

---

Ред Монитор  
Версия 0.6.1  
Руководство пользователя

# Содержание

1	Общие сведения о программе	4
1.1	Назначение программы	4
1.2	Минимальный состав аппаратных средств	5
1.3	Минимальный состав программных средств	5
2	Установка и настройка Ред Монитор	6
2.1	Сервер СУБД	6
2.1.1	Ред База Данных	6
2.1.2	Экспортёр	6
	Установка экспортёра	6
	Настройка экспортёра	7
2.2	Сервер мониторинга	11
2.2.1	Prometheus	11
	Установка Prometheus	11
	Основные настройки Prometheus	12
	Секция global	13
	Секция rule_files	13
	Секция scrape_configs	13
	Секция alerting	14
	Запуск Prometheus	14
2.2.2	Alertmanager	14
	Установка Alertmanager	14
	Основные настройки Alertmanager	15
	Секция global	16
	Секция route	16
	Секция receivers	17
	Секция inhibit_rules	17
	Включение уведомлений	17
	Запуск Alertmanager	18
2.2.3	Grafana	18
2.2.4	Ред База Данных	19
2.3	Установка на один сервер	19
3	Просмотр метрик	20
3.1	Просмотр метрик с помощью Prometheus	20
3.2	Просмотр метрик с помощью Grafana	21
3.3	Обзорная страница	22
3.4	Информация о выбранном сервере	23
3.4.1	Индекс здоровья	24
3.4.2	Базы данных	25
3.4.3	Информация о выбранной базе данных	26
	SQL запросы	27
	Соединения	28
3.4.4	Процессы	30
	Обзор процесса	30
3.4.5	SQL-запросы	31
3.4.6	Экспортёр	32
Приложение А	Собираемые метрики	34
А.1	Метрики таблиц мониторинга	34
А.2	Метрики агрегатного аудита	39

A.3	Метрики утилиты <code>rdb_lock_print</code> . . . . .	41
A.4	Метрики репликации . . . . .	43
A.5	Метрики операционной системы . . . . .	44
A.6	Другие метрики . . . . .	47

## Глава 1

# Общие сведения о программе

## 1.1 Назначение программы

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

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

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

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

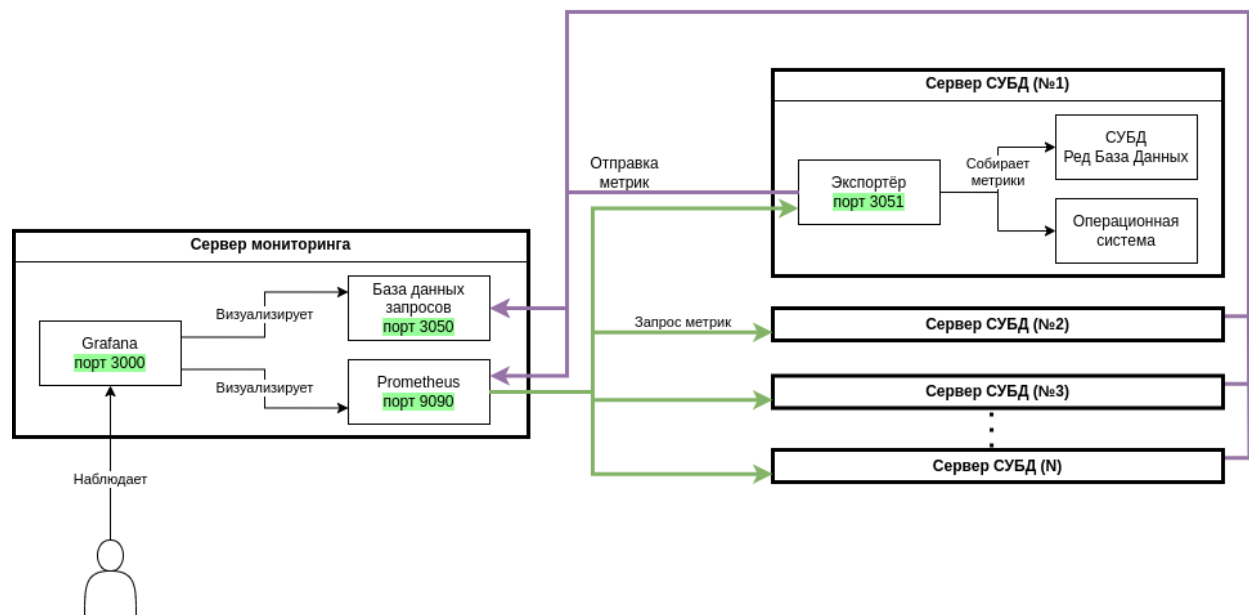


Рисунок 1.1 — Схема взаимодействия компонентов Ред Монитора

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

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

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

Prometheus и Базы данных пользовательских запросов.

Порты, используемые по умолчанию:

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

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

## 1.2 Минимальный состав аппаратных средств

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

## 1.3 Минимальный состав программных средств

Перед началом работы с Ред Монитор необходимо:

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

## Глава 2

# Установка и настройка Ред Монитор

## 2.1 Сервер СУБД

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

### 2.1.1 Ред База Данных

На Наблюдаемой СУБД для сбора метрик пользовательских запросов необходимо:

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

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

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

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

### 2.1.2 Экспортёр

#### Установка экспортёра

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

```
tar -xzf <архив>
```

Перейдите в распакованную папку:

```
cd <папка>
```

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

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

- Опция `-o` указывает, какую операцию нужно выполнить: установку или обновление.
- Опция `-p` определяет, что нужно установить/обновить: экспортёр, плагины для Grafana или всё сразу.
- Опция `-d` устанавливает/обновляет плагины для Grafana с путями по умолчанию. Конфигу-

густационный файл Grafana будет установлен по пути `/etc/grafana/grafana.ini`, а папка с плагинами Grafana по пути `/var/lib/grafana/plugins/`.

- Опция `-h` выводит справку о доступных опциях.

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

Для удаления Ред Монитора выполните команду:

```
sudo ./uninstall.sh [<опции>]
```

```
<опции>:=  
  exporter  
| grafana_plugins  
| full
```

## Настройка экспортёра

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

Структура файла `exporter_conf.json`:

```
{  
  "exporter": {  
    "host": "<хост экспортёра>",  
    "port": <порт экспортёра>,  
    "scrape_rdb_lock_print": {  
      "enabled": true  
    },  
    "scrape_system": {  
      "enabled": false  
    },  
    "scrape_mon_tables": {  
      "enabled": true  
    },  
    "scrape_log": {  
      "enabled": true,  
      "start_from_end": true  
    },  
    "scrape_atrace": {  
      "enabled": true  
    }  
  },  
  "dictserver": {  
    "host": "<хост базы данных>",  
    "port": <порт базы данных>,  
    "database": "/db/statements.fdb",  
    "user": "<пользователь>",  
    "password": "<пароль>",  
    "Auth_plugins": ["<плагин>", ..., "<плагин>"]  
  },  
  "RedDatabase": {  
    "host": "<хост СУБД>",
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
"port": <порт СУБД>,
"user": "<логин>",
"password": "<пароль>",
"folder_path": "<путь к папке установки RedDatabase>",
"Auth_plugins": ["<плагин>", ..., "<плагин>"]
},
"databases": {
  "<псевдоним>": {
    "alias": "<алиас>",
    "path": "<путь к базе данных>"
  }
}
}
```

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

Параметры конфигурации:

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

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

Секция `dictserver`:

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



квадратные скобки;

Секция `RedDatabase`:

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

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

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

Где:

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

Пример настроенного файла `exporter_conf.json`:

```
{
  "exporter": {
    "host": "127.0.0.1",
    "port": 3051,
    "scrape_rdb_lock_print": {
      "enabled": true
    },
    "scrape_system":{
      "enabled": false
    },
    "scrape_mon_tables":{
      "enabled": true
    },
    "scrape_log": {
      "enabled": true,
      "start_from_end": true
    },
    "scrape_atrace": {
      "enabled": false
    }
  }
}
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
}
},
"dictserver": {
  "host": "172.17.0.2",
  "port": 3050,
  "database": "/db/statements.fdb",
  "user": "SYSDBA",
  "password": "masterkey",
  "Auth_plugins": ["Srp"]
},
"RedDatabase": {
  "host": "127.0.0.1",
  "port": 3050,
  "user": "SYSDBA",
  "password": "masterkey",
  "folder_path": "/opt/RedDatabase",
  "Auth_plugins": ["Srp"]
},
"databases": {
  "employee_db": {
    "alias": "employee",
    "path": "/opt/RedDatabase/examples/empbuild/employee.fdb"
  }
}
}
```

Перед запуском экспортёра убедитесь, что сервер СУБД запущен.

Запуск и остановка утилиты осуществляется с помощью `systemd`:

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

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

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

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

## 2.2 Сервер мониторинга

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

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

### 2.2.1 Prometheus

#### Установка Prometheus

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

Для установки Prometheus выполните команду:

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

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

1. Распаковать скачанный архив:

```
tar -xf <путь_к_архиву>
```

2. Переместить распакованный архив в предварительно созданную папку:

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

3. Создать пользователя и группу prometheus:

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

4. Назначить права на папку /opt/prometheus:

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

5. Создать unit-файл для запуска через службу:

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

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

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
--storage.tsdb.retention.time=30d
```

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

После создания unit-файла выполнить:

```
sudo systemctl daemon-reload
```

## Основные настройки Prometheus

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

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

Для настройки Prometheus используется файл `prometheus.yml`.

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

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

rule_files:
  - rules.yml

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

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

## Секция `global`

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

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

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

Параметр `scrape_timeout` устанавливает время ожидания получения метрик.

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

## Секция `rule_files`

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

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

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

## Секция `scrape_configs`

В секции `scrape_configs` задаются настройки для сбора метрик:

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

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

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

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

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

## Секция alerting

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

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

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

## Запуск Prometheus

Запуск `Prometheus` осуществляется следующей командой:

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

Или можно запустить службу:

```
sudo systemctl start prometheus
```

Более подробно о настройке `Prometheus` можно узнать на официальном сайте – [prometheus.io](https://prometheus.io).

### 2.2.2 Alertmanager

Если уведомления не нужны, то этот пункт можно пропустить.

## Установка Alertmanager

Скачать `Alertmanager` можно с официального сайта -- [prometheus.io](https://prometheus.io).

Для установки нужно выполнить следующие действия:

1. Распаковать скачанный архив:

```
tar -xf <путь_к_архиву>
```

2. Переместить распакованный архив в предварительно созданную папку:

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

3. Создать пользователя и группу `alertmanager`:

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

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

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

5. Создать unit-файл для запуска через службу:

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

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

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

После создания unit-файла выполнить:

```
sudo systemctl daemon-reload
```

## Основные настройки Alertmanager

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

Пример конфигурации Alertmanager:

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

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

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

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

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

### Секция `global`

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

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

Параметр `smtp_require_tls` указывает, использовать ли протокол TLS.

Параметр `smtp_smarthost` устанавливает адрес сервера почты.

Параметр `smtp_from` указывает почту отправителя.

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

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

### Секция `route`

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

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

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

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

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

Параметр `repeat_interval` задаёт время между повторной отправкой уведомлений.

Параметр `receiver` определяет получателя.



## Секция `receivers`

В секции `receivers` указывается список получателей уведомлений.

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

Параметр `name` указывает имя получателя.

Параметр `to` определяет почту получателя.

## Секция `inhibit_rules`

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

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

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

## Включение уведомлений

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

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

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

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

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

## Запуск Alertmanager

Запуск Alertmanager осуществляется следующей командой:

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

Или можно запустить службу Alertmanager:

```
sudo systemctl start alertmanager
```

### 2.2.3 Grafana

На Ред ОС Grafana можно установить через пакетный менеджер:

```
sudo dnf install grafana
```

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

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

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

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

```
Введите путь до папки плагинов: /var/lib/grafana/plugins
```

```
Копирование файлов...
```

```
Успешно!
```

```
Редактировать файл /etc/grafana/grafana.ini для включения плагинов в  
список доверенных?(д/н) да
```

```
Успешно!
```

По-умолчанию проверяется наличие файла `/etc/grafana/grafana.ini`, если он существует, то нужно разрешить его редактирование. Если файла не существует, то нужно будет ввести путь до другого файла с плагинами Grafana.

В случае успешной установки в файле `grafana.ini` будут указаны плагины `redsoft-redmonitor-app` и `redsoft-rdbconnector-datasource`:

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

Плагины для Grafana будут установлены на сервер мониторинга.

Запуск Grafana выполняется командой:

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

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

для доступа к Prometheus (по умолчанию `http://localhost:9090/`) и нажмите на кнопку **Save & test**.

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

## 2.2.4 Ред База Данных

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

Пример настройки скрипта:

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

Запуск скрипта:

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

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

## 2.3 Установка на один сервер

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

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

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

## Глава 3

# Просмотр метрик

Собираемые метрики описаны в приложении *Собираемые метрики*.

## 3.1 Просмотр метрик с помощью Prometheus

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

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

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

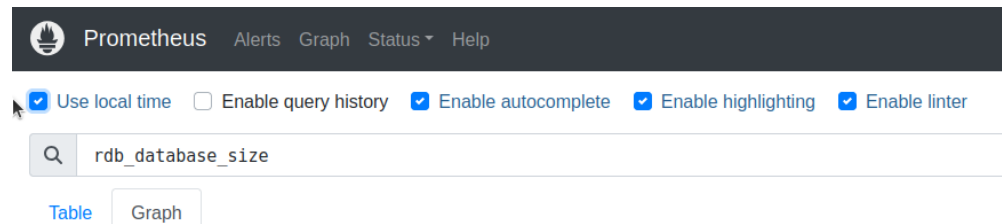


Рисунок 3.1 — Использование локального времени

Чтобы узнать значение конкретной метрики, нужно ввести PromQL запрос.

Примеры PromQL запросов:

1. Вывод объёма базы данных:

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

2. Вывод скорости записи на диск в секунду:

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

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

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

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

## 3.2 Просмотр метрик с помощью Grafana

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

Для отображения собранной информации необходимо включить плагин RedMonitor:

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

После этого в меню Apps будет добавлен плагин RedMonitor:

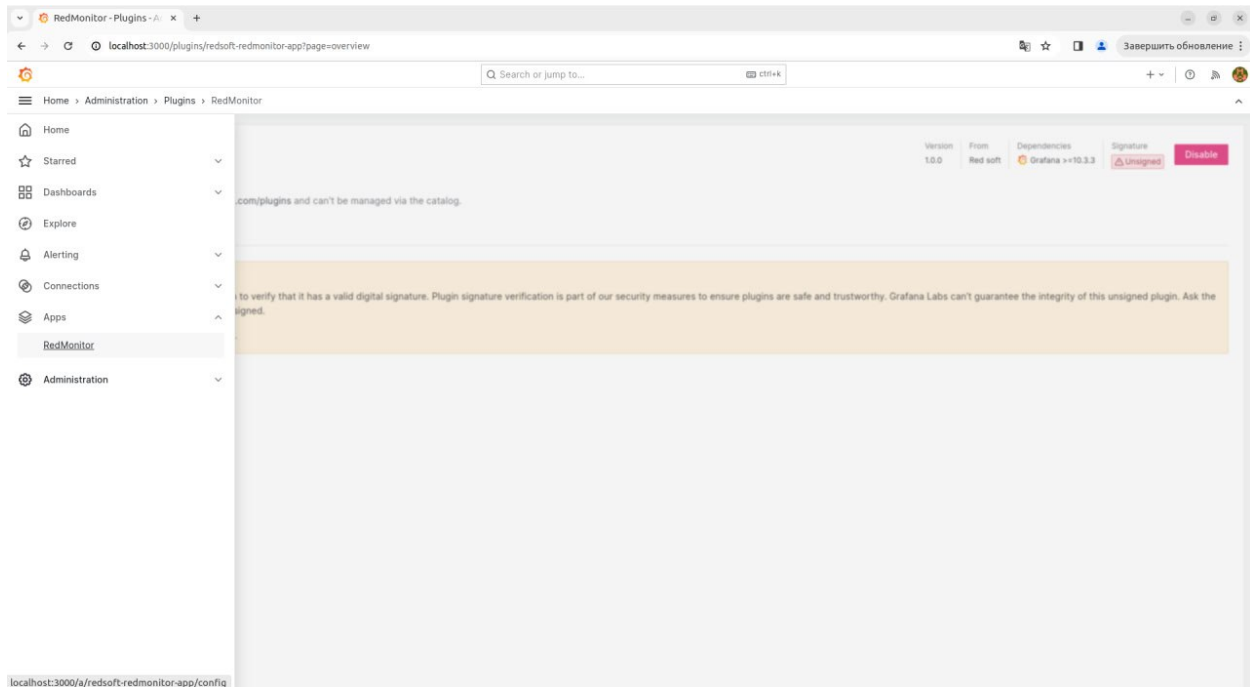


Рисунок 3.2 — Плагин Ред Монитор

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

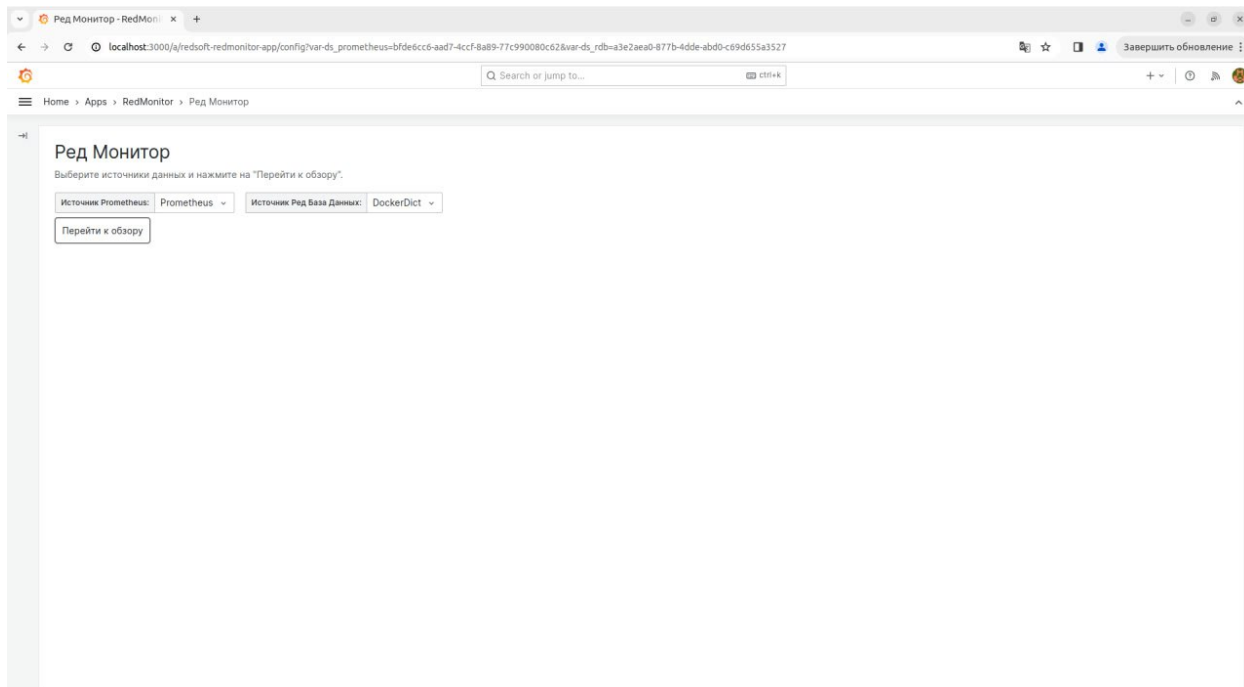


Рисунок 3.3 — Настройка плагина Ред Монитор

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

## 3.3 Обзорная страница

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

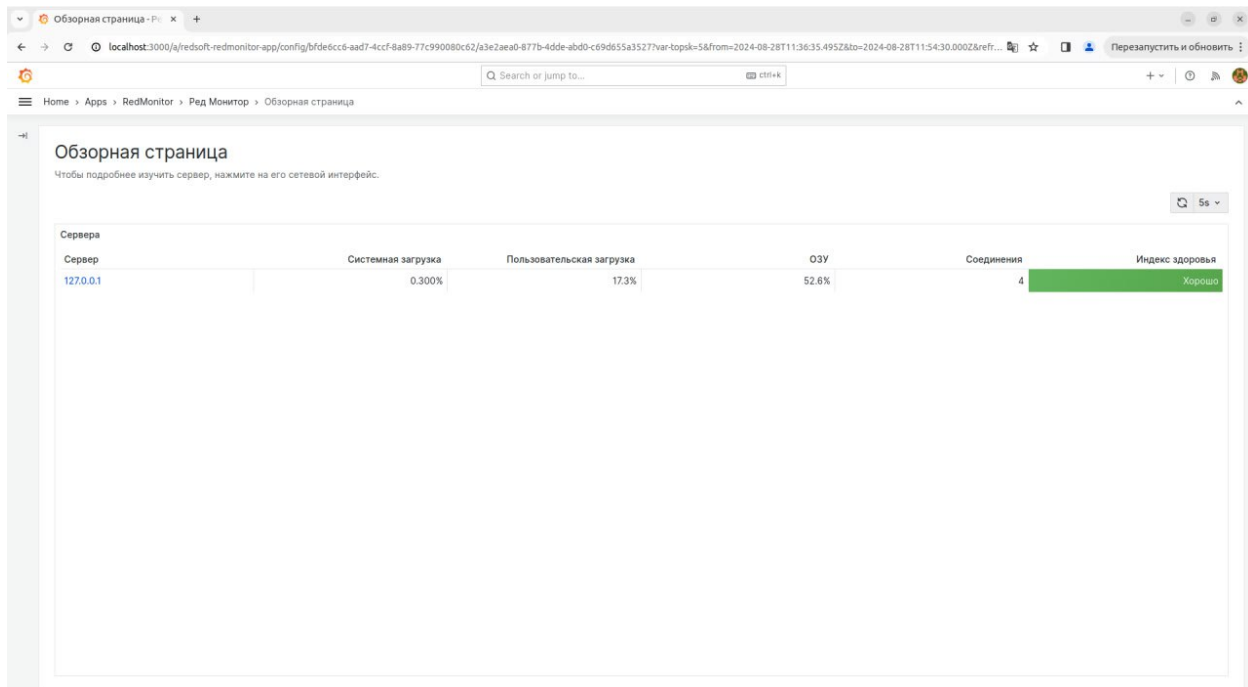


Рисунок 3.4 — Обзорная страница

## 3.4 Информация о выбранном сервере

На этой странице находится подробная информация о выбранном сервере.

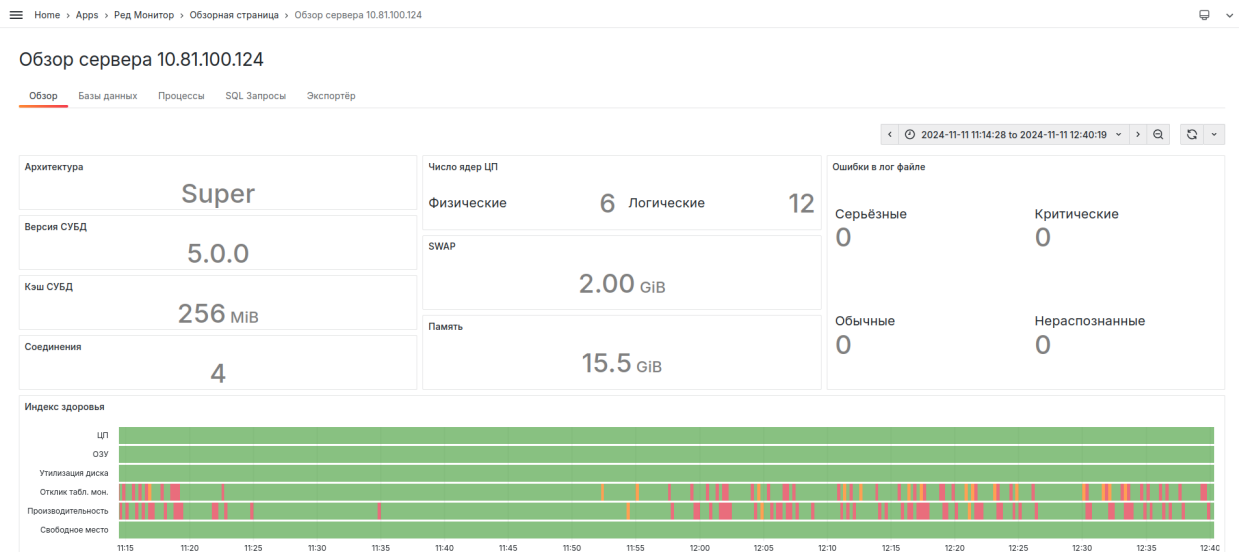


Рисунок 3.5 — Обзор выбранного сервера

- **Кэш** – размер, выделенный в оперативной памяти для кэша;
- **Архитектура** – архитектура СУБД;
- **Соединения** - количество соединений, установленных в данный момент;

- Ошибки в лог файле - количество ошибок, записанных в лог файл.

График Количество соединений показывает количество активных соединений в зависимости от времени.

График Ошибки в лог файле по группам отображает количество ошибок разной критичности в зависимости от времени.

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

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

График Количество прерываний показывает сколько прерываний процессора было совершено за последний интервал сбора метрик.

График Чтение/запись показывает количество выполненных операций чтения с дисков и записи на них в зависимости от времени.

График Переключения контекста показывает сколько переключений контекста процессора было совершено за последний интервал сбора метрик.

График SWAP показывает использование SWAP в зависимости от времени.

График Тактовая частота процессора показывает текущую, максимальную и минимальную частоту процессора в зависимости от времени.

График Swapping показывает объём информации, который был загружен в SWAP и выгружен из него за время, прошедшее с предыдущего измерения.

### 3.4.1 Индекс здоровья

Индекс здоровья отображает общую информацию о состоянии сервера. Зелёные зоны говорят о нормальной работе сервера, жёлтые, что есть незначительное отклонение от нормы, а красные показывают возможное наличие проблемы.

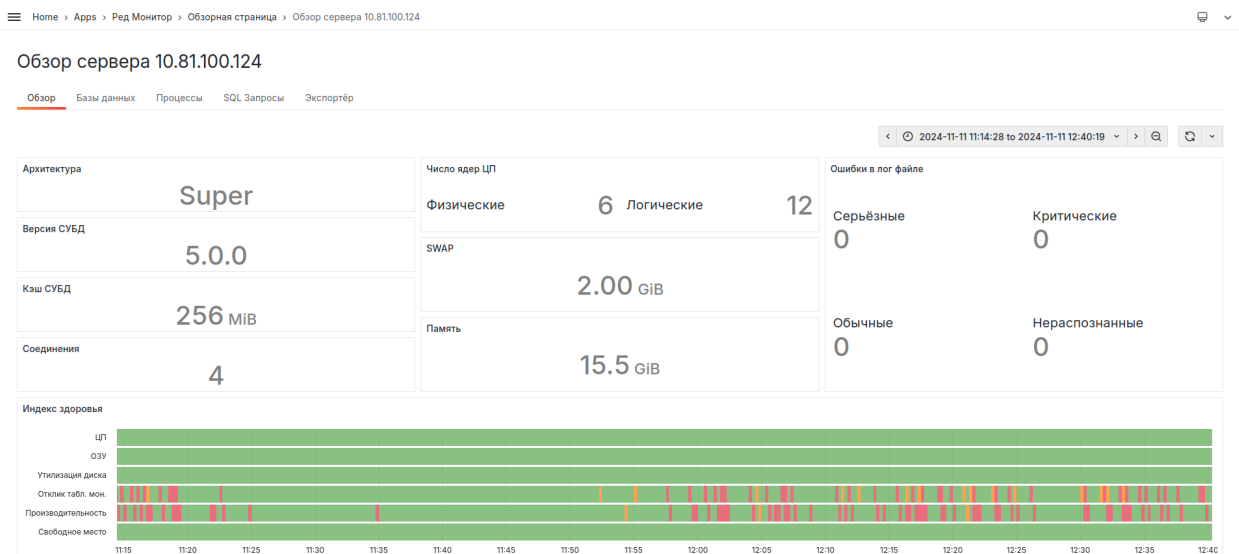


Рисунок 3.6 — Индекс здоровья

В строке ЦП показана системная загрузка центрального процессора. Если она выше 20%, то область будет закрашена красным.

ОЗУ отображает использование оперативной памяти, если оно больше 90%, то область будет за-



крашена красным.

Утилизация диска считается, как процент времени использования диска для чтения/записи с момента прошлого обращения к серверу для сбора метрик. Если диск использовался для чтения/записи более 75% времени с момента прошлого обращения к серверу, то область будет закрашена красным.

Отклик таблиц мониторинга считается, как отношение времени отклика в текущий момент к среднему времени отклика за выбранный отрезок времени:

- $0 < \text{отклик таблиц мониторинга} \leq 1$  - зелёная зона;
- $1 < \text{отклик таблиц мониторинга} \leq 1.5$  - жёлтая зона;
- $1.5 < \text{отклик таблиц мониторинга}$  - красная зона.

Производительность считается, как отношение скорости выполнения запроса в текущий момент к средней скорости выполнения за выбранный отрезок времени:

- $0 < \text{производительность} \leq 1$  - зелёная зона;
- $1 < \text{производительность} \leq 1.5$  - жёлтая зона;
- $1.5 < \text{производительность}$  - красная зона.

Свободное место определяет наличие свободного места на дисках, если на каком-либо диске его осталось менее 10%, то область будет закрашена красным.

## 3.4.2 Базы данных

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

Обзор сервера 10.81.100.124

Обзор Базы данных Процессы SQL Запросы Экспортёр

2024-11-11 11:14:28 to 2024-11-11 12:40:19

Базы данных	Соединения	Занимает на диске	DMT	Выполнено запросов
employee_db	4	0.000306%	343	1717

Рисунок 3.7 — Обзор баз данных

### 3.4.3 Информация о выбранной базе данных

На этой странице находится подробная информация о базе данных.

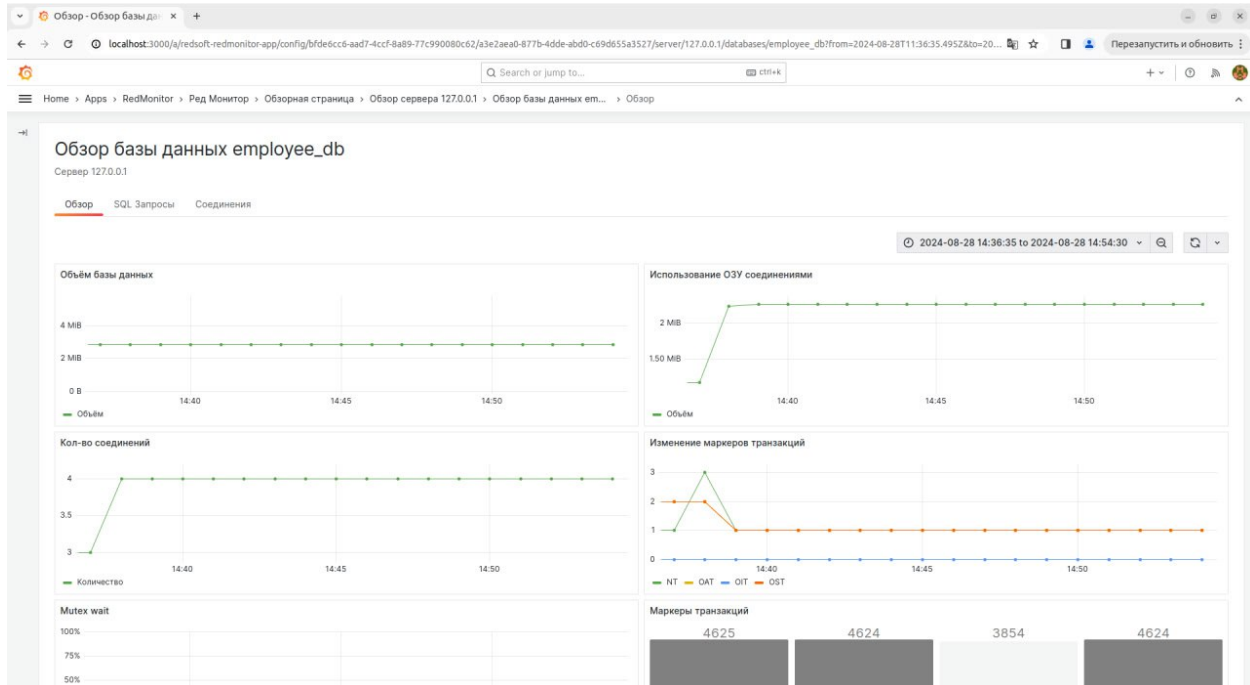


Рисунок 3.8 — Обзор выбранной базы данных

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 **Маркеры транзакций** отражает значения маркеров транзакций.

График **Mutex wait** показывает процент попыток, которые были заблокированы, когда владелец старался обратиться к таблице блокировок.

График **Выполнение запросов** показывает количество выполненных запросов в зависимости от времени.

График **Количество блокировок** показывает количество блокировок в зависимости от времени.

## SQL запросы

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

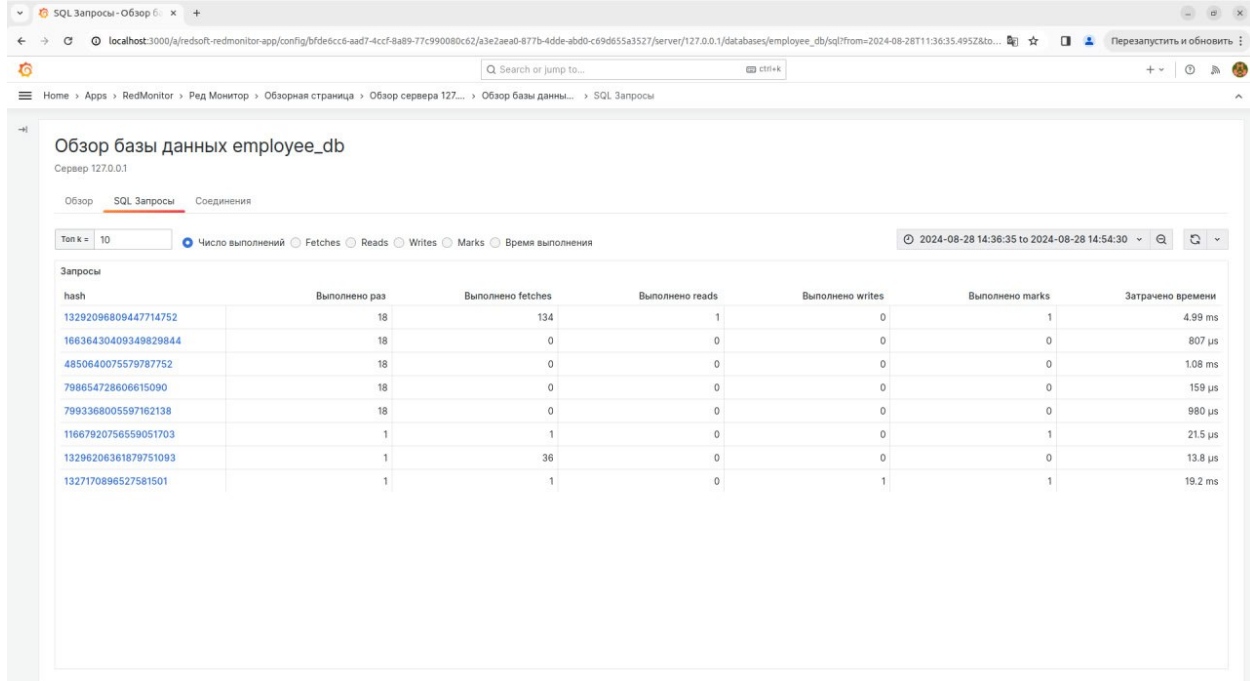


Рисунок 3.9 — SQL запросы выбранной базы данных

## Обзор запроса

На данной странице находится подробная информация о выбранном запросе.

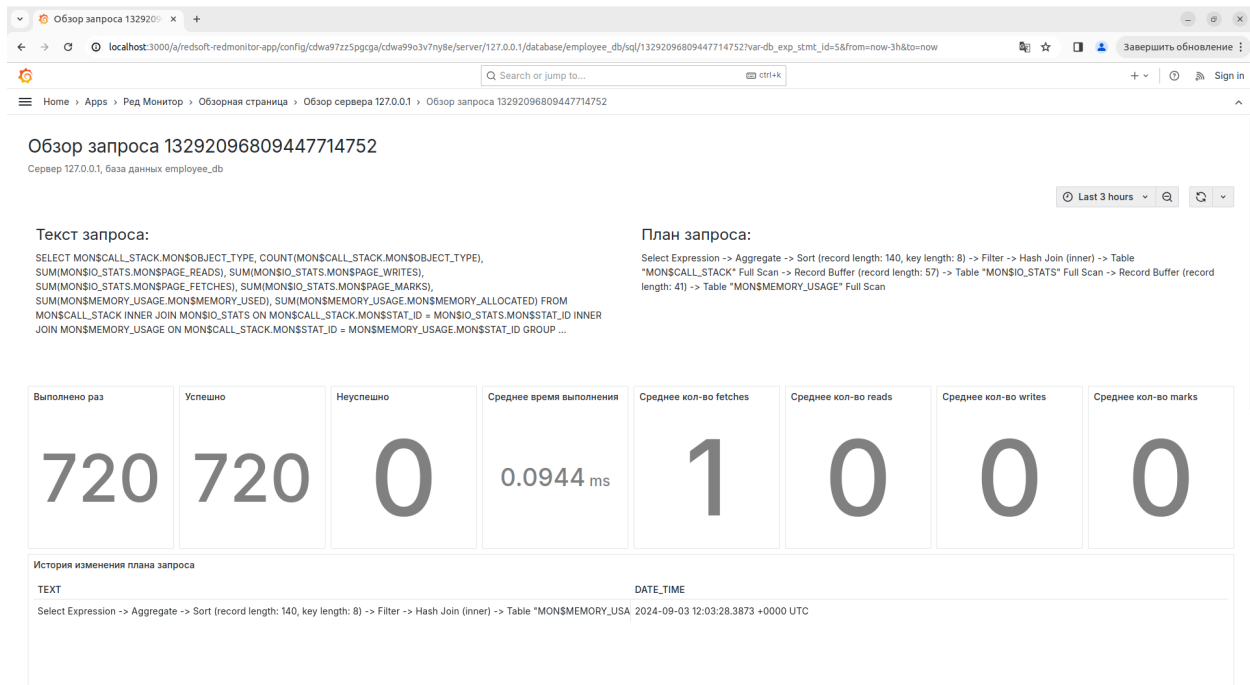


Рисунок 3.10 — Обзор запроса

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

## 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.

The screenshot shows a web interface for monitoring database connections. The page title is 'Обзор базы данных employee\_db' and the server is 'Сервер 127.0.0.1'. The 'Соединения' (Connections) tab is active. A table displays the top 10 connections, with columns for connection ID, user name, client address, client process, server process ID, connection duration, and various operation counts (Fetches, Reads, Writes, Marks).

ID Соединения	Имя пользователя	Адрес клиента	Процесс клиента	ID процесса сервера	Длительность соедине	Операции fetches	Операции reads	Операции writes	Операции marks
900	SYSDBA	127.0.0.1/52260	/home/eyedm/.pyenv/vers	13935	2.25 mins	8	0	2	4
901	Cache Writer			13935	2.25 mins	0	0	0	0
902	Garbage Collector			13935	2.25 mins	0	0	0	0

Рисунок 3.11 — Соединения с базой данных

## Обзор соединения

На данной странице находится подробная информация о выбранном соединении.

The screenshot shows a detailed view of connection 900. The page title is 'Обзор соединения 900' and the server is 'Сервер 127.0.0.1, база данных employee\_db'. The 'Длительность соединения' (Connection Duration) is 4.25 mins. Below this, a table lists connection parameters. At the bottom, four large numbers represent the number of operations performed: Fetches (2450), Marks (35), Reads (99), and Writes (17).

Имя пользователя	Роль пользователя	Системное подключение	ID процесса сервера	ID процесса клиента	Адрес клиента	Процесс клиента
SYSDBA	NONE	false	13935	20463	127.0.0.1/52260	/home/eyedm/.pyenv/versions/3.8.18/l

Работа со страницами (всего)

Fetches	Marks	Reads	Writes
2450	35	99	17

Рисунок 3.12 — Обзор соединения

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

### 3.4.4 Процессы

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

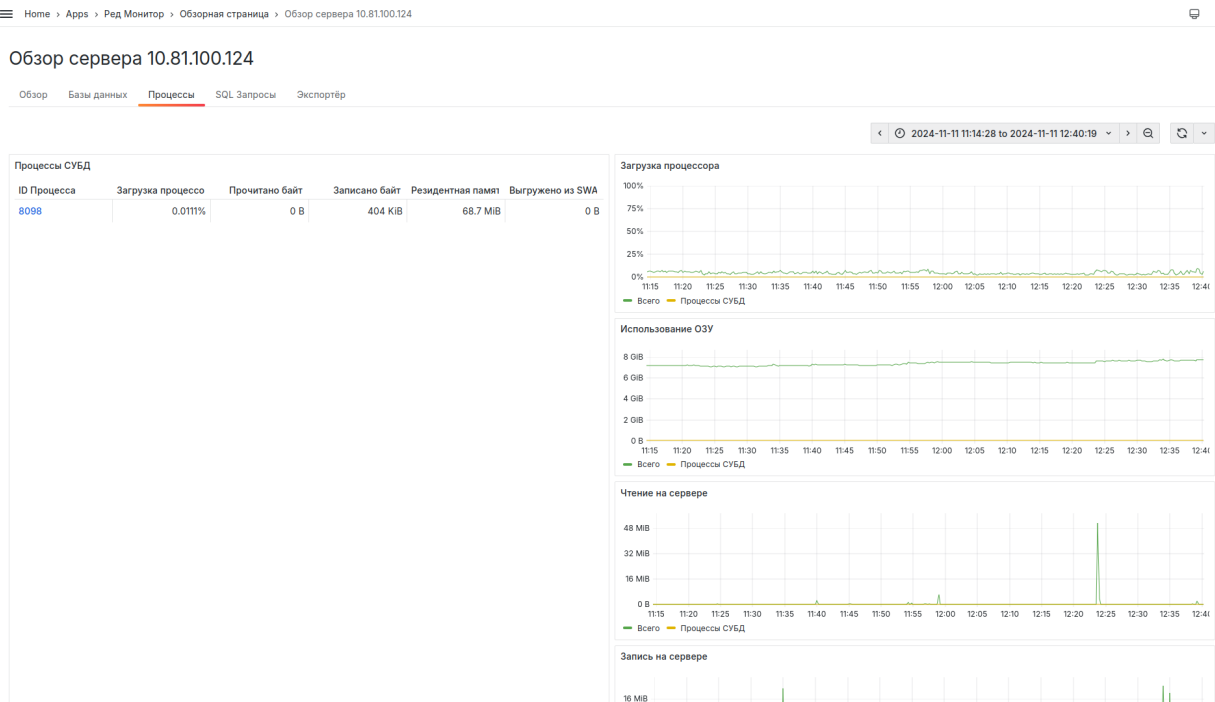


Рисунок 3.13 — Процессы СУБД

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

### Обзор процесса

На данной странице находится подробная информация о выбранном процессе.

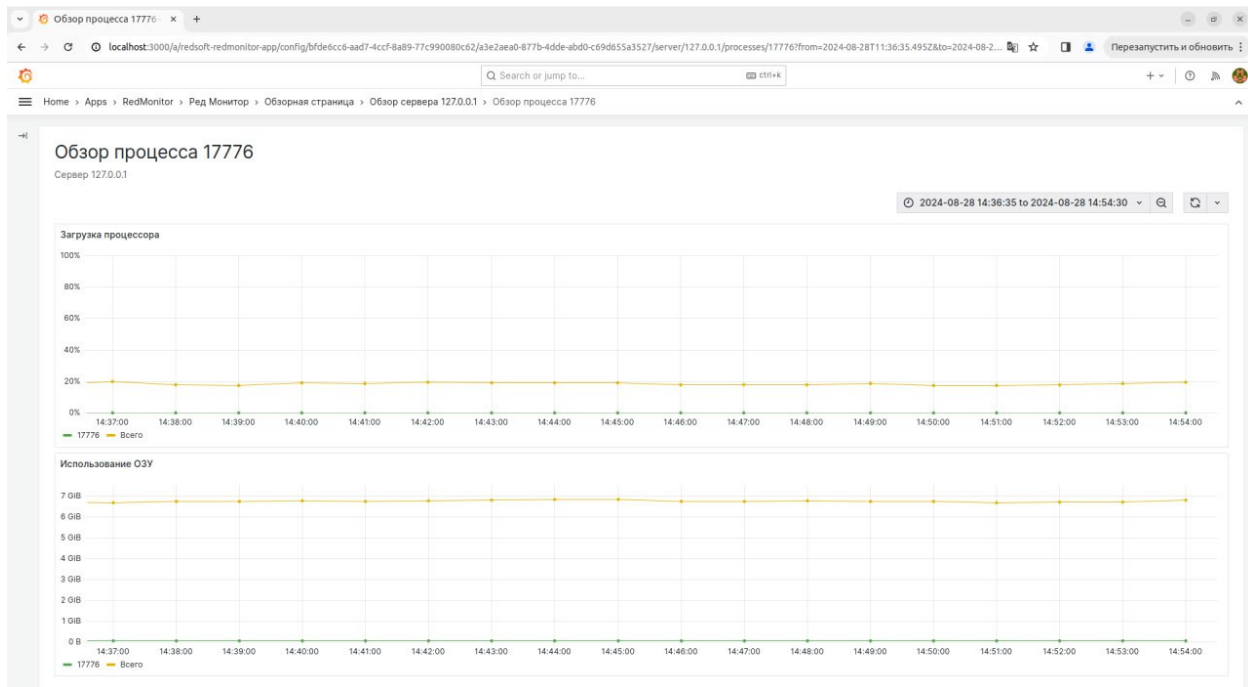


Рисунок 3.14 — Обзор процесса

### 3.4.5 SQL-запросы

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

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

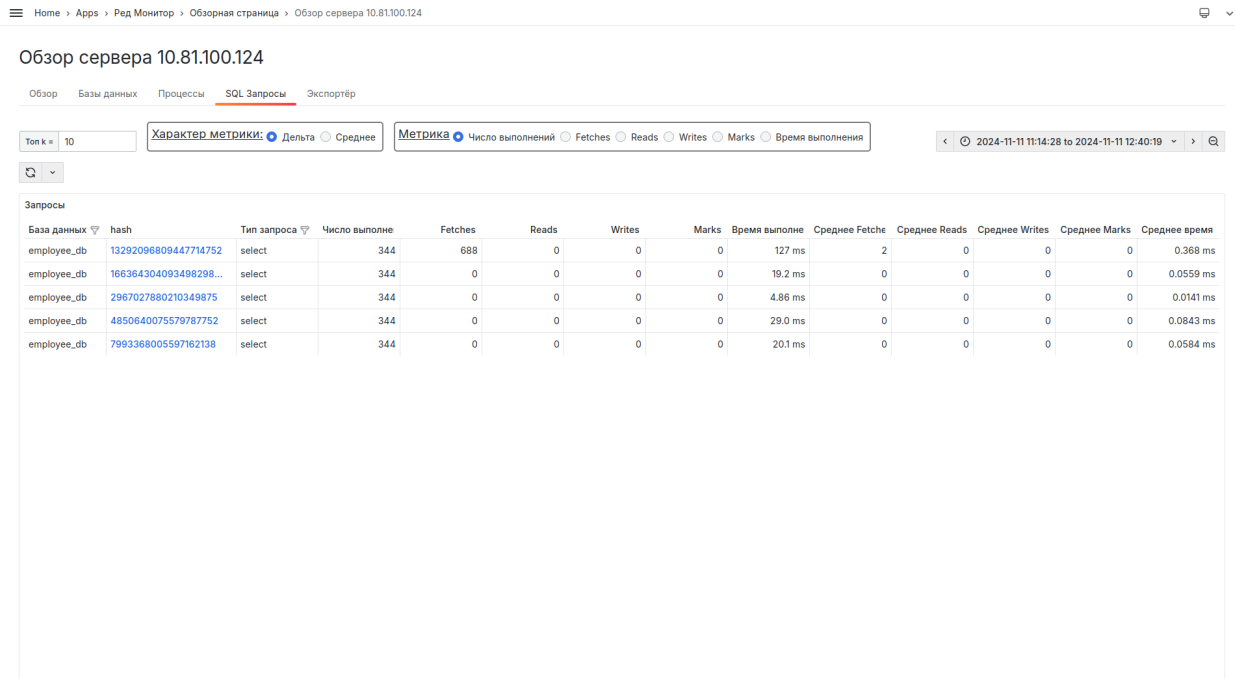


Рисунок 3.15 — Запросы

### 3.4.6 Экспортёр

На странице находится информация о сборе метрик экспортёром.

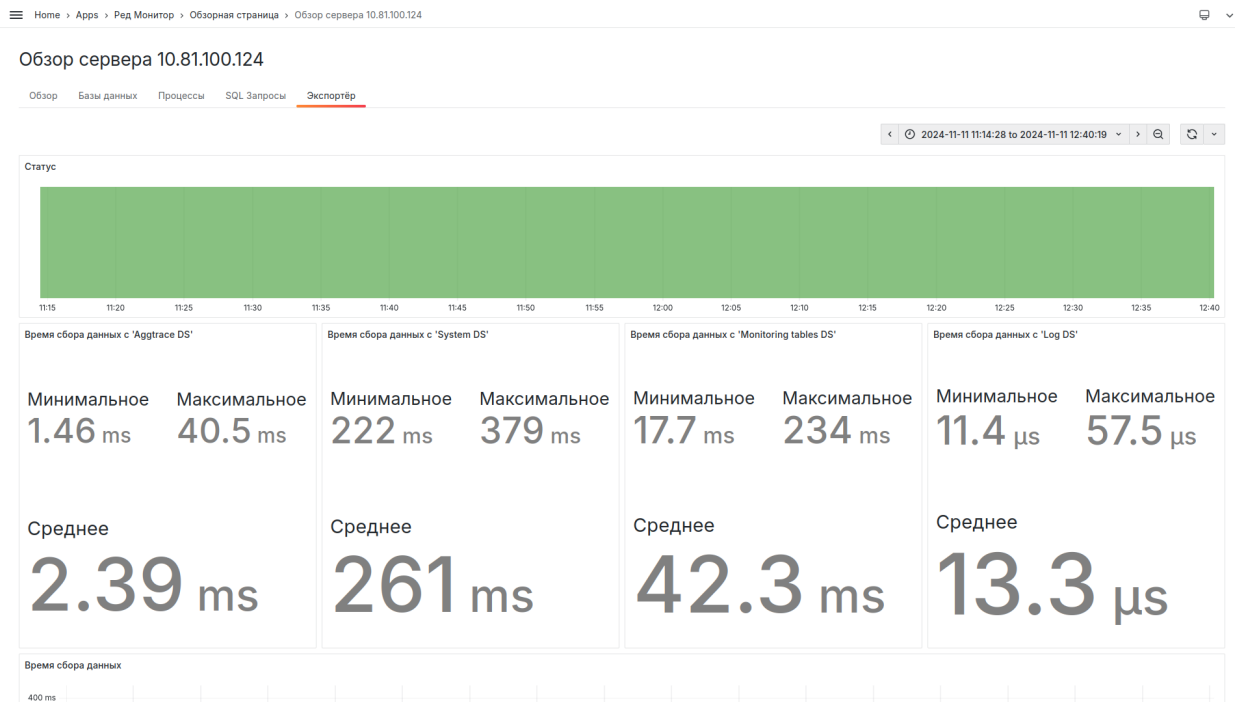


Рисунок 3.16 — Экспортёр



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

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

## 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	<p>database - база данных.            is_active - активна ли транзакция.            isolation_mode - уровень изоляции:</p> <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> <p>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:</p> <ul style="list-style-type: none"> <li>• used - объём используемой памяти.</li> <li>• allocated - количество выделенной памяти.</li> </ul>	<p>Объём памяти (в байтах), используемой транзакциями, выполняющимися в момент сбора информации.</p>
rdb_transactions_count	<p>database - база данных.            is_active - активна ли транзакция.            isolation_mode - уровень изоляции:</p> <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> <p>read_only - выполняется ли транзакция в режиме read_only.            auto_commit - используется ли режим автоматической фиксации.            auto_undo - используется ли автоматическая отмена транзакции.</p>	<p>Количество транзакций, выполняющихся в момент сбора информации.</p>

(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. Read-only - значение 1, read-write значение 0.
rdb_database_forced_writes	database - база данных.	Указывает, установлен для базы режим синхронного вывода (forced writes, значение 1) или режим асинхронного вывода (значение 0).
rdb_database_backup_state	database - база данных.	Указывает состояние бэкапа базы данных: <ul style="list-style-type: none"> <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 (shutdown) базы данных: <ul style="list-style-type: none"> <li>• 0 — база данных активна (online);</li> <li>• 1 — остановлена для нескольких пользователей (multi-user shutdown);</li> <li>• 2 — остановлена для одного пользователя (single-user shutdown);</li> <li>• 3 — полностью остановка (full shutdown).</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 экспортёра 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_aggtrace_stmt_pages	database - база данных; hash - хэш запроса; q_operation - тип запроса: <ul style="list-style-type: none"> <li>• select</li> <li>• insert</li> <li>• update</li> <li>• create</li> <li>• delete</li> <li>• drop</li> </ul> operation - тип операции: <ul style="list-style-type: none"> <li>• reads;</li> <li>• fetches;</li> <li>• marks;</li> <li>• writes.</li> </ul> marker - количество операций: <ul style="list-style-type: none"> <li>• min;</li> <li>• max;</li> <li>• avg.</li> </ul>	Количество операций, выполненных запросом.
rdb_aggtrace_stmt_times	database - база данных; hash - хэш запроса; q_operation - тип запроса: <ul style="list-style-type: none"> <li>• select</li> <li>• insert</li> <li>• update</li> <li>• create</li> <li>• delete</li> <li>• drop</li> </ul> marker - затраченное время: <ul style="list-style-type: none"> <li>• min;</li> <li>• max;</li> <li>• avg.</li> </ul>	Время, затраченное на выполнение запросов.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_aggtrace_stmt_count	database - база данных; hash - хэш запроса; q_operation - тип запроса: <ul style="list-style-type: none"> <li>• select</li> <li>• insert</li> <li>• update</li> <li>• create</li> <li>• delete</li> <li>• drop</li> </ul> marker - результат выполнения: <ul style="list-style-type: none"> <li>• failed</li> <li>• succeed</li> <li>• total</li> </ul>	Количество выполненных запросов выбранного типа.
rdb_aggtrace_sort_mu_total	database - база данных; hash - хэш запроса; q_operation - тип запроса: <ul style="list-style-type: none"> <li>• select</li> <li>• insert</li> <li>• update</li> <li>• create</li> <li>• delete</li> <li>• drop</li> </ul> marker - результат выполнения: <ul style="list-style-type: none"> <li>• failed</li> <li>• succeed</li> <li>• total</li> </ul>	Объём памяти (в байтах), выделенной под сортировку.
rdb_aggtrace_sort_mu_disk	database - база данных; hash - хэш запроса; q_operation - тип запроса: <ul style="list-style-type: none"> <li>• select</li> <li>• insert</li> <li>• update</li> <li>• create</li> <li>• delete</li> <li>• drop</li> </ul> marker - результат выполнения: <ul style="list-style-type: none"> <li>• failed</li> <li>• succeed</li> <li>• total</li> </ul>	Объём памяти (в байтах), выделенной под сортировку на диске.

(разрыв таблицы)



(разрыв таблицы)

Название метрики	Метки	Описание метрики
<code>rdb_aggtrace_sort_mu_cache</code>	<p><code>database</code> - база данных;</p> <p><code>hash</code> - хэш запроса;</p> <p><code>q_operation</code> - тип запроса:</p> <ul style="list-style-type: none"> <li>• <code>select</code></li> <li>• <code>insert</code></li> <li>• <code>update</code></li> <li>• <code>create</code></li> <li>• <code>delete</code></li> <li>• <code>drop</code></li> </ul> <p><code>marker</code> - результат выполнения:</p> <ul style="list-style-type: none"> <li>• <code>failed</code></li> <li>• <code>succeed</code></li> <li>• <code>total</code></li> </ul>	Объём памяти (в байтах), выделенной под сортировку в кэше.

## A.3 Метрики утилиты `rdb_lock_print`

Таблица A.3 — Метрики утилиты `rdb_lock_print`

Название метрики	Метки	Описание метрики
<code>rdb_lock_table_memory_usage</code>	<p><code>database</code> - база данных;</p> <p><code>usage</code>:</p> <ul style="list-style-type: none"> <li>• <code>length</code> - общий объем памяти, выделенный таблице блокировок (в байтах);</li> <li>• <code>used</code> - наибольшая величина смещения в таблице блокировок, которая используется в настоящий момент.</li> </ul>	Объем памяти, выделенный таблице блокировок (в байтах).
<code>rdb_lock_table_requests_count</code>	<p><code>database</code> - база данных;</p> <p><code>kind</code> - тип запроса:</p> <ul style="list-style-type: none"> <li>• <code>enqs</code> - число запросов, полученных на блокировку (не включает запросы, которые пришли и ушли);</li> <li>• <code>converts</code> - запросы на повышение уровня блокировки;</li> <li>• <code>rejects</code> - запросы, которые не могут быть удовлетворены;</li> <li>• <code>blocks</code> - Запросы, которые не могут быть удовлетворены немедленно.</li> </ul>	Количество запросов выбранного типа.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_lock_table_deadlocks_count	database - база данных; counter: <ul style="list-style-type: none"> <li>• scans - количество проверок на взаимные блокировки;</li> <li>• found - количество найденных взаимных блокировок.</li> </ul>	Подсчёт взаимных блокировок.
rdb_lock_table_scan_interval	database - база данных;	Время (в секундах), которое ожидает менеджер блокировок до того как запустить к поиску взаимных блокировок.
rdb_lock_table_acquires	database - база данных; counter: <ul style="list-style-type: none"> <li>• requires - сколько раз владелец запрашивает исключительное управление таблицей блокировок, чтобы выполнить изменения;</li> <li>• blocks - сколько раз владелец находился в состоянии ожидания при запросе исключительного управления таблицей блокировок.</li> </ul>	Запросы на исключительное управление таблицей блокировок.
rdb_lock_table_mutex_wait	database - база данных;	Процент попыток, которые были заблокированы, когда владелец старался обратиться к таблице блокировок.
rdb_lock_table_hash_slots_count	database - база данных;	Число слотов кэширования блокировок.
rdb_lock_table_hash_slots_lengths	database - база данных; length: <ul style="list-style-type: none"> <li>• min;</li> <li>• max;</li> <li>• avg.</li> </ul>	Длина цепочки кэширования.
rdb_lock_table_owners_count	database - база данных; kind - тип владельца: <ul style="list-style-type: none"> <li>• process</li> <li>• database</li> <li>• connection</li> <li>• transaction</li> <li>• dummy_process</li> </ul>	Количество владельцев, соединенных с таблицей блокировок.
rdb_lock_table_owners_requests_avg	database - база данных;	Среднее количество групп запросов, которые были освобождены и не использованы повторно.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_lock_table_locks_count	database - база данных; series - тип ресурса; is_pended - ожидается ли кем-то.	Количество блокировок указанного типа.
rdb_lock_table_locks_memory	database - база данных; series - тип ресурса; is_pended - ожидается ли кем-то; usage - использование памяти: <ul style="list-style-type: none"> <li>• used - объём используемой памяти.</li> <li>• allocated - количество выделенной памяти.</li> </ul>	Объём выделенной памяти в байтах для указанной блокировки.
rdb_lock_table_pending_req_count_avg	database - база данных; series - тип ресурса;	Среднее количество ожидающих владельцев для указанного типа ресурса.

## А.4 Метрики репликации

Таблица А.4 — Метрики репликации

Название метрики	Метки	Описание метрики
rdb_master_segments_count	database - база данных; state - состояние сегмента: <ul style="list-style-type: none"> <li>• full</li> <li>• arch</li> <li>• free</li> <li>• used</li> </ul>	Количество сегментов указанного состояния.
rdb_master_current_sequence	database - база данных.	Номер текущего сегмента
rdb_master_segments_size_in_use	database - база данных.	Общий размер журнала репликации в байтах.
rdb_slave_current_segment	database - база данных.	Номер текущего сегмента
rdb_slave_oldest_segment	database - база данных.	Сегмент, который начал самую раннюю транзакцию, которая не была завершена во время обработки последней последовательности.
rdb_slave_active_transaction	database - база данных.	Количество активных транзакций.
rdb_slave_segments_in_queue	database - база данных.	Количество сегментов в очереди.
rdb_master_pings	from to	Задержка ответа между серверами в секундах.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_repl_log_size	database - база данных.	Общий размер журнала репликации в байтах.

## А.5 Метрики операционной системы

Таблица А.5 — Метрики операционной системы

Название метрики	Метки	Описание метрики
rdb_disks_io_counters	database - база данных. 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	database - база данных. 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_times	database - база данных. disk - диск. operation - тип операции: <ul style="list-style-type: none"> <li>• reads</li> <li>• writes</li> </ul>	Время (в миллисекундах), затраченное на выполнение операций чтения/записи на диск.
rdb_disks_io_wtime	database - база данных. disk - диск.	Взвешенное время, потраченное на операции ввода-вывода.
rdb_disks_io_time	database - база данных. disk - диск.	Время, потраченное на фактический ввод-вывод.
rdb_mount_point_usage	database - база данных. point - точка монтирования. usage: <ul style="list-style-type: none"> <li>• used</li> <li>• free</li> <li>• total</li> </ul>	Использование точки мониторинга.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_other_procs_memory_usage	usage - тип операции: <ul style="list-style-type: none"> <li>• resident - объём резидентной памяти, используемой процессами, не относящимся к СУБД.</li> <li>• virtual - объём виртуальной памяти, выделенной под процессы, не относящиеся к СУБД.</li> <li>• swapped_out - количество записанных на диск байт при нехватке ОЗУ.</li> </ul>	Количество прочитанной и записанной информации в байтах процессами, которые не относятся к СУБД.
rdb_other_procs_io_bytes	operation - тип операции: <ul style="list-style-type: none"> <li>• reads</li> <li>• writes</li> </ul>	Количество прочитанной и записанной информации в байтах процессами, которые не относятся к СУБД.
rdb_opened_temp_files_size	temp_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_opened_temp_files_count	temp_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_usage	pid	Загрузка процессора процессами СУБД.
rdb_procs_memory_usage	pid usage: <ul style="list-style-type: none"> <li>• resident</li> <li>• virtual</li> <li>• swapped_out</li> </ul>	Объём резидентной, виртуальной и выгруженной в swap памяти в байтах.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_procs_io_bytes	pid 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_count	is_rdb - относится ли процесс к СУБД.	Количество процессов.
rdb_system_memory	field: <ul style="list-style-type: none"> <li>• used</li> <li>• availabel</li> <li>• total</li> </ul>	Использование оперативной памяти.
rdb_system_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_system_swapped	direction: <ul style="list-style-type: none"> <li>• in</li> <li>• out</li> </ul>	Объём данных (в байтах), затронутый swap.
rdb_system_cpu_load	group: <ul style="list-style-type: none"> <li>• user</li> <li>• system</li> </ul>	Загрузка процессора.
rdb_system_cpu_freq	marker: <ul style="list-style-type: none"> <li>• max</li> <li>• min</li> <li>• current</li> </ul>	Частота работы процессора.
rdb_system_cpu_ctx		Число изменений контекста процессора.
rdb_system_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_tmp_disks_io_bytes	disk operation: <ul style="list-style-type: none"> <li>• reads</li> <li>• writes</li> </ul>	Показывает количество байт прочитанных с диска и записанных на него.

(разрыв таблицы)

(разрыв таблицы)

Название метрики	Метки	Описание метрики
rdb_tmp_disks_io_counters	disk operation: <ul style="list-style-type: none"> <li>• reads</li> <li>• writes</li> </ul>	Показывает количество операций чтения с диска и записи на него.
rdb_tmp_disks_io_times	disk operation: <ul style="list-style-type: none"> <li>• reads</li> <li>• writes</li> </ul>	Показывает количество времени в миллисекундах, потраченного на чтение с диска и запись на него.
rdb_tmp_dir_size	name point	Показывает объём временных файлов СУБД.

## А.6 Другие метрики

Таблица А.6 — 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.