**RT.WIDESTORE**

**ИНСТРУКЦИЯ АДМИНИСТРАТОРА**

2025

# 1. ВВЕДЕНИЕ

## 1.1 Наименование и условное обозначение

Наименование системы: RT.WideStore.

## [¶](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#h-12-%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F)1.2 Область применения

RT.WideStore – столбцовая система управления базами данных, предназначенная для онлайн обработки аналитических запросов (OLAP).

### 1.3 Термины, определения и сокращения

В настоящем документе использованы и определены следующие термины и сокращения:

| ***Термин/ Сокращение*** | ***Определение*** |
| --- | --- |
| Docker | Платформа, которая упрощает процесс сборки, запуска, управления и распространения приложений с помощью виртуализации операционной системы, на которой он установлен. |
| OLAP | (англ. online analytical processing, интерактивная аналитическая обработка) – технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу. |
| RAM | Оперативная память. |
| RBAC | (англ. Role Based Access Contro) Управление доступом на основе ролей, развитие политики избирательного управления доступом, при этом права доступа субъектов системы на объекты группируются с учётом специфики их применения, образуя роли.  |
| SQL | (англ. structured query language, язык структурированных запросов) –декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных. |
| Воркфлоу | Поток работ (англ. workflow) – графическое представление потока задач в процессе и связанных с ним подпроцессах, включая специфические работы, информационные зависимости и последовательность решений и работ. |
| ОС | Операционная система. |
| Система | Система «RT.WideStore». |
| СУБД | Система управления базами данных. |
| Шард | Сервер входящий в кластер. |

## 1.4 Эксплуатационные требования

### 1.4.1 Требования к Процессору

RT.WideStore реализует параллельную обработку данных и использует все доступные аппаратные ресурсы. При выборе процессора учитывайте, что RT.WideStore работает более эффективно в конфигурациях с большим количеством ядер, но с более низкой тактовой частотой, чем в конфигурациях с меньшим количеством ядер и более высокой тактовой частотой. Например, 16 ядер с 2600 MHz предпочтительнее, чем 8 ядер с 3600 MHz.

Рекомендуется использовать технологии Turbo Boost и hyper-threading. Их использование существенно улучшает производительность при типичной нагрузке.

### 1.4.2 Требования к оперативной памяти

Мы рекомендуем использовать как минимум 4 ГБ оперативной памяти, чтобы иметь возможность выполнять нетривиальные запросы. Сервер RT.WideStore может работать с гораздо меньшим объёмом RAM, память требуется для обработки запросов.

Необходимый объём RAM зависит от:

сложности запросов,

объёма данных, обрабатываемых в запросах.

Для расчета объёма RAM необходимо оценить размер промежуточных данных для операций GROUP BY, DISTINCT, JOIN, а также других операций, которыми вы пользуетесь.

RT.WideStore может использовать внешнюю память для промежуточных данных. Подробнее смотрите в разделе GROUP BY во внешней памяти.

### 1.4.3 Файл подкачки

Отключайте файл подкачки в продуктовых средах.

### 1.4.4 Требования к подсистеме хранения

Для установки RT.WideStore необходимо 2ГБ свободного места на диске.

Объём дискового пространства, необходимый для хранения ваших данных, необходимо рассчитывать отдельно. Расчёт должен включать:

Приблизительную оценку объёма данных.

Можно взять образец данных и получить из него средний размер строки. Затем умножьте полученное значение на количество строк, которое вы планируете хранить.

Оценку коэффициента сжатия данных.

Чтобы оценить коэффициент сжатия данных, загрузите некоторую выборку данных в RT.WideStore и сравните действительный размер данных с размером сохранённой таблицы. Например, данные типа clickstream обычно сжимаются в 6-10 раз.

Для оценки объёма хранилища, примените коэффициент сжатия к размеру данных. Если вы планируете хранить данные в нескольких репликах, то необходимо полученный объём умножить на количество реплик.

### 1.4.5 Требование к подсистеме передачи данных

По возможности, используйте сети 10G и более высокого класса.

Пропускная способность сети критически важна для обработки распределенных запросов с большим количеством промежуточных данных. Также, скорость сети влияет на задержки в процессах репликации.

### 1.4.6 Требование к операционной системе

RT.WideStore разработан для семейства операционных систем Linux. Рекомендуемый дистрибутив Linux – Ubuntu. В системе должен быть установлен пакет **tzdata**.

RT.WideStore может работать и в других семействах операционных систем, например, в FreeBSD или Mac OS X.

## 1.5 Требования к квалификации администратора

Администратор Системы должен иметь навыки работы со следующими продуктами:

Опыт администрирования ОС, на которой устанавливается Система.

Навыки развертывания и администрирование СУБД.

Опыт управление безопасностью, задание групповых политик, создание пользователей, управление удаленным доступом и т.п.

Знание языка запросов (SQL).

Навыки диагностики и мониторинга сетевого оборудования.

Умение работать с большим объёмом информации.

Владение скриптингом bash, PowerShell для оптимизации и автоматизации повседневных рутинных функций системного администрирования.

Опыт конфигурирования Apache Kafka, Zookeeper.

Опыт масштабирования СУБД.

Навыки технической поддержки пользователей.

Умение выполнять ремонт и обслуживание компьютерного и периферийного оборудования.

# 2. ЭКСПЛУАТАЦИЯ RT.WIDESTORE

## 2.1 Мониторинг RT.WideStore

Вы можете отслеживать:

использование аппаратных ресурсов,

метрики сервера RT.WideStore.

### 2.1.1 Использование аппаратных ресурсов

RT.WideStore не отслеживает состояние аппаратных ресурсов самостоятельно.

Рекомендуем контролировать:

Загрузку и температуру процессоров.

Например, можно использовать команды операционной системы: ***dmesg***– для вывода сообщений драйверов устройств и ***turbostat***– для вывода данных о топологии процессора, частоты, статистики состояния питания в режиме ожидания, температуре и мощности на X86 процессорах.

Использование системы хранения, оперативной памяти и сети.

### 2.1.2 Метрики сервера RT.WideStore

Сервер RT.WideStore имеет встроенные инструменты мониторинга.

Для отслеживания событий на сервере используйте логи. Параметры логгирования настраиваются секции ***logger***конфигурационного файла.

RT.WideStore собирает:

различные метрики того, как сервер использует вычислительные ресурсы,

общую статистику обработки запросов.

Метрики находятся в таблицах:

[***system.metrics***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2#:~:text=24.%20system.metrics) – содержит метрики, которые могут быть рассчитаны мгновенно или имеют текущее значение. Например, число одновременно обрабатываемых запросов или текущее значение задержки реплики. Эта таблица всегда актуальна.).

[***system.events***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2#:~:text=17.%20system.events) – содержит информацию о количестве событий, произошедших в системе. Например, в таблице можно найти, сколько запросов SELECT обработано с момента запуска сервера RT.WideStore.

[***system.asynchronous\_metrics***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2#:~:text=2.%20system.asynchronous_metrics) – содержит метрики, которые периодически вычисляются в фоновом режиме. Например, объём используемой оперативной памяти.

Можно настроить экспорт метрик из RT.WideStore в [Graphite](https://github.com/graphite-project), для отображения в виде графика. Смотрите секцию graphite конфигурационного файла RT.WideStore. Перед настройкой экспорта метрик необходимо настроить Graphite, как указано в [официальном руководстве](https://graphite.readthedocs.io/en/latest/install.html).

Можно настроить экспорт метрик из RT.WideStore в [Prometheus](https://prometheus.io/). Смотрите секцию prometheus конфигурационного файла RT.WideStore. Перед настройкой экспорта метрик необходимо настроить Prometheus, как указано в [официальном руководстве](https://prometheus.io/docs/prometheus/latest/installation/).

Также, можно отслеживать доступность сервера через HTTP API. Отправьте HTTP GET к ресурсу **/ping**. Если сервер доступен, он отвечает 200 OK.

Для мониторинга серверов в кластерной конфигурации необходимо установить параметр [***max\_replica\_delay\_for\_distributed\_queries***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_4#:~:text=55.%20max_replica_delay_for_distributed_queries)и использовать HTTP ресурс **/replicas\_status**. Если реплика доступна и не отстаёт от других реплик, то запрос к ***/replicas\_status*** возвращает 200 OK. Если реплика отстаёт, то запрос возвращает 503 HTTP\_SERVICE\_UNAVAILABLE, включая информацию о размере отставания.

## 2.2 Устранение неисправностей

Основные типы проблем, связанных с установкой и эксплуатацией RT.WideStore:

установка дистрибутива,

соединение с сервером,

обработка запросов,

скорость обработки запросов.

### 2.2.1 Соединение с сервером

Возможные проблемы соединения с сервером:

сервер не запущен,

неожиданные или неправильные параметры конфигурации.

### 2.2.2 Сервер не запущен

Если сервер не запущен:

**Проверьте, запущен ли сервер командой:**

$ sudo service clickhouse-server status

Copy

Если сервер не запущен, запустите его с помощью команды:

$ sudo service clickhouse-server start

Copy

**Проверьте журналы:**

Основной лог clickhouse-server по умолчанию – /var/log/clickhouse-server/clickhouse-server.log.

Copy

В случае успешного запуска вы должны увидеть строки, содержащие:

<Information> Application: starting up. – сервер запускается.

<Information> Application: Ready for connections. – сервер запущен и готов принимать соединения.

Copy

Если **clickhouse-server** не запустился из-за ошибки конфигурации вы увидите <Error> строку с описанием ошибки. Например:

2019.01.11 15:23:25.549505 [ 45 ] {} <Error> ExternalDictionaries: Failed reloading

'event2id' external dictionary: Poco::Exception. Code: 1000, e.code() = 111, e.displayText() =

Connection refused, e.what() = Connection refused

Copy

Если вы не видите ошибки в конце файла, просмотрите весь файл начиная со строки:

<Information> Application: starting up.

Copy

При попытке запустить второй экземпляр ***clickhouse-server*** журнал выглядит следующим образом:

2019.01.11 15:25:11.151730 [ 1 ] {} <Information> : Starting ClickHouse 19.1.0 with revision

54413

2019.01.11 15:25:11.154578 [ 1 ] {} <Information> Application: starting up

2019.01.11 15:25:11.156361 [ 1 ] {} <Information> StatusFile: Status file ./status already

exists - unclean restart. Contents:

PID: 8510

Started at: 2019-01-11 15:24:23

Revision: 54413

2019.01.11 15:25:11.156673 [ 1 ] {} <Error> Application: DB::Exception: Cannot lock file

./status. Another server instance in same directory is already running.

2019.01.11 15:25:11.156682 [ 1 ] {} <Information> Application: shutting down

2019.01.11 15:25:11.156686 [ 1 ] {} <Debug> Application: Uninitializing subsystem: Logging

Subsystem

2019.01.11 15:25:11.156716 [ 2 ] {} <Information> BaseDaemon: Stop SignalListener thread

Copy

**Проверьте логи system.d:**

Если из логов ***clickhouse-server*** вы не получили необходимой информации или логов нет, то вы можете посмотреть логи ***system.d*** командой:

$ sudo journalctl -u clickhouse-server

Copy

**Запустите clickhouse-server в интерактивном режиме:**

$ sudo -u clickhouse /usr/bin/clickhouse-server --config-file /etc/clickhouse-server/config.xml

Copy

Эта команда запускает сервер как интерактивное приложение со стандартными параметрами скрипта автозапуска. В этом режиме clickhouse-server выводит сообщения в консоль.

### 2.2.3 Параметры конфигурации

Проверьте:

Настройки Docker:

При запуске ClickHouse в Docker в сети IPv6 убедитесь, что установлено `network=host`.

Copy

Параметры ***endpoint***:

Проверьте настройки

[listen\_host](server\_configuration\_parameters/settings.md#server\_configuration\_parameters-

listen\_host)

и

[tcp\_port](server\_configuration\_parameters/settings.md#server\_configuration\_parameters-

tcp\_port).

По умолчанию, сервер ClickHouse принимает только локальные подключения.

Copy

Настройки протокола HTTP:

Проверьте настройки протокола для HTTP API.

Copy

Параметры безопасного подключения:

Проверьте:

- Настройку `tcp\_port\_secure`.

- Параметры для SSL-сертификатов.

Используйте правильные параметры при подключении. Например, используйте параметр

`port\_secure` при использовании `clickhouse\_client`.

Copy

Настройки пользователей:

Возможно, вы используете неверное имя пользователя или пароль.

Copy

### 2.2.4 Обработка запросов

Если RT.WideStore не может обработать запрос, он отправляет клиенту описание ошибки. В clickhouse-client вы получаете описание ошибки в консоли. При использовании интерфейса HTTP, RT.WideStore отправляет описание ошибки в теле ответа. Например:

$ curl 'http://localhost:8123/' --data-binary "SELECT a"

Code: 47, e.displayText() = DB::Exception: Unknown identifier: a. Note that there are no

tables (FROM clause) in your query, context: required\_names: 'a' source\_tables:

table\_aliases: private\_aliases: column\_aliases: public\_columns: 'a' masked\_columns:

array\_join\_columns: source\_columns: , e.what() = DB::Exception

Copy

Если вы запускаете ***clickhouse-client*** c параметром stack-trace, то RT.WideStore возвращает описание ошибки и соответствующий стек вызовов функций на сервере.

Может появиться сообщение о разрыве соединения. В этом случае необходимо повторить запрос. Если соединение прерывается каждый раз при выполнении запроса, следует проверить журналы сервера на наличие ошибок.

### 2.2.5 Скорость обработки запросов

Если вы видите, что RT.WideStore работает слишком медленно, необходимо профилировать загрузку ресурсов сервера и сети для ваших запросов.

Для профилирования запросов можно использовать утилиту ***clickhouse-benchmark***. Она показывает количество запросов, обработанных за секунду, количество строк, обработанных за секунду и процент или времени обработки запросов.

## 2.3 Управление доступом

RT.WideStore поддерживает управление доступом на основе подхода RBAC, то есть основанную на управление доступом на основе ролей.

Объекты системы доступа в RT.WideStore:

аккаунт пользователя,

роль,

политика доступа к строкам,

профиль настроек,

квота.

Вы можете настроить объекты системы доступа, используя:

SQL-ориентированный воркфлоу.

Функциональность необходимо включить управление доступом (см. [п.2.3.9](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.3.9%20%D0%92%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)).

Конфигурационные файлы сервера: ***users.xml*** и ***config.xml***.

Рекомендуется использовать SQL-воркфлоу. Оба метода конфигурации работают одновременно, поэтому, если для управления доступом вы используете конфигурационные файлы, вы можете плавно перейти на SQL-воркфлоу.

***Внимание:****Нельзя одновременно использовать оба метода для управления одним и тем же объектом системы доступа.*

Чтобы посмотреть список всех пользователей, ролей, профилей и пр., а также все привилегии, используйте запрос [***SHOW ACCESS***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=7.18%20SHOW%20ACCESS).

### 2.3.1 Использование

По умолчанию сервер RT.WideStore предоставляет аккаунт пользователя default, для которого выключена функция SQL-ориентированного управления доступом, но у него есть все права и разрешения. Аккаунт ***default***используется во всех случаях, когда имя пользователя не определено. Например, при входе с клиента или в распределенных запросах. При распределенной обработке запроса ***default***используется, если в конфигурации сервера или кластера не указаны свойства ***user***и ***password***.

Если вы начали пользоваться RT.WideStore недавно, попробуйте следующий сценарий:

[Включите SQL-ориентированное управление доступом](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.3.9%20%D0%92%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B5%20SQL-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC) для пользователя ***default***.

Войдите под пользователем default и создайте всех необходимых пользователей. Не забудьте создать аккаунт администратора (GRANT ALL ON \*.\* TO admin\_user\_account WITH GRANT OPTION).

[Ограничьте разрешения](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.10.2%20%D0%A0%D0%B0%D0%B7%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%B4%D0%BB%D1%8F%20%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81%D0%BE%D0%B2) для пользователя ***default*** и отключите для него SQL-ориентированное управление доступом.

### 2.3.2 Особенности реализации

Существуют следующие особенности реализации выдачи разрешений:

Вы можете выдавать разрешения на базы данных или таблицы, даже если они не существуют.

При удалении таблицы все связанные с ней привилегии не отзываются. Если вы затем создадите новую таблицу с таким же именем, все привилегии останутся действительными. Чтобы отозвать привилегии, связанные с удаленной таблицей, необходимо выполнить, например, запрос REVOKE ALL PRIVILEGES ON db.table FROM ALL.

У привилегий нет настроек времени жизни.

### 2.3.3 Аккаунт пользователя

Аккаунт пользователя – это объект системы доступа, позволяющий авторизовать кого-либо в RT.WideStore. Аккаунт содержит:

Идентификационную информацию.

Привилегии, определяющие область действия запросов, которые могут быть выполнены пользователем.

Хосты, которые могут подключаться к серверу RT.WideStore.

Назначенные роли и роли по умолчанию.

Настройки и их ограничения, которые применяются по умолчанию при входе пользователя.

Присвоенные профили настроек.

[Привилегии](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=8.4%20%D0%9F%D1%80%D0%B8%D0%B2%D0%B8%D0%BB%D0%B5%D0%B3%D0%B8%D0%B8) присваиваются аккаунту пользователя с помощью запроса [***GRANT***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=8.%20GRANT)или через назначение [ролей](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.3.5%20%D0%A0%D0%BE%D0%BB%D1%8C). Отозвать привилегию можно с помощью запроса [***REVOKE***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=10.%20REVOKE). Чтобы вывести список присвоенных привилегий, используется выражение [***SHOW GRANTS***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=7.6%20SHOW%20GRANTS).

Запросы управления:

[CREATE USER](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=3.6%20CREATE%20USER),

[ALTER USER](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=4.%20ALTER),

[DROP USER](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=15.4%20DROP%20USER),

[SHOW CREATE USER](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=7.7%20SHOW%20CREATE%20USER).

### 2.3.4 Применение настроек

Настройки могут быть заданы разными способами: для аккаунта пользователя, для назначенных ему ролей или в профилях настроек. При входе пользователя, если настройка задана для разных объектов системы доступа, значение настройки и ее ограничения применяются в следующем порядке (от высшего приоритета к низшему):

Настройки аккаунта.

Настройки ролей по умолчанию для аккаунта. Если настройка задана для нескольких ролей, порядок применения не определен.

Настройки из профилей настроек, присвоенных пользователю или его ролям по умолчанию. Если настройка задана в нескольких профилях, порядок применения не определен.

Настройки, которые по умолчанию применяются ко всему серверу, или настройки из профиля по умолчанию (профили настроек ***default\_profile*** находятся в файле, указанном в параметре ***user\_config*** см. [Конфигурационные параметры сервера](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.9%20%D0%9A%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5%20%D0%BF%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%80%D1%8B%20%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%B0)).

### 2.3.5 Роль

Роль – это контейнер объектов системы доступа, которые можно присвоить аккаунту пользователя.

Роль содержит:

[Привилегии](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=8.4%20%D0%9F%D1%80%D0%B8%D0%B2%D0%B8%D0%BB%D0%B5%D0%B3%D0%B8%D0%B8) (привилегия – разрешение на выполнение определенного типа запросов, она имеет иерархическую структуру. Набор разрешенных запросов зависит от области действия привилегии).

Настройки и ограничения.

Список назначенных ролей.

Запросы управления:

[CREATE ROLE](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=3.7%20CREATE%20ROLE),

[ALTER ROLE](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=4.%20ALTER),

[DROP ROLE](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=15.5%20DROP%20ROLE),

[SET ROLE](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=22.%20SET%20ROLE),

[SET DEFAULT ROLE](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=22.1%20SET%20DEFAULT%20ROLE),

[SHOW CREATE ROLE](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=7.8%20SHOW%20CREATE%20ROLE).

Привилегии можно присвоить роли с помощью запроса [***GRANT***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=8.%20GRANT). Для отзыва привилегий у роли RT.WideStore предоставляет запрос [***REVOKE***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=10.%20REVOKE).

### 2.3.6 Политика доступа к строкам

Политика доступа к строкам – это фильтр, определяющий, какие строки доступны пользователю или роли. Политика содержит фильтры для конкретной таблицы, а также список ролей и/или пользователей, которые должны использовать данную политику.

Запросы управления:

[CREATE ROW POLICY](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=3.8%20CREATE%20ROW%20POLICY),

[ALTER ROW POLICY](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=4.%20ALTER),

[DROP ROW POLICY](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=15.6%20DROP%20ROW%20POLICY),

[SHOW CREATE ROW POLICY](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=7.9%20SHOW%20CREATE%20ROW%20POLICY).

### 2.3.7 Профиль настроек

Профиль настроек – это набор настроек. Профиль настроек содержит настройки и ограничения, а также список ролей и/или пользователей, по отношению к которым применяется данный профиль.

Запросы управления:

[CREATE SETTINGS PROFILE](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=3.10%20CREATE%20SETTINGS%20PROFILE),

[ALTER SETTINGS PROFILE](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=4.%20ALTER),

[DROP SETTINGS PROFILE](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=15.8%20DROP%20SETTINGS%20PROFILE),

[SHOW CREATE SETTINGS PROFILE](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=7.11%20SHOW%20CREATE%20SETTINGS%20PROFILE).

### 2.3.8 Квота

Квота ограничивает использование ресурсов. См. [п. 2.6](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.6%20%D0%9A%D0%B2%D0%BE%D1%82%D1%8B).

Квота содержит набор ограничений определенной длительности, а также список ролей и/или пользователей, на которых распространяется данная квота.

Запросы управления:

[CREATE QUOTA](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=3.9%20CREATE%20QUOTA),

[ALTER QUOTA](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=4.%20ALTER),

[DROP QUOTA](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=15.7%20DROP%20QUOTA),

[SHOW CREATE QUOTA](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_1#:~:text=7.10%20SHOW%20CREATE%20QUOTA).

### 2.3.9 Включение SQL-ориентированного управления доступом

Для включения SQL-ориентированного управления доступом:

Настройте каталог для хранения конфигураций.

RT.WideStore хранит конфигурации объектов системы доступа в каталоге, установленном в конфигурационном параметре сервера access\_control\_path.

Включите SQL-ориентированное управление доступом как минимум для одного аккаунта.

По умолчанию управление доступом на основе SQL выключено для всех пользователей. Вам необходимо настроить хотя бы одного пользователя в файле конфигурации users.xml и присвоить значение 1 параметру access\_management.

## 2.4 Резервное копирование данных

[Репликация](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_6#:~:text=2.1.8%20%D0%A0%D0%B5%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)обеспечивает защиту от аппаратных сбоев, но не защищает от человеческих ошибок: случайного удаления данных, удаления не той таблицы, которую надо было, или таблицы на не том кластере, а также программных ошибок, которые приводят к неправильной обработке данных или их повреждению. Во многих случаях подобные ошибки влияют на все реплики. RT.WideStore имеет встроенные средства защиты для предотвращения некоторых типов ошибок – например, по умолчанию не получится удалить таблицы \*MergeTree, содержащие более 50 Гб данных, одной командой. Однако эти средства защиты не охватывают все возможные случаи и могут быть обойдены.

Для того чтобы эффективно уменьшить возможные человеческие ошибки, следует тщательно подготовить стратегию резервного копирования и восстановления данных **заранее**.

Каждая компания имеет различные доступные ресурсы и бизнес-требования, поэтому нет универсального решения для резервного копирования и восстановления RT.WideStore, которое будет подходить в каждой ситуации. То, что работает для одного гигабайта данных, скорее всего, не будет работать для десятков петабайт. Существует множество возможных подходов со своими плюсами и минусами, которые будут рассмотрены ниже. Рекомендуется использовать несколько подходов вместо одного, чтобы компенсировать их различные недостатки.

***Примечание:****Имейте в виду, что, если вы создали резервную копию чего-то и никогда не пытались восстановить её, скорее всего, восстановление не будет работать должным образом, когда вам это действительно понадобится (или, по крайней мере, это займет больше времени, чем будет приемлемо для бизнеса). Поэтому, какой бы подход к резервному копированию вы ни выбрали, обязательно автоматизируйте процесс восстановления и регулярно запускайте его на резервном кластере RT.WideStore.*

### 2.4.1 Дублирование данных

Часто данные, которые поступают в RT.WideStore, доставляются через некоторую отказоустойчивую очередь, например [Apache Kafka](https://kafka.apache.org/). В этом случае можно настроить дополнительный набор подписчиков, которые будут считывать один и тот же поток данных во время записи в RT.WideStore и хранить его в холодном хранилище. Большинство компаний уже имеют некоторые рекомендуемые по умолчанию холодные хранилища, которые могут быть хранилищем объектов или распределенной файловой системой, например [HDFS](https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html).

### 2.4.2 Снимки файловой системы

Некоторые локальные файловые системы позволяют делать снимки (например, [ZFS](https://en.wikipedia.org/wiki/ZFS))), но они могут быть не лучшим выбором для обслуживания живых запросов. Возможным решением является создание дополнительных реплик с такой файловой системой и исключение их из [Distributed](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_6#:~:text=2.5.1%20Distributed)таблиц, используемых для запросов SELECT. Снимки на таких репликах будут недоступны для запросов, изменяющих данные. В качестве бонуса, эти реплики могут иметь особые конфигурации оборудования с большим количеством дисков, подключенных к серверу, что будет экономически эффективным.

### 2.4.3 clickhouse-copier

clickhouse-copier – это универсальный инструмент, который изначально был создан для перешардирования таблиц с петабайтами данных. Его также можно использовать для резервного копирования и восстановления, поскольку он надёжно копирует данные между таблицами и кластерами RT.WideStore.

Для небольших объёмов данных можно применять "INSERT INTO ... SELECT ..." в удалённые таблицы.

### 2.4.4 Манипуляции с партициями

RT.WideStore позволяет использовать запрос "ALTER TABLE ... FREEZE PARTITION ..." для создания локальной копии партиций таблицы. Это реализуется с помощью жестких ссылок (hardlinks) на каталог ***/var/lib/clickhouse/shadow/***, поэтому такая копия обычно не занимает дополнительное место на диске для старых данных. Созданные копии файлов не обрабатываются сервером RT.WideStore, поэтому вы можете просто оставить их там: у вас будет простая резервная копия, которая не требует дополнительной внешней системы, однако при аппаратных проблемах вы можете утратить и актуальные данные и сохраненную копию. По этой причине, лучше удаленно скопировать их в другое место, а затем удалить локальную копию. Распределенные файловые системы и хранилища объектов по-прежнему являются хорошими вариантами для этого, однако можно использовать и обычные присоединенные файловые серверы с достаточно большой ёмкостью (в этом случае передача будет происходить через сетевую файловую систему или, возможно, [rsync](https://en.wikipedia.org/wiki/Rsync)).

Для автоматизации этого подхода доступен инструмент от сторонних разработчиков: [clickhouse-backup](https://github.com/AlexAkulov/clickhouse-backup) (инструмент для простого резервного копирования и восстановления RT.WideStore с поддержкой облачных хранилищ).

## 2.5 Конфигурационные файлы

RT.WideStore поддерживает многофайловое управление конфигурацией. Основной конфигурационный файл сервера — **/etc/clickhouse-server/config.xml** или **/etc/clickhouse-server/config.yaml**. Остальные файлы должны находиться в директории **/etc/clickhouse-server/config.d**. Обратите внимание, что конфигурационные файлы могут быть записаны в форматах XML или YAML, но смешение этих форматов в одном файле не поддерживается. Например, можно хранить основные конфигурационные файлы как **config.xml** и **users.xml**, а дополнительные файлы записать в директории **config.d** и **users.d** в формате **.yaml**.

Все XML файлы должны иметь одинаковый корневой элемент, обычно **<clickhouse>**. Для YAML элемент **clickhouse:** должен отсутствовать, так как парсер вставляет его автоматически.

### ****2.5.1 Переопределение****

Некоторые настройки, определенные в основном конфигурационном файле, могут быть переопределены в других файлах:

У элементов этих конфигурационных файлов могут быть указаны атрибуты **replace** или **remove**.

Если ни один атрибут не указан, сервер объединит содержимое элементов рекурсивно, заменяя совпадающие значения дочерних элементов.

Если указан атрибут **replace**, сервер заменит весь элемент на указанный.

Если указан атрибут **remove**, сервер удалит элемент.

Также возможно указать атрибуты как переменные среды с помощью **from\_env="VARIABLE\_NAME"**:

<clickhouse>

<macros>

<replica from\_env="REPLICA" />

<layer from\_env="LAYER" />

<shard from\_env="SHARD" />

</macros>

</clickhouse>

Copy

### ****2.5.2 Подстановки****

В конфигурационном файле могут быть указаны «подстановки». Если у элемента присутствует атрибут **incl**, то в качестве значения будет использована соответствующая подстановка из файла. По умолчанию путь к файлу с подстановками - **/etc/metrika.xml**. Он может быть изменён в конфигурации сервера в элементе include\_from. Значения подстановок указываются в элементах **/clickhouse/имя\_подстановки** этого файла. Если подстановка, заданная в **incl**, отсутствует, то делается соответствующая запись в лог. Чтобы RT.WideStore фиксировал в логе отсутствие подстановки, необходимо указать атрибут **optional="true"** (например, настройки для macros).

Если нужно заменить весь элемент подстановкой, можно использовать **include** как имя элемента.

Пример подстановки XML:

<clickhouse>

<!-- Appends XML subtree found at `/profiles-in-zookeeper` ZK path to `<profiles>` element. -->

<profiles from\_zk="/profiles-in-zookeeper" />

<users>

<!-- Replaces `include` element with the subtree found at `/users-in-zookeeper` ZK path. -->

<include from\_zk="/users-in-zookeeper" />

<include from\_zk="/other-users-in-zookeeper" />

</users>

</clickhouse>

Copy

Подстановки могут также выполняться из ZooKeeper. Для этого укажите у элемента атрибут **from\_zk = "/path/to/node"**. Значение элемента заменится на содержимое узла **/path/to/node** в ZooKeeper. В ZooKeeper-узел также можно положить целое XML-поддерево, оно будет целиком вставлено в исходный элемент.

В элементе **users\_config** файла **config.xml** можно указать относительный путь к конфигурационному файлу с настройками пользователей, профилей и квот. Значение **users\_config** по умолчанию — **users.xml**. Если **users\_config** не указан, то настройки пользователей, профилей и квот можно задать непосредственно в **config.xml**.

Настройки пользователя могут быть разделены в несколько отдельных файлов аналогичных **config.xml** и **config.d\**. Имя директории задаётся также как **users\_config**. Имя директории задаётся так же, как имя файла в **users\_config**, с подстановкой **.d** вместо **.xml**/**.yaml**. Директория **users.d** используется по умолчанию, также как **users.xml** используется для **users\_config**. Например, можно иметь по отдельному конфигурационному файлу для каждого пользователя:

$ cat /etc/clickhouse-server/users.d/alice.xml

Copy

<clickhouse>

<users>

<alice>

<profile>analytics</profile>

<networks>

<ip>::/0</ip>

</networks>

<password\_sha256\_hex>...</password\_sha256\_hex>

<quota>analytics</quota>

</alice>

</users>

</clickhouse>

Copy

Для каждого конфигурационного файла, сервер при запуске генерирует также файлы **file-preprocessed.xml**. Эти файлы содержат все выполненные подстановки и переопределения, и предназначены для информационных целей. Если в конфигурационных файлах были использованы ZooKeeper-подстановки, но при старте сервера ZooKeeper недоступен, то сервер загрузит конфигурацию из preprocessed-файла.

Сервер следит за изменениями конфигурационных файлов, а также файлов и ZooKeeper-узлов, которые были использованы при выполнении подстановок и переопределений, и перезагружает настройки пользователей и кластеров на лету. То есть, можно изменять кластера, пользователей и их настройки без перезапуска сервера.

### ****2.5.3 Примеры записи конфигурации на YAML****

Здесь можно рассмотреть пример реальной конфигурации записанной на YAML: [config.yaml.example](https://github.com/ClickHouse/ClickHouse/blob/master/programs/server/config.yaml.example).

Между стандартами XML и YAML имеются различия, поэтому в этом разделе будут перечислены некоторые подсказки для написания конфигурации на YMAL.

Для записи обычной пары ключ-значение следует использовать Scalar:

key: value

Copy

Для создания тега, содержащего подтеги следует использовать Map:

map\_key:

key1: val1

key2: val2

key3: val3

Copy

Для создания списка значений или подтегов, расположенных по определенному ключу, следует использовать Sequence:

seq\_key:

- val1

- val2

- key1: val3

- map:

key2: val4

key3: val5

Copy

В случае, если необходимо объявить тег, аналогичный XML-атрибуту, необходимо задать скаляр, имеющий ключ с префиксом @ и заключенный в кавычки:

map:

"@attr1": value1

"@attr2": value2

key: 123

Copy

Из такой Map мы получим после конвертации:

<map attr1="value1" attr2="value2">

<key>123</key>

</map>

Copy

Помимо Map, можно задавать атрибуты для Sequence:

seq:

- "@attr1": value1

- "@attr2": value2

- 123 - abc

Copy

Таким образом получая аналог следующей записи на XML:

<seq attr1="value1" attr2="value2">123</seq>

<seq attr1="value1" attr2="value2">abc</seq>

Copy

### ****2.5.4 Детали реализации****

При старте сервера для каждого конфигурационного файла создаются файлы предобработки **file-preprocessed.xml**. Они содержат все выполненные подстановки и переопределения (эти сведения записываются просто для информации). Если в конфигурационном файле настроены подстановки ZooKeeper, но при старте сервера ZooKeeper не доступен, то сервер загружает конфигурацию из соответствующего файла предобработки.

Сервер отслеживает как изменения в конфигурационных файлах, так и файлы и узы ZooKeeper, которые были использованы при выполнении подстановок и переопределений, и на ходу перезагружает настройки для пользователей и кластеров. Это означает, что можно изменять кластеры, пользователей и их настройки без перезапуска сервера.

## 2.6 Квоты

Квоты позволяют ограничить использование ресурсов за некоторый интервал времени, или просто подсчитывать использование ресурсов.

Квоты настраиваются в конфиге пользователей. Обычно это***users.xml***.

В системе есть возможность ограничить сложность одного запроса. Для этого смотрите раздел «Ограничения на сложность запроса».

В отличие от них, квоты:

ограничивают не один запрос, а множество запросов, которые могут быть выполнены за интервал времени;

при распределённой обработке запроса, учитывают ресурсы, потраченные на всех удалённых серверах.

Рассмотрим фрагмент файла ***users.xml***, описывающего квоты:

<!-- Квоты. -->

<quotas>

<!-- Имя квоты. -->

<default>

<!-- Ограничения за интервал времени. Можно задать много интервалов с разными ограничениями. -->

<interval>

<!-- Длина интервала. -->

<duration>3600</duration>

<!-- Без ограничений. Просто считать соответствующие данные за указанный интервал. -->

<queries>0</queries>

<errors>0</errors>

<result\_rows>0</result\_rows>

<read\_rows>0</read\_rows>

<execution\_time>0</execution\_time>

</interval>

</default>

Copy

Видно, что квота по умолчанию просто считает использование ресурсов за каждый час, но не ограничивает их.

Подсчитанное использование ресурсов за каждый интервал, выводится в лог сервера после каждого запроса:

<statbox>

<!-- Ограничения за интервал времени. Можно задать много интервалов с разными ограничениями. -->

<interval>

<!-- Длина интервала. -->

<duration>3600</duration>

<queries>1000</queries>

<errors>100</errors>

<result\_rows>1000000000</result\_rows>

<read\_rows>100000000000</read\_rows>

<execution\_time>900</execution\_time>

</interval>

<interval>

<duration>86400</duration>

<queries>10000</queries>

<errors>1000</errors>

<result\_rows>5000000000</result\_rows>

<read\_rows>500000000000</read\_rows>

<execution\_time>7200</execution\_time>

</interval>

</statbox>

Copy

Для квоты с именем ***statbox***заданы ограничения за каждый час и за каждые 24 часа (86 400 секунд). Интервал времени считается, начиная от некоторого ***implementation defined***фиксированного момента времени. То есть, интервал длины 24 часа начинается не обязательно в полночь.

Когда интервал заканчивается, все накопленные значения сбрасываются. То есть, в следующий час, расчёт квоты за час, начинается заново.

Рассмотрим величины, которые можно ограничить:

***queries***– общее количество запросов;

***errors***– количество запросов, при выполнении которых было выкинуто исключение;

***result\_rows***– суммарное количество строк, отданных в виде результата;

***read\_rows***– суммарное количество исходных строк, прочитанных из таблиц, для выполнения запроса, на всех удалённых серверах;

***execution\_time***– суммарное время выполнения запросов, в секундах (wall time).

Если за хотя бы один интервал, ограничение превышено, то кидается исключение с текстом о том, какая величина превышена, за какой интервал, и когда начнётся новый интервал (когда снова можно будет задавать запросы).

Для квоты может быть включена возможность указывать «ключ квоты», чтобы производить учёт ресурсов для многих ключей независимо. Рассмотрим это на примере:

<!-- Для глобального конструктора отчётов. -->

<web\_global>

<!-- keyed - значит в параметре запроса передаётся "ключ" quota\_key,

и квота считается по отдельности для каждого значения ключа.

Например, в качестве ключа может передаваться логин пользователя в Метрике,

и тогда квота будет считаться для каждого логина по отдельности.

Имеет смысл использовать только если quota\_key передаётся не пользователем, а программой.

Также можно написать <keyed\_by\_ip /> - тогда в качестве ключа квоты используется IP-адрес.

(но стоит учесть, что пользователь может достаточно легко менять IPv6-адрес)

-->

<keyed />

Copy

Квота прописывается для пользователей в секции users конфига. Смотрите раздел «Права доступа».

При распределённой обработке запроса, накопленные величины хранятся на сервере-инициаторе запроса. То есть, если пользователь пойдёт на другой сервер - там квота будет действовать «с нуля».

При перезапуске сервера, квоты сбрасываются.

## 2.7 Оптимизация производительности

### 2.7.1 Профилировщик запросов

RT.WideStore запускает профилировщик выборки, позволяющий анализировать выполнение запроса. Используя профилировщик, вы можете найти процедуры исходного кода, которые наиболее часто используются во время выполнения запроса. Вы можете отслеживать время процессора и реальное время, включая время простоя.

Чтобы использовать профилировщик:

Настройте раздел trace\_log конфигурации сервера.

В этом разделе настраивается системная таблица ***trace\_log***, содержащая результаты работы профилировщика. Он настроен по умолчанию. Помните, что данные в этой таблице действительны только для работающего сервера. После перезапуска сервера RT.WideStore не очищает таблицу, и весь сохраненный адрес виртуальной памяти может стать недействительным.

Установите параметры [***query\_profiler\_cpu\_time\_period\_ns***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_4#:~:text=121.%20query_profiler_cpu_time_period_ns)или [***query\_profiler\_real\_time\_period\_ns***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_4#:~:text=120.%20query_profiler_real_time_period_ns). Обе настройки можно использовать одновременно.

Эти параметры позволяют настраивать таймеры профилировщика. Поскольку это настройки сеанса, вы можете получить различную частоту выборки для всего сервера, отдельных пользователей или профилей пользователей, для вашего интерактивного сеанса и для каждого отдельного запроса.

Частота дискретизации по умолчанию – одна выборка в секунду, при этом включены как ЦП, так и таймеры реального времени. Такая частота позволяет собрать достаточно информации о кластере RT.WideStore. При этом, работая с такой периодичностью, профилировщик не влияет на производительность сервера RT.WideStore. Если вам нужно профилировать каждый отдельный запрос, попробуйте использовать более высокую частоту выборки.

Чтобы проанализировать системную таблицу ***trace\_log***:

Установите ***clickhouse-common-static-dbg*** пакет.

Разрешите функции самоанализа с помощью параметра [***allow\_introspection\_functions***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_4#:~:text=122.%20allow_introspection_functions).

По соображениям безопасности функции самоанализа по умолчанию отключены.

Используйте функции самоанализа ***аaddressToLine***, ***addressToSymbol***, чтобы получить имена функций и их позиции в коде RT.WideStore. Чтобы получить профиль для какого-либо запроса, вам нужно агрегировать данные из таблицы ***trace\_log***. Вы можете агрегировать данные по отдельным функциям или по всем трассировкам стека.

Если вам нужно визуализировать ***trace\_log***информацию, используйте ***flamegraph***и ***speedscope***.

Пример использования профилировщика запросов представлен в [***Приложении 1***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_1).

## 2.8 Системные таблицы

[Системные таблицы](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2) содержат следующую информацию:

состоянии сервера, процессов и окружении,

внутренние процессы сервера.

Системные таблицы:

находятся в базе данных system,

доступны только для чтения,

не могут быть удалены или изменены, но их можно отключить.

Большинство системных таблиц хранят свои данные в оперативной памяти. Сервер RT.WideStore создает эти системные таблицы при старте.

В отличие от других системных таблиц, таблицы с системными логами [***metric\_log***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2#:~:text=23.%20system.metric_log), [***query\_log***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2#:~:text=34.%20system.query_log), [***query\_thread\_log***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2#:~:text=35.%20system.query_thread_log), [***trace\_log***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2#:~:text=54.%20system.trace_log), [***part\_log***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2#:~:text=30.%20system.part_log), [***crash\_log***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2#:~:text=6.%20system.crash_log)и [***text\_log***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2#:~:text=53.%20system.text_log)используют движок таблиц [***MergeTree***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_6#:~:text=2.1.1%20MergeTree)и по умолчанию хранят свои данные в файловой системе. Если удалить таблицу из файловой системы, сервер RT.WideStore снова создаст пустую таблицу во время следующей записи данных. Если схема системной таблицы изменилась в новом релизе, то RT.WideStore переименует текущую таблицу и создаст новую.

Таблицы с системными логами log можно настроить, создав конфигурационный файл с тем же именем, что и таблица в разделе ***/etc/clickhouse-server/config.d/***, или указав соответствующие элементы в ***/etc/clickhouse-server/config.xml***. Настраиваться могут следующие элементы:

***database***– база данных, к которой принадлежит системная таблица. Эта опция на текущий момент устарела. Все системные таблицы находятся в базе данных ***system***.

***table***– таблица для добавления данных.

***partition\_by***– [ключ партиционирования](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_UG_Attachment_6#:~:text=2.1.9%20%D0%9F%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9%20%D0%BA%D0%BB%D1%8E%D1%87%20%D0%BF%D0%B0%D1%80%D1%82%D0%B8%D1%86%D0%B8%D0%BE%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F).

***ttl***– время жизни таблицы.

***flush\_interval\_milliseconds***– интервал сброса данных на диск, в миллисекундах.

***engine***– полное имя движка (начиная с ENGINE =) с параметрами. Эта опция противоречит ***partition\_by***и ***ttl***. Если указать оба параметра вместе, сервер вернет ошибку и завершит работу.

Пример:

<widestore>

<query\_log>

<database>system</database>

<table>query\_log</table>

<partition\_by>toYYYYMM(event\_date)</partition\_by>

<ttl>event\_date + INTERVAL 30 DAY DELETE</ttl>

<!--

<engine>ENGINE = MergeTree PARTITION BY toYYYYMM(event\_date) ORDER BY (event\_date, event\_time) SETTINGS index\_granularity = 1024</engine>

-->

<flush\_interval\_milliseconds>7500</flush\_interval\_milliseconds>

</query\_log>

</widestore>

Copy

По умолчанию размер таблицы не ограничен. Управлять размером таблицы можно используя TTL для удаления устаревших записей журнала. Также вы можете использовать функцию партиционирования для таблиц MergeTree.

Описание системных таблиц представлено в [***Приложении 2***](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_2)***.***

### 2.8.1 Источники системных показателей

Для сбора системных показателей сервер RT.WideStore использует:

возможности ***CAP\_NET\_ADMIN***,

***procfs***(только Linux).

Если для сервера RT.WideStore не включено ***CAP\_NET\_ADMIN***, он пытается обратиться к ***ProcfsMetricsProvider***. ***ProcfsMetricsProvider***позволяет собирать системные показатели для каждого запроса (для CPU и I/O).

Если ***procfs***поддерживается и включена в системе, то сервер RT.WideStore собирает следующие системные показатели:

OSCPUVirtualTimeMicroseconds,

OSCPUWaitMicroseconds,

OSIOWaitMicroseconds,

OSReadChars,

OSWriteChars,

OSReadBytes,

OSWriteBytes.

## 2.9 Конфигурационные параметры сервера

Раздел содержит описания настроек сервера, которые не могут изменяться на уровне сессии или запроса.

Рассмотренные настройки хранятся в файле ***config.xml*** сервера RT.WideStore.

Прочие настройки описаны в разделе «[Настройки](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.10%20%D0%9D%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B8)».

Перед изучением настроек ознакомьтесь с разделом [Конфигурационные файлы](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.5%20%D0%9A%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5%20%D1%84%D0%B0%D0%B9%D0%BB%D1%8B), обратите внимание на использование подстановок (атрибуты ***incl*** и ***optional***).

## 2.10 Настройки

Все настройки, описанные ниже, могут быть заданы несколькими способами.

Настройки задаются послойно, т.е. каждый следующий слой перезаписывает предыдущие настройки.

Способы задания настроек, упорядоченные по приоритету:

Настройки в конфигурационном файле сервера ***users.xml***.

Устанавливаются в элементе ***<profiles>***.

Настройки для сессии.

Из консольного клиента RT.WideStore в интерактивном режиме отправьте запрос ***SET setting=value***.

Аналогично можно использовать RT.WideStore-сессии в HTTP-протоколе, для этого необходимо указывать HTTP-параметр ***session\_id***.

Настройки для запроса:

При запуске консольного клиента RT.WideStore в не интерактивном режиме установите параметр запуска ***--setting=value***.

При использовании HTTP API передавайте cgi-параметры (URL?setting\_1=value&setting\_2=value...).

Укажите необходимые настройки в секции ***SETTINGS*** запроса ***SELECT***. Эти настройки действуют только в рамках данного запроса, а после его выполнения сбрасываются до предыдущего значения или значения по умолчанию.

Настройки, которые можно задать только в конфигурационном файле сервера, в разделе не рассматриваются.

### 2.10.1 Пользовательские настройки

В дополнение к общим настройкам, пользователи могут определять собственные настройки.

Название пользовательской настройки должно начинаться с одного из предопределённых префиксов. Список этих префиксов должен быть задан в параметре ***custom\_settings\_prefixes*** конфигурационного файла сервера.

<custom\_settings\_prefixes> custom\_</custom\_settings\_prefixes>

Copy

Чтобы задать значение пользовательской настройке, используйте команду ***SET***:

SET custom\_a = 123;

Copy

Чтобы получить текущее значение пользовательской настройки, используйте функцию ***getSetting()***:

SELECT getSetting('custom\_a');

Copy

### 2.10.2 Разрешения для запросов

Запросы в RT.WideStore можно разделить на несколько типов:

Запросы на чтение данных: **SELECT**, **SHOW**, **DESCRIBE**, **EXISTS**.

Запросы за запись данных: **INSERT**, **OPTIMIZE**.

Запросы на изменение настроек: **SET**, **USE**.

[Запросы DDL](https://ru.wikipedia.org/wiki/Data_Definition_Language): **CREATE**, **ALTER**, **RENAME**, **ATTACH**, **DETACH**, **DROP** **TRUNCATE**.

**KILL QUERY**.

Разрешения пользователя по типу запроса регулируются параметрами:

***KILL QUERY***.

Разрешения пользователя по типу запроса регулируются параметрами:

***readonly*** – ограничивает разрешения для всех типов запросов, кроме DDL.

***allow\_ddl*** – ограничивает разрешения для DDL запросов.

KILL QUERY выполняется с любыми настройками.

### 2.10.3 readonly

Ограничивает разрешения для запросов на чтение данных, запись данных и изменение параметров.

Разделение запросов по типам смотрите по тексту выше по тексту.

Возможные значения:

0 – разрешены все запросы.

1 – разрешены только запросы на чтение данных.

2 – разрешены запросы на чтение данных и изменение настроек.

После установки readonly = 1 или 2 пользователь не может изменить настройки ***readonly*** и ***allow\_ddl*** в текущей сессии.

При использовании метода ***GET*** в ***HTTP*** интерфейсе, ***readonly = 1*** устанавливается автоматически. Для изменения данных используйте метод ***POST***.

Установка ***readonly = 1*** запрещает изменение всех настроек. Существует способ запретить изменения только некоторых настроек, см. [ограничения на изменение настроек](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.14%20%D0%9E%D0%B3%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BD%D0%B0%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B5%D0%BA).

Значение по умолчанию: 0.

### 2.10.4 allow\_ddl

Разрешает/запрещает ***DDL*** запросы.

Разделение запросов по типам смотрите по тексту выше по тексту.

Возможные значения:

0 – ***DDL*** запросы не разрешены.

1 – ***DDL*** запросы разрешены.

Если allow\_ddl = 0, то невозможно выполнить SET allow\_ddl = 1 для текущей сессии.

Значение по умолчанию: 1.

## 2.11 Ограничения на сложность запроса

Ограничения на сложность запроса - часть настроек.

Используются, чтобы обеспечить более безопасное исполнение запросов из пользовательского интерфейса.

Почти все ограничения действуют только на SELECT-ы.

При распределённой обработке запроса, ограничения действуют на каждом сервере по отдельности.

Ограничения проверяются на каждый блок обработанных данных, а не на каждую строку. В связи с этим, ограничения могут быть превышены на размер блока.

Ограничения вида «максимальное количество чего-нибудь» могут принимать значение 0, которое обозначает «не ограничено».

Для большинства ограничений также присутствует настройка вида overflow\_mode - что делать, когда ограничение превышено.

Оно может принимать одно из двух значений: throw или break; а для ограничения на агрегацию (***group\_by\_overflow\_mode***) есть ещё значение any.

throw – кинуть исключение (по умолчанию).

break – прервать выполнение запроса и вернуть неполный результат, как будто исходные данные закончились.

any (только для ***group\_by\_overflow\_mode***) – продолжить агрегацию по ключам, которые успели войти в набор, но не добавлять новые ключи в набор.

Параметры ограничения на сложность запросов представлены в [**Приложении 3**](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_3).

## 2.12 Другие настройки

#### Описание дополнительных настроек RT.WideStore представлены в [*Приложении 4*](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_AI_Attachment_4).

## 2.13 Профили настроек

Профиль настроек – это набор настроек, сгруппированных под одним именем.

***Примечание:****Для управления профилями настроек рекомендуется использовать SQL-ориентированный воркфлоу, который также поддерживается в RT.WideStore.*

Название профиля может быть любым. Вы можете указать один и тот же профиль для разных пользователей. Самое важное, что можно прописать в профиле – ***readonly=1***, это обеспечит доступ только на чтение.

Профили настроек поддерживают наследование. Это реализуется указанием одной или нескольких настроек profile перед остальными настройками, перечисленными в профиле. Если одна настройка указана в нескольких профилях, используется последнее из значений.

Все настройки профиля можно применить, установив настройку profile.

Пример:

Установить профиль web:

SET profile = 'web'

Copy

Профили настроек объявляются в конфигурационном файле пользователей. Обычно это ***users.xml***.

Пример:

<!-- Settings profiles -->

<profiles>

<!-- Default settings -->

<default>

<!-- The maximum number of threads when running a single query. -->

<max\_threads>8</max\_threads>

</default>

<!-- Settings for quries from the user interface -->

<web>

<max\_rows\_to\_read>1000000000</max\_rows\_to\_read>

<max\_bytes\_to\_read>100000000000</max\_bytes\_to\_read>

<max\_rows\_to\_group\_by>1000000</max\_rows\_to\_group\_by>

<group\_by\_overflow\_mode>any</group\_by\_overflow\_mode>

<max\_rows\_to\_sort>1000000</max\_rows\_to\_sort>

<max\_bytes\_to\_sort>1000000000</max\_bytes\_to\_sort>

<max\_result\_rows>100000</max\_result\_rows>

<max\_result\_bytes>100000000</max\_result\_bytes>

<result\_overflow\_mode>break</result\_overflow\_mode>

<max\_execution\_time>600</max\_execution\_time>

<min\_execution\_speed>1000000</min\_execution\_speed>

<timeout\_before\_checking\_execution\_speed>15</timeout\_before\_checking\_execution\_speed>

<max\_columns\_to\_read>25</max\_columns\_to\_read>

<max\_temporary\_columns>100</max\_temporary\_columns>

<max\_temporary\_non\_const\_columns>50</max\_temporary\_non\_const\_columns>

<max\_subquery\_depth>2</max\_subquery\_depth>

<max\_pipeline\_depth>25</max\_pipeline\_depth>

<max\_ast\_depth>50</max\_ast\_depth>

<max\_ast\_elements>100</max\_ast\_elements>

<readonly>1</readonly>

</web>

</profiles>

Copy

В примере задано два профиля: ***default*** и ***web***.

Профиль ***default*** имеет специальное значение – он обязателен и применяется при запуске сервера. Профиль ***default*** содержит настройки по умолчанию.

Профиль ***web*** – обычный профиль, который может быть установлен с помощью запроса SET или параметра URL при запросе по HTTP.

## 2.14 Ограничения на изменение настроек

Ограничения на изменение настроек могут находиться внутри секции ***profiles*** файла ***user.xml*** и запрещают пользователю менять некоторые настройки с помощью запроса SET.

Выглядит это следующим образом:

<profiles>

<имя\_пользователя>

<constraints>

<настройка\_1>

<min>нижняя\_граница</min>

</настройка\_1>

<настройка\_2>

<max>верхняя\_граница</max>

</настройка\_2>

<настройка\_3>

<min>нижняя\_граница</min>

<max>верхняя\_граница</max>

</настройка\_3>

<настройка\_4>

<readonly/>

</настройка\_4>

</constraints>

</имя\_пользователя>

</profiles>

Copy

Если пользователь пытается выйти за пределы, установленные этими ограничениями, то кидается исключение и настройка сохраняет прежнее значение.

Поддерживаются три типа ограничений: ***min***, ***max*** и ***readonly***. Ограничения ***min*** и ***max*** указывают нижнюю и верхнюю границы для числовых настроек и могут использоваться вместе.

Ограничение ***readonly*** указывает, что пользователь не может менять настройку.

Пример:

Пусть файл ***users.xml*** содержит строки:

<profiles>

<default>

<max\_memory\_usage>10000000000</max\_memory\_usage>

<force\_index\_by\_date>0</force\_index\_by\_date>

...

<constraints>

<max\_memory\_usage>

<min>5000000000</min>

<max>20000000000</max>

</max\_memory\_usage>

<force\_index\_by\_date>

<readonly/>

</force\_index\_by\_date>

</constraints>

</default>

</profiles>

Copy

Каждый из следующих запросов кинет исключение:

SET max\_memory\_usage=20000000001;

SET max\_memory\_usage=4999999999;

SET force\_index\_by\_date=1;

Code: 452, e.displayText() = DB::Exception: Setting max\_memory\_usage should not be greater than 20000000000.

Code: 452, e.displayText() = DB::Exception: Setting max\_memory\_usage should not be less than 5000000000.

Code: 452, e.displayText() = DB::Exception: Setting force\_index\_by\_date should not be changed.

Copy

***Примечание:****профиль с именем***default***обрабатывается специальным образом: все ограничения на изменение настроек из этого профиля становятся дефолтными и влияют на всех пользователей, кроме тех, где эти ограничения явно переопределены.*

## 2.15 Настройки пользователей в файле user.xml

Раздел users конфигурационного файла user.xml содержит настройки для пользователей.

***Примечание:****Для управления пользователями рекомендуется использовать SQL-ориентированный воркфлоу, который также поддерживается в RT.WideStore.*

Структура раздела users:

<users>

<!-- If user name was not specified, 'default' user is used. -->

<user\_name>

<password></password>

<!-- Or -->

<password\_sha256\_hex></password\_sha256\_hex>

<access\_management>0|1</access\_management>

<networks incl="networks" replace="replace">

</networks>

<profile>profile\_name</profile>

<quota>default</quota>

<databases>

<database\_name>

<table\_name>

<filter>expression</filter>

<table\_name>

</database\_name>

</databases>

</user\_name>

<!-- Other users settings -->

</users>

Copy

### 2.15.1 user\_name/password

Пароль можно указать в текстовом виде или в виде SHA256 (шестнадцатеричный формат):

Чтобы назначить пароль в текстовом виде (**не рекомендуем**), поместите его в элемент password.

Например, `<password\_sha256\_hex>65e84be33532fb784c48129675f9eff3a682b27168c0ea744b2cf58ee02337c5

</password\_sha256\_hex>`.

Пример создания пароля в командной строке:

```

PASSWORD=$(base64 < /dev/urandom | head -c8); echo "$PASSWORD"; echo -n "$PASSWORD" | sha256sum | tr -d '-'

```

Первая строка результата — пароль. Вторая строка — соответствующий ему хэш SHA256.

Copy

Для совместимости с клиентами MySQL, пароль можно задать с помощью двойного хэша SHA1, поместив его в элемент ***password\_double\_sha1\_hex***.

Например, `<password\_double\_sha1\_hex>08b4a0f1de6ad37da17359e592c8d74788a83eb0</password\_double\_sha1\_hex>`.

Пример создания пароля в командной строке:

```

PASSWORD=$(base64 < /dev/urandom | head -c8); echo "$PASSWORD"; echo -n "$PASSWORD" | sha1sum |

tr -d '-' | xxd -r -p | sha1sum | tr -d '-'

```

Первая строка результата — пароль. Вторая строка — соответствующий ему двойной хэш SHA1.

Copy

### 2.15.2 access\_management

Включает или выключает SQL-ориентированное управление доступом для пользователя.

Возможные значения:

0 – выключено.

1 – включено.

Значение по умолчанию: 0.

### 2.15.3 user\_name/networks

Список сетей, из которых пользователь может подключиться к серверу RT.WideStore.

Каждый элемент списка имеет одну из следующих форм:

<ip> — IP-адрес или маска подсети.

Примеры: `213.180.204.3`, `10.0.0.1/8`, `10.0.0.1/255.255.255.0`, `2a02:6b8::3`,

`2a02:6b8::3/64`, `2a02:6b8::3/ffff:ffff:ffff:ffff::`.

Copy

<host> — Имя хоста.

Пример: `example01.host.ru`.

Для проверки доступа выполняется DNS-запрос, и все возвращенные IP-адреса сравниваются с

адресом клиента.

Copy

<host\_regexp> — Регулярное выражение для имен хостов.

Пример, `^example\d\d-\d\d-\d\.host\.ru$`

Для проверки доступа выполняется [DNS запрос PTR](https://en.wikipedia.org/wiki/Reverse\_DNS\_lookup)

для адреса клиента, а затем применяется заданное регулярное выражение. Затем, для результатов запроса

PTR выполняется другой DNS-запрос и все полученные адреса сравниваются с адресом клиента.

Рекомендуем завершать регулярное выражение символом $.

Copy

Все результаты DNS-запросов кэшируются до перезапуска сервера.

Примеры:

Чтобы открыть доступ пользователю из любой сети, укажите:

<ip>::/0</ip>

Copy

***Внимание****: Открывать доступ из любой сети небезопасно, если у вас нет правильно настроенного брандмауэра или сервер не отключен от интернета.*

Чтобы открыть только локальный доступ, укажите:

<ip>::1</ip>

<ip>127.0.0.1</ip>

Copy

### 2.15.4 user\_name/profile

Пользователю можно назначить профиль настроек. Профили настроек конфигурируются в отдельной секции файла users.xml. Подробнее читайте в [п. 2.13](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.13%20%D0%9F%D1%80%D0%BE%D1%84%D0%B8%D0%BB%D0%B8%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B5%D0%BA).

### 2.15.5 user\_name/quota

Квотирование позволяет отслеживать или ограничивать использование ресурсов в течение определённого периода времени. Квоты настраиваются в разделе ***quotas*** конфигурационного файла ***users.xml***.

Пользователю можно назначить квоты. Подробное описание настройки квот смотрите в [п. 2.6](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_WideStore/RT_WideStore_Admin_Instructions#:~:text=2.6%20%D0%9A%D0%B2%D0%BE%D1%82%D1%8B).

### 2.15.6 user\_name/databases

В этом разделе вы можете ограничить выдачу RT.WideStore запросами SELECT для конкретного пользователя, таким образом реализуя базовую защиту на уровне строк.

Пример:

Следующая конфигурация задаёт, что пользователь user1 в результате запросов SELECT может получать только те строки table1, в которых значение поля id равно 1000.

<user1>

<databases>

<database\_name>

<table1>

<filter>id = 1000</filter>

</table1>

</database\_name>

</databases>

</user1>

Copy

Элемент ***filter*** содержать любое выражение, возвращающее значение типа UInt8. Обычно он содержит сравнения и логические операторы. Строки ***database\_name.table1***, для которых фильтр возвращает 0 не выдаются пользователю. Фильтрация несовместима с операциями PREWHERE и отключает оптимизацию WHERE→PREWHERE.