

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

### Обзорная страница

Чтобы подробнее изучить сервер, нажмите на его сетевой интерфейс.



Сервер	Системная нагрузка	Пользовательская нагрузка	ОЗУ	Соединения	Индекс здоровья
10.0.2.2	3.25%	12.6%	88.9%	4	3
127.0.0.1	Нет данных	Нет данных	Нет данных	4	0

Рисунок 6.3 — Обзорная страница

## 6.6 Информация о выбранном сервере

На этой странице находится подробная информация о выбранном сервере.

### Обзор сервера 127.0.0.1

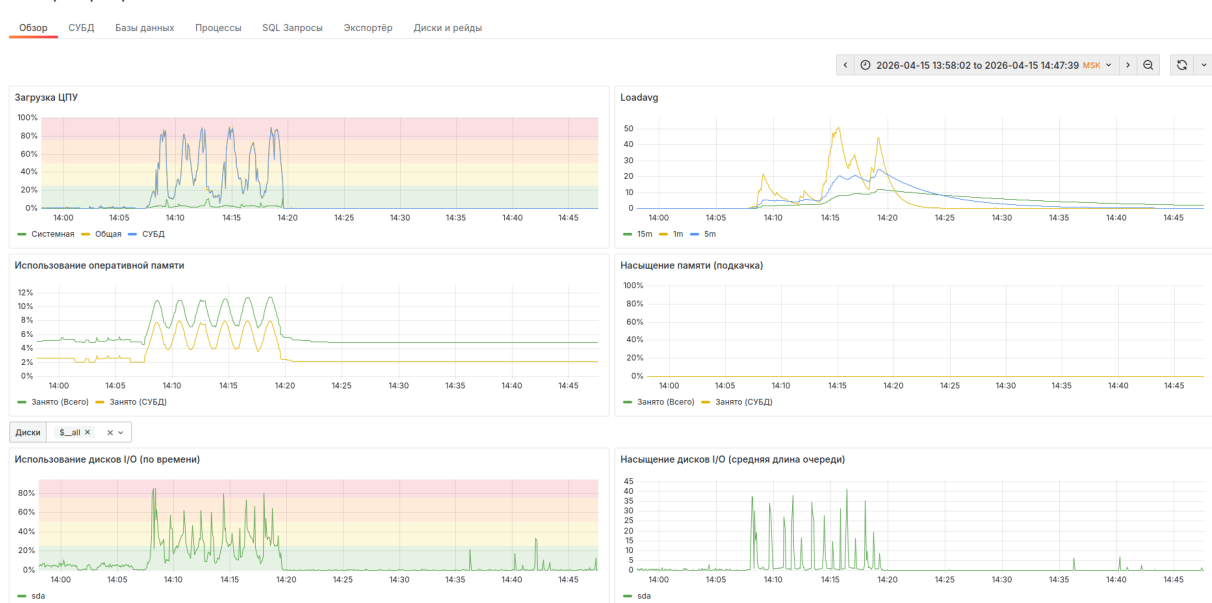


Рисунок 6.4 — Обзор выбранного сервера - Часть 1

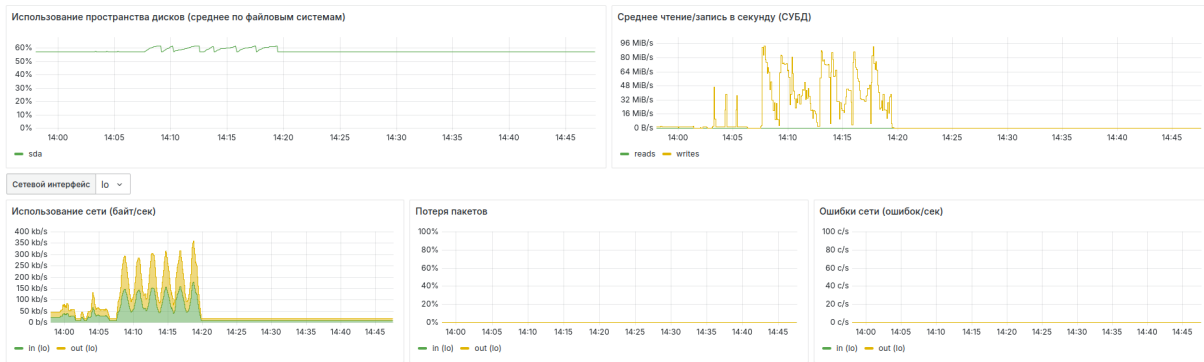


Рисунок 6.5 — Обзор выбранного сервера - Часть 2

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.

### 6.6.1 СУБД

Обзор сервера 127.0.0.1

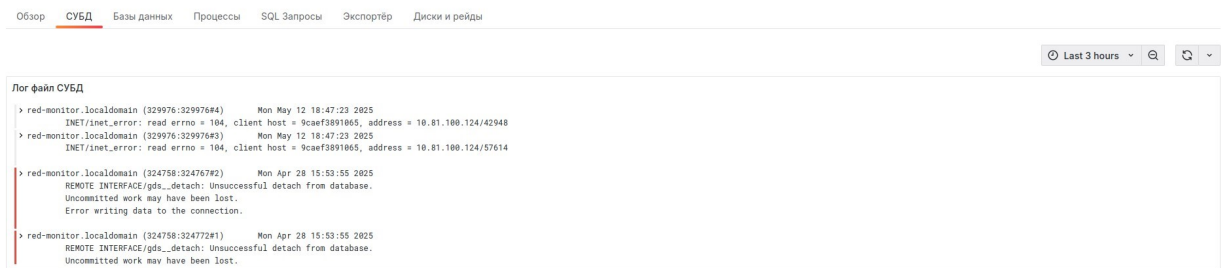


Рисунок 6.6 — Лог СУБД

Панель Лог файл СУБД показывает содержание файла firebird.log.

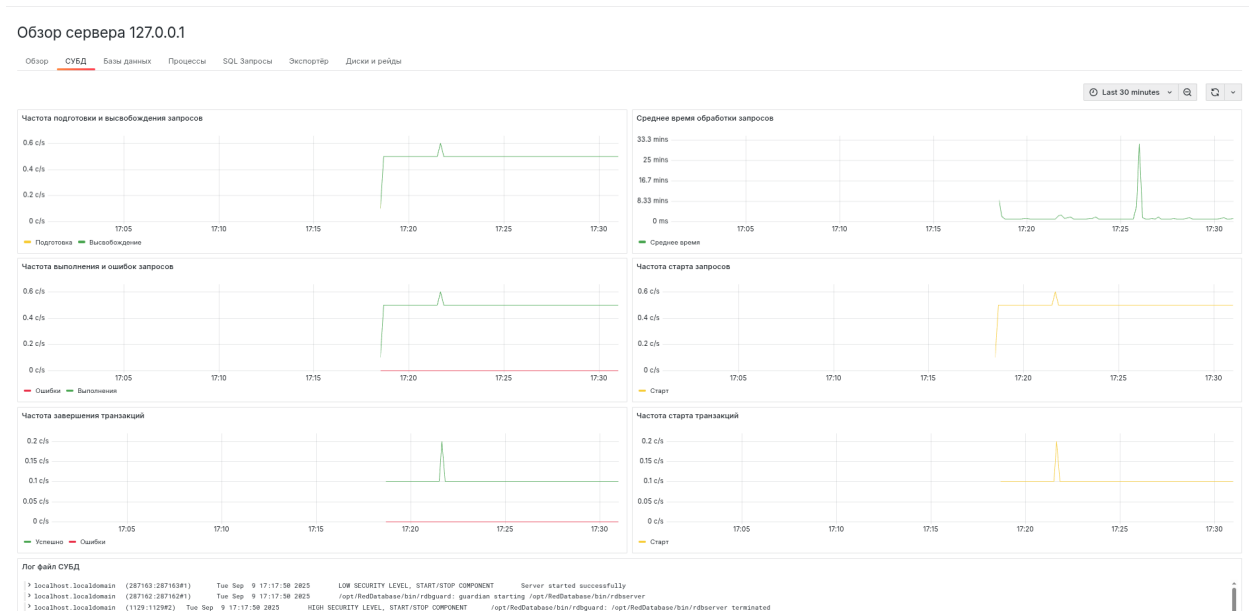


Рисунок 6.7 — Общая информация о СУБД

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.

## 6.6.2 Базы данных

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

Home > Apps > Обзорная страница > Обзор сервера 127.0.0.1

Обзор сервера 127.0.0.1

Обзор СУБД **Базы данных** Процессы SQL Запросы Экспортёр Диски и рейды

Last 3 hours

Базы данных	База данных	Размер базы даннь	Диск	Раздел	Точка монтировани	Занято файл. сис.	Размер файл. сис.	Соединения	Старт транз/сек	Активные транзакц	Завершение транз/	Запросов/сек
employee_db		2.87 MiB	sda	ro_redos-root	/	0 B	0 B	Нет данных	0	1	0	0.133

Рисунок 6.8 — Обзор баз данных

### 6.6.3 Информация о выбранной базе данных

На этой странице находится подробная информация о базе данных.

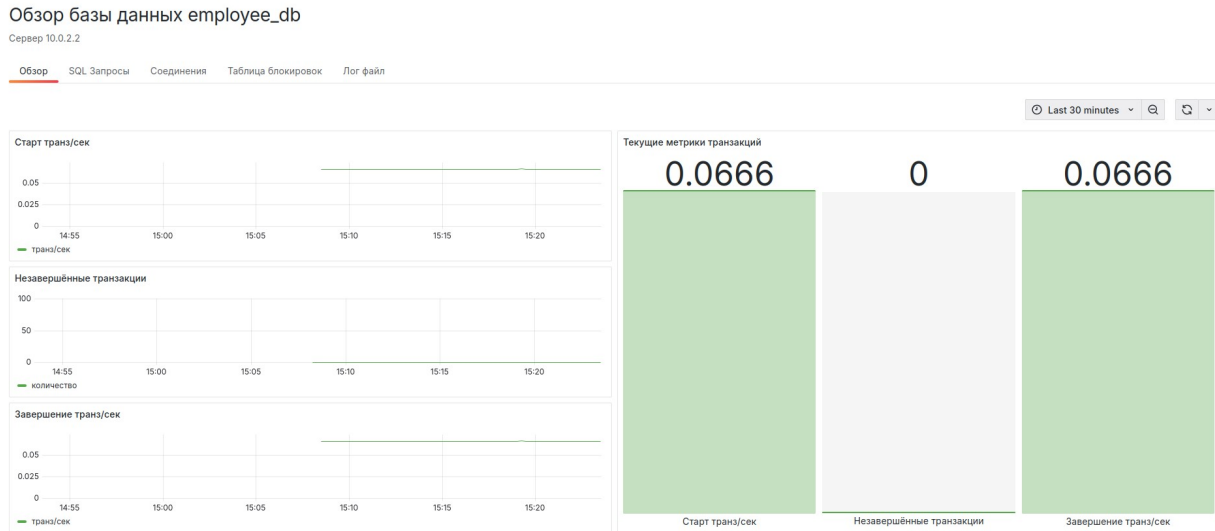


Рисунок 6.9 — Обзор выбранной базы данных - Часть 1

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 6.10 — 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.

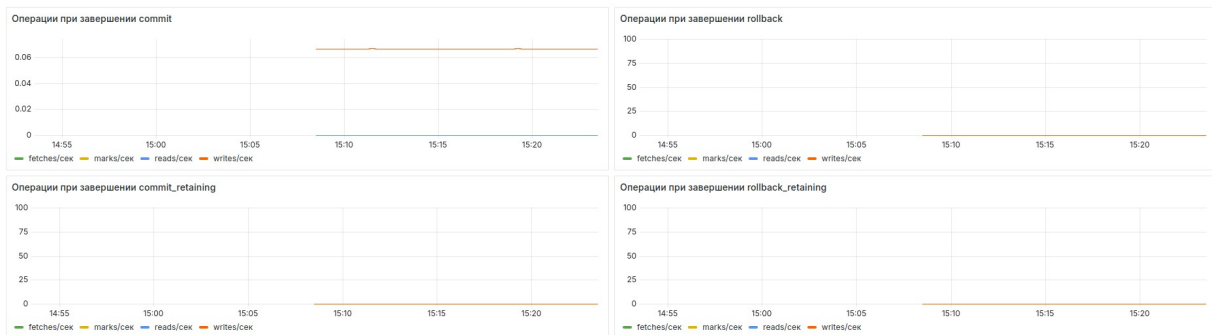


Рисунок 6.11 — Обзор выбранной базы данных - Часть 3

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.

Обзор базы данных employee\_db  
Сервер 127.0.0.1

Обзор SQL Запросы Соединения Таблица блокировок Лог файл

Тип к: 10 Характер метрики За интервал: Метрика: Число выполнений

Top запросов

Хеш	Число выполнений	Fetches	Reads	Writes	Marks	Время выполнения	Число выполнений	Среднее Fetches	Среднее Reads	Среднее Writes	Среднее Marks	Среднее время выпол
10388932830054856...	93	382	2	2	0	36.6 ms	1	4.11	0.0215	0.0215	0	0.393 ms
12826355705395996037	93	0	0	0	0	4.98 ms	1	0	0	0	0	0.0535 ms
13090851087413721210	93	0	0	0	0	4.93 ms	1	0	0	0	0	0.0530 ms
2717352643558139388	93	0	0	0	0	4.43 ms	1	0	0	0	0	0.0477 ms
2913147644868675017	2	232	2	2	0	1.09 ms	1	116	1	1	0	0.546 ms
655285844143685659	93	0	0	0	0	1.34 ms	1	0	0	0	0	0.0144 ms

Рисунок 6.12 — SQL запросы выбранной базы данных

## Обзор запроса

На данной странице находится подробная информация о выбранном запросе.

### Обзор запроса 680d0a53b06027a9

Сервер 10.0.2.2, база данных test\_db

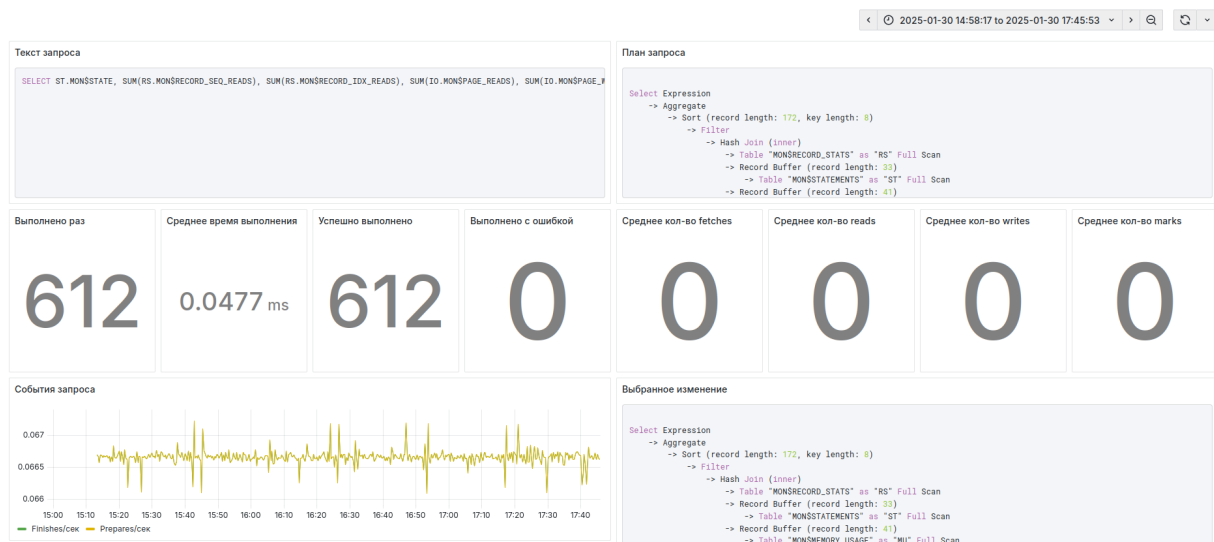


Рисунок 6.13 — Обзор запроса - Часть 1

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

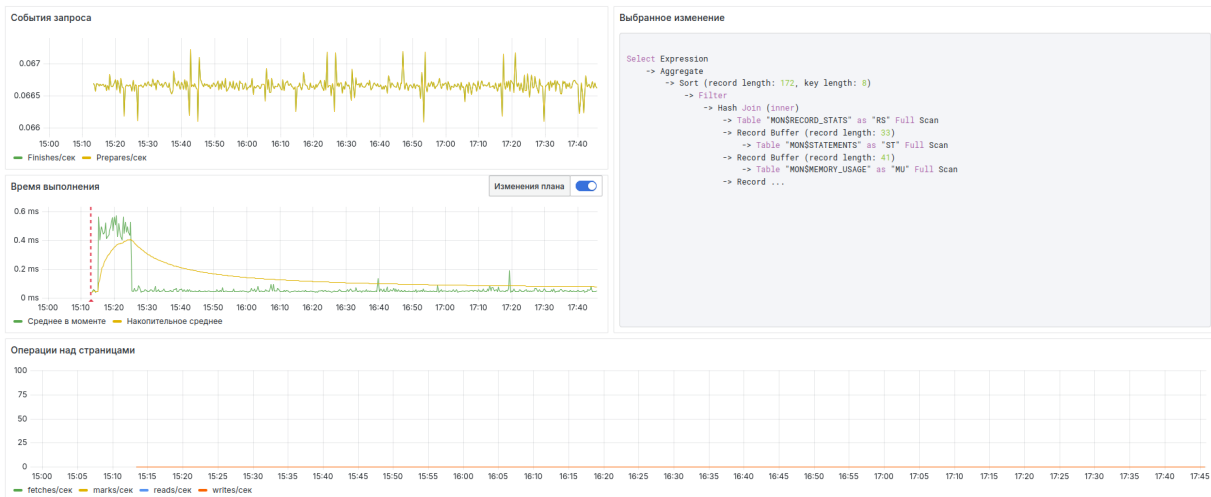


Рисунок 6.14 — Обзор запроса - Часть 2

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.

### Обзор базы данных test\_db

Сервер 10.0.2.2

Обзор SQL Запросы **Соединения** Таблица блокировок Лог файл

Топ k = 10  Метрики  Fetches  Reads  Writes  Marks  Длительность соединения Last 24 hours

ID Соединения	Имя пользователя	Адрес клиента	Процесс клиента	ID процесса сервера	Fetches	Reads	Writes	Marks	Длительность соединен
39	'SYSDBA '	'127.0.0.1/55806'	"/home/eyedm/Projects/re	29962	227	0	50	101	12.5 mins
40	'Cache Writer '			29962	0	0	0	0	12.5 mins
41	'Garbage Collector '			29962	0	0	0	0	12.5 mins
42	'SYSDBA '	'127.0.0.1/34614'	"/home/eyedm/Projects/re	35165	0	0	0	0	12.5 mins

Рисунок 6.15 — Соединения с базой данных

## Обзор соединения

На данной странице находится подробная информация о выбранном соединении.

### Обзор соединения 39

Сервер 10.0.2.2, база данных test\_db

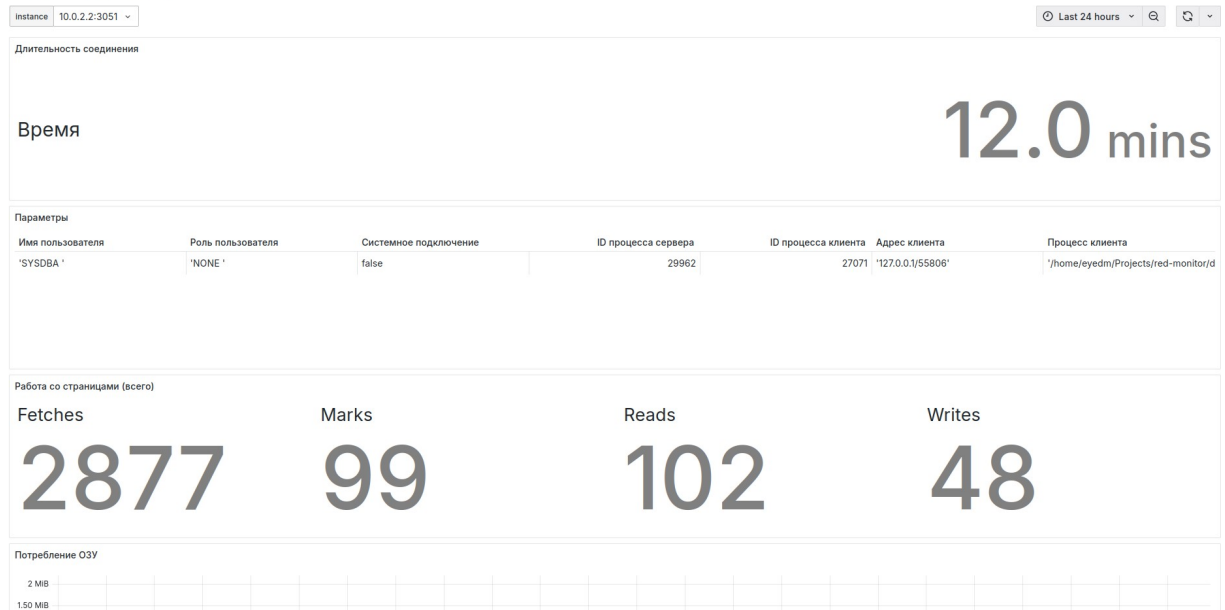


Рисунок 6.16 — Обзор соединения

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.

### Обзор базы данных employee\_db

Сервер 127.0.0.1

Обзор SQL Запросы Соединения **Таблица блокировок** Лог файл

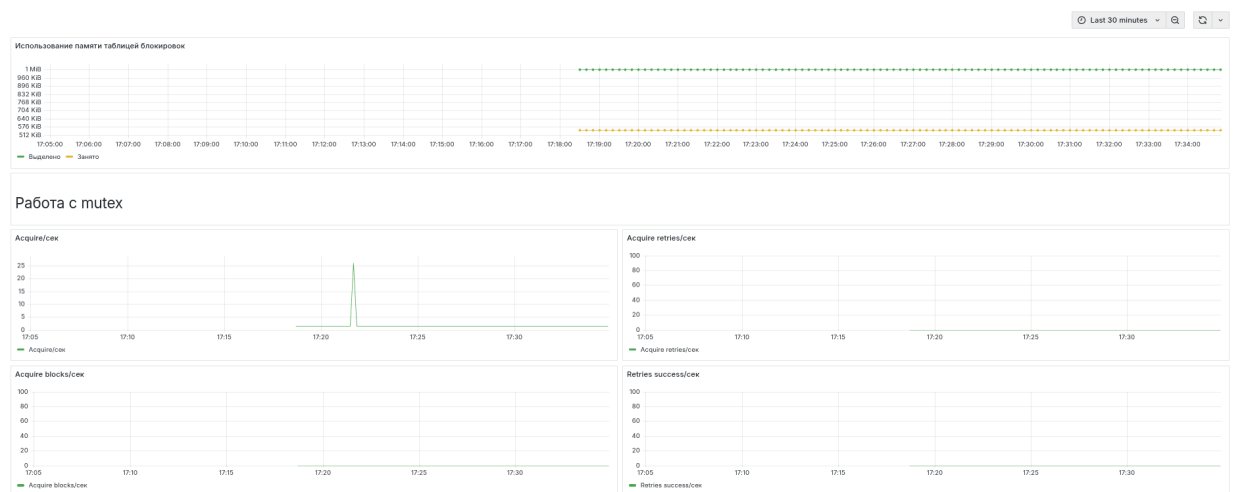


Рисунок 6.17 — Использование памяти таблицей блокировок

## Лог файл

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

### Обзор базы данных employee\_db

Сервер 10.81.100.124

Обзор SQL Запросы Соединения Таблица блокировок **Лог файл**

Last 7 days

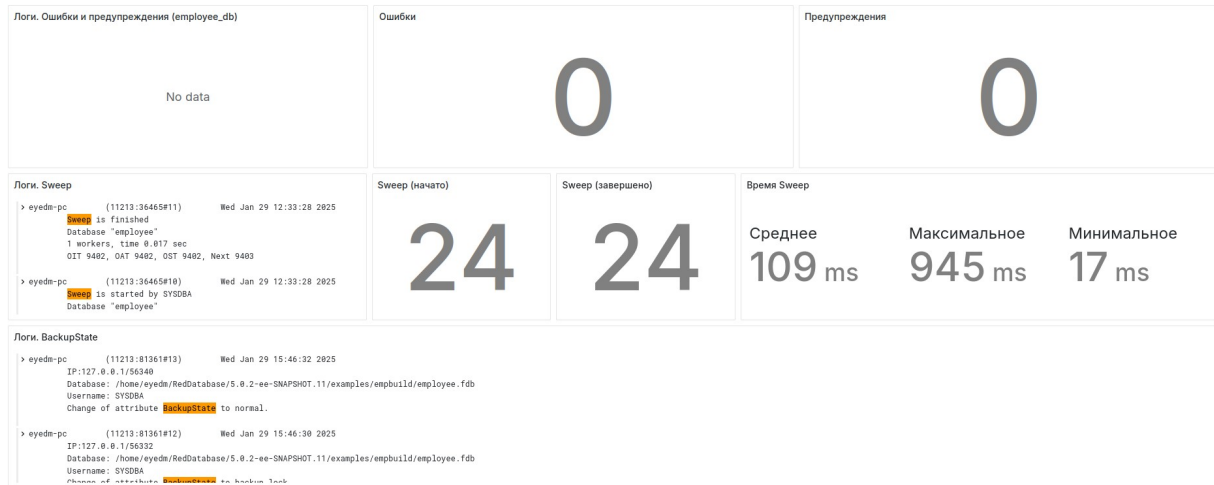


Рисунок 6.18 — Анализ лог файла

## 6.6.4 Процессы

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

### Обзор сервера 127.0.0.1

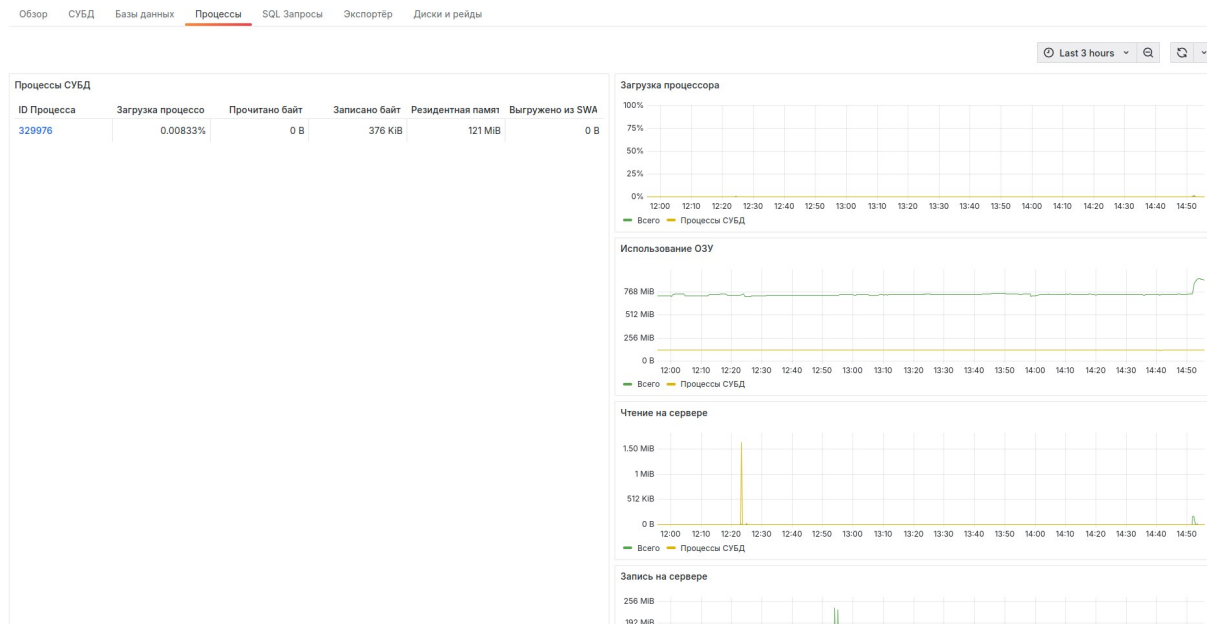


Рисунок 6.19 — Процессы СУБД

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.

Обзор процесса 359413  
Сервер 10.0.2.2



Рисунок 6.20 — 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.

### 6.6.5 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](#).

Обзор сервера 127.0.0.1

Обзор СУБД Базы данных Процессы **SQL Запросы** Экспортёр Диски и рейды

Топ N: 10 Характер метрики Дельта Метрика Число выполнений Last 5 minutes

База данных	ID Запроса	Число выполнений	Fetches	Reads	Writes	Marks	Время выполнен	Среднее Fetches	Среднее Reads	Среднее Writes	Среднее Marks	Среднее время в
employee_db	10015	20	0	0	0	0	0.789 ms	0	0	0	0	0.0395 ms
employee_db	10012	20	0	0	0	0	1.66 ms	0	0	0	0	0.0830 ms

Рисунок 6.21 — Запросы

### 6.6.6 Агент мониторинга

На странице находится информация о сборе метрик агентом мониторинга.

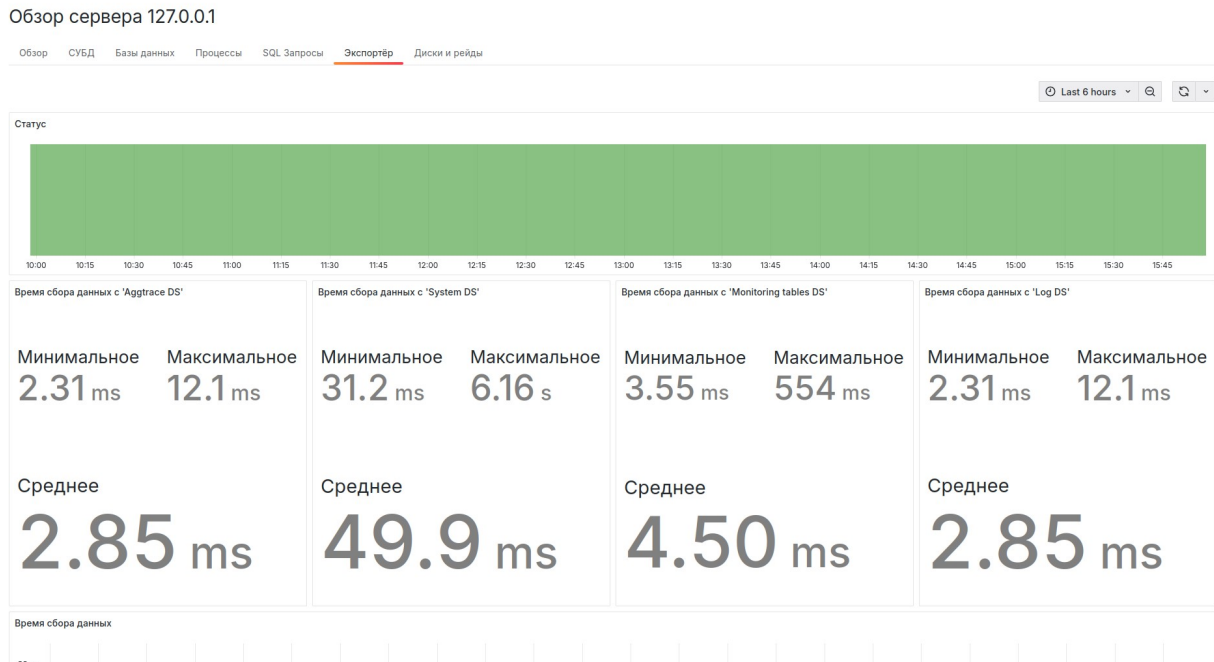


Рисунок 6.22 — Агент мониторинга

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.

### 6.6.7 Диски и рейды

На странице отображается информация о выбранных дисках.

## Обзор сервера 10.0.2.15

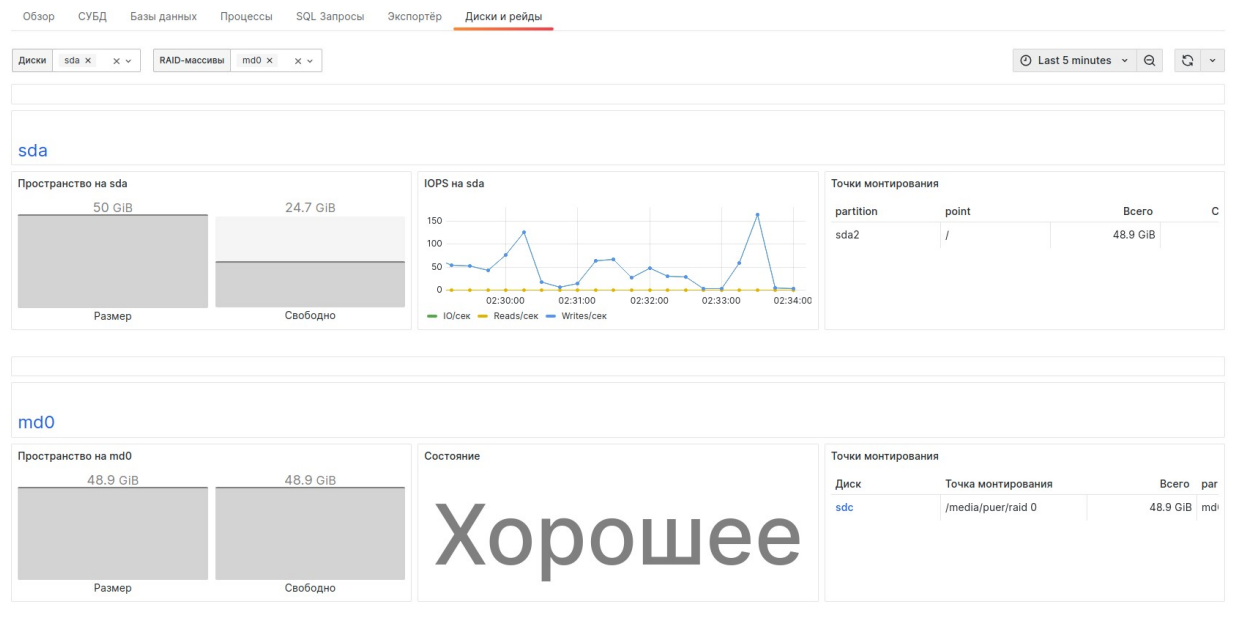


Рисунок 6.23 — Информация о диске

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.

## Обзор RAID-массива md0

Сервер 10.0.2.15

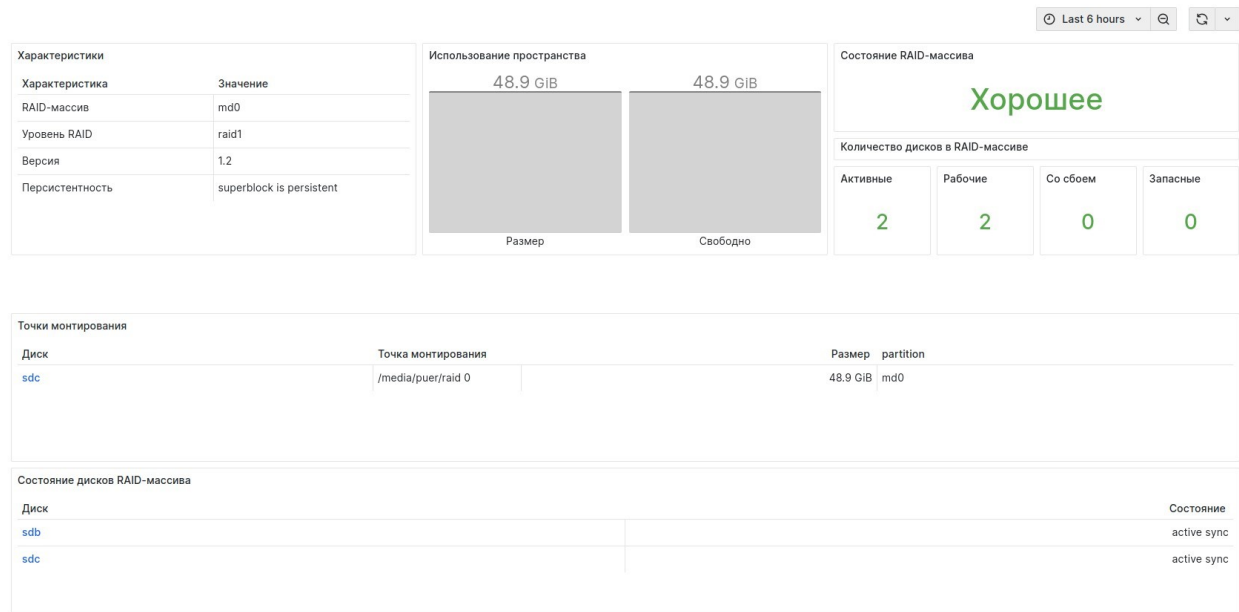


Рисунок 6.24 — Информация о RAID-массиве

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".

## 6.7 Мониторинг кластеров

### 6.7.1 Необходимые настройки

Необходимо установить Сервер СУБД на каждый узел кластера по инструкции.

В конфигурационном файле `prometheus.yml` в параметре `targets` нужно указать сокет каждого узла. Подробнее о настройке Prometheus см. Основные настройки Prometheus.

## 6.7.2 Обзор кластеров

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

### Обзор кластеров

Страница обзора кластеров

Last 6 hours 🔍 ↻

Кластеры		
Имя кластера	Активные узлы	Кол-во узлов
<a href="#">single_cluster</a>	1	1
<a href="#">rdbcluster_monitor</a>	3	3

Рисунок 6.25 — Обзор кластеров

## 6.7.3 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](#).

### Обзор кластера rdbcluster\_monitor

Страница обзора кластеров

Обзор Временные метрики

Last 6 hours 🔍 ↻

Узлы							
Сервер	Имя узла	Роль	Время работы	Может стать ведущим	Режим работы	Синхронизирован	Обслуживание
10.81.100.161	mini-pc0	Ведущий	2.01 weeks	Уже является ведущим	Синхронный	✓	Выключено
10.81.100.194	mini-pc1	Реплика	1.11 weeks	Да	Синхронный	✓	Выключено
10.81.100.107	mini-pc2	Реплика	2.01 weeks	Да	Синхронный	✓	Выключено

Базы данных					
Псевдоним	Имя узла	Путь к базе данных	Поколение репликации	Номер журнала репликации	Синхронизация
test	mini-pc0	/db/test_database.fdb	1	0	✓
test	mini-pc1	/db/test_database.fdb	1	0	✓
test	mini-pc2	/db/test_database.fdb	1	0	✓

Рисунок 6.26 — Информация о выбранном кластере

### Временные метки

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

## Обзор кластера rdbcluster\_monitor

Страница обзора кластеров

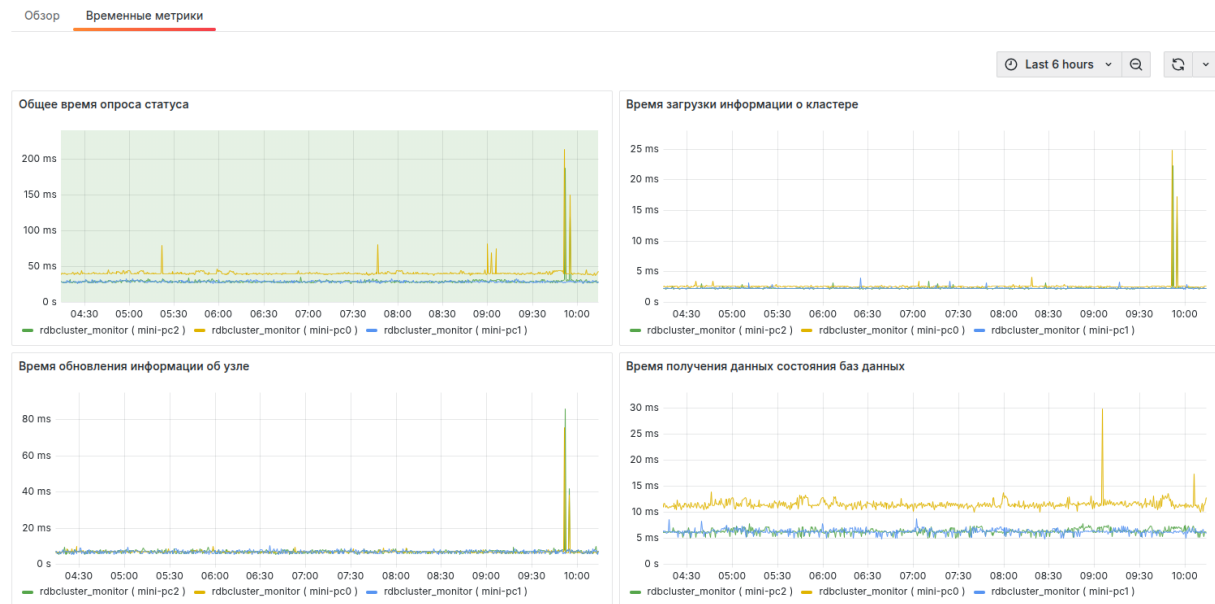


Рисунок 6.27 — Временные метрики

### 6.7.4 Информация об узле кластера

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

## Обзор сервера 10.81.100.161

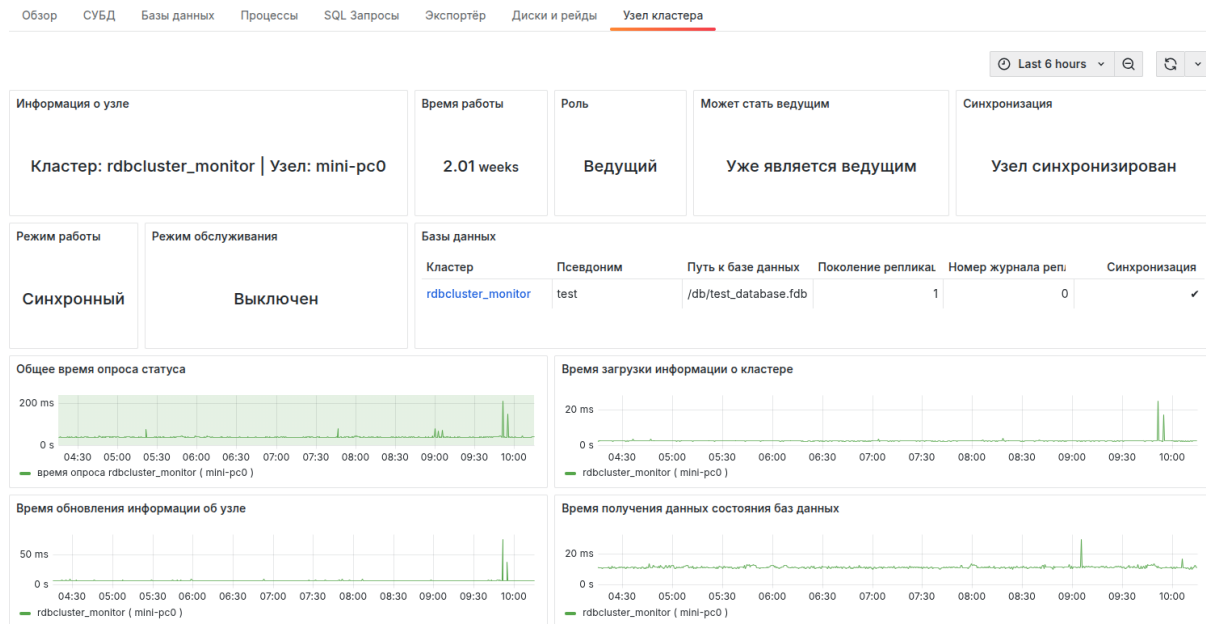


Рисунок 6.28 — Информация об узле кластера

## Приложение А Собираемые метрики

### А.1 Метрики таблиц мониторинга

Таблица А.1 — Метрики таблиц мониторинга

Название метрики	Метки	Описание метрики
rdb_call_stack_pages	<p>database - база данных.</p> <p>object_type - тип объекта базы данных:</p> <ul style="list-style-type: none"> <li>• procedure;</li> <li>• trigger;</li> <li>• function.</li> </ul> <p>operation - тип операции:</p> <ul style="list-style-type: none"> <li>• reads;</li> <li>• fetches;</li> <li>• marks;</li> <li>• writes.</li> </ul>	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"> <li>• procedure;</li> <li>• trigger;</li> <li>• function;</li> </ul> <p>usage - использование памяти:</p> <ul style="list-style-type: none"> <li>• used - объём используемой памяти;</li> <li>• allocated - количество выделенной памяти.</li> </ul>	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"> <li>• procedure;</li> <li>• trigger;</li> <li>• function;</li> </ul>	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"> <li>• reads;</li> <li>• fetches;</li> <li>• marks;</li> <li>• writes.</li> </ul>	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"> <li>• used - объём используемой памяти;</li> <li>• allocated - количество выделенной памяти.</li> </ul>	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"> <li>• consistency</li> <li>• concurrency</li> <li>• read_committed_rec_ver</li> <li>• read_committed_no_rec_ver</li> <li>• read_committed_read_consistency</li> </ul> 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"> <li>• reads;</li> <li>• fetches;</li> <li>• marks;</li> <li>• writes.</li> </ul>	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"> <li>• consistency</li> <li>• concurrency</li> <li>• read_committed_rec_ver</li> <li>• read_committed_no_rec_ver</li> <li>• read_committed_read_consistency</li> </ul> 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"> <li>• used - объём используемой памяти.</li> <li>• allocated - количество выделенной памяти.</li> </ul>	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"> <li>• consistency</li> <li>• concurrency</li> <li>• read_committed_rec_ver</li> <li>• read_committed_no_rec_ver</li> <li>• read_committed_read_consistency</li> </ul> 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"> <li>• idle — бездействующий;</li> <li>• active — активный;</li> <li>• stalled — приостановленный, то есть запрос или курсор "живой", но в данный момент не выполняется.</li> </ul> <p>operation - тип операции:</p> <ul style="list-style-type: none"> <li>• reads;</li> <li>• fetches;</li> <li>• marks;</li> <li>• writes.</li> </ul>	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"> <li>• idle — бездействующий;</li> <li>• active — активный;</li> <li>• stalled — приостановленный, то есть запрос или курсор "живой", но в данный момент не выполняется.</li> </ul> <p>usage - использование памяти:</p> <ul style="list-style-type: none"> <li>• used - объём используемой памяти.</li> <li>• allocated - количество выделенной памяти.</li> </ul>	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"> <li>• idle — бездействующий;</li> <li>• active — активный;</li> <li>• stalled — приостановленный, то есть запрос или курсор "живой", но в данный момент не выполняется.</li> </ul>	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. <code>Read-only</code> - значение 1, <code>read-write</code> значение 0.
rdb_database_forced_writes	database - база данных.	Указывает, установлен для базы режим синхронного вывода ( <code>forced writes</code> , 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"> <li>• 0 — база не затронута бэкапом,</li> <li>• 1 — база заблокирована для резервирования,</li> <li>• 2 — объединение временного файла дельты и основного файла базы данных.</li> </ul>
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 ( <code>shutdown</code> ) базы данных: <ul style="list-style-type: none"> <li>• 0 — база данных активна (<code>online</code>);</li> <li>• 1 — остановлена для нескольких пользователей (<code>multi-user shutdown</code>);</li> <li>• 2 — остановлена для одного пользователя (<code>single-user shutdown</code>);</li> <li>• 3 — полностью остановка (<code>full shutdown</code>).</li> </ul>
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: <ul style="list-style-type: none"> <li>• 0 - соединение отсутствует;</li> <li>• 1 - соединение установлено.</li> </ul>

## 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.

## А.3 Метрики утилиты rdb\_lock\_print

Таблица А.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_convertts	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: <ul style="list-style-type: none"> <li>• reads</li> <li>• writes</li> </ul>	Количество операций чтения и записи на диск.
rdb_disks_io_bytes	disk - диск. operation - тип операции: <ul style="list-style-type: none"> <li>• reads</li> <li>• writes</li> </ul>	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"> <li>• normal - время, затраченное на фактический ввод/вывод;</li> <li>• weighted - взвешенное время, затраченное на фактический ввод/вывод.</li> </ul>	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"> <li>• used</li> <li>• free</li> <li>• total</li> </ul>	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"> <li>• table</li> <li>• blob</li> <li>• undo</li> <li>• redbuf</li> <li>• merge</li> <li>• sort</li> <li>• tpc</li> <li>• snap</li> </ul>	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"> <li>• table</li> <li>• blob</li> <li>• undo</li> <li>• redbuf</li> <li>• merge</li> <li>• sort</li> <li>• tpc</li> <li>• snap</li> </ul>	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"> <li>• system</li> <li>• user</li> <li>• iowait</li> </ul>	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"> <li>• resident - резидентная память</li> <li>• virtual - виртуальная память</li> <li>• swapped_out - занимаемая память в пространстве swap</li> <li>• shared - разделяемая память: text, lib, data, dirty, uss, pss.</li> </ul>	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"> <li>• reads</li> <li>• writes</li> </ul>	Количество операций ввода/вывода, выполняемых процессом СУБД.
rdb_procs_io_bytes	pid - идентификатор процесса СУБД operation - тип операции: <ul style="list-style-type: none"> <li>• reads</li> <li>• writes</li> </ul>	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"> <li>• free</li> <li>• used</li> <li>• total</li> </ul>	Использование оперативной памяти.
rdb_swap_usage	usage: <ul style="list-style-type: none"> <li>• used</li> <li>• free</li> <li>• total</li> </ul>	Использование памяти, выделенной для swap.
rdb_swapped	direction: <ul style="list-style-type: none"> <li>• in</li> <li>• out</li> </ul>	Объём данных (в байтах), затронутый swap.
rdb_cpu_freq	marker: <ul style="list-style-type: none"> <li>• current</li> </ul>	Частота работы процессора.
rdb_cpu_times	kind: <ul style="list-style-type: none"> <li>• system</li> <li>• user</li> <li>• iowait</li> <li>• irq</li> <li>• soft_irq</li> </ul>	Время, потраченное процессором.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_cpu_ctx		Число изменений контекста процессора.
rdb_cpu_cores	kind: <ul style="list-style-type: none"> <li>• logical</li> <li>• physical</li> </ul>	Количество ядер процессора.
rdb_cpu_interrupts	marker: <ul style="list-style-type: none"> <li>• soft</li> <li>• casual</li> </ul>	Число прерываний процессора.
rdb_tmp_mount_point_usage	point usage: <ul style="list-style-type: none"> <li>• total</li> <li>• used</li> <li>• free</li> </ul>	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 Метрики RAID-массивов

Таблица А.5 — Метрики RAID-массивов

Название метрики	Метки	Описание метрики
rdb_raid_disks_count	raid - имя массива; state - состояние диска: <ul style="list-style-type: none"> <li>• active - количество активных дисков</li> <li>• working - количество рабочих дисков</li> <li>• failed - количество неисправных дисков</li> <li>• spare - количество запасных дисков</li> </ul>	Количество дисков, находящихся в конкретном состоянии.
rdb_raid_state	raid - имя массива.	Текущее состояние RAID-массива: <ul style="list-style-type: none"> <li>• 1 - clean</li> <li>• 2 - degraded</li> <li>• 3 - resyncing</li> <li>• 4 - recovering</li> <li>• 5 - faulty</li> </ul>
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: <ul style="list-style-type: none"> <li>• 0 - unknown (неизвестное состояние диска)</li> <li>• 1 - absent (диск отсутствует)</li> <li>• 2 - active (диск активен)</li> <li>• 3 - sync (диск синхронизирован)</li> <li>• 4 - spare (запасной диск)</li> <li>• 5 - write-mostly (диск используется преимущественно для записи)</li> <li>• 6 - blocked (диск заблокирован)</li> <li>• 7 - faulty (диск неисправен)</li> </ul> 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_transactions _markers	database - база данных; marker: <ul style="list-style-type: none"> <li>• NT</li> <li>• OST</li> <li>• OAT</li> <li>• OIT</li> </ul>	Значения маркеров транзакций в момент сбора информации.
rdb_log_errors	kind - тип ошибки: <ul style="list-style-type: none"> <li>• fatal</li> <li>• critical</li> <li>• normal</li> <li>• event</li> </ul>	Количество ошибок указанного типа.
rdb_server_version	kind: major, minor, fix	Версия СУБД.