MyDB Сервер для MySQL

Руководство пользователя для версии 8.4.3-3.1 (2025-01-28)

Команда технической документации *MyDB*

docs@mydb.ru

Оглавление

docs@mydb.ru

1. MyDB Сервер для MySQL 8.4 — Документация

Эта документация относится к последней версии: MyDB Сервер для MySQL 8.4.3-3.1 (Примечания к выпуску).

MyDB Сервер для MySQL — это свободно доступная, распространяемая с открытым исходным кодом, полностью совместимая и расширенная замена любой базы данных MySQL. Продукт обеспечивает превосходную и оптимизированную производительность, более высокую масштабируемость и доступность, а также улучшенные функции резервного копирования, мониторинга и управления.

Предприятия доверяют MyDB Сервер для MySQL, поскольку он обеспечивает лучшую производительность, надёжность и масштабируемость для самых требовательных рабочих нагрузок.

Вы можете использовать Краткое руководство, чтобы начать использовать MyDB Сервер для MySQL.

Последнее обновление: 2024-04-30

2. Примечания к выпускам

2.1 Указатель примечаний к выпускам MyDB Сервер для MySQL 8.0

Последнее обновление: 2024-04-30

2.2 MyDB Сервер для MySQL 8.4.0-1.1 (05.10.2024)

Начните работу с Краткого руководства по MyDB Сервер для MySQL.

MyDB Сервер для MySQL 8.4.0-1.1 включает в себя все функции и исправления ошибок, доступные в MySQL 8.4.0, все функции и исправления ошибок, доступные в Percona Server for MySQL 8.4.0-1, а также функции и исправления, разработанные в компании MyDB и описанные в этом разделе.

2.2.1 Основные моменты выпуска

- Удалена поддержка модуля телеметрии, который собирал анонимные данные об установках Percona Server и отправлял в компанию Percona

Исправления ошибок в MyDB Сервер для MySQL 8.4.0-1.1

Новые функции в MyDB Сервер для MySQL 8.4.0-1.1

Улучшения и исправления ошибок, включённые в Percona Server for MySQL 8.4.0-1:

Полный список исправлений ошибок и изменений можно найти в Примечаниях к выпуску Percone Server for MySQL 8.4.0-1.

Улучшения и исправления ошибок, включённые в MySQL 8.4.0

Полный список исправлений ошибок и изменений можно найти в Примечаниях к выпуску MySQL 8.4.0.

Последнее обновление: 2025-01-28

2.3 MyDB Сервер для MySQL 8.4.3-3.1 (28.01.2025)

Начните работу с Краткого руководства по MyDB Сервер для MySQL.

MyDB Сервер для MySQL 8.4.3-3.1 включает в себя все функции и исправления ошибок, доступные в MySQL 8.4.3 и все функции и исправления ошибок, доступные в Percona Server for MySQL 8.4.3-3.

2.3.1 Основные моменты выпуска

- Мог возвращаться пустой результат при выборке из пространственного (`SPATIAL`) индекса со столбцом, содержащим идентификатор пространственной ссылки (`SRID`). Использование `FORCE INDEX` для сканирования этого индекса вызывало аварийную остановку сервера.
- Улучшена стабильность для установок с более 8000 таблиц.
- Ускорен старт сервера во время востановления после сбоя на установках с большим количеством табличных пространств (ошибка #110402)
- Изменения в MySQL 8.0.33 приводили к снижению производительности для запросов. Эти участки кода были оптимизированы, чтобы поднять производительность на прежний уровень.
- Повышена надёжность и стабильность алгоритма `INSTANT` для `ALTER TABLE`.
- Импорт таблицы, созданной с другим значением `sql_mode`, иногда приводил к несоответствиям схемы, что создавало риск повреждения данных во вторичных индексах. Исправление включает проверки целостности импортированного табличного пространства.
- Исправлено снижение производительности при перестроении таблиц с вторичными индексами, при котором создавалось больше операций ввода-вывода файлов по сравнению с MySQL 8.0.26.

Улучшения и исправления ошибок, включённые в Percona Server for MySQL 8.4.3-3:

Полный список исправлений ошибок и изменений можно найти в Примечаниях к выпуску Percone Server for MySQL 8.4.3-3.

Улучшения и исправления ошибок, включённые в MySQL 8.4.3

Полный список исправлений ошибок и изменений можно найти в Примечаниях к выпуску MySQL 8.4.3.

Последнее обновление: 2025-03-25

3. Инструкции по началу работы

3.1 Обзор

MyDB Сервер для MySQL — это свободно доступная, распространяемая с открытым исходным кодом, полностью совместимая и расширенная замена любой базы данных MySQL. Продукт обеспечивает превосходную и оптимизированную производительность, более высокую масштабируемость и доступность, а также улучшенные функции резервного копирования, мониторинга и управления.

Чтобы вы могли быстро начать работу с MyDB Сервер для MySQL, в этом кратком руководстве основное внимание уделяется использованию Docker.

Вы можете изучить альтернативные варианты установки в разделе Установка документации MyDB Сервер для MySQL.

3.1.1 Цель инструкций

Этот документ проведет вас через процесс первоначальной настройки, включая установку пароля администратора, а также создание базы данных.

3.1.2 Шаги для начинающих пользователей

Следующие разделы руководства проведут вас через процесс установки и работы с базой данных для разработчика. Выберите метод установки, который лучше всего подходит для вашей среды.

3.1.3 Следующие шаги

Запустите MyDB Сервер для MySQL 8.4 в контейнере Docker →

Выберите следующие шаги →

Последнее обновление: 2024-04-30

3.2 Запуск MyDB Сервер для MySQL в контейнере Docker

В следующих инструкциях вы можете называть любые элементы в соответствии со стандартами вашей организации или использовать структуру и данные вашей таблицы. Если вы это сделаете, результаты могут отличаться от примеров.

3.2.1 Предварительные условия

- Docker Engine установлен и работает.
- Стабильное подключение к Интернету
- Базовое понимание интерфейса командной строки (CLI).

Всегда адаптируйте команды и конфигурации к вашей конкретной среде и требованиям безопасности.

3.2.2 Запускаем Docker-контейнер

Чтобы использовать команду `docker run`, укажите имя или идентификатор образа, который вы хотите использовать, и, при необходимости, некоторые флаги и аргументы, которые изменяют поведение контейнера. Команда имеет следующие параметры:

Параметр	Описание
`-d`	Запускает контейнер в режиме `detached`, позволяя контейнеру работать в фоновом режиме.
`-p 3306:3306`	Отображает порт MySQL контейнера (3306) на тот же порт вашего хоста, обеспечивая внешний доступ.
`--name mydb`	Предоставляет контейнеру осмысленное имя. Если вы не укажете эту опцию, Docker добавит случайное имя.
`-e MYSQL_ROOT_PASSWORD=secret`	Добавляет переменную среды и меняет пароль по умолчанию.
`--v myvol:/var/lib/mysql`	Подключает каталог хоста (myvol) в качестве тома данных контейнера, обеспечивая постоянное хранилище базы данных между жизненными циклами контейнера.
`registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:8.4.3-3.1`	Изображение с тегом (8.4.3-3.1) для указания конкретного релиза.

Для доступа к базе данных необходимо предоставить хотя бы одну переменную среды, например `MYSQL_ROOT_PASSWORD`, `MYSQL_DATABASE`, `MYSQL_USER` и `MYSQL_PASSWORD`, иначе экземпляр откажется инициализироваться.

При необходимости вы можете заменить пароль `secret` на более надежный пароль.

В этом документе мы используем тег `8.4.3-3.1`. В Docker тег — это метка, присвоенная образу и используемая для поддержки различных версий образа. Если мы не добавили тег, Docker

использует `latest` в качестве тега по умолчанию и загружает последний образ `mydb` в реестре контейнеров Gitflic .

Чтобы запустить версию MyDB Сервер для MySQL в контейнере Docker на архитектуре ARM64, используйте тег `8.4.3-3.1-aarch64` вместо `8.4.3-3.1`.

```
$ docker run -d -p 3306:3306 --name mydb \
-e MYSQL_ROOT_PASSWORD=secret \
-v myvol:/var/lib/mysql \
registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:8.4.3-3.1
```

Ожидаемый результат

```
Unable to find image 'registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:8.4.3-3.1'
locally
8.4.3-3.1: Pulling from project/mydb-ru/mydb-server/mydb
a480a496ba95: Pull complete
d8ed145ee679: Pull complete
3cbe6fb8cf6a: Pull complete
86ebfbf5fab7: Pull complete
99146b7d5d41: Pull complete
2d346ef8e515: Pull complete
e806e7182b53: Pull complete
196f11ae5ace: Pull complete
c7ff140a6d79: Pull complete
70b17cc4c4b9: Pull complete
Digest: sha256:0ca2ddb4021385d419a1fa0be9511b65c5609246b970204f357cd8c144171d52
Status: Downloaded newer image for registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:
8.4.3-3.1
5d10afc763795d7807b7d79bcfdb5721f8474475b66f59908e7f5cd7de79db78
```

3.2.3 Подключаемся к экземпляру базы данных

Чтобы подключиться к базе данных MySQL в контейнере, используйте команду `docker exec` с командой подключения экземпляра базы данных. Вы должны знать имя или идентификатор контейнера, на котором работает сервер базы данных, а также учётную запись базы данных.

Команда `docker exec` запускает указанную команду в работающем контейнере. Команда подключения экземпляра базы данных подключается к серверу MySQL с использованием имени пользователя и пароля.

Для этого примера используются следующие аргументы:

Вариант	Описание
`-it`	Использовать интерактивный режим и выделить псевдотерминал
`mydb`	Имя работающего контейнера
`mysql`	Имя клиента командной строки
`-u`	Указывает учетную запись пользователя, используемую для подключения
`-p`	Указывает пароль при подключении

Вы должны ввести пароль, когда сервер предложит вам это сделать.

Пример подключения к экземпляру базы данных

```
$ docker exec -it mydb mysql -uroot -p
```

Вам будет предложено ввести пароль, в качестве которого в нашем примере служит слово `secret`. Если вы изменили пароль, используйте свой пароль. Во время ввода вы не увидите никаких символов.

```
Enter password:
```

Вы должны увидеть следующий результат.

Ожидаемый результат

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.4.3-3.1 MyDB Server (GPL), Release '1.1', Revision '723681b1'

Copyright (c) 2024-2024 MyDB LLC and/or its affiliates
Copyright (c) 2009-2024 Percona LLC and/or its affiliates
Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.


mysql>
```

3.2.4 Создаем базу данных

Чтобы создать базу данных, используйте команду `CREATE DATABASE`. При желании вы можете указать в команде набор символов и параметры сортировки для базы данных. После создания базы данных выберите ее с помощью команды `USE` или опции `-D` в клиенте MySQL.

```
mysql> CREATE DATABASE mydb;
```

Ожидаемый результат

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> USE mydb;
```

Ожидаемый результат

```
Database changed
```

3.2.5 Создаем таблицу

Создайте таблицу, используя команду `CREATE TABLE`. Для каждого столбца вы можете укзать тип данных, ограничения и использовать ключевое слово `DEFAULT` для столбцов со значениями по умолчанию. Вы также можете указать индексы и другие параметры создаваемой таблицы.

```
mysql> CREATE TABLE employees (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    name VARCHAR(255) DEFAULT NULL,
    email VARCHAR(255) DEFAULT NULL,
    country VARCHAR(100) DEFAULT NULL,
    PRIMARY KEY (id));
```

Ожидаемый результат

```
Query OK, 0 rows affected (0,03 sec)
```

3.2.6 Заполняем таблицу данными

Вставьте данные в таблицу с помощью SQL-команды `INSERT`. Эта команда может добавлять несколько записей в таблицу.

```
mysql> INSERT INTO employees (name, email, country)
VALUES
    ("Erasmus Richardson","posuere.cubilia.curae@outlook.net","England"),
    ("Jenna French","rhoncus.donec@hotmail.couk","Canada"),
    ("Alfred Dejesus","interdum@aol.org","Austria"),
    ("Hamilton Puckett","dapibus.quam@outlook.com","Canada"),
    ("Michal Brzezinski","magna@icloud.pl","Poland"),
    ("Zofia Lis","zofial00@hotmail.pl","Poland"),
    ("Aisha Yakubu","ayakubu80@outlook.com","Nigeria"),
    ("Miguel Cardenas","euismod@yahoo.com","Peru"),
    ("Luke Jansen","nibh@hotmail.edu","Netherlands"),
    ("Roger Pettersen","nunc@protonmail.no","Norway");
```

Ожидаемый результат

```
Query OK, 10 rows affected (0,05 sec)
Записей: 10  Дубликатов: 0  Предупреждений: 0
```

3.2.7 Запускаем запрос командой `SELECT`

Запросы в виде команд `SELECT` извлекают данные из одной или нескольких таблиц на основе заданных критериев. Это наиболее распространенный тип запросов, который можно использовать для различных целей, например для отображения, фильтрации, сортировки, агрегирования или объединения данных. Команды `SELECT` не изменяют данные в базе данных, но могут повлиять на производительность, если запрос включает большие или сложные наборы данных.

```
mysql> SELECT id, name, email, country FROM employees WHERE country = 'Poland';
```

Ожидаемый результат

```
+----+-------------------+---------------------+---------+
| id | name              | email               | country |
+----+-------------------+---------------------+---------+
|  5 | Michal Brzezinski | magna@icloud.pl     | Poland  |
|  6 | Zofia Lis         | zofial00@hotmail.pl | Poland  |
+----+-------------------+---------------------+---------+
2 rows in set (0.00 sec)
```

3.2.8 Запускаем команду `UPDATE`

Команды `UPDATE` изменяют существующие данные в таблице. Они используются для изменения или исправления информации, хранящейся в базе данных. Команды `UPDATE` могут обновлять один или несколько столбцов и строк одновременно, в зависимости от указанных условий. Они также могут потерпеть неудачу, если нарушат какие-либо ограничения или правила, определенные в таблице.

Пример команды `UPDATE` с последующим запуском `SELECT` с указанием `WHERE` для проверки обновления.

```
mysql> UPDATE employees SET name = 'Zofia Niemec' WHERE id = 6;
```

Ожидаемый результат

```
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> SELECT name FROM employees WHERE id = 6;
```

Ожидаемый результат

```
+--------------+
| name         |
+--------------+
| Zofia Niemec |
+--------------+
1 row in set (0.00 sec)
```

3.2.9 Запускаем команду `INSERT`

Команды `INSERT` добавляют новые данные в таблицу. Они используются для заполнения базы данных новой информацией. Команды `INSERT` могут вставлять одну или несколько строк одновременно, в зависимости от синтаксиса. Запрос может завершиться неудачей, если он нарушает какие-либо ограничения или правила, определенные для таблицы, например первичные ключи, внешние ключи, уникальные индексы или триггеры.

Вставьте строку в таблицу, а затем запустите SELECT с указанием `WHERE`, чтобы убедиться, что запись была вставлена.

```
mysql> INSERT INTO employees (name, email, country)
VALUES
("Kenzo Sasaki","KenSasaki@outlook.com","Japan");
```

Ожидаемый результат

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT id, name, email, country FROM employees WHERE id = 11;
```

Ожидаемый результат

```
+----+--------------+-----------------------+---------+
| id | name         | email                 | country |
+----+--------------+-----------------------+---------+
| 11 | Kenzo Sasaki | KenSasaki@outlook.com | Japan   |
+----+--------------+-----------------------+---------+
1 row in set (0.00 sec)
```

3.2.10 Запускаем команду `DELETE`

Команды `DELETE` удаляют существующие данные из таблицы. Они используются для очистки информации, которая больше не нужна или неактуальна в базе данных. Запросы `DELETE` могут удалять одну или несколько строк одновременно, в зависимости от указанных условий.

Они также могут инициировать каскадное удаление связанных таблиц, если применяются ограничения внешнего ключа.

Удалите строку в таблице и запустите `SELECT` с указанием `WHERE`, чтобы подтвердить удаление.

```
mysql> DELETE FROM employees WHERE id >= 11;
```

Ожидаемый результат

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT id, name, email, country FROM employees WHERE id > 10;
```

Ожидаемый результат

```
Empty set (0.00 sec)
```

3.2.11 Завершение работы

Дальнейшие шаги выполняют следующие действия:

- Выход из командной оболочки MySQL и Docker-контейнера.
- Удаление Docker-контейнера и Docker-образа.
- Удаление тома Docker.

Подробное описание шагов:

1. Чтобы выйти из оболочки командного клиента MySQL, мы используем `exit`. Вы также можете использовать команды `\q` или `quit`. Выполнение этих команд также закрывает соединение.

    • Пример выхода из командной оболочки MySQL и закрытия соединения.

    ```
    mysql> exit
    ```

    Ожидаемый результат

    ```
    Bye
    ```

2. Возможно, вы захотите удалить Docker-контейнер и образ, если они больше не нужны, или чтобы освободить место на диске. Чтобы удалить контейнер Docker, используйте команду `docker rm` и укажите идентификатор или имя контейнера (в нашем примере это `mydb`). Чтобы удалить образ Docker, используйте команду `docker rmi` и укажите идентификатор

или имя изображения + тег (в нашем примере это `registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:8.4.3-3.1`)

- Пример удаления Docker-контейнера.

```
$ docker container rm mydb -f
```

Ожидаемый результат

```
mydb
```

- Пример удаления образа Docker. Если вы используете версию MyDB Сервер для ARM64, отредактируйте команду Docker, указав тег `8.4.3-3.1-aarch64`. Это изменение меняет команду на `docker image rmi mydbru/mydb:8.4.3-3.1-aarch64`

```
$ docker image rmi registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:
8.4.3-3.1
```

Ожидаемый результат

```
Untagged: registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:8.4.3-3.1
Untagged: registry.gitflic.ru/project/mydb-ru/mydb-server/
mydb@sha256:0ca2ddb4021385d419a1fa0be9511b65c5609246b970204f357cd8c144171d52
Deleted: sha256:25306cf85b900d9242ef730fc67e4c57df3f92ccf09ead453c9ef5b46ffe11ed
Deleted: sha256:500bc2200c437bce438a3da0e1198cf0307d46ef2e0fca73dacdaae8ee9191d3
Deleted: sha256:ff0d7308fd9681fbb58b994480795d6bc4b0322c5a818def55b8badb04942b45
Deleted: sha256:703128937b40f7d2d8a77c5c11242e964a514b4e7f377b89d28b2c7c18dfa927
Deleted: sha256:99d4adcaf6838f31fe4a7da1910727d082672ebefd05992019e77376a4687540
Deleted: sha256:8072775536fec6061816c612716d6620c52711d7aa702f1012b175b9a3327b24
Deleted: sha256:afede41a81b020c008b988fe2f7747eaeea16036ddb6bd47e2fa7f4b8a759a38
Deleted: sha256:9c8f0e05344e5f3cb114c8303743bc2f2f83e9af1b205d62c934de330c35c697
Deleted: sha256:7b67d8b006c83d6985140a980b598fbb66f87b87a635a182207ce0781a2e60f3
Deleted: sha256:90e2f9f206af0aaef7f6ecd707dad85b7c686f5fe53318acfbafe1b216766046
Deleted: sha256:98b5f35ea9d3eca6ed1881b5fe5d1e02024e1450822879e4c13bb48c9386d0ad
```

3. Удалите том Docker, если контейнер не использует этот том и он вам больше не нужен.

- Пример удаления тома Docker.

```
$ docker volume rm myvol
```

Ожидаемый результат

```
myvol
```

3.2.12 Устранение неисправностей

- Отказ в подключении: убедитесь, что Docker запущен и контейнер активен. Убедитесь, что порт 3306 доступен по IP-адресу контейнера.
- Неправильные учетные данные: перепроверьте пароль `root`, который вы установили во время запуска контейнера.
- Потеря данных. Всегда регулярно создавайте резервные копии данных за пределами тома контейнера.

3.2.13 Меры безопасности

- Надежные пароли: используйте сложные, уникальные пароли для пользователя `root` и любых дополнительных учетных записей, созданных в контейнере. Буквенно-цифровой пароль должен содержать не менее 12 символов. Пароль должен состоять из прописных и строчных букв, цифр и символов.
- Сетевые ограничения: Ограничьте сетевой доступ к контейнеру, ограничив доступ только авторизованными IP-адресами с помощью правил брандмауэра.
- Периодические обновления: регулярно обновляйте образ MyDB Сервер и Docker Engine для устранения известных уязвимостей.
- Шифрование данных: рассмотрите возможность шифрования каталога данных внутри тома контейнера для дополнительного уровня безопасности.
- Мониторинг журналов: активно отслеживайте журналы контейнеров на предмет подозрительной активности или ошибок.

Помните, что ответственное управление контейнерами и надежные методы обеспечения безопасности имеют решающее значение для защиты вашего развертывания MyDB. Следуя этим рекомендациям, вы сможете воспользоваться преимуществами Docker и MyDB Сервер, уделяя при этом внимание целостности и безопасности ваших данных.

3.2.14 Следующий шаг

Выберите следующие шаги →

Последнее обновление: 2024-10-18

docs@mydb.ru

3.3 Следующие шаги

Создав базу данных и выполнив запросы, вы сделали первые шаги, чтобы стать начинающим разработчиком MySQL. Однако вам еще предстоит многому научиться и попрактиковаться, чтобы улучшить свои навыки и знания. Дальнейшие шаги, которые вы можете предпринять:

- Ознакомьтесь с различными типами данных, такими как целые числа, строки, даты и логические значения, и выберите те, которые подходят для ваших данных.
- Создавайте и используйте индексы для оптимизации производительности ваших запросов и снижения нагрузки на сервер базы данных.
- Объединяйте данные из нескольких таблиц и источников, используя соединения, подзапросы и объединения.
- Используйте функции, процедуры, триггеры и представления для инкапсуляции логики, автоматизации задач и создания повторно используемых компонентов.
- Используйте транзакции, блокировки и уровни изоляции для обеспечения целостности и согласованности данных в параллельных операциях.
- Используйте инструменты резервного копирования и восстановления, чтобы защитить ваши данные от потери или повреждения.
- Используйте функции безопасности, такие как пользователи, роли, привилегии и шифрование, чтобы защитить ваши данные от несанкционированного доступа или изменения.
- Используйте инструменты отладки и тестирования, такие как журналы, сообщения об ошибках, точки останова и контрольные проверки, чтобы выявлять и исправлять ошибки в вашем коде или запросах.
- Используйте инструменты документации, такие как комментарии, диаграммы, схемы и руководства, чтобы объяснять и документировать свой код или запросы.

Эти задачи расширят ваши знания и навыки использования MyDB Сервер для MySQL и позволят вам стать более уверенными и опытными в разработке приложений баз данных.

Дополнительную информацию можно найти в документации MyDB Сервер для MySQL.

Последнее обновление: 2024-04-30

4. Установка

4.1 Установка MyDB Сервер для MySQL

Перед установкой ознакомьтесь с Примечаниями к выпуску MyDB Сервер для MySQL 8.4.

Чтобы быстро начать работу, воспользуйтесь Инструкциями по началу работы . Вы можете использовать инструкции для Docker или установки с помощью менеджера пакетов.

Последнее обновление: 2024-04-30

4.2 Перед началом работы

4.2.1 Инструкции по загрузке продуктов MyDB

Выберите программное обеспечение

Для выбора программного обеспечения выполните следующие действия:

1. Откройте страницу Загрузки продуктов MyDB.
2. Выберите каталог с программное обеспечение MyDB, например `mydb-server-8.4`.
3. Выберите каталог с версией продукта, например `mydb-server-8.4.3-3.1`
4. Выберите каталог `binary` для загрузки бинарных пакетов или `source` для загрузки пакетов с исходным кодом.
5. В случае бинарных пакетов, выберите каталог, соответствующий нужной ОС.

Самый простой способ — загрузить все пакеты для выбранной ОС.

Последнее обновление: 2024-04-30

4.2.2 Используйте mydb-release

Пользователь MyDB Сервер для MySQL отдает приоритет эффективности и надежности. Установка программного обеспечения напрямую предлагает базовое решение, но для повышения контроля, удобства и безопасности используйте репозитории программного обеспечения MyDB и инструмент mydb-release . Использование инструмента `mydb-release` обеспечивает упрощенную, безопасную и эффективную установку и работу MyDB Сервер для MySQL.

Простое управление репозиторием

`mydb-release` позволяет с помощью одной команды настраить вашу систему для доступа к официальным репозиториям MyDB, устраняя необходимость вручную определять и добавлять отдельные источники. При добавлении нового продукта MyDB `mydb-release` выполняет функции простой дополнительной настройки репозитория.

Целевые репозитории

Выбирайте из различных репозиториев, включая конкретные версии, каналы стабильности (стабильные, тестовые, экспериментальные) и отдельные продукты MyDB. Такой детальный контроль гарантирует получение точной желаемой конфигурации программного обеспечения.

Автоматические обновления

Воспользуйтесь преимуществами автоматического обновления списков репозиториев, гарантируя доступ к новейшим пакетам по мере их выпуска MyDB.

Оптимизированное управление пакетами

Используйте собственный менеджер пакетов вашей системы (подходит для Debian/Ubuntu, yum для Red Hat/CentOS) для установки MyDB Сервер для MySQL и связанных компонентов. Этот знакомый интерфейс упрощает процесс, устраняя необходимость загружать отдельные пакеты и управлять ими.

Разрешение зависимостей

`mydb-release` учитывает зависимости пакетов, гарантируя автоматическую установку всех необходимых компонентов вместе с MyDB Сервер для MySQL. Больше не нужно бороться с отсутствующими библиотеками или проблемами совместимости.

Согласованная конфигурация

`mydb-release` гарантирует согласованность версий пакетов в ваших системах за счет поддержки централизованных репозиториев. Эта функция упрощает управление конфигурацией и снижает риск несогласованности, возникающей при установке вручную.

Проверка ключа GPG

`mydb-release` использует ключи GPG для криптографической проверки подлинности и целостности пакетов, загруженных из репозиториев MyDB. Эта мера безопасности защищает вас от вредоносного программного обеспечения или поддельных пакетов, гарантируя, что устанавливаемое вами программное обеспечение является подлинным и безопасным.

Подписанные пакеты

Все пакеты в репозиториях MyDB имеют цифровую подпись, что дополнительно повышает безопасность и предотвращает несанкционированный доступ. Этот дополнительный уровень защиты обеспечивает уверенность в том, что установленное вами программное обеспечение не было скомпрометировано.

Регулярные обновления безопасности

MyDB активно выпускает обновления безопасности для своего программного обеспечения. Используя `mydb-release`, вы получаете доступ к этим критически важным обновлениям, когда они становятся доступными, что помогает вам поддерживать безопасную и стабильную среду базы данных.

Последнее обновление: 2024-04-30

4.3 Использование бинарных архивов

4.3.1 Установка MyDB Сервер для MySQL 8.4 из бинарного архива.

Бинарный архив содержит группу файлов, включая исходный код, объединенных в один файл с помощью команды tar и сжатых с помощью gzip.

См. список бинарных архивов для MyDB Сервер для MySQL , чтобы выбрать подходящий архив для вашей среды.

Вы можете загрузить бинарные архивы из раздела «Linux — Любой» на странице загрузки.

Загрузите и извлеките правильный бинарный архив. Например, для Debian 12:

```
$ wget https://downloads.mydb.ru/downloads/mydb-server-8.4/mydb-
server-8.4.3-3.1/binary/tarball/mydb-server-8.4.3-3.1-
Linux.x86_64.glibc2.36.tar.gz
```

Последнее обновление: 2024-04-30

4.3.2 Соответствие имён бинарных архивов и версии MyDB Сервер для MySQL

Бинарные архивы для Linux именуются по версии `glibc2`. Вы можете узнать эту версию в своей операционной системе с помощью следующей команды:

```
$ ldd --version
```

Ожидаемый результат

```
ldd (Ubuntu GLIBC 2.35-0ubuntu3.1) 2.35
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.
```

Если версия glibc вашей операционной системы не указана в списке ниже, значит, этот выпуск MyDB Сервер для MySQL не поддерживает вашу операционную систему.

Организация имен файлов в бинарных архиве

Ниже перечислены платформы и соответствующие им полные имена бинарных файлов.

Платформа	Имя архива MyDB Сервер для MySQL	версия glibc2
Ubuntu 24.04	mydb-server-8.4.3-3.1-Linux.x86_64.glibc2.39.tar.gz	glibc2.35
Ubuntu 22.04	mydb-server-8.4.3-3.1-Linux.x86_64.glibc2.35.tar.gz	glibc2.35
Ubuntu 20.04	mydb-server-8.4.3-3.1-Linux.x86_64.glibc2.31.tar.gz	glibc2.31
Red Hat Enterprise 9	mydb-server-8.4.3-3.1-Linux.x86_64.glibc2.34.tar.gz	glibc2.34
Red Hat Enterprise 8	mydb-server-8.4.3-3.1-Linux.x86_64.glibc2.28.tar.gz	glibc2.28

Типы файлов следующие:

Тип	Имя	Описание
Полный	mydb-server-<номер-версии>-Linux.x86_64.<версия-glibc2>.tar.gz	Содержит все доступные файлы
Минимальный	mydb-server-<номер-версии>-Linux.x86_64.<версия-glibc2>-minimal.tar.gz	Содержит бинарные файлы и библиотеки
Отладочный	mydb-server-<номер-версии>-Linux.x86_64.<версия-glibc2>-debug.tar.gz	Содержит минимальные файлы сборки и тестовые файлы, а также символы отладки

Последнее обновление: 2024-04-30

4.4 Сборка из исходников

4.4.1 Установка MyDB Сервер для MySQL из архива с исходным кодом

Загрузите и распакуйте архив с исходным кодом.

Например:

```
$ wget https://downloads.mydb.ru/downloads/mydb-server-8.4/mydb-
server-8.4.3-3.1/source/mydb-server-8.4.3-3.1.tar.gz
$ tar xfz mydb-server-8.4.3-3.1.tar.gz
```

Чтобы завершить установку, следуйте инструкциям в разделе Сборка MyDB Сервер для MySQL из исходного кода.

Последнее обновление: 2024-04-30

4.4.2 Сборка MyDB Сервер для MySQL 8.0 из исходного кода

Следующие инструкции описывают сборку и установку MyDB Сервер для MySQL 8.0 из исходного кода.

Получение исходного кода из Git

MyDB использует Gitflic для разработки открытых проектов. Чтобы собрать последнюю версию MyDB Сервер для MySQL из исходного кода, вам понадобится установленный в вашей системе `git`.

Вы можете получить последнюю версию дерева исходного кода MyDB Сервер для MySQL 8.0 с помощью следующих команд:

```
$ git clone https://giflic.ru/mydb/mydb-server.git
$ cd mydb-server
$ git checkout 8.0
$ git submodule init
$ git submodule update
```

Если вы собираетесь вносить изменения в MyDB Сервер для MySQL 8.0 и хотите распространять результаты этой работы, вы можете создать новый архив с исходным кодом – точно так же, как мы это делаем для релиза:

```
$ cmake .
$ make dist
```

После получения исходного репозитория или извлечения исходного кода из архива (полученного от MyDB или созданного вами самостоятельно), вам нужно будет настроить и собрать MyDB Сервер для MySQL.

Сначала запустите CMake, чтобы настроить сборку. Здесь можно указать все обычные параметры сборки аналогично сборке MySQL. В зависимости от параметров, с которыми вы хотите скомпилировать MyDB Сервер для MySQL, вам могут понадобиться другие библиотеки, установленные в вашей системе. Вот пример, использующий конфигурационные параметры, аналогичные параметрам, которые MyDB использует для создания собственных сборок:

```
$ cmake . -DCMAKE_BUILD_TYPE=RelWithDebInfo -DBUILD_CONFIG=mysql_release -
DFEATURE_SET=community
```

Сборка проект из исходного кода

Теперь соберите проект с помощью `make`:

```
$ make
```

И установите:

```
$ make install
```

По завершении этого шага MyDB Сервер для MySQL 8.0 будет установлен в вашей системе.

Последнее обновление: 2024-04-30

4.5 Docker

4.5.1 Запуск MyDB Сервер для MySQL в контейнере Docker

MyDB Сервер для *MySQL* имеет официальный образ Docker, размещенный в реестре контейнеров Gitflic. Если вам нужна последняя версия, используйте тег `latest`. Вы можете указать конкретную версию, используя фильтр тегов Docker для версий 8.4.

Убедитесь, что вы используете последнюю версию Docker. Версии, установленные из репозиториев `apt` и `yum` могут быть устаревшими и вызывать ошибки. Установите самую свежую версию Docker в вашей системе.

Запуск фонового контейнера

Вы можете запустить фоновый контейнер с опцией `--detached` или `-d`, которая запускает контейнер в режиме "detached". В этом режиме контейнер завершает работу, когда завершается корневой процесс, используемый для запуска контейнера.

В следующем примере контейнер с именем `mydb` запускается с последней версией *MyDB* Сервер для *MySQL* 8.4. Это действие также создает пользователя `root` и использует `root` в качестве пароля. Обратите внимание, что `root` не является безопасным паролем.

```
$ docker run -d \
  --name mydb \
  -e MYSQL_ROOT_PASSWORD=root \
  registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:8.4
```

Ожидаемый результат

```
Unable to find image 'registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:8.4' locally
8.4: Pulling from project/mydb-ru/mydb-server/mydb
```

По умолчанию Docker извлекает образ из реестра контейнеров, если он недоступен локально.

Чтобы просмотреть журналы контейнера, используйте следующую команду:

```
$ docker logs mydb --follow
```

Ожидаемый результат

```
2024-10-17T16:19:30.048168Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld
8.4.3-3.1) starting as process 1
2024-10-17T16:19:30.164911Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has
started.
...
2024-10-17T16:19:31.327168Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for
connections. Version: '8.4.3-3.1'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306  MyDB
Server (GPL), Release '1.1', Revision '723681b1'.
```

Вы можете получить доступ к серверу, когда увидите в журнале информацию о готовности к подключениям (“ready for connections”).

Передача параметров MyDB

Вы можете передать параметры MyDB с помощью команды `docker run`. Например, следующая команда использует UTF-8 в качестве настройки по умолчанию для набора символов и параметров сортировки для всех баз данных:

```
$ docker run -d \
 --name mydb \
 -e MYSQL_ROOT_PASSWORD=root \
 registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:8.4 \
    --character-set-server=utf8 \
 --collation-server=utf8_general_ci
```

Доступ к контейнеру MyDB

Команда `docker exec` позволяет запустить интерпретатор командной строки внутри контейнера. С этой командой часто указывают опции `-it`, которые используют стандартный ввод как интерактивный терминал.

Пример доступа к фоновому контейнеру:

```
$ docker exec -it mydb /bin/bash
```

Если вам нужно устранить неполадки, журнал ошибок можно найти в `/var/log/` или `/var/log/mysql/`. Имя файла может быть `error.log` или `mysqld.log`.

Поиск неисправностей

Посмотреть журнал ошибок можно с помощью следующей команды:

```
[mysql@mydb] $ more /var/log/mysql/error.log
```

Ожидаемый результат

```
...
2024-08-29T04:20:22.190474Z 0 [Warning] 'NO_ZERO_DATE', 'NO_ZERO_IN_DATE' and
'ERROR_FOR_DIVISION_BY_ZERO' sql modes should be used with strict mode. They will be
merged with strict mode in a future release.
2024-08-29T04:20:22.190520Z 0 [Warning] 'NO_AUTO_CREATE_USER' sql mode was not set.
...
```

Доступ к базе данных

Вы можете получить доступ к базе данных либо с помощью `docker exec`, либо с помощью команды `mysql` в оболочке контейнера.

Пример использования `docker exec` для доступа к базе данных:

```
$ docker exec -it mydb mysql -uroot -proot
```

Ожидаемый результат

```
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
...
```

Остановка процесса MyDB приведёт к выходу из контейнера.

Вы также можете запустить клиент командной строки MySQL в оболочке контейнера для доступа к базе данных:

```
[mysql@mydb] $ mysql -uroot -proot
```

Ожидаемый результат

```
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
...
```

Доступ к серверу из приложения в другом контейнере

Образ открывает снаружи контейнера стандартный порт MySQL 3306, поэтому связывание контейнеров делает экземпляр MyDB Сервер доступным из других контейнеров.

Чтобы связать контейнер, в котором работает ваше приложение (в данном случае из образа с именем `app/image`) с контейнером MyDB Сервер, запустите его с помощью следующей команды:

```
$ docker run -d \
  --name app \
  --link mydb \
  app/image:latest
```

Этот контейнер приложения сможет получить доступ к контейнеру MyDB Сервер через порт 3306.

Хранение данных

Существует два способа хранения данных, используемых приложениями, работающими в контейнерах Docker:

- Можно разрешить Docker управлять хранилищем ваших данных путем записи файлов базы данных на диск хост-системы, используя свое внутреннее управление томами.
- Можно Создать каталог данных в хост-системе на высокопроизводительном хранилище и смонтировать его в каталог, видимый из контейнера. Этот метод помещает файлы базы данных в заранее известное место на хост-системе и упрощает доступ к этим файлам для инструментов и приложений на хост-системе. Пользователь должен убедиться, что каталог существует, что учетные записи пользователей имеют необходимые разрешения и что все другие механизмы безопасности в хост-системе настроены правильно.

Например, если вы создаете каталог данных на подходящем томе в вашей хост-системе с именем `/local/datadir`, то вам нужно запустить контейнер с помощью следующей команды:

```
$ docker run -d \
  --name mydb \
  -e MYSQL_ROOT_PASSWORD=root \
  -v /local/datadir:/var/lib/mysql \
  registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:8.4
```

Параметр `-v /local/datadir:/var/lib/mysql` монтирует каталог `/local/datadir` на хосте в каталог `/var/lib/mysql` в контейнере, который является каталогом данных по умолчанию, используемый *MyDB* Сервер для *MySQL*.

Не добавляйте `MYSQL_ROOT_PASSWORD` к команде `docker run`, если каталог данных содержит подкаталоги, файлы или данные.

Примечание

Если у вас включен SELinux, назначьте соответствующий тип политики новому каталогу данных, чтобы контейнеру был разрешен доступ к нему:

```
$ chcon -Rt svirt_sandbox_file_t /local/datadir
```

Перенаправление порта

Docker позволяет отображать порты контейнера на порты хост-системы, используя опцию `-p`. Если вы запустите контейнер с этой опцией, вы можете подключиться к базе данных, подключив свой клиент к указанному вами порту на хост-машине.

Эта возможность упрощает консолидацию экземпляров на одном хосте.

Чтобы отобразить стандартный порт MySQL 3306 на порт 6603 на хосте:

```
$ docker run -d \
 --name mydb \
 -e MYSQL_ROOT_PASSWORD=root \
 -p 6603:3306 \
 registry.gitflic.ru/project/mydb-ru/mydb-server/mydb:8.4
```

Выход из контейнера

Если вы находитесь в интерактивной оболочке, используйте «CTRL-D» или команду `exit`, чтобы выйти из сеанса.

Если у вас запущен процесс, не являющийся оболочкой, прервите его, нажав «CTRL-C», прежде чем использовать «CTRL-D» или `exit`.

Остановка контейнера

Команда контейнера `docker stop` отправляет сигнал `TERM`, затем ждет 10 секунд и отправляет сигнал `KILL`. Следующий пример останавливает контейнер `mydb`:

```
$ docker stop mydb
```

По умолчанию период времени до остановки контейнера составляет 10 секунд. Очень большой экземпляр может не успеть сбросить данные из памяти на диск за это время. Для экземпляра такого типа добавьте параметр `--time` или `-t` для команды `docker stop`:

```
$ docker stop mydb -t 600
```

Удаление контейнера

Чтобы удалить остановленный контейнер, используйте команду `docker rm`:

```
$ docker rm mydb
```

Для дополнительной информации

Просмотрите Документацию Docker.

Последнее обновление: 2025-02-11

4.5.2 Переменные среды Docker

При запуске контейнера Docker с MyDB Сервер вы можете настроить конфигурацию экземпляра при помощи добавления переменных среды в команду `docker run`.

Эти переменные ни на что не повлияют, если вы запустите контейнер с каталогом данных, который уже содержит базу данных. Любая ранее существовавшая база данных остается нетронутой при запуске контейнера.

Переменные являются необязательными, но необходимо указать хотя бы одно из следующих значений:

- `MYSQL_DATABASE` — имя схемы базы данных, которая создается при запуске контейнера.
- `MYSQL_USER` — создать учетную запись пользователя при запуске контейнера
- `MYSQL_PASSWORD` — используется с `MYSQL_USER` для создания пароля для этой учетной записи пользователя.
- `MYSQL_ALLOW_EMPTY_PASSWORD` — создает пользователя root с пустым паролем. Этот вариант небезопасен, и его следует использовать только для тестирования или проверки концепции, когда базу данных можно впоследствии удалить. Кто угодно может подключиться как root.
- `MYSQL_ROOT_PASSWORD` — этот пароль используется для учетной записи пользователя `root`. Этот вариант не рекомендуется для производственной среды.
- `MYSQL_RANDOM_ROOT_PASSWORD` — установите эту переменную вместо `MYSQL_ROOT_PASSWORD`, если вы хотите, чтобы сервер MyDB сгенерировал для вас пароль. Сгенерированный пароль доступен в логах контейнера только при первом запуске. Используйте `docker logs`. После первого запуска восстановить пароль невозможно.

Для дополнительной защиты вашего экземпляра используйте переменную `MYSQL_ONETIME_PASSWORD`.

Эти переменные видны любому, кто может запустить `docker inspect`.

```
$ docker inspect mydb
```

Ожидаемый результат

```
...
 "Env": [
                "MYSQL_ROOT_PASSWORD=root",
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
                "MYDB_VERSION=8.4.3-3.1",
                "OS_VER=el8",
                "FULL_MYDB_VERSION=8.4.3-3.1.el8"
               ]
...
```

Если это нежелательно, вместо этих переменных вам следует использовать Docker-секреты или Docker-тома.

MyDB Сервер для MySQL также позволяет добавлять суффикс `_FILE` к имени переменной. Этот суффикс позволяет добавить значение из файла по указанному пути, чтобы его нельзя было увидеть снаружи контейнера.

Последнее обновление: 2024-04-30

5. Обновление

5.1 Обзор обновления с 8.0 до 8.4

Обновление вашего сервера до версии 8.4 имеет следующие преимущества:

Преимущества	Описание
Исправления безопасности	Эти исправления и обновления защищают ваши данные от кибератак и устраняют уязвимости и ошибки в программном обеспечении баз данных.
Новые или улучшенные функции	У вас есть доступ к новым или улучшенным функциям, которые повышают функциональность, производительность и доступность базы данных.
Сокращение ручной работы	Вы можете автоматизировать некоторые рутинные задачи.
Актуальность	У ваших клиентов и заинтересованных сторон меняются потребности и ожидания. Использование последней версии может помочь быстрее предоставлять решения.
Снижение эксплуатационных расходов	Модернизированный сервер базы данных может помочь снизить эксплуатационные расходы, поскольку сервер имеет повышенную эффективность и масштабируемость.

Отказ от обновления базы данных может повлечь за собой следующие риски:

Риски	Описание
Риски безопасности	Ваш сервер базы данных уязвим для кибератак, поскольку вы не получаете исправлений безопасности. Эти атаки могут привести к утечке данных, потере и повреждению данных. Эти действия могут нанести вред репутации организации и привести к потере денег.
Сервисные риски	Вы не получаете выгоды от новых или улучшенных функций. Этот риск может привести к ухудшению пользовательского опыта, снижению производительности и увеличению времени простоя.
Риски поддержки	Вы ограничены в доступе к поддержке. Этот риск может привести к увеличению времени разрешения проблем, нерешенным проблемам и более высоким затратам на поддержку.
Риски совместимости	У вас могут возникнуть проблемы совместимости с оборудованием, операционной системой или приложениями, поскольку старая версия не поддерживается на новых платформах.
Риск отказа	Сбой в оборудовании, операционной системе или приложении может привести к необходимости обновления в неподходящее время.

Создайте тестовую среду для проверки обновления перед обновлением рабочих серверов. Тестовая среда имеет решающее значение для успеха обновления. Не существует поддерживаемой процедуры перехода на более раннюю версию . Вы можете попытаться реплицировать версию 8.0 до версии 5.7 или восстановить резервную копию.

Несколько инструментов из набора Percona Toolkit могут помочь в процессе обновления.

Мы рекомендуем обновиться до последней версии.

Просмотрите документацию на предмет других изменений между версиями 8.0 и 8.4.

Просмотрите Стратегии обновления, чтобы получить обзор основных стратегий.

В следующем списке обобщен ряд изменений серии выпусков 8.4 и приведены полезные инструкции, которые помогут вам выполнить плавное обновление. Настоятельно рекомендуем прочитать эту информацию:

- Обновление MySQL
- Обновление MySQL: Прежде чем начать
- Пути обновления
- Полный список изменений в MySQL 8.4
- Подготовка установки к обновлению
- Утилиты Percona, которые упрощают обновление основных версий MySQL
- Примечания к выпуску MyDB Сервер для MySQL 8.0
- Устранение неполадок при обновлении
- Перестроение или восстановление таблиц или индексов

Последнее обновление: 2025-01-28

5.2 Планируем обновление

Обновление базы данных должно быть запланировано и протестировано . Требования к времени простоя являются основным фактором в стратегии обновления.

Следующие шаги должны быть рассмотрены и протестированы:

Шаг	Описание
Обзор текущего окружения	Изучите, какие серверные компоненты/плагины и какие версии приложений используются, количество пользовательских подключений и когда база данных имеет самую большую нагрузку.
Требования к аппаратному и программному обеспечению	Проверьте, требуются ли какие-либо обновления инфраструктуры для поддержки обновления базы данных. Подумайте, необходимо ли обновлять аппаратное и программное обеспечение.
Проведите регрессионное тестирование	Выполните регрессионное тестирование в тестовой среде.
Последовательность обновления	Просмотрите процессы с высоким приоритетом, чтобы спланировать тестирование обновления
Резервное копирование и откат	Спланируйте, что произойдет, если обновление не удастся. Создайте резервную копию важных файлов, включая файлы конфигурации. Имейте в виду, что после обновления сервера базы данных понижение версии не поддерживается.
Примечания к выпуску	Ознакомьтесь с примечаниями к выпуску новой версии.

Последнее обновление: 2024-04-30

5.3 Стратегии обновления

Существуют различные стратегии, которые следует учитывать при обновлении 8.0 до 8.4.

5.3.1 Обновление «на месте»

Обновление на 8.4 не позволяет откатиться. Стратегия обновления «на месте» не рекомендуется, и ее следует использовать только в крайнем случае.

Обновление «на месте» включает в себя завершение работы сервера 8.0 и замену бинарных файлов или пакетов сервера новыми. На этом этапе новую версию сервера можно запустить в существующем каталоге данных. Обратите внимание, что сервер должен быть настроен на медленное завершение работы, т.е. должна быть установлена системная переменная `innodb_fast_shutdown=0` перед выключением.

Особенности обновления «на месте»:

- Меньше дополнительных затрат на инфраструктуру по сравнению с новой средой, но узлы необходимо тестировать.
- Обновление большого кластера может выполняться постепенно в течение нескольких недель с паузами между обновлениями узла кластера, предназначенных для чтения.
- Требуется аварийное переключение производственного трафика, и для минимального времени простоя необходимо иметь хорошие инструменты высокой доступности.

Если вы используете XA-транзакции с InnoDB, запуск `XA RECOVER` перед обновлением проверяет наличие незафиксированных XA-транзакций. Если эта команда возвращает непустой результат, зафиксируйте или откатите XA-транзакции, выполнив команды `XA COMMIT` или `XA ROLLBACK`.

5.3.2 Новое окружение с переключением

Обновление с помощью нового окружения предполагает создание дублирующей среды с тем же количеством серверов, с теми же характеристиками оборудования и той же операционной системой, что и на текущих производственных узлах.

На новом оборудовании устанавливается целевая версия MySQL. После этого настраивается новое окружение и восстанавливаются производственные данные. Помните, что вы можете использовать утилиту `pt-config-diff` из пакета Percona Toolkit для сверки конфигураций MySQL.

Затем настраивается репликация из текущего окружения во вновь созданное.

Во время переключения с текущего окружения на новое вся запись в текущее окружение останавливается, и трафик приложения необходимо перенаправить в новый окружение. Переключение можно выполнить с помощью виртуального IP-адреса или вручную перенаправив само приложение. Как только записи будут получены в новой окружении, вы окажетесь в ситуации невозможности возврата, и на этом этапе можно отключить старое окружение.

Особенности обновления с помощью нового окружения:

- Этот подход требует дополнительных затрат на инфраструктуру, поскольку необходимо построить совершенно новое окружение.
- Возможность апгрейда одновременно и ОС, и СУБД.
- Позволяет легко обновлять оборудование.
- Требуется только одно окно переключения.

Последнее обновление: 2024-04-30

docs@mydb.ru

5.4 Изменения

5.4.1 Общие изменения

- MySQL теперь включает в себя транзакционный словарь данных, в котором хранится информация об объектах базы данных.
- Функция «атомарные DDL-команды» объединяет обновления словаря данных, операции в механизме хранения и записи бинарного журнала, связанные с операцией DDL, в одну атомарную транзакцию.
- Сервер MySQL теперь автоматически выполняет все необходимые задачи обновления при следующем запуске для обновления системных таблиц в схеме `mysql`, а также объектов в других схемах, таких как `sys` и пользовательских схемах. Начиная с версии 8.0.16 больше не требуется вручную вызывать утилиту `mysql_upgrade`.
- Сервер MySQL теперь поддерживает повторное использование сеанса SSL по умолчанию с настройкой тайм-аута в кэше подключений, в течение которого клиенту разрешено запрашивать повторное использование сеанса для новых соединений.
- MySQL теперь поддерживает создание групп ресурсов и управление ими, а также позволяет назначать потоки, выполняемые на сервере, определенным группам так, чтобы указанные потоки выполнялись в соответствии с ресурсами, доступными указанной группе.
- Шифрованием таблиц MySQL теперь можно управлять глобально, определяя и применяя параметры шифрования по умолчанию. Переменная `default_table_encryption` определяет значение шифрования по умолчанию для вновь созданных схем и общего табличного пространства. Эти значения по умолчанию применяются путем включения переменной `table_encryption_privilege_check`.
- Набор символов по умолчанию изменился с `latin1` на `utf8mb4`. В наборе символов `utf8mb4` доступно несколько новых параметров сортировки.
- MySQL поддерживает использование выражений в качестве значений по умолчанию для столбцов с типами данных BLOB, TEXT, GEOMETRY и JSON.
- MySQL теперь имеет блокировку резервного копирования, которая разрешает использование команд DML во время резервного копирования на лету, одновременно предотвращая операции, которые могут привести к несогласованному моментальному снимку.
- MySQL теперь позволяет настраивать порт TCP/IP специально для административных подключений. Этот административный порт доступен, даже если соединения из лимита `max_connections` уже установлены на основном порту.
- MySQL теперь поддерживает невидимые индексы, которые не используются оптимизатором, и позволяет тестировать эффект от удаления индекса, не удаляя его фактически.
- MySQL теперь поддерживает функциональность «Document Store» для разработки приложений, сочетающих работу с данными в SQL и NoSQL моделях с использованием единой базы данных.

- MySQL 8.0 позволяет сохранять глобальные динамические переменные сервера в конфигурационном файле с помощью команды `SET PERSIST` вместо обычной команды `SET GLOBAL`.

Примечание

С полным и актуальным списком изменений в MySQL 8.0 можно ознакомиться в соответствующем разделе официальной документации.

Последнее обновление: 2024-04-30

5.4.2 Изменения в InnoDB

- Максимальное значение счетчика автоинкремента теперь сохраняется при перезапуске сервера.
- При обнаружении повреждения дерева индексов InnoDB записывает флаг повреждения в журнал REDO, что делает флаг повреждения устойчивым к остановке сервера.
- Новая динамическая переменная `innodb_deadlock_detect` может использоваться для отключения обнаружения взаимных блокировок.
- Временные таблицы InnoDB теперь создаются в общем временном табличном пространстве `ibtmp1`.
- Системные таблицы и таблицы словаря данных теперь создаются в одном файле табличного пространства InnoDB с именем mysql.ibd в каталоге данных MySQL.
- По умолчанию журналы отмены теперь располагаются в двух табличных пространствах отмены, а не в системном табличном пространстве, и создаются при инициализации экземпляра MySQL.
- Новая переменная `innodb_dedicated_server`, отключенная по умолчанию, может использоваться для автоматической настройки InnoDB нескольких параметров на основе обнаруженной памяти сервера.
- Файлы табличного пространства можно переместить или восстановить в новое место, пока сервер находится в автономном режиме, с помощью опции `innodb_directories`.

Примечание

С полным и актуальным списком изменений в MySQL 8.0 можно ознакомиться в соответствующем разделе официальной документации.

Последнее обновление: 2024-04-30

5.4.3 Изменения в безопасности и управлении учетными записями в MySQL 8.0

- Таблицы прав доступа в системной базе данных `mysql` теперь являются (транзакционными) таблицами InnoDB.
- Доступен новый плагин аутентификации `caching_sha2_password`. Как и плагин `sha256_password`, `caching_sha2_password` реализует хеширование паролей SHA-256, но использует кеширование для устранения проблем с задержкой во время подключения.
- MySQL теперь поддерживает роли, которые являются именованными наборами прав доступа. Роли можно создавать и удалять. Ролям могут быть предоставлены и отозваны привилегии. Роли можно назначать и отзывать из учетных записей пользователей.
- MySQL теперь включает концепцию категорий учетных записей пользователей, при этом системные и обычные пользователи различаются в зависимости от того, имеют ли они привилегию `SYSTEM_USER`.
- MySQL теперь хранит информацию об истории паролей, что позволяет ограничить повторное использование предыдущих паролей.
- MySQL теперь поддерживает режим FIPS, если он скомпилирован с использованием OpenSSL, а библиотека OpenSSL и объектный модуль FIPS доступны во время выполнения.
- MySQL теперь позволяет администраторам настраивать учетные записи пользователей таким образом, чтобы слишком много последовательных неудачных попыток входа из-за неправильных паролей приводили к временной блокировке учетной записи.
- Начиная с MySQL 8.0.27, MySQL поддерживает многофакторную аутентификацию (MFA), что позволяет создавать учетные записи, имеющие до трех методов аутентификации.

Примечание

С полным и актуальным списком изменений в MySQL 8.0 можно ознакомиться в соответствующем разделе официальной документации.

Последнее обновление: 2024-04-30

5.4.4 Устаревшая функциональность в версии 8.0

- Набор символов `utf8mb3` устарел. Вместо этого используйте `utf8mb4`. Набор символов `utf8mb3` действителен в MySQL 8.0, однако рекомендуется использовать `utf8mb4` для улучшенной поддержки Unicode. Прочитайте Migrating to utf8mb4: Things to Consider для получения дополнительной информации.
- `caching_sha2_password` — это плагин аутентификации по умолчанию в MySQL 8.0, который предоставляет расширенный набор возможностей плагина аутентификации `sha256_password`. `sha256_password` считается устаревшим. Новый плагин аутентификации по умолчанию `caching_sha2_password` предлагает более безопасное хеширование паролей и улучшенную аутентификацию клиентских подключений, `mysql_native_password` который использовался по умолчанию раньше. Существующие пользователи, созданные с указанием плагина `mysql_native_password`, по-прежнему могут использоваться и входить в БД. Новые пользователи будут созданы с помощью плагина `caching_sha2_password`, если вы не измените плагин аутентификации по умолчанию.
- Утилита `mysql_upgrade` устарела, поскольку её функции по обновлению системных таблиц в системной схеме `mysql` и объектов в других схемах были перенесены в сервер MySQL. Начиная с MySQL 8.0.16, сервер выполняет все задачи, которые ранее выполнялись `mysql_upgrade`.

  Процесс обновления автоматически начинается с запуска нового бинарного файла MySQL со старым каталогом данных.

  Текстовый файл `mysql_upgrade_info`, который создается в каталоге данных, используется для хранения номера версии MySQL. Также в каталоге данных создаются новые файлы InnoDB.

  После установки новой версии MySQL сервер теперь автоматически выполняет все необходимые задачи обновления при следующем запуске и не требует вызова `mysql_upgrade` администратором. Кроме того, сервер обновляет содержимое справочных таблиц (чего не делал `mysql_upgrade`).

  Новая опция `--upgrade` обеспечивает контроль над тем, как сервер выполняет автоматические операции по словарю данных и обновлению сервера при запуске.
- Плагин `validate_password` был переработан с использованием инфраструктуры серверных компонента. Старая реализация плагина `validate_password` все еще доступна, но устарела.
- Указание `ENGINE` для команд `ALTER TABLESPACE` и `DROP TABLESPACE`.
- SQL-режим `PAD_CHAR_TO_FULL_LENGTH` (см. раздел “Server SQL Modes” в официальной документации)
- Поддержка `AUTO_INCREMENT` устарела для столбцов типа `FLOAT` и `DOUBLE` (и любых их синонимов). Рассмотрите возможность удаления атрибута `AUTO_INCREMENT` из таких столбцов или преобразуйте их в целочисленный тип.
- Атрибут `UNSIGNED` устарел для столбцов типа `FLOAT`, `DOUBLE` и `DECIMAL` (и любых синонимов). Вместо этого рассмотрите возможность использования простого ограничения CHECK для таких столбцов.

- Синтаксис `FLOAT(M,D)` и `DOUBLE(M,D)` для указания количества цифр для столбцов типа `FLOAT` и `DOUBLE` (и любых синонимов) является нестандартным расширением MySQL. Этот синтаксис устарел.
- Нестандартные операторы в стиле языка C `&&`, `||` и `!`, являющиеся синонимами стандартных SQL-операторов `AND`, `OR` и `NOT` соответственно, считаются устаревшими. Приложения, использующие нестандартные операторы, следует исправить для использования стандартных.
- Системная переменная `relay_log_info_file` и опция `--master-info-file` устарели. Ранее они использовались для указания имени файла метаданных для журнала ретрансляции и ведущего сервера, когда были заданы значения `relay_log_info_repository=FILE` и `master_info_repository=FILE`, но эти параметры устарели. Использование файлов метаданных для журнала ретрансляции и ведущего сервера было заменено защищенными от сбоев таблицами, которые используются по умолчанию в MySQL 8.0.
- Использование переменной среды `MYSQL_PWD` для указания пароля MySQL устарело.

Примечание

С полным и актуальным списком изменений в MySQL 8.0 можно ознакомиться в соответствующем разделе официальной документации.

Последнее обновление: 2024-04-30

5.4.5 Удалённая функциональность в версии 8.0

- Системная переменная `innodb_locks_unsafe_for_binlog` удалена. Уровень изоляции `READ COMMITTED` обеспечивает аналогичную функциональность.
- Использование `GRANT` для создания пользователей. Вместо этого используйте `CREATE USER`. Следование этой практике делает SQL-режим `NO_AUTO_CREATE_USER` несущественным для команды `GRANT`, поэтому он тоже удаляется, и теперь в журнал сервера записывается ошибка, когда наличие этого значения для опции `sql_mode` в файле параметров предотвращает запуск `mysqld`.
- Использование `GRANT` для изменения свойств учетной записи, кроме назначения привилегий. Сюда входят свойства аутентификации, SSL и ограничения ресурсов. Вместо этого установите такие свойства во время создания учетной записи с помощью `CREATE USER` или измените их позже с помощью `ALTER USER`.
- Указание `IDENTIFIED BY PASSWORD 'auth_string'` для `CREATE USER` и `GRANT`. Вместо этого используйте `IDENTIFIED WITH auth_plugin AS 'auth_string'` для `CREATE USER` и `ALTER USER`, где значение `'auth_string'` находится в формате, совместимом с указанным плагином.
- Функция `PASSWORD()`.

  Кроме того, удаление `PASSWORD()` означает, что синтаксис `SET PASSWORD … = PASSWORD('auth_string')` больше недоступен.
- Системная переменная `old_passwords`.

- Кэш запросов был удален. Удалена следующая функциональность:
  - Команды `FLUSH QUERY CACHE` и `RESET QUERY CACHE`
  - Следующие системные переменные:
    - `query_cache_limit`
    - `query_cache_min_res_unit`
    - `query_cache_size`
    - `query_cache_type`
    - `query_cache_wlock_invalidate`
  - Следующе переменные состояния:
    - `Qcache_free_blocks`
    - `Qcache_free_memory`
    - `Qcache_hits`
    - `Qcache_inserts`
    - `Qcache_lowmem_prunes`
    - `Qcache_not_cached`
    - `Qcache_queries_in_cache`
    - `Qcache_total_blocks`
  - Удаление кэша запросов также привело к удалению следующих состояний соединения:
    - `checking privileges on cached query`
    - `checking query cache for a query`
    - `invalidating query cache entries`
    - `sending cached result to the client`
    - `storing result in the query cache`
    - `Waiting for query cache lock`
- Системные переменные `tx_isolation` и `tx_read_only` были удалены. Вместо этого используйте `transaction_isolation` и `transaction_read_only`.
- Системная переменная `sync_frm` удалена, поскольку файлы `.frm` больше не используются.
- Системная переменная `secure_auth` и клиентская опция `--secure-auth` были удалены. Опция `MYSQL_SECURE_AUTH` для функции C API `mysql_options()` была удалена.
- Системная переменная `log_warnings` и опция сервера `-–log-warnings` были удалены. Вместо этого используйте системную переменную `log_error_verbosity`.
- Глобальная область видимости системной переменной `sql_log_bin` была удалена. `sql_log_bin` теперь имеет только область видимости сеанса, и приложения, которые полагаются на доступ к `@@GLOBAL.sql_log_bin`, должны быть скорректированы.
- Неиспользуемые системные переменные `date_format`, `datetime_format`, `time_format` и `max_tmp_tables` удалены.
- Устаревшие квалификаторы `ASC` или `DESC` для указаний `GROUP BY` удалены. Запросы, которые ранее полагались на сортировку `GROUP BY`, могут возвращать результаты,

отличающиеся от предыдущих версий MySQL. Чтобы создать заданный порядок сортировки, добавьте указание `ORDER BY`.

- Синтаксический анализатор больше не считает `N` синонимом `NULL` в операторах SQL. Вместо этого используйте `NULL`. Это изменение не влияет на операции импорта или экспорта текстовых файлов, выполняемые с помощью `LOAD DATA` или `SELECT … INTO OUTFILE`, для которых `NULL` продолжает обозначаться как `N`.
- Параметры `--ssl` и `--ssl-verify-server-cert` в клиентских утилитах были удалены. Используйте `--ssl-mode=REQUIRED` вместо `--ssl=1` или `--enable-ssl`. Используйте `--ssl-mode=DISABLED` вместо `--ssl=0`, `--skip-ssl` или `--disable-ssl`. Используйте `--ssl-mode=VERIFY_IDENTITY` вместо параметров `--ssl-verify-server-cert`.
- Программа `mysql_install_db` была удалена из дистрибутивов MySQL. Инициализацию каталога данных следует выполнять путем вызова `mysqld` с опцией `--initialize` или `--initialize-insecure`. Кроме того, был удален параметр `--bootstrap` для `mysqld`, который использовался программой `mysql_install_db`, а также был удален параметр CMake `INSTALL_SCRIPTDIR`, который контролировал место установки `mysql_install_db`.
- Утилита `mysql_plugin` была удалена. Альтернативы включают загрузку плагинов при запуске сервера с использованием опции `--plugin-load`, или `--plugin-load-add`, или во время выполнения с использованием инструкции `INSTALL PLUGIN`.
- Утилита `resolveip` удалена. Вместо этого можно использовать `nslookup`, `host` или `dig`.

Примечание

С полным и актуальным списком изменений в MySQL 8.0 можно ознакомиться в соответствующем разделе официальной документации.

Последнее обновление: 2024-04-30

5.5 Инструменты из пакета Percona Toolkit, которые могут помочь с обновлением

В наборе инструментов Percona Toolkit есть несколько инструментов, которые могут помочь в планировании обновления, а также значительно упростить весь процесс и снизить вероятность простоев или проблем.

Имя	Описание
pt-upgrade	помогает выполнять запросы SELECT к приложению и генерирует отчеты о том, как каждый шаблон запроса выполняется на серверах в разных версиях MySQL
pt-query-digest	Поскольку передовая практика требует сбора и тестирования всех запросов приложений путем активации журнала медленных запросов на определенный период времени, большинство компаний в конечном итоге получат огромное количество данных из журнала медленных запросов. Инструмент pt-query-digest может помочь в подготовке дайджеста запроса для тестирования обновления.
pt-config-diff	Инструмент pt-config-diff помогает определить различия в настройках MySQL между файлами и переменными сервера. Это позволяет сравнивать обновленную версию с предыдущей версией и проверять различия в конфигурации.
pt-show-grants	Инструмент pt-show-grants извлекает, упорядочивает и затем печатает права доступа для учетных записей пользователей MySQL. Это может помочь экспортировать и создать резервную копию ваших прав доступпа в MySQL перед обновлением или позволить вам легко реплицировать пользователей с одного сервера на другой, просто извлекая права доступа с первого сервера и передавая выходные данные непосредственно на другой сервер.

Дополнительную информацию см. в документации Percona Toolkit.

Последнее обновление: 2024-04-30

5.6 Руководство по обновлению с MySQL 5.7 или Percona Server 5.7 до MyDB Сервер для MySQL 8.0

Важно

Обновление «на месте» не рекомендуется. Используйте обновление на новом окружении с переключением.

Обновление «на месте» включает в себя завершение работы сервера 5.7 и замену бинарных файлов или пакетов сервера новыми. На этом этапе новую версию сервера можно запустить в существующем каталоге данных. Обратите внимание, что сервер должен быть настроен на медленное завершение работы, т.е. должна быть установлена системная переменная `innodb_fast_shutdown=0` перед выключением. Хотя обновление на месте может подойти не для всех сред, особенно для сред, в которых необходимо учитывать множество переменных, в большинстве случаев обновление должно работать.

Особенности обновления «на месте»:

- Меньше дополнительных затрат на инфраструктуру по сравнению с новой средой, но узлы необходимо тестировать.
- Обновление большого кластера может выполняться постепенно в течение нескольких недель с паузами между обновлениями узла кластера, предназначенных для чтения.
- Требуется аварийное переключение производственного трафика, и для минимального времени простоя необходимо иметь хорошие инструменты высокой доступности.

Прежде чем начать процесс обновления, рекомендуется сделать полную резервную копию вашей базы данных.

Скопируйте файл конфигурации базы данных, например, `my.cnf`, в другой каталог, чтобы сохранить его.

Предупреждение

Не обновляйтесь с версии 5.7 до 8.0 на поврежденном экземпляре. Если экземпляр сервера вышел из строя, запустите восстановление после сбоя прежде чем продолжить обновление.

Переменная `encrypt-binlog` удалена, а соответствующий параметр командной строки `--encrypt-binlog` не поддерживается. Важно удалить переменную `encrypt-binlog` из вашего файла конфигурации перед попыткой обновления с другой версии *Percona Server 5.7*. В противном случае генерируется ошибка загрузки сервера с сообщением о неизвестной переменной.

Реализованное шифрование файла бинарного журнала совместимо со старыми версиями формата. Зашифрованный бинарный файл журнала, использовавшийся в Percona Server 5.7 поддерживаются MyDB Сервер для MySQL 8.0.

Вы можете выбрать один из следующих способов обновления с MySQL 5.7 или Percona Server 5.7 до MyDB Сервер для MySQL 8.0:

- Обновление с помощью репозиториев MyDB.
- Обновление с систем, использующих MyRocks или TokuDB Storage Engine и секционированные таблицы.
- Обновление с помощью отдельных пакетов

Последнее обновление: 2024-04-30

5.7 Обновление путем миграции в новое окружение с переключением

Обновление с помощью нового окружения предполагает создание дублирующей среды с тем же количеством серверов, с теми же характеристиками оборудования и той же операционной системой, что и на текущих производственных узлах.

На новом оборудовании устанавливается целевая версия MySQL. После этого настраивается новое окружение и восстанавливаются производственные данные. Помните, что вы можете использовать утилиту `pt-config-diff` из пакета Percona Toolkit для сверки конфигураций MySQL.

Затем настраивается репликация из текущего окружения во вновь созданное.

Во время переключения с текущего окружения на новое вся запись в текущее окружение остановливается, и трафик приложения необходимо перенаправить в новый окружение. Переключение можно выполнить с помощью виртуального IP-адреса или вручную перенаправив само приложение. Как только записи будут получены в новой окружении, вы окажетесь в ситуации невозможности возврата, и на этом этапе можно отключить старое окружение.

Особенности обновления с помощью нового окружения:

- Этот подход требует дополнительных затрат на инфраструктуру, поскольку необходимо построить совершенно новое окружение.
- Возможность апгрейда одновременно и ОС, и СУБД.
- Позволяет легко обновлять оборудование.
- Требуется только одно окно переключения.

Последнее обновление: 2024-04-30

5.8 Обновление систем с механизмами хранения MyRocks или TokuDB и секционированными таблицами

Из-за ограничений, налагаемых *MySQL*, поддержка секционирования таблиц обеспечивается механизмом хранения. *MySQL* 8.0 обеспечивает поддержку секционированных таблица только для механизма хранения *InnoDB*.

Если вы используете секционированные таблицы с механизмом хранения *MyRocks* или *TokuDB*, обновление может завершиться неудачно, если вы не включите собственное секционирование, предоставляемое механизмом хранения.

Предупреждение

Механизм хранения *TokuDB* больше не поддерживается. Для получения дополнительной информации см. TokuDB changes in Percona Server for MySQL by version в документации Percona Server.

Прежде чем попытаться выполнить обновление, проверьте, есть ли у вас таблицы, не использующие собственное секционирование.

```
$ mysqlcheck -u root --all-databases --check-upgrade
```

Если таблицы найдены, `mysqlcheck` выдает предупреждение:

Пример вывода `mysqlcheck`

```
Enable either the rocksdb_enable_native_partition variable or
the tokudb_enable_native_partition variable depending on the storage
engine and restart the server.
```

Следующим шагом будет изменение таблиц, которые не используют собственное секционирование, с помощью указания `UPGRADE PARTITIONING`:

```
ALTER TABLE <имя-таблицы> UPGRADE PARTITIONING;
```

Выполните эти шаги для каждой таблицы, указанной в списке `mysqlcheck`. В противном случае обновление до версии 8.0 завершится неудачно, и в журнале ошибок появятся следующие сообщения:

Журнал ошибок при неудачном обновлении

```
2018-12-17T18:34:14.152660Z 2 [ERROR] [MY-013140] [Server] The 'partitioning' feature is
not available; you need to remove '--skip-partition' or use MySQL built with '-
DWITH_PARTITION_STORAGE_ENGINE=1'
2018-12-17T18:34:14.152679Z 2 [ERROR] [MY-013140] [Server] Can't find file: './comp_test/
t1_RocksDB_lz4.frm' (errno: 0 - Success)
2018-12-17T18:34:14.152691Z 2 [ERROR] [MY-013137] [Server] Can't find file: './comp_test/
t1_RocksDB_lz4.frm' (OS errno: 0 - Success)
```

5.8.1 Обновление дистрибутива Linux «на месте» в системе с установленными пакетами MyDB.

Ниже приведен рекомендуемый процесс выполнения обновления дистрибутива в системе с установленными пакетами MyDB:

1. Запишите установленные пакеты MyDB.
2. Создайте резервную копию данных и конфигураций.
3. Удалите пакеты MyDB, не удаляя файл конфигурации или данные.
4. Выполните обновление, следуя инструкциям по обновлению дистрибутива.
5. Перезагрузите систему.
6. Установите пакеты MyDB, предназначенные для обновленной версии дистрибутива.

Последнее обновление: 2024-04-30

5.9 Понижение версии MyDB Сервер для MySQL

Следующая таблица перечисляет пути понижения версии:

Путь	Примеры	Методы
Внутри корректирующего или LTS-релиза	8.0.35 до 8.0.34	понижение “на месте”, логический экспорт/импорт, асинхронная репликация, плагин MySQL Clone
Корректирующий или LTS-релиз до последнего корректирующего или LTS-релиза	8.4.x до 8.0.x	логический экспорт/импорт, асинхронная репликация
Корректирующий или LTS-релиз до инновационного релиза после последней серии LTS	8.4.x до 8.3.0	логический экспорт/импорт, асинхронная репликация
Инновационный релиз до другого инновационного релиза после последней серии LTS	8.3.0 до 8.2.0	логический экспорт/импорт, асинхронная репликация

Мы не поддерживаем понижения с любым релизом 8.0.x ниже 8.0.34. Релизы в диапазоне выше 8.0.34 могут быть понижены до любого релиза в этом диапазоне, включая 8.0.34.

5.9.1 Риски понижения версии

Понижение версии имеет следующие риски:

Риск	Описание
Потеря данных	Если в процессе понижения версии возникнут проблемы, вы можете потерять свои данные. Важно сделать резервную копию ваших данных перед попыткой понизить версию.
Несовместимость	Если вы используете какие-либо функции или улучшения в последней версии, понижение версии может привести к проблемам несовместимости.
Производительность	Понижение версии может привести к снижению производительности.
Безопасность	Новые версии имеют обновления безопасности, которые недоступны в старых версиях, что может привести к уязвимости.

Последнее обновление: 2025-01-28

6. После установки

6.1 AppArmor

6.1.1 Настройка безопасной установки MyDB Сервер для MySQL с помощью AppArmor

Операционная система имеет систему дискреционного контроля доступа (DAC). AppArmor дополняет DAC системой обязательного контроля доступа (MAC). AppArmor — это модуль безопасности по умолчанию для систем Ubuntu или Debian, который использует профили для определения того, как программы получают доступ к ресурсам.

AppArmor основан на путях к файлам и ограничивает процессы с помощью профилей. Каждый профиль содержит набор правил политики. Некоторые приложения могут устанавливать свой профиль вместе с приложением. Если при установке не устанавливается такой профиль, то это приложение не является частью подсистемы AppArmor. Вы также можете создавать свои профили, поскольку они представляют собой простые текстовые файлы, хранящиеся в каталоге `/etc/apparmor.d`.

AppArmor улучшает безопасность системы, обеспечивая строгий контроль доступа и защищая от несанкционированного доступа и потенциальных угроз. Это достигается путем определения профилей, которые указывают, как программы взаимодействуют с системными ресурсами. Эти профили действуют как набор правил, определяющих действия программы и ресурсы, к которым она может получить доступ. Ограничивая каждую программу ее назначенным профилем, AppArmor ограничивает ущерб в случае компрометации и предотвращает несанкционированное повышение привилегий. Кроме того, AppArmor предоставляет детализированный контроль над поведением программ, что позволяет администраторам настраивать политики безопасности в соответствии с конкретными требованиями приложений и минимизировать поверхность атаки. В целом, AppArmor играет важную роль в укреплении безопасности системы для разработчиков MySQL, поддержании целостности системы и снижении рисков, связанных с нарушениями безопасности.

См. также:

Профили AppArmor
Управление профилями AppArmor
Отключение AppArmor
Конфигурация AppArmor
Устранение неполадок в AppArmor

Последнее обновление: 2024-04-30

docs@mydb.ru

6.1.2 Профили AppArmor

Режимы профиля AppArmor определяют, как приложения взаимодействуют с ресурсами системы. Вы можете комбинировать профили в режиме `Enforce` и профили в режиме `Complain` на вашем сервере.

Режим	Описание
`Enforce`	Ограничивает процессы MySQL в соответствии с правилами, определёнными в профиле. Любое действие, нарушающее эти правила, отклоняется.
`Complain`	Позволяет процессам MySQL выполнять ограниченные действия, но регистрирует эти действия для последующего анализа.
`Disabled`	Полностью отключает ограничения профиля, позволяя процессам MySQL выполнять любые действия без регистрации.

Понимание этих режимов помогает разработчикам MySQL гарантировать, что их приложения могут получать доступ к необходимым ресурсам, при этом поддерживая безопасность системы.

Преимущества

Преимущество	Описание
Повышенная безопасность	Режимы профиля AppArmor, такие как `Enforce` и `Complain`, помогают обеспечивать выполнение политик безопасности для предотвращения несанкционированного доступа.
Удобство устранения неполадок	Режимы профиля предоставляют гибкость в диагностике проблем с доступом, позволяя разработчикам переключаться между режимами.

НЕДОСТАТКИ

Недостаток	Описание
Ограниченная гибкость	Профили могут ограничивать определенные действия или доступ, что потенциально снижает функциональность приложений MySQL.
Сложность	Понимание и управление различными профилями может быть сложным для начинающих разработчиков, что может приводить к ошибкам.
Проблемы с отладкой	Устранение проблем, связанных с профилями, таких как записи DENIED в логах, может требовать дополнительных знаний.

См. также:

AppArmor
Управление профилями AppArmor
Отключение AppArmor
Конфигурация AppArmor
Устранение неполадок в AppArmor

Последнее обновление: 2024-04-30

6.1.3 Управление профилями AppArmor

Понимание рисков AppArmor в разработке MySQL

Хотя профили AppArmor помогают защитить ваш сервер MySQL, неверная конфигурация может привести к неожиданному поведению и потенциальным уязвимостям безопасности. Вот почему тщательный обзор и тестирование имеют решающее значение при внесении изменений:

Потенциальные риски неправильно настроенных профилей AppArmor

Неправильная конфигурация	Описание
Чрезмерно жесткие профили	Эти профили могут помешать MySQL получать доступ к необходимым файлам или ресурсам, что затрудняет его функциональность и вызывает ошибки. Представьте себе профиль, который случайно блокирует MySQL от записи в его журналы, что делает их бесполезными для устранения неполадок.
Чрезмерно мягкие профили	Профили с недостаточными ограничениями могут позволить несанкционированный доступ к файлам или функциональным возможностям MySQL. Это создает риск безопасности, так как злоумышленник, используя уязвимость, может воспользоваться более свободным профилем для получения большего контроля над сервером.
Неправильное назначение профиля	Назначение неправильного профиля процессу может вызвать любую из вышеупомянутых проблем. Например, случайное назначение MySQL профиля, предназначенного для другой службы, может иметь непредвиденные последствия.

Важность тщательного контроля и тестирования

Тщательно проверяя и тестируя изменения в вашем профиле AppArmor, вы можете минимизировать риски, связанные с неправильной конфигурацией, и обеспечить безопасную и функциональную среду для MySQL.

- Тщательно просмотрите свои изменения: Дважды проверьте изменения в профиле AppArmor, чтобы убедиться, что они предоставляют MySQL необходимые разрешения при сохранении уровня безопасности.
- Тестируйте свои изменения в безопасной среде: Прежде чем применять изменения на сервере в рабочем состоянии, протестируйте их в тестовой среде, которая имитирует ваше рабочее окружение. Этот тест позволяет выявить и исправить любые проблемы, вызванные изменениями профиля AppArmor, без воздействия на вашу активную установку MySQL.

Установка утилит для управления AppArmor

Установите пакет apparmor-utils для работы с профилями. Используйте эти утилиты для создания, обновления, принудительного применения, переключения в режим `Complain` и отключения профилей по мере необходимости:

```
$ sudo apt install apparmor-utils
```

Ожидаемый результат

```
Reading package lists... Done
Building dependency tree
...
The following additional packages will be installed:
    python3-apparmor python3-libapparmor
...
```

Добавление профиля `mysqld`

Добавьте профиль `mysqld`, выполнив следующую процедуру:

1. Загрузите актуальную версию AppArmor:

```
$ wget https://gitflic.ru/project/mydb/mydb-server/blob/raw?file=8.0/build-ps/
debian/extra/apparmor.d/usr.sbin.mysqld.in
```

Ожидаемый результат

```
...
Saving to 'apparamor-profile`
...
```

1. Переместите файл в `/etc/apparmor.d/usr.sbin.mysqld`:

```
$ sudo mv apparmor-profile /etc/apparmor.d/usr.sbin.mysqld
```

1. Создайте пустой файл для редактирования:

```
$ sudo touch /etc/apparmor.d/local/usr.sbin.mysqld
```

1. Загрузите профиль:

```
$ sudo apparmor_parser -r -T -W /etc/apparmor.d/usr.sbin.mysqld
```

1. Перезапустите *MyDB* Сервер для *MySQL*:

```
$ sudo systemctl restart mysql
```

1. Проверьте статус профиля:

```
$ sudo aa-status
```

Ожидаемый результат

```
```{.text .no-copy}
...
processes are in enforce mode
...
/usr/sbin/mysqld (100840)
...
```
```

Проверка текущего статуса

Под пользователем `root` или с помощью `sudo` вы можете проверить статус AppArmor:

```
$ sudo aa-status
```

Ожидаемый результат

```
apparmor module is loaded.
34 profiles are loaded.
32 profiles in enforce mode.
...
    /usr/sbin/mysqld
...
2 profiles in complain mode.
...
3 profiles have profiles defined.
...
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

Переключение профиля в режим `Complain`

Переключите профиль в режим `Complain`, когда программа находится в текущем каталоге, с помощью этой команды:

```
$ sudo aa-complain <программа>
```

При необходимости укажите путь к программе в команде, например:

```
$ sudo aa-complain /sbin/<программа>
```

Если профиль не сохранен в `/etc/apparmor.d/`, используйте следующую команду:

```
$ sudo aa-complain /путь/к/профилям/<программа>
```

Переключение профиля в режим `Enforce`

Переключите профиль в режим `Enforce`, когда программа находится в текущем каталоге, с помощью этой команды:

```
$ sudo aa-enforce <программа>
```

При необходимости укажите путь к программе в команде, например:

```
$ sudo aa-enforce /sbin/<программа>
```

Если профиль не сохранен в `/etc/apparmor.d/`, используйте следующую команду:

```
$ sudo aa-enforce /путь/к/профилям/<программа>
```

Отключение профиля

Вы можете отключить профиль, но рекомендуется переключить профиль в режим `Complain`.

Используйте любой из следующих методов, чтобы отключить профиль:

```
$ sudo ln -s /etc/apparmor.d/usr.sbin.mysqld /etc/apparmor.d/disable/
$ sudo apparmor_parser -R /etc/apparmor.d/usr.sbin.mysqld
```

или

```
$ aa-disable /etc/apparmor.d/usr.sbin.mysqld
```

Перезагрузка всех профилей

Выполните любую из следующих команд, чтобы перезагрузить все профили:

```
$ sudo service apparmor reload
```

или

```
$ sudo systemctl reload apparmor.service
```

Перезагрузка одного профиля

Чтобы перезагрузить один профиль, выполните следующее:

```
$ sudo apparmor_parser -r /etc/apparmor.d/<профиль>
```

Чтобы некоторые изменения вступили в силу, возможно, потребуется перезапустить программу.

См. также:

AppArmor
Профили AppArmor
Отключение AppArmor
Конфигурация AppArmor
Устранение неполадок в AppArmor

Последнее обновление: 2024-04-30

6.1.4 Отключение AppArmor

Риски отключения AppArmor

Использование AppArmor может показаться излишним, но если вы отключите его, ваш сервер может столкнуться с проблемами безопасности.

Не отключайте AppArmor в рабочих окружениях. Это действие может иметь следующие риски:

- **Увеличенная поверхность атаки.** Отключение AppArmor убирает ограничения безопасности, потенциально позволяя несанкционированный доступ к файлам и функциональности MyDB Сервер для MySQL. Это создает привлекательную цель для злоумышленников, стремящихся воспользоваться уязвимостями или получить контроль над вашей базой данных. |
- **Непредвиденные уязвимости.** AppArmor может помочь смягчить даже неизвестные уязвимости, ограничивая неожиданные поведения. Отключение его делает вашу систему более восприимчивой к этим скрытым уязвимостям.
- **Случайные ошибки конфигурации.** Даже с хорошими намерениями ручная настройка контроля доступа может быть подвержена ошибкам. AppArmor обеспечивает предопределенный уровень безопасности, уменьшая риск человеческой ошибки в управлении разрешениями. |

Вместо этого используйте функции безопасности AppArmor и конфигурируйте их в соответствии с вашими требованиями.

Процедура отключения

Если AppArmor необходимо отключить, выполните следующие команды:

1. Проверьте статус.

```
$ sudo apparmor_status
```

2. Остановите и отключите AppArmor.

```
$ sudo systemctl stop apparmor
$ sudo systemctl disable apparmor
```

См. также:

AppArmor
Профили AppArmor
Управление профилями AppArmor
Конфигурация AppArmor
Устранение неполадок в AppArmor

Последнее обновление: 2025-01-28

6.1.5 Конфигурация AppArmor

Редактирование профиля

Отредактируйте только `/etc/apparmor.d/local/usr.sbin.mysql`.

Мы рекомендуем переключить профиль в режим `Complain` перед редактированием файла. Отредактируйте файл в любом текстовом редакторе. Когда работа будет завершена, перезагрузите один профиль и переключите профиль в режим `Enforce`.

Настройка нестандартного расположения пользовательского каталога данных

Вы можете изменить каталог данных на нестандартный, например `/var/lib/mysqlcustom`. Вам следует включить режим аудита, чтобы фиксировать все действия, и отредактировать профиль, чтобы разрешить доступ к настраиваемому расположению.

```
$ cat /etc/mysql/mysql.conf.d/mysqld.cnf
```

Ожидаемый результат

```
The MyDB 8.4 configuration file.

For explanations see
https://dev.mysql.com/doc/mysql/en/server-system-variables.html

[mysqld]
pid-file    = /var/run/mysqld/mysqld.pid
socket        = /var/run/mysqld/mysqld.sock
*datadir    = /var/lib/mysqlcustom*
log-error    = /var/log/mysql/error.log
```

Включите режим аудита для `mysqld`. В этом режиме применяется политика безопасности, и весь доступ протоколируется.

```
$ aa-audit mysqld
```

Перезапустите MyDB Сервер для MySQL:

```
$ sudo systemctl mysql restart
```

Если перезапуск не удался, значит AppArmor заблокировал доступ к каталогу пользовательских данных. Чтобы диагностировать проблему, проверьте журналы на наличие следующих ключевых слов:

- `ALLOWED` — событие журнала, когда профиль находится в режиме `Complain` и действие нарушает политику.
- `DENIED` — событие журнала, когда профиль находится в режиме `Enforce` и действие заблокировано.

Например, следующие записи журнала показывают `DENIED`:

Ожидаемый результат

```
...
Dec 07 12:17:08 ubuntu-s-4vcpu-8gb-nyc1-01-aa-ps audit[16013]: AVC apparmor="DENIED"
operation="mknod" profile="/usr/sbin/mysqld" name="/var/lib/mysqlcustom/binlog.index"
pid=16013 comm="mysqld" requested_mask="c" denied_mask="c" fsuid=111 ouid=111
Dec 07 12:17:08 ubuntu-s-4vcpu-8gb-nyc1-01-aa-ps kernel: audit: type=1400
audit(1607343428.022:36): apparmor="DENIED" operation="mknod" profile="/usr/sbin/mysqld"
name="/var/lib/mysqlcustom/mysqld_tmp_file_case_insensitive_test.lower-test" pid=16013
comm="mysqld" requested_mask="c" denied_mask="c" fsuid=111 ouid=111
...
```

Откройте файл `/etc/apparmor.d/local/usr.sbin.mysqld` в текстовом редакторе и отредактируйте следующие записи в разделе `Allow data dir access`.

```
Allow data dir access
/var/lib/mysqlcustom/ r,
/var/lib/mysqlcustom/** rwk,
```

Закомментируйте, используя символ `#`, текущие записи в разделе `Allow data dir access section`. Этот шаг не является обязательным. Если вы пропустите этот шаг, mysqld продолжит обращаться к расположению каталога данных по умолчанию.

Примечание

Отредактируйте локальную версию файла вместо основного профиля. Разделение изменений упрощает обслуживание.

Перезагрузить профиль:

```
$ apparmor_parser -r -T /etc/apparmor.d/usr.sbin.mysqld
```

Перезапустите MySQL:

```
$ systemctl restart mysqld
```

Настройка нестандартного расположения журнала

Чтобы переместить журналы в произвольное расположение, вам необходимо отредактировать файл конфигурации my.cnf, а затем отредактировать локальный профиль, чтобы разрешить доступ:

```
cat /etc/mysql/mysql.conf.d/mysqld.cnf
```

Ожидаемый результат

```
The MyDB 8.0 configuration file.

For explanations see
https://dev.mysql.com/doc/mysql/en/server-system-variables.html

[mysqld]
pid-file    = /var/run/mysqld/mysqld.pid
socket        = /var/run/mysqld/mysqld.sock
datadir    = /var/lib/mysql
log-error    = /*custom-log-dir*/mysql/error.log
```

Убедитесь, что пользовательский каталог существует.

```
$ ls -la /custom-log-dir/
```

Ожидаемый результат

```
total 12
drwxrwxrwx  3 root root 4096 Dec  7 13:09 .
drwxr-xr-x 24 root root 4096 Dec  7 13:07 ..
drwxrwxrwx  2 root root 4096 Dec  7 13:09 mysql
```

Перезапустите сервер MyDB.

```
$ service mysql start
```

Ожидаемый результат

```
Job for mysql.service failed because the control process exited with error code.
See "systemctl status mysql.service" and "journalctl -xe" for details.
```

```
$ journalctl -xe
```

Ожидаемый результат

```
...
AVC apparmor="DENIED" operation="mknod" profile="/usr/sbin/mysqld" name="/custom-log-dir/
mysql/error.log"
...
```

AppArmor запретил доступ. Отредактируйте локальный профиль в разделе `Allow log file access`, чтобы разрешить доступ к нестандартному расположению журнала.

```
$ cat /etc/apparmor.d/local/usr.sbin.mysqld
```

Ожидаемый результат

```
 Site-specific additions and overrides for usr.sbin.mysqld..
 For more details, please see /etc/apparmor.d/local/README.

 Allow log file access
 /custom-log-dir/mysql/ r,
 /custom-log-dir/mysql/** rw,
```

Перезагрузить профиль:

```
$ apparmor_parser -r -T /etc/apparmor.d/usr.sbin.mysqld
```

Перезапустите сервер MyDB:

```
$ systemctl restart mysqld
```

Установка расположения каталога `secure_file_priv`

По умолчанию `secure_file_priv` указывает на следующее местоположение:

```
mysql> mysqlshow variables like 'secure_file_priv';
```

Ожидаемый результат

```
+------------------+-----------------------+
| Variable_name    | Value                 |
+------------------+-----------------------+
| secure_file_priv | /var/lib/mysql-files/ |
+------------------+-----------------------+
```

Чтобы разрешить доступ к другому местоположению, в текстовом редакторе откройте локальный профиль. Проверьте настройки в разделе `Allow data dir access`:

```
Allow data dir access
/var/lib/mysql/ r,
/var/lib/mysql/** rwk,
```

Отредактируйте локальный профиль в текстовом редакторе, чтобы разрешить доступ к произвольному расположению.

```
$ cat /etc/apparmor.d/local/usr.sbin.mysqld
```

Ожидаемый результат

```
Site-specific additions and overrides for usr.sbin.mysqld..
For more details, please see /etc/apparmor.d/local/README.

Allow data dir access
/var/lib/mysqlcustom/ r,
/var/lib/mysqlcustom/** rwk,
```

Перезагрузить профиль:

```
$ apparmor_parser -r -T /etc/apparmor.d/usr.sbin.mysqld
```

Перезапустите MyDB Сервер для MySQL:

```
$ systemctl restart mysqld
```

См. также:

AppArmor
Профили AppArmor
Управление профилями AppArmor
Отключение AppArmor
Устранение неполадок в AppArmor

Последнее обновление: 2024-04-30

6.1.6 Устранение неполадок в AppArmor

Устранение неполадок профилей AppArmor обеспечивает доступ приложений к необходимым ресурсам без ущерба для безопасности системы.

Режимы профилей

Профили AppArmor работают в разных режимах:

Режим	Описание
`Enforce`	Ограничивает процессы MySQL в соответствии с правилами, определёнными в профиле. Любое действие, нарушающее эти правила, отклоняется.
`Complain`	Позволяет процессам MySQL выполнять ограниченные действия, но регистрирует эти действия для последующего анализа.
`Disabled`	Полностью отключает ограничения профиля, позволяя процессам MySQL выполнять любые действия без регистрации.

ПРОВЕРКА СТАТУСА

Используйте команду `aa-status`, чтобы проверить текущее состояние профилей AppArmor. Эта проверка помогает определить, находятся ли профили в режиме `Enforce` или в режиме `Complain`.

ПЕРЕКЛЮЧЕНИЕ РЕЖИМОВ

Вам может понадобиться переключать профили между режимами при устранении неполадок. Используйте `aa-enforce` для переключения в режим `Enforce` и `aa-complain` для переключения в режим `Complain`.

ОТКЛЮЧЕНИЕ ПРОФИЛЕЙ

При необходимости профили могут быть временно отключены. Однако это не рекомендуется по соображениям безопасности. Используйте такие команды `ln -s` или `aa-disable`, чтобы отключить профили.

ПЕРЕЗАГРУЗКА ПРОФИЛЕЙ

После внесения изменений в профили или переключения режимов важно перезагрузить профили, чтобы изменения вступили в силу. Используйте такие команды `service apparmor reload` или `apparmor_parser -r`, чтобы перезагрузить профили.

ПРОСМОТР ЗАПИСЕЙ ЛОГОВ

Отслеживайте записи логов на наличие действий DENIED (отклонено) или ALLOWED (разрешено). Записи DENIED указывают на то, что профиль блокирует действие, в то время как записи ALLOWED предполагают, что действие разрешено.

РЕДАКТИРОВАНИЕ ПРОФИЛЕЙ

Вам может потребоваться редактировать профили AppArmor для устранения проблем с доступом и разрешения определенных действий. Отредактируйте файлы профиля в директории `/etc/apparmor.d/`, чтобы изменить разрешения доступа.

См. также:

- AppArmor
- Профили AppArmor
- Управление профилями AppArmor
- Отключение AppArmor
- Конфигурация AppArmor

Последнее обновление: 2024-04-30

6.2 Улучшения бинарного журнала и репликации

Благодаря постоянному развитию в *MyDB* Сервер для *MySQL* вошли ряд улучшений, связанные с репликацией и обработкой бинарных журналов. Это привело к особенностям репликации, отличающим ее от *MySQL*.

6.2.1 Безопасность операторов с указанием `LIMIT`

MySQL рассматривает все команды `UPDATE/DELETE/INSERT ... SELECT` с указанием `LIMIT` как небезопасные. Эти команды переключают режим бинарного журнала с режима `statement-based` на `row-based`, если `binlog_format` установлен в `MIXED`.

Вот почему:

- Указание `LIMIT` без `ORDER BY` делает результат недетерминированным
- Одна и та же команда может затрагивать разные строки даннных на первичном сервере и на репликах

```
UPDATE table1 LIMIT 10 SET col1 = 'value';
DELETE FROM table1 LIMIT 5;
INSERT INTO table2 SELECT * FROM table1 LIMIT 3;
```

Чтобы сделать эти команды безопасными для `statement-based` репликации, нужно сделать одно из следующих действий:

- Убрать указание `LIMIT`
- Добавить указание `ORDER BY`, чтобы сделать результаты детерминированными

```
UPDATE table1 SET col1 = 'value' ORDER BY id LIMIT 10;
DELETE FROM table1 ORDER BY id LIMIT 5;
INSERT INTO table2 SELECT * FROM table1 ORDER BY id LIMIT 3;
```

Исключение составляет случай, когда указание `LIMIT` используется с `ORDER BY` по уникальному ключу — в этом случае команда становится детерминированной и безопасной для `statement-based` репликации.

MyDB Сервер для *MySQL* более точен, он рассматривает такие команды как безопасные, если они содержат укахание `ORDER BY PK` или условие `WHERE`.

6.2.2 Улучшение производительности при обновлении положения журнала ретрансляции

MySQL всегда обновляет позицию журнала ретрансляции в настройках репликации с несколькими источниками независимо от того, была ли уже выполнены зафиксированная транзакция или нет. MyDB Сервер пропускает обновления позиции журнала ретрансляции для уже зарегистрированных GTID.

6.2.3 Детали позиции журнала ретрансляции

В частности, такие безусловные обновления позиции журнала ретрансляции вызывали дополнительные операции fsync в случае `relay-log-info-repository=TABLE` и с большим количеством каналов, передающих такие дублирующие (уже выполненные) транзакции, ситуация становилась хуже пропорционально. Исправлена ошибка #1786 (MySQL #85141).

6.2.4 Улучшение производительности при обновлении источника и статуса соединения

Реплики, настроенные для обновления состояния источника и информации о соединении только при ротации файла журнала, не наблюдали ожидаемого снижения нагрузки. *MySQL* дополнительно обновлял эту информацию в случае наличия нескольких источников репликации, когда реплике приходилось пропустить уже выполненное событие GTID.

Конфигурация с `master_info_repository=TABLE` и `sync_master_info=0` заставляет реплику обновлять информацию о состоянии источника и соединения в специальной таблице только при ротации файлов журнала, а не после каждых `sync_master_info` событий. Но эта функциональность не работала в конфигурациях репликации с несколькими источниками. Сигналы Heartbeat, отправляемые на реплику, чтобы пропустить события GTID, которые она уже выполнила, ранее оценивались как события ротации журнала ретрансляции и приводили к синхронизации таблицы `mysql.slave_master_info`. Эта неточность могла привести к огромному (вплоть до 5 раз при некоторых настройках) увеличению нагрузки на запись на реплике до того, как проблема была исправлена в *Percona Server for MySQL*. Исправлена ошибка #1812 (MySQL #85158).

6.2.5 Запись команд `FLUSH` в бинарный журнал

Команды `FLUSH`, такие как `FLUSH SLOW LOGS`, не записываются в бинарный журнал, если системная переменная `binlog_skip_flush_commands` установлена в **ON**.

Кроме того, были реализованы следующие изменения в поведении режимов `read_only` и `super_read_only`:

- Когда для `read_only` установлено значение **ON**, любая команда `FLUSH ...`, выполняемая обычным пользователем (без привилегии `SUPER`), не записывается в бинарный журнал независимо от значения переменной `binlog_skip_flush_commands`.
- Когда для параметра `super_read_only` установлено значение **ON**, любая команда `FLUSH ...`, выполняемая любым пользователем (даже обладающим привилегией `SUPER`), не записывается в бинарный журнал независимо от значения переменной `binlog_skip_flush_commands`.

Попытка запустить команду `FLUSH` без привилегий `SUPER` или `RELOAD` приводит к ошибке `ER_SPECIFIC_ACCESS_DENIED_ERROR` независимо от значения переменной `binlog_skip_flush_commands`.

`binlog_skip_flush_commands`

Свойство	Значение
Командная строка	Да
Конфигурационный файл	Да
Область видимости	Глобальная
Динамическая	Да
По умолчанию	`OFF`

Когда для `binlog_skip_flush_commands` установлено значение **ON**, команды `FLUSH ...` не записываются в бинарный журнал.

Установка `binlog_skip_flush_commands` не влияет на следующие команды, потому что они не записываются в бинарный журналЖ

- `FLUSH LOGS`
- `FLUSH BINARY LOGS`
- `FLUSH TABLES WITH READ LOCK`
- `FLUSH TABLES ... FOR EXPORT`

Команда `FLUSH` не записывается в бинарный журнал и игнорирует значение `binlog_skip_flush_commands`, когда она запускается с ключевым словом `NO_WRITE_TO_BINLOG` (или его синонимом `LOCAL`).

6.2.6 Сохранение комментариев в командах DDL

Когда вы выполняете команды DDL, например `DROP TABLE`, сервер выполняет следующие действия в отношении бинарного журнала:

Действие	Описание
Удаляет комментарии	Сервер удаляет любые комментарии в изначальной команде. Например, если вы используете `DROP TABLE my_table /* This is a comment */;`, в бинарном журнале не сохраняется комментарий.
Добавляет кавычки	Сервер добавляет кавычки вокруг имени таблицы. То есть, если вы запускаете `DROP TABLE my_table;`, он записывает в журнал `DROP TABLE "my_table";`.

Эти действия упрощают формат журналирования, но иногда вам нужен изначальный формат.

binlog_ddl_skip_rewrite

Свойство	Значение
Командная строка	Да
Конфигурационный файл	Да
Область видимости	Глобальная
Динамическая	Да
По умолчанию	`OFF`

Если переменная выключена (по умолчанию), сервер удаляет комментарии и добавляет кавычки.

Если переменная включена, все DDL команды `DROP TABLE`, затрагивающие одну таблицу, записываются в бинарный журнал в следующем виде:

• Комментарии сохраняются, то есть, любые заметки, которые вы добавляете в команду, остаются в бинарном журнале.

• Добавляются кавычки.

Вы можете включать `binlog_ddl_skip_rewrite` на лету:

```
-- Проверяем текущее значение
SHOW VARIABLES LIKE 'binlog_ddl_skip_rewrite';

-- Включить переменную
SET GLOBAL binlog_ddl_skip_rewrite = ON;

-- Выключить переменную
SET GLOBAL binlog_ddl_skip_rewrite = OFF;
```

Вы можете включить её на постоянной основе, добавив её в конфигурационный файл

```
[mysqld]
binlog_ddl_skip_rewrite = ON
```

После добавления в конфигурационный файл перезапустите сервис MySQL.

DDL команды `DROP TABLE` с несколькими таблицами не поддерживаются и возвращают ошибку.

```
SET binlog_ddl_skip_rewrite = ON;
/*comment at start*/DROP TABLE t /*comment at end*/;
```

6.2.7 Пользовательские функции бинарного журнала

Чтобы реализовать восстановление на определенный момент времени (point in time recovery), мы добавили `binlog_utils_udf`. Включены следующие определяемые пользователем функции:

Имя	Возвращает	Описание
get_binlog_by_gtid()	Имя файла бинлога в виде строки	Возвращает имя файла бинлога, содержащего указанный GTID
get_last_gtid_from_binlog()	GTID в виде строки	Возвращает последний GTID в указанном бинлоге
get_gtid_set_by_binlog()	Набор GTID в виде строки	Возвращает все GTID в указанном бинлоге
get_binlog_by_gtid_set()	Имя файла бинлога в виде строки	Возвращает имя файла бинлога, который содержит хотя бы один GTID из указанного набора
get_first_record_timestamp_by_binlog()	Временную метку в виде целого числа	Возвращает временную метку первого события в указанном бинлоге
get_last_record_timestamp_by_binlog()	Временную метку в виде целого числа	Возвращает временную метку последнего события в указанном бинлоге

Все функции, возвращающие временные метки, возвращают свои значения в формате UNIX-времени с точностью до микросекунды. Другими словами, они представляют собой количество микросекунд, прошедших с 1 января 1970 года.

Все функции, принимающие имя бинарного журнала в качестве параметра, принимают только короткие имена без компонента пути. Если во входных данных обнаружен разделитель пути (`/`), возвращается ошибка. Это требуется для ограничения путей, из которых можно читать бинарные журналы. Они всегда считываются из текущего каталога для бинарных журналов (системная переменная @@log_bin_basename).

Все функции, возвращающие имена файлов бинарного журнала, возвращают имя в краткой форме, без компонента пути.

Основной синтаксис `get_binlog_by_gtid()` следующий:

```
* get_binlog_by_gtid(string) [AS] alias
```

Использование: `SELECT get_binlog_by_gtid(string) [AS] alias`

Пример использования команды `get_binlog_gtid`:

```
CREATE FUNCTION get_binlog_by_gtid RETURNS STRING SONAME 'binlog_utils_udf.so';
SELECT get_binlog_by_gtid("F6F54186-8495-47B3-8D9F-011DDB1B65B3:1") AS result;
```

Ожидаемый результат

```
+--------------+
| result       |
+==============+
| binlog.00001 |
+--------------+
```

```
DROP FUNCTION get_binlog_by_gtid;
```

Основной синтаксис `get_last_gtid_from_binlog()` следующий:

```
* get_last_gtid_from_binlog(string) [AS] alias
```

Использование: `SELECT get_last_gtid_from_binlog(string) [AS] alias`

Например использования команды `get_last_gtid_from_binlog`:

```
CREATE FUNCTION get_last_gtid_from_binlog RETURNS STRING SONAME
'binlog_utils_udf.so';
SELECT get_last_gtid_from_binlog("binlog.00001") AS result;
```

Ожидаемый результат

```
+----------------------------------------+
| result                                 |
+========================================+
| F6F54186-8495-47B3-8D9F-011DDB1B65B3:10 |
+----------------------------------------+
```

```
DROP FUNCTION get_last_gtid_from_binlog;
```

Основной синтаксис `get_gtid_set_by_binlog()` следующий:

```
* get_gtid_set_by_binlog(string) [AS] alias
```

Использование: `SELECT get_gtid_set_by_binlog(string) [AS] alias`

Например использование команды `get_gtid_set_by_binlog`:

```
CREATE FUNCTION get_gtid_set_by_binlog RETURNS STRING SONAME
'binlog_utils_udf.so';
SELECT get_gtid_set_by_binlog("binlog.00001") AS result;
```

Ожидаемый результат

```
+-------------------------+
| result                  |
+=========================+
| 11ea-b9a7:7,11ea-b9a7:8 |
+-------------------------+
```

```
DROP FUNCTION get_gtid_set_by_binlog;
```

Основной синтаксис `get_binlog_by_gtid_set()` следующий:

```
* get_binlog_by_gtid_set(string) [AS] alias
```

Использование: `SELECT get_binlog_by_gtid_set(string) [AS] alias`

Пример использования команды `get_binlog_by_gtid_set`:

```
CREATE FUNCTION get_binlog_by_gtid_set RETURNS STRING SONAME
'binlog_utils_udf.so';
SELECT get_binlog_by_gtid_set("11ea-b9a7:7,11ea-b9a7:8") AS result;
```

Ожидаемый результат

```
+----------------------------------------------------------------+
| result                                                         |
+================================================================+
| bin.000003                                                     |
+----------------------------------------------------------------+
```

```
DROP FUNCTION get_binlog_by_gtid_set;
```

Основной синтаксис `get_first_record_timestamp_by_binlog()` следующий:

```
* get_first_record_timestamp_by_binlog(TIMESTAMP) [AS] alias
```

Использование: `SELECT get_first_record_timestamp_by_binlog(TIMESTAMP) [AS] alias`

Пример использования команды `get_first_record_timestamp_by_binlog`:

```
CREATE FUNCTION get_first_record_timestamp_by_binlog RETURNS INTEGER SONAME
'binlog_utils_udf.so';
SELECT FROM_UNIXTIME(get_first_record_timestamp_by_binlog("bin.00003") DIV
1000000) AS result;
```

Ожидаемый результат

```
+---------------------+
| result              |
+=====================+
| 2020-12-03 09:10:40 |
+---------------------+
```

```
DROP FUNCTION get_first_record_timestamp_by_binlog;
```

Основной синтаксис `get_last_record_timestamp_by_binlog()` следующий:

```
* get_last_record_timestamp_by_binlog(TIMESTAMP) [AS] alias
```

Использование: `SELECT get_last_record_timestamp_by_binlog(TIMESTAMP) [AS] alias`

Пример использования команды `get_last_record_timestamp_by_binlog`:

```
CREATE FUNCTION get_last_record_timestamp_by_binlog RETURNS INTEGER SONAME
'binlog_utils_udf.so';
SELECT FROM_UNIXTIME(get_last_record_timestamp_by_binlog("bin.00003") DIV
1000000) AS result;
```

Ожидаемый результат

```
+---------------------+
| result              |
+=====================+
| 2020-12-04 04:18:56 |
+---------------------+
```

```
DROP FUNCTION get_last_record_timestamp_by_binlog;
```

6.2.8 Ограничения

Для следующих переменных не присваивайте значения с одним или несколькими символами точки (`.`):

- log_bin
- log_bin_index

Значение с использованием точки (`.`) обрабатывается по-разному в *MySQL* и **Percona XtraBackup** и может привести к непредсказуемому поведению.

Последнее обновление: 2025-01-28

docs@mydb.ru

6.3 После установки

В зависимости от типа установки вам может потребоваться выполнить следующие задачи:

6.3.1 Установка с использованием бинарных файлов или сборки из исходного кода.

Задача
Инициализация каталога данных
Тестирование сервера
Установка запуска службы во время загрузки

Инициализируем каталог данных

Если вы устанавливаете сервер, используя либо дистрибутив с исходным кодом, либо из бинарного архива TAR, каталог данных не инициализируется и после установки необходимо запустить процесс инициализации.

Запустите `mysqld` с параметром `--initialize` или параметром `--initialize-insecure`.

Выполнение `mysqld` с любым из этих вариантов делает следующее:

- Проверяет существование каталога данных
- Инициализирует системное табличное пространство и связанные с ним структуры.
- Создает системные таблицы, включая таблицы прав доступа, таблицы часовых поясов и справочные таблицы на стороне сервера.
- Создает `root@localhost`

Вам следует выполнить следующие шаги под пользователем `mysql`.

1. Перейдите в каталог MySQL. В примере используется расположение по умолчанию.

```
$ cd /usr/local/mysql
```

2. Создайте каталог для файлов MySQL. Системная переменная `secure_file_priv` использует путь к каталогу в качестве значения.

```
$ mkdir mydata
```

   Каталог должен принадлежать системному пользователю `mysql` и иметь разрешения `drwxr-x---`. Четыре раздела в этой нотации определяют разрешения; файл/каталог, пользователь, группа и другие.

   Первый символ обозначает тип записи: файл или каталог. Первый символ — `d` для каталога.

   Остальные разделы задаются наборами из трех символов.

Разрешение	Пользователь	Группа	Другое
Чтение	Да	Да	Нет
Запись	Да	Нет	Нет
Выполнение	Да	Да	Нет

3. Запустите команду для инициализации каталога данных.

   f4b67873 (Translate select chapters to Russian)

```
$ bin/mysqld --initialize
```

Тестируем сервер

После инициализации каталога данных и запуска сервера вы можете запускать тесты на сервере.

В этом разделе предполагается, что вы использовали настройки установки по умолчанию. Если вы изменили расположение установки, перейдите к каталогу установки. Вы также можете добавить местоположение, Настроив переменные среды.

Вы можете использовать клиентскую утилиту mysqladmin для доступа к серверу.

Если у вас возникли проблемы с подключением к серверу, используйте пользователя `root` и пароль учетной записи `root`.

```
$ sudo mysqladmin -u root -p version
```

Ожидаемый результат

```
Enter password:
mysql Ver 8.4.3-3.1 for debian-linux-gnu on x86_64 (MyDB (GPL), Release '1.1', Revision
'723681b1')
...
Server version      8.4.3-3.1
Protocol version    10
Connection          Localhost via UNIX socket
UNIX socket         /var/run/mysqld/mysqld.sock
Uptime:             4 hours 58 min 10 section

Threads:    2 Questions:    16 Slow queries: 0 Opens: 139 Flush tables: 3
Open tables: 59  Queries per second avg: 0.0000
```

Используйте `mysqlshow` для отображения информации о базе данных и таблицах.

```
$ sudo mysqlshow -u root -p
```

Ожидаемый результат

```
Enter password:

+--------------------+
|      Databases     |
+====================+
| information_schema |
+--------------------+
| mysql              |
+--------------------+
| performance_schema |
+--------------------+
| sys                |
+--------------------+
```

Настраиваем запуск службы во время загрузки

После установки из бинарного архива или собранных самостоятельно исполняемых файлов настройте поддержку `systemd`.

Следующие команды запускают, проверяют состояние и останавливают сервер:

```
$ sudo systemctl start mysqld
$ sudo systemctl status mysqld
$ sudo systemctl stop mysqld
```

Выполните следующую команду, чтобы запустить службу во время загрузки:

```
$ sudo systemctl enable mysqld
```

Выполните следующую команду, чтобы запретить запуск службы во время загрузки:

```
$ sudo systemctl disable mysqld
```

6.3.2 Все типы установок

Задача
Обновить пароль root
Защитить сервер
Заполнить таблицы часовых поясов

Обновляем пароль root

Во время установки в Debian/Ubuntu вам будет предложено ввести пароль root. В Red Hat Enterprise Linux и его производных пароль root обновляется после установки.

Перезапустите сервер с опцией `--skip-grant-tables`, чтобы разрешить доступ без пароля. Этот вариант небезопасен. Эта опция также отключает удаленные подключения.

```
$ sudo systemctl stop mysqld
$ sudo systemctl set-environment MYSQLD_OPTS="--skip-grant-tables"
$ sudo systemctl start mysqld
$ mysql
```

Перезагрузите таблицы разрешений, чтобы иметь возможность выполнить команду `ALTER USER`. Введите пароль, соответствующий текущей политике.

```
mysql> FLUSH PRIVILEGES;
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'rootPassword_12';
mysql> exit
```

Если при добавлении пароля MySQL возвращает `ERROR 1819 (HY000) Your password does not satisfy the current policy`, выполните следующую команду, чтобы увидеть требования политики.

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
```

Повторите свой пароль, чтобы удовлетворить требованиям.

Остановите сервер, удалите параметр `--skip-grant-tables`, запустите сервер и войдите на сервер с обновленным паролем.

```
$ sudo systemctl stop mysqld
$ sudo systemctl unset-environment MYSQLD_OPTS
$ sudo systemctl start mysqld
$ mysql -u root -p
```

Защищаем сервер

Скрипт `mysql_secure_installation` повышает безопасность установки.

Скрипт делает следующее:

- Изменяет пароль `root`
- Запрещает удаленный вход в систему для учетных записей `root`.
- Удаляет анонимных пользователей
- Удаляет тестовую базу данных
- Перезагружает таблицы привилегий

Следующий команда запускает скрипт:

```
$ mysql_secure_installation
```

6.3.3 Заполняем таблицы часовых поясов

Системные таблицы часовых поясов:

- `time_zone`
- `time_zone_leap_second`
- `time_zone_name`
- `time_zone_transition`
- `time_zone_transition_type`

Если вы устанавливаете сервер, используя исходный дистрибутив либо дистрибутив бинарных архивов TAR, то при установке создаются таблицы часовых поясов, но эти таблицы не заполняются.

Программа `mysql_tzinfo_to_sql` заполняет таблицы из данных каталога `zoneinfo` присутствующего в Linux системах.

Распространенный метод заполнения таблиц — передать путь к каталогу `zoneinfo` в `mysql_tzinfo_to_sql`, а затем отправить вывод в `mysql_system_schema`.

В примере предполагается, что вы запускаете команду с учетной записью root. Учетная запись должна иметь права на изменение системной схемы `mysql`.

```
$ mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root -p -D mysql
```

Последнее обновление: 2024-04-30

6.4 Настройка SELinux

SELinux (Security-Enhanced Linux) – это модуль безопасности Linux (LSM), который реализует списки контроля доступа. Он определяет, как и с какими ограничениями процессы взаимодействуют с файлами, сетевыми портами, каталогами, другими процессами и дополнительными компонентами сервера.

Политика SELinux определяет набор правил, «типы» файлов и «домены» для процессов. Правила определяют, как процесс взаимодействует с другим типом. SELinux решает, разрешить или запретить действие, исходя из контекста субъекта, того, какой объект инициирует действие и какой объект является целью действия.

Метка представляет контекст для администраторов и пользователей.

CentOS 7 и CentOS 8 содержат политику MySQL SELinux. *MyDB* Сервер для *MySQL* является полной заменой MySQL и может использовать эту политику без изменений.

6.4.1 Пример контекста SELinux

Чтобы просмотреть контекст SELinux, добавьте ключ `-Z`, который поддерживается многими системными утилитами. Вот пример контекста `mysqld`:

```
$ ps -eZ | grep mysqld_t
```

Ожидаемый результат

```
system_u:system_r:mysqld_t:s0    3356 ?        00:00:01 mysqld
```

Контекст имеет следующие свойства:

- Пользователь — system_u
- Роль — system_r
- Тип или домен — mysqld_t
- Уровень чувствительности - s0 3356

Большинство правил политики SELinux основаны на типе или домене.

6.4.2 Список типов или доменов SELinux, связанных с файлами

Свойство безопасности, на которое опирается SELinux, — это свойство безопасности `Type`. Имя типа часто заканчивается на `_t`. Группа объектов с одинаковым значением безопасности типа принадлежит одному домену.

Чтобы просмотреть типы mysqldb_t, связанные с каталогами и файлами MySQL, выполните следующую команду:

```
$ ls -laZ /var/lib/ | grep mysql
```

Ожидаемый результат

```
drwxr-x--x. mysql   mysql   system_u:object_r:mysqld_db_t:s0 mysql
drwxr-x---. mysql   mysql   system_u:object_r:mysqld_db_t:s0 mysql-files
drwxr-x---. mysql   mysql   system_u:object_r:mysqld_db_t:s0 mysql-keyring
```

Примечание

Если тип политики не определяет свойство типа для объекта, значением по умолчанию является `unconfined_t`.

6.4.3 Режимы SELinux

SELinux имеет следующие режимы:

- `Disabled` — модули политики SELinux не загружены, что отключает политики. Ничего не сообщается.
- `Permissive` — SELinux активен, но модули политики не применяются. Сообщается о нарушении политики, но это не останавливает действие.
- `Enforcing` — SELinux активен, о нарушениях сообщается и действия отклоняются. Если нет правила, разрешающего доступ к ограниченному ресурсу, SELinux запрещает доступ.

6.4.4 Типы политик

SELinux имеет несколько типов политик:

- `Targetd` — большинство процессов работают без ограничений. Конкретные службы содержатся в доменах безопасности и определяются политиками.
- `Strict` — все процессы содержатся в доменах безопасности и определяются политиками.

SELinux управляет ограниченными процессами, которые выполняются в домене, и ограничивает всё, если это явно не разрешено. Неограниченному процессу в неограниченном домене разрешен почти весь доступ.

MySQL — это ограниченный процесс, и модуль политики определяет, какие файлы читаются, какие порты открываются и т. д. SELinux предполагает, что установка *MyDB* Сервер для *MySQL* использует расположение файлов и порты по умолчанию.

Если вы измените значение по умолчанию, вам также необходимо отредактировать политику. Если вы не обновите политику, SELinux в принудительном режиме откажет в доступе ко всем ресурсам, отличным от настроек по умолчанию.

6.4.5 Проверка режима SELinux

Чтобы проверить текущий режим SELinux, используйте любую из следующих команд:

```
$ sestatus
```

Ожидаемый результат

```
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      31
```

или

```
$ grep ^SELINUX= /etc/selinux/config
```

Ожидаемый результат

```
SELINUX=enforcing
```

Примечание

Добавьте параметр `-b` в `sestatus`, чтобы отобразить логические значения политики. Отображаются логические значения для каждого параметра. Пример использования параметра `-b`:

```
$ sestatus -b | grep mysql
```

Ожидаемый результат

```
mysql_connect_any                           off
selinuxuser_mysql_connect_enabled
```

Файл `/etc/selinux/config` определяет, включен или отключен SELinux, а если включен, то будет ли SELinux работать в принудительном (`Enforcing`) или разрешающем (`Permissive`) режиме.

6.4.6 Отключение SELinux

Если вы планируете использовать принудительный режим впоследствии, используйте разрешительный режим вместо отключения SELinux. Пока SELinux отключен, система может содержать объекты с неправильной маркировкой или объекты без метки. Если вы повторно включите SELinux и планируете включить принудительный режим, вам необходимо выполнить действия по перемаркировке всей файловой системы.

Чтобы отключить SELinux при загрузке, установите параметр ядра `selinux=0`. В этом случае ядро не загружает инфраструктуру SELinux. Эта опция имеет тот же эффект, что и добавление инструкции `SELINUX=disabled` в файле конфигурации с последующей перезагрузкой системы.

6.4.7 Дополнительные инструменты SELinux

При необходимости установите инструменты управления SELinux, такие как `semanage` или `search`.

В RHEL 7 или совместимых операционных системах используйте следующую команду от имени пользователя root:

```
$ yum -y install policycoreutils-python
```

В RHEL 8 или совместимых операционных системах используйте следующую команду от имени пользователя root:

```
$ yum -y install policycoreutils-python-utils
```

Примечание

Для запуска команд управления SELinux вам могут потребоваться права root.

6.4.8 Переключение режима в файле конфигурации

Переключение между режимами может помочь при устранении неполадок или изменении правил.

Чтобы навсегда изменить режим, отредактируйте файл `/etc/selinux/config` и измените значение `SELINUX=`. Вам также следует проверить изменение.

```
$ cat /etc/selinux/config | grep SELINUX= | grep -v ^#
```

Ожидаемый результат

```
SELINUX=enforcing
SELINUX=enforcing
```

```
$ sudo sed -i 's/^SELINUX=.*/SELINUX=permissive/g' /etc/selinux/config

$ cat /etc/selinux/config | grep SELINUX= | grep -v ^#
```

Ожидаемый результат

```
SELINUX=permissive
SELINUX=permissive
```

Перезагрузите систему после изменений.

При переключении из отключенного или разрешающего режима в принудительное использование см. раздел Перемаркировка всей файловой системы.

6.4.9 Переключение режима до следующей перезагрузки

Чтобы изменить режим до следующей перезагрузки, используйте любую из следующих команд от имени пользователя root:

```
$ setenforce Enforcing
```

или

```
$ setenforce 1
```

Доступны следующие параметры `setenforce`:

параметры `setenforce`	Также допустимо
`0`	`Permissive`
`1`	`Enforcing`

Вы можете просмотреть текущий режим, выполнив любую из следующих команд:

```
$ getenforce
```

Ожидаемый результат

```
Enforcing
```

или

```
$ sestatus | grep -i mode
```

Ожидаемый результат

```
Current mode:                   permissive
Mode from config file:          enforcing
```

6.4.10 Переключение режима службы

Вы можете переместить одну или несколько служб в разрешительный домен. Остальные службы остаются в принудительном режиме.

Чтобы добавить службу в разрешительный домен, запустите от имени пользователя root:

```
$ sudo semanage permissive -a mysqld_t
```

Чтобы просмотреть текущие разрешительные домены, выполните следующую команду:

```
$ sudo semanage permissive -l
```

Ожидаемый результат

```
...
Customized Permissive Types

mysqld_t

Builtin Permissive Types
...
```

Чтобы удалить службу из разрешительного домена, выполните следующую команду:

```
$ sudo semanage permissive -d mysqld_t
```

Служба возвращается в системный режим SELinux. Обязательно выполните действия по перемаркировке всей файловой системы.

6.4.11 Перемаркировка всей файловой системы

Переход от отключенного или разрешительного режима к принудительному требует дополнительных действий. Для функционирования принудительного режима требуются правильные контексты или метки. Разрешительный режим позволяет пользователям и процессам неправильно помечать файлы и системные объекты. Отключенный режим не загружает инфраструктуру SELinux и не маркирует ресурсы или процессы.

В RHEL и совместимых системах для изменения меток используйте приложение fixfiles. Вы можете перемаркировать всю файловую систему или контексты файлов приложения.

Для одного приложения выполните следующую команду:

```
$ fixfiles -R mysqld restore
```

Чтобы перемаркировать файловую систему без перезагрузки системы, используйте следующую команду:

```
$ fixfiles -f -F relabel
```

Другой вариант перемаркировывает файловую систему во время перезагрузки. Вы можете добавить сенсорный файл, прочитать его во время перезагрузки или настроить параметр загрузки ядра. Завершение операции изменения метки автоматически удаляет сенсорный файл.

Добавьте сенсорный файл от имени root:

```
$ touch /.autorelabel
```

Чтобы настроить ядро, добавьте параметр ядра `autorelabel=1` в список параметров загрузки. Параметр вызывает перемаркировку системы. Перезагрузитесь в разрешительном режиме, чтобы процесс завершился, прежде чем переходить к принудительному использованию.

Примечание

Перемаркировка всей файловой системы требует времени. После завершения перемаркировки система снова перезагружается.

6.4.12 Установка пользовательского каталога данных

Если вы не используете настройки по умолчанию, SELinux в принудительном режиме блокирует доступ к системе.

Например, при установке вы использовали следующую конфигурацию:

```
datadir=/var/lib/mysqlcustom
socket=/var/lib/mysqlcustom/mysql.sock
```

Перезапустите службу.

```
$ service mysqld restart
```

Ожидаемый результат

```
Redirecting to /bin/systemctl restart mysqld.service
Job for mysqld.service failed because the control process exited with error code.
See "systemctl status mysqld.service" and "journalctl -xe" for details.
```

Проверьте журнал, чтобы увидеть код ошибки.

```
$ journalctl -xe
```

Ожидаемый результат

```
...
SELinux is preventing mysqld from getattr access to the file /var/lib/mysqlcustom/ibdata1.
...
```

Проверьте типы SELinux в `/var/lib/mysqlcustom`.

```
ls -1aZ /var/lib/mysqlcustom
```

Ожидаемый результат

```
  total 164288
  drwxr-x--x.  6 mysql mysql system_u:object_r:var_lib_t:s0       4096 Dec  2 07:58  .
  drwxr-xr-x. 38 root  root  system_u:object_r:var_lib_t:s0       4096 Dec  1 14:29  ..
  ...
  -rw-r-----.  1 mysql mysql system_u:object_r:var_lib_t:s0   12582912 Dec  1 14:29
ibdata1
  ...
```

Для решения проблемы используйте следующие методы:

- Установите правильные метки для файлов `mysqlcustom`.
- Измените политику SELinux mysqld, чтобы разрешить mysqld доступ к файлам `var_lib_t`.

Рекомендуемое решение — установить правильные метки. В следующей процедуре предполагается, что вы уже создали и установили владельца для расположения пользовательского каталога данных:

1. Чтобы изменить контекст SELinux, используйте `semanage fcontext`. На этом этапе вы определяете, как SELinux будет работать с пользовательскими путями:

   ```
   $ semanage fcontext -a -e /var/lib/mysql /var/lib/mysqlcustom
   ```

   SELinux применяет к пользовательскому каталогу ту же схему маркировки, которая определена в политике mysqld для каталога `/var/lib/mysql`. Файлы, созданные в пользовательском каталоге, помечаются так, как если бы они находились в `/var/lib/mysql`.

2. Команда `restorecon` применяет изменения.

   ```
   $ restorecon -R -v /var/lib/mysqlcustom
   ```

3. Перезапустите службу mysqld:

   ```
   $ service mysqld start
   ```

6.4.13 Установка нестандартного местоположения журнала

Если вы не используете настройки по умолчанию, SELinux в принудительном режиме блокирует доступ к местоположению. Измените местоположение журнала на произвольное в my.cnf:

```
log-error=/logs/mysqld.log
```

Проверьте расположение журнала с помощью следующей команды:

```
$ ls -laZ /
```

Ожидаемый результат

```
...
drwxrwxrwx.   2 root root unconfined_u:object_r:default_t:s0    6 Dec  2 09:16 logs
...
```

Запуск MySQL возвращает следующее сообщение:

```
$ service mysql start
```

Ожидаемый результат

```
Redirecting to /bin/systemctl start mysql.service
Job for mysqld.service failed because the control process exited with error code.
See "systemctl status mysqld.service" and "journalctl -xe" for details.

$ journalctl -xe
...
SELinux is preventing mysqld from write access to the directory logs.
...
```

Политика SELinux по умолчанию позволяет mysqld записывать журналы в местоположение, помеченное `var_log_t`, которое является местоположением `/var/log`. Решить проблему можно одним из следующих способов:

- Правильно пометьте местоположение `/logs`
- Отредактируйте политику SELinux, чтобы разрешить mysqld доступ ко всем каталогам.

Рекомендуется пометить пользовательское местоположение `/logs`, поскольку из-за него блокируется доступ. Выполните следующие команды, чтобы пометить произвольное местоположение:

```
$ semanage fcontext -a -t var_log_t /logs
$ restorecon -v /logs
```

Возможно, вы не сможете изменить метку каталога `/logs`. Например, другие приложения со своими правилами используют тот же каталог.

Чтобы настроить политику SELinux при совместном использовании каталога, выполните следующие действия:

1. Создайте локальную политику:

```
ausearch -c 'mysqld' --raw | audit2allow -M my-mysqld
```

2. Эта команда создает файлы `my-mysqld.te` и `my-mysqld.pp`. `mysqld.te` — это файл политики соблюдения типов. `my-mysqld.pp` — это модуль политики, загружаемый в виде бинарного файла в подсистему SELinux.

   Пример файла `my-myslqd.te`:

```
module my-mysqld 1.0;

require {
    *type mysqld_t*;
    type var_lib_t;
    *type default_t*;
    class file getattr;
    *class dir write*;
}

============== mysqld_t ==============
*allow mysqld_t default_t:dir write*;
allow mysqld_t var_lib_t:file getattr;
```

   Эта политика содержит правила для пользовательского каталога данных и пользовательского каталога журналов. Мы установили правильные метки для местоположения каталога данных, и применение этой автоматически сгенерированной политики ослабило бы нашу защиту, разрешив mysqld доступ к тегам `var_lib_t`.

3. События, генерируемые SELinux, преобразуются в правила. Созданная политика может содержать правила для недавних нарушений, а также включать несвязанные правила. Несвязанные правила создаются в результате действий, таких как изменение местоположения каталога данных, которые не связаны с каталогом журналов. Добавьте параметр `--start`, чтобы использовать события журнала через определенное время для фильтрации нежелательных событий. Этот параметр фиксирует события, когда отметка времени равна указанному времени или позже. SELinux генерирует политику для текущих действий.

```
$ ausearch --start 10:00:00 -c 'mysqld' --raw | audit2allow -M my-mysqld
```

4. Эта политика позволяет mysqld записывать данные в каталоги с тегами. Откройте файл my_mysqld:

```
module my-mysqld 1.0;

require {
    type mysqld_t;
    type default_t;
    class dir write;
}
```

```
============= mysqld_t ==============
allow mysqld_t default_t:dir write;
```

5. Установите модуль политики SELinux:

```
$ semodule -i my-mysqld.pp
```

Перезапустите службу. В случае сбоя проверьте журнал и выполните ту же процедуру.

Если SELinux запрещает mysql создавать файл журнала внутри каталога, вы можете просмотреть все нарушения, изменив режим SELinux на `Permissive` и затем запустив mysqld. Все нарушения фиксируются в журнале. После этого запуска вы можете сгенерировать

модуль локальной политики, установить его и переключить SELinux обратно в режим `Enforcing`. Выполните следующую процедуру:

1. Выгрузите текущий локальный модуль политики my-mysqld:

```
$ semodule -r my-mysqld
```

2. Вы можете перевести один домен в разрешительный режим. Другие домены в системе останутся в принудительном режиме. Используйте `semanage permissive` с параметром `-a`, чтобы перевести mysqld_t в разрешительный режим:

```
$ semanage permissive -a mysqld_t
```

3. Проверьте изменение режима:

```
$ semdule -l | grep permissive
```

Ожидаемый результат

```
...
permissive_mysqld_t
...
```

4. Чтобы облегчить поиск по логу, напечатайте время:

```
$ date
```

5. Запустите службу.

```
$ service mysqld start
```

6. MySQL запускается, и SELinux регистрирует нарушения в журнале. Проверьте журнал:

```
$ journalctl -xe
```

7. Остановите службу:

```
$ service mysqld stop
```

8. Создайте локальную политику mysqld, используя время, полученное на шаге 4:

```
$ ausearch --start <date-c 'mysqld' --raw | audit2allow -M my-mysqld
```

9. Просмотрите политику (созданная вами политика может отличаться):

```
$ cat my-mysqld.te
```

Ожидаемый результат

```
module my-mysqld 1.0;

require {
type default_t;
    type mysqld_t;
    class dir { add_name write };
    class file { append create open };
}

============= mysqld_t ==============
allow mysqld_t default_t:dir { add_name write };
allow mysqld_t default_t:file { append create open };
```

10. Установите политику:

```
$ semodule -i my-mysqld.pp
```

11. Используйте semanage permissive с параметром -d, который удаляет разрешительный домен для службы:

```
$ semanage permissive -d mysqld_t
```

12. Перезапустите службу:

```
$ service mysqld start
```

Примечание

Используйте эту процедуру для настройки локального модуля политики mysqld. Вам следует просмотреть внесенные изменения, чтобы убедиться, что правила не слишком толерантны.

6.4.14 Установить каталог `secure_file_priv`

При необходимости обновите теги SELinux для каталога `/var/lib/mysql-files/`, используемого для `SELECT ... INTO OUTFILE` или подобных операций. Серверу требуется только доступ на чтение/запись к каталогу назначения.

Чтобы настроить `secure_file_priv` для использования этого каталога, выполните следующие команды для установки контекста:

```
$ semanage fcontext -a -t mysqld_db_t "/var/lib/mysql-files/(/.*)?"
$ restorecon -Rv /var/lib/mysql-files
```

При необходимости отредактируйте путь для другого местоположения.

Последнее обновление: 2024-04-30

7. Управление

7.1 Просмотр полных привилегий с помощью `SHOW EFFECTIVE GRANTS`

В MySQL команда `SHOW GRANTS` имеет следующие ограничения:

- Отображает только явно назначенные привилегии
- Не отображает унаследованные привилегии анонимной учётной записи
- Не отображает привилегии, унаследовонные через роли, если не задано указание `USING`

Другие привилегии могут присутствовать в учётной записи, но не отображаться. Например:

```
-- Создаёт именованного и анонимного пользователей
mysql> CREATE USER 'user1'@'localhost';
mysql> CREATE USER ''@'localhost';

-- Назначаем привилегии анонимному пользователю
mysql> GRANT SELECT ON db.* TO ''@'localhost';
```

```
-- Проверяем привилегии пользователя user1
mysql> SHOW GRANTS FOR 'user1'@'localhost';
```

Ожидаемый результат

```
GRANT USAGE ON *.* TO 'user1'@'localhost'
```

Несмотря на то, что `'user1'@'localhost'` может использовать `SELECT on db.*`, эта привилегия не присутствует в выводе `SHOW GRANTS`.

В *MyDB* Сервер для *MySQL* команда `SHOW EFFECTIVE GRANTS` предоставляет полное описание привилегий пользователя. Она показывает не только привилегии, назначенные пользователю напрямую, но и все унаследованные из других учётных записей, таких как анонимные пользователи или роли. Это описание включет привилегии на уровне системы, базы данных и таблицы и даёт полную картину пользовательских прав внутри базы.

Преимущества:

- Показывает полную картину привилегий
- Помогает найти источники привилегий
- Упрощает аудит безопасности
- Упрощает устранение проблем
- Раскрывает унаследованные привилегии

В Oracle *MySQL* `SHOW GRANTS` отображает только явно назначенные привилегии для указанной учетной записи. Для учетной записи могут быть доступны и другие привилегии, но они не отображаются. Например, если существует анонимная учетная запись, указанная учетная запись может иметь возможность использовать привилегии анонимной, но `SHOW GRANTS` не отобразит их. *MyDB* Сервер для *MySQL* предлагает команду `SHOW EFFECTIVE GRANTS` для отображения всех действующих привилегий для учетной записи, включая те, что назначены неявно через другие учётные записи.

7.1.1 Пример

Если мы создадим следующих пользователей:

```
mysql> CREATE USER grantee@localhost IDENTIFIED BY 'grantee1';
```

Ожидаемый результат

```
Query OK, 0 rows affected (0.50 sec)
```

```
mysql> CREATE USER grantee IDENTIFIED BY 'grantee2';
```

Ожидаемый результат

```
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> CREATE DATABASE db2;
```

Ожидаемый результат

```
Query OK, 1 row affected (0.20 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON db2.* TO grantee WITH GRANT OPTION;
```

Ожидаемый результат

```
Query OK, 0 rows affected (0.12 sec)
```

- Вывод `SHOW GRANTS` перед изменением:

```
mysql> SHOW GRANTS;
```

Ожидаемый результат

```
+-----------------------------------------------------------------------------------------------
----------------------+
| Grants for
grantee@localhost
         |
+-----------------------------------------------------------------------------------------------
----------------------+
| GRANT USAGE ON *.* TO 'grantee'@'localhost' IDENTIFIED BY PASSWORD
'*9823FF338D44DAF02422CF24DD1F879FB4F6B232' |
+-----------------------------------------------------------------------------------------------
----------------------+
1 row in set (0.04 sec)
```

Хотя разрешение на базу данных `db2` не показано, у пользователя `grantee` достаточно прав для создания таблицы в этой базе данных:

```
user@trusty:~$ mysql -ugrantee -pgrantee1 -h localhost
```

```
mysql> CREATE TABLE db2.t1(a int);
```

Ожидаемый результат

```
Query OK, 0 rows affected (1.21 sec)
```

- Вывод `SHOW EFFECTIVE GRANTS` показывает все привилегии для пользователя `grantee`:

```
mysql> SHOW EFFECTIVE GRANTS;
```

Ожидаемый результат

```
+-------------------------------------------------------------------+
| Grants for grantee@localhost                                      |
+-------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'grantee'@'localhost' IDENTIFIED BY PASSWORD|
| '*9823FF338D44DAF02422CF24DD1F879FB4F6B232'                       |
| GRANT ALL PRIVILEGES ON `db2`.* TO 'grantee'@'%' WITH GRANT OPTION|
+-------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

Последнее обновление: 2025-01-28

7.2 `UNINSTALL COMPONENT`

Команда `UNINSTALL COMPONENT` выполняет следующие действия:

- Деактивирует компонент
- Удаляет компонент

Команда не отменяет сохранение каких-либо переменных.

Если возникает ошибка, например неправильное написание имени компонента, команда завершается неудачей и ничего не происходит.

Вы можете удалить несколько компонентов одновременно.

7.2.1 Требуемая привилегия

Для этой команды требуется привилегия `DELETE` для системной таблицы `mysql.component`. Выполнение команды удаляет регистрационную строку из этой таблицы.

7.2.2 Пример

Ниже приведен пример команды `UNINSTALL COMPONENT`.

```
mysql > UNINSTALL COMPONENT 'file://componentA' ;
```

Последнее обновление: 2024-04-30

8. Справочная информация

8.1 Зарезервированные ключевые слова

MyDB использует зарезервированные ключевые слова, которые определяют или управляют функцией или особенностью базы данных. Добавьте эти слова в список зарезервированных ключевых слов MySQL при использовании MyDB Сервер для MySQL.

Если вам необходимо использовать зарезервированное ключевое слово в качестве идентификатора, заключите это слово в набор обратных кавычек (`).

Ниже приведен список зарезервированных ключевых слов, специфичных для MyDB:

- CLIENT_STATISTICS
- CLUSTERING
- COMPRESSION_DICTIONARY
- EFFECTIVE
- INDEX_STATISTICS
- SEQUENCE_TABLE
- TABLE_STATISTICS
- THREAD_STATISTICS
- USER_STATISTICS

Последнее обновление: 2024-04-30

8.2 Список переменных, добавленных в MyDB Сервер для MySQL 8.0

В этом разделе перечислены переменные, которые присутствуют только в MyDB, но отсутствуют в MySQL.

8.2.1 Системные переменные

Имя	Командная строка	Файл конфигурации	Область видимости	Динамическая
audit_log_buffer_size	Да	Да	Глобальная	Нет
audit_log_file	Да	Да	Глобальная	Нет
audit_log_flush	Да	Да	Глобальная	Да
audit_log_format	Да	Да	Глобальная	Нет
audit_log_handler	Да	Да	Глобальная	Нет
audit_log_policy	Да	Да	Глобальная	Да
audit_log_rotate_on_size	Да	Да	Глобальная	Нет
audit_log_rotations	Да	Да	Глобальная	Нет
audit_log_strategy	Да	Да	Глобальная	Нет
audit_log_syslog_facility	Да	Да	Глобальная	Нет
audit_log_syslog_ident	Да	Да	Глобальная	Нет
audit_log_syslog_priority	Да	Да	Глобальная	Нет
csv_mode	Да	Да	Глобальная/ Сессионная	Да
enforce_storage_engine	Да	Да	Глобальная	Нет
expand_fast_index_creation	Да	Нет	Глобальная/ Сессионная	Да
extra_max_connections	Да	Да	Глобальная	Да
extra_port	Да	Да	Глобальная	Нет
have_backup_locks	Да	Нет	Глобальная	Нет
have_backup_safe_binlog_info	Да	Нет	Глобальная	Нет
have_snapshot_cloning	Да	Нет	Глобальная	Нет
innodb_cleaner_lsn_age_factor	Да	Да	Глобальная	Да
innodb_corrupt_table_action	Да	Да	Глобальная	Да
innodb_empty_free_list_algorithm	Да	Да	Глобальная	Да
innodb_encrypt_online_alter_logs	Да	Да	Глобальная	Да
innodb_encrypt_tables	Да	Да	Глобальная	Да
innodb_kill_idle_transaction	Да	Да	Глобальная	Да
innodb_max_bitmap_file_size	Да	Да	Глобальная	Да
innodb_max_changed_pages	Да	Да	Глобальная	Да
innodb_print_lock_wait_timeout_info	Да	Да	Глобальная	Да
innodb_show_locks_held	Да	Да	Глобальная	Да
innodb_temp_tablespace_encrypt	Да	Да	Глобальная	Нет

Имя	Командная строка	Файл конфигурации	Область видимости	Динамическая
innodb_track_changed_pages	Да	Да	Глобальная	Нет
keyring_vault_config	Да	Да	Глобальная	Да
keyring_vault_timeout	Да	Да	Глобальная	Да
log_slow_filter	Да	Да	Глобальная/ Сессионная	Да
log_slow_rate_limit	Да	Да	Глобальная/ Сессионная	Да
log_slow_rate_type	Да	Да	Глобальная	Да
log_slow_sp_statements	Да	Да	Глобальная	Да
log_slow_verbosity	Да	Да	Глобальная/ Сессионная	Да
log_warnings_suppress	Да	Да	Глобальная	Да
proxy_protocol_networks	Да	Да	Глобальная	Нет
query_response_time_flush	Да	Нет	Глобальная	Нет
query_response_time_range_base	Да	Да	Глобальная	Да
query_response_time_stats	Да	Да	Глобальная	Да
slow_query_log_always_write_time	Да	Да	Глобальная	Да
slow_query_log_use_global_control	Да	Да	Глобальная	Да
thread_pool_high_prio_mode	Да	Да	Глобальная/ Сессионная	Да
thread_pool_high_prio_tickets	Да	Да	Глобальная/ Сессионная	Да
thread_pool_idle_timeout	Да	Да	Глобальная	Да
thread_pool_max_threads	Да	Да	Глобальная	Да
thread_pool_oversubscribe	Да	Да	Глобальная	Да
thread_pool_size	Да	Да	Глобальная	Да
thread_pool_stall_limit	Да	Да	Глобальная	Нет
thread_statistics	Да	Да	Глобальная	Да
tokudb_alter_print_error				
tokudb_analyze_delete_fractionref				
tokudb_analyze_in_background	Да	Да	Глобальная/ Сессионная	Да
tokudb_analyze_mode	Да	Да	Глобальная/ Сессионная	Да
tokudb_analyze_throttle	Да	Да	Глобальная/ Сессионная	Да

Имя	Командная строка	Файл конфигурации	Область видимости	Динамическая
`tokudb_analyze_time`	Да	Да	Глобальная/ Сессионная	Да
`tokudb_auto_analyze`	Да	Да	Глобальная/ Сессионная	Да
`tokudb_block_size`				
`tokudb_bulk_fetch`				
`tokudb_cache_size`				
`tokudb_cachetable_pool_threads`	Да	Да	Глобальная	Нет
`tokudb_cardinality_scale_percent`				
`tokudb_check_jemalloc`				
`tokudb_checkpoint_lock`				
`tokudb_checkpoint_on_flush_logs`				
`tokudb_checkpoint_pool_threads`	Да	Да	Глобальная	Нет
`tokudb_checkpointing_period`				
`tokudb_cleaner_iterations`				
`tokudb_cleaner_period`				
`tokudb_client_pool_threads`	Да	Да	Глобальная	Нет
`tokudb_commit_sync`				
`tokudb_compress_buffers_before_eviction`	Да	Да	Глобальная	Нет
`tokudb_create_index_online`				
`tokudb_data_dir`				
`tokudb_debug`				
`tokudb_directio`				
`tokudb_disable_hot_alter`				
`tokudb_disable_prefetching`				
`tokudb_disable_slow_alter`				
`tokudb_empty_scan`				
`tokudb_enable_partial_eviction`	Да	Да	Глобальная	Нет
`tokudb_fanout`	Да	Да	Глобальная/ Сессионная	Да
`tokudb_fs_reserve_percent`				
`tokudb_fsync_log_period`				
`tokudb_hide_default_row_format`				
`tokudb_killed_time`				

Имя	Командная строка	Файл конфигурации	Область видимости	Динамическая
`tokudb_last_lock_timeout`				
`tokudb_load_save_space`				
`tokudb_loader_memory_size`				
`tokudb_lock_timeout`				
`tokudb_lock_timeout_debug`				
`tokudb_log_dir`				
`tokudb_max_lock_memory`				
`tokudb_optimize_index_fraction`				
`tokudb_optimize_index_name`				
`tokudb_optimize_throttle`				
`tokudb_pk_insert_mode`				
`tokudb_prelock_empty`				
`tokudb_read_block_size`				
`tokudb_read_buf_size`				
`tokudb_read_status_frequency`				
`tokudb_row_format`				
`tokudb_rpl_check_readonly`				
`tokudb_rpl_lookup_rows`				
`tokudb_rpl_lookup_rows_delay`				
`tokudb_rpl_unique_checks`				
`tokudb_rpl_unique_checks_delay`				
`tokudb_strip_frm_data`	Да	Да	Глобальная	Нет
`tokudb_support_xa`				
`tokudb_tmp_dir`				
`tokudb_version`				
`tokudb_write_status_frequency`				
`userstat`	Да	Да	Глобальная	Да
`version_comment`	Да	Да	Глобальная	Да
`version_suffix`	Да	Да	Глобальная	Да

8.2.2 Переменные состояния

Имя	Тип	Область видимости
`Binlog_snapshot_file`	Строковый	Глобальная
`Binlog_snapshot_position`	Числовой	Глобальная
`Com_lock_binlog_for_backup`	Числовой	Глобальная/ Сессионная
`Com_lock_tables_for_backup`	Числовой	Глобальная/ Сессионная
`Com_show_client_statistics`	Числовой	Глобальная/ Сессионная
`Com_show_index_statistics`	Числовой	Глобальная/ Сессионная
`Com_show_table_statistics`	Числовой	Глобальная/ Сессионная
`Com_show_thread_statistics`	Числовой	Глобальная/ Сессионная
`Com_show_user_statistics`	Числовой	Глобальная/ Сессионная
`Com_unlock_binlog`	Числовой	Глобальная/ Сессионная
`Innodb_background_log_sync`	Числовой	Глобальная
`Innodb_buffer_pool_pages_LRU_flushed`	Числовой	Глобальная
`Innodb_buffer_pool_pages_made_not_young`	Числовой	Глобальная
`Innodb_buffer_pool_pages_made_young`	Числовой	Глобальная
`Innodb_buffer_pool_pages_old`	Числовой	Глобальная
`Innodb_checkpoint_age`	Числовой	Глобальная
`Innodb_checkpoint_max_age`	Числовой	Глобальная
`Innodb_ibuf_free_list`	Числовой	Глобальная
`Innodb_ibuf_segment_size`	Числовой	Глобальная
`Innodb_lsn_current`	Числовой	Глобальная
`Innodb_lsn_flushed`	Числовой	Глобальная
`Innodb_lsn_last_checkpoint`	Числовой	Глобальная
`Innodb_master_thread_active_loops`	Числовой	Глобальная
`Innodb_master_thread_idle_loops`	Числовой	Глобальная
`Innodb_max_trx_id`	Числовой	Глобальная
`Innodb_mem_adaptive_hash`	Числовой	Глобальная
`Innodb_mem_dictionary`	Числовой	Глобальная

Имя	Тип	Область видимости
`Innodb_oldest_view_low_limit_trx_id`	Числовой	Глобальная
`Innodb_purge_trx_id`	Числовой	Глобальная
`Innodb_purge_undo_no`	Числовой	Глобальная
`Open_tables_with_triggers`	Числовой	Глобальная
`Threadpool_idle_threads`	Числовой	Глобальная
`Threadpool_threads`	Числовой	Глобальная
`Tokudb_DB_OPENS`		
`Tokudb_DB_CLOSES`		
`Tokudb_DB_OPEN_CURRENT`		
`Tokudb_DB_OPEN_MAX`		
`Tokudb_LEAF_ENTRY_MAX_COMMITTED_XR`		
`Tokudb_LEAF_ENTRY_MAX_PROVISIONAL_XR`		
`Tokudb_LEAF_ENTRY_EXPANDED`		
`Tokudb_LEAF_ENTRY_MAX_MEMSIZE`		
`Tokudb_LEAF_ENTRY_APPLY_GC_BYTES_IN`		
`Tokudb_LEAF_ENTRY_APPLY_GC_BYTES_OUT`		
`Tokudb_LEAF_ENTRY_NORMAL_GC_BYTES_IN`		
`Tokudb_LEAF_ENTRY_NORMAL_GC_BYTES_OUT`		
`Tokudb_CHECKPOINT_PERIOD`		
`Tokudb_CHECKPOINT_FOOTPRINT`		
`Tokudb_CHECKPOINT_LAST_BEGAN`		
`Tokudb_CHECKPOINT_LAST_COMPLETE_BEGAN`		
`Tokudb_CHECKPOINT_LAST_COMPLETE_ENDED`		
`Tokudb_CHECKPOINT_DURATION`		
`Tokudb_CHECKPOINT_DURATION_LAST`		
`Tokudb_CHECKPOINT_LAST_LSN`		
`Tokudb_CHECKPOINT_TAKEN`		
`Tokudb_CHECKPOINT_FAILED`		
`Tokudb_CHECKPOINT_WAITERS_NOW`		
`Tokudb_CHECKPOINT_WAITERS_MAX`		
`Tokudb_CHECKPOINT_CLIENT_WAIT_ON_MO`		
`Tokudb_CHECKPOINT_CLIENT_WAIT_ON_CS`		

Имя	Тип	Область видимости
Tokudb_CHECKPOINT_BEGIN_TIME		
Tokudb_CHECKPOINT_LONG_BEGIN_TIME		
Tokudb_CHECKPOINT_LONG_BEGIN_COUNT		
Tokudb_CHECKPOINT_END_TIME		
Tokudb_CHECKPOINT_LONG_END_TIME		
Tokudb_CHECKPOINT_LONG_END_COUNT		
Tokudb_CACHETABLE_MISS		
Tokudb_CACHETABLE_MISS_TIME		
Tokudb_CACHETABLE_PREFETCHES		
Tokudb_CACHETABLE_SIZE_CURRENT		
Tokudb_CACHETABLE_SIZE_LIMIT		
Tokudb_CACHETABLE_SIZE_WRITING		
Tokudb_CACHETABLE_SIZE_NONLEAF		
Tokudb_CACHETABLE_SIZE_LEAF		
Tokudb_CACHETABLE_SIZE_ROLLBACK		
Tokudb_CACHETABLE_SIZE_CACHEPRESSURE		
Tokudb_CACHETABLE_SIZE_CLONED		
Tokudb_CACHETABLE_EVICTIONS		
Tokudb_CACHETABLE_CLEANER_EXECUTIONS		
Tokudb_CACHETABLE_CLEANER_PERIOD		
Tokudb_CACHETABLE_CLEANER_ITERATIONS		
Tokudb_CACHETABLE_WAIT_PRESSURE_COUNT		
Tokudb_CACHETABLE_WAIT_PRESSURE_TIME		
Tokudb_CACHETABLE_LONG_WAIT_PRESSURE_COUNT		
Tokudb_CACHETABLE_LONG_WAIT_PRESSURE_TIME		
Tokudb_CACHETABLE_POOL_CLIENT_NUM_THREADS		
Tokudb_CACHETABLE_POOL_CLIENT_NUM_THREADS_ACTIVE		
Tokudb_CACHETABLE_POOL_CLIENT_QUEUE_SIZE		
Tokudb_CACHETABLE_POOL_CLIENT_MAX_QUEUE_SIZE		
Tokudb_CACHETABLE_POOL_CLIENT_TOTAL_ITEMS_PROCESSED		
Tokudb_CACHETABLE_POOL_CLIENT_TOTAL_EXECUTION_TIME		
Tokudb_CACHETABLE_POOL_CACHETABLE_NUM_THREADS		

Имя	Тип	Область видимости
Tokudb_CACHETABLE_POOL_CACHETABLE_NUM_THREADS_ACTIVE		
Tokudb_CACHETABLE_POOL_CACHETABLE_QUEUE_SIZE		
Tokudb_CACHETABLE_POOL_CACHETABLE_MAX_QUEUE_SIZE		
Tokudb_CACHETABLE_POOL_CACHETABLE_TOTAL_ITEMS_PROCESSED		
Tokudb_CACHETABLE_POOL_CACHETABLE_TOTAL_EXECUTION_TIME		
Tokudb_CACHETABLE_POOL_CHECKPOINT_NUM_THREADS		
Tokudb_CACHETABLE_POOL_CHECKPOINT_NUM_THREADS_ACTIVE		
Tokudb_CACHETABLE_POOL_CHECKPOINT_QUEUE_SIZE		
Tokudb_CACHETABLE_POOL_CHECKPOINT_MAX_QUEUE_SIZE		
Tokudb_CACHETABLE_POOL_CHECKPOINT_TOTAL_ITEMS_PROCESSED		
Tokudb_CACHETABLE_POOL_CHECKPOINT_TOTAL_EXECUTION_TIME		
Tokudb_LOCKTREE_MEMORY_SIZE		
Tokudb_LOCKTREE_MEMORY_SIZE_LIMIT		
Tokudb_LOCKTREE_ESCALATION_NUM		
Tokudb_LOCKTREE_ESCALATION_SECONDS		
Tokudb_LOCKTREE_LATEST_POST_ESCALATION_MEMORY_SIZE		
Tokudb_LOCKTREE_OPEN_CURRENT		
Tokudb_LOCKTREE_PENDING_LOCK_REQUESTS		
Tokudb_LOCKTREE_STO_ELIGIBLE_NUM		
Tokudb_LOCKTREE_STO_ENDED_NUM		
Tokudb_LOCKTREE_STO_ENDED_SECONDS		
Tokudb_LOCKTREE_WAIT_COUNT		
Tokudb_LOCKTREE_WAIT_TIME		
Tokudb_LOCKTREE_LONG_WAIT_COUNT		
Tokudb_LOCKTREE_LONG_WAIT_TIME		
Tokudb_LOCKTREE_TIMEOUT_COUNT		
Tokudb_LOCKTREE_WAIT_ESCALATION_COUNT		
Tokudb_LOCKTREE_WAIT_ESCALATION_TIME		
Tokudb_LOCKTREE_LONG_WAIT_ESCALATION_COUNT		
Tokudb_LOCKTREE_LONG_WAIT_ESCALATION_TIME		
Tokudb_DICTIONARY_UPDATES		
Tokudb_DICTIONARY_BROADCAST_UPDATES		

Имя	Тип	Область видимости
Tokudb_DESCRIPTOR_SET		
Tokudb_MESSAGES_IGNORED_BY_LEAF_DUE_TO_MSN		
Tokudb_TOTAL_SEARCH_RETRIES		
Tokudb_SEARCH_TRIES_GT_HEIGHT		
Tokudb_SEARCH_TRIES_GT_HEIGHTPLUS3		
Tokudb_LEAF_NODES_FLUSHED_NOT_CHECKPOINT		
Tokudb_LEAF_NODES_FLUSHED_NOT_CHECKPOINT_BYTES		
Tokudb_LEAF_NODES_FLUSHED_NOT_CHECKPOINT_UNCOMPRESSED_BYTES		
Tokudb_LEAF_NODES_FLUSHED_NOT_CHECKPOINT_SECONDS		
Tokudb_NONLEAF_NODES_FLUSHED_TO_DISK_NOT_CHECKPOINT		
Tokudb_NONLEAF_NODES_FLUSHED_TO_DISK_NOT_CHECKPOINT_BYTES		
Tokudb_NONLEAF_NODES_FLUSHED_TO_DISK_NOT_CHECKPOINT_UNCOMPRESSE		
Tokudb_NONLEAF_NODES_FLUSHED_TO_DISK_NOT_CHECKPOINT_SECONDS		
Tokudb_LEAF_NODES_FLUSHED_CHECKPOINT		
Tokudb_LEAF_NODES_FLUSHED_CHECKPOINT_BYTES		
Tokudb_LEAF_NODES_FLUSHED_CHECKPOINT_UNCOMPRESSED_BYTES		
Tokudb_LEAF_NODES_FLUSHED_CHECKPOINT_SECONDS		
Tokudb_NONLEAF_NODES_FLUSHED_TO_DISK_CHECKPOINT		
Tokudb_NONLEAF_NODES_FLUSHED_TO_DISK_CHECKPOINT_BYTES		
Tokudb_NONLEAF_NODES_FLUSHED_TO_DISK_CHECKPOINT_UNCOMPRESSED_BY		
Tokudb_NONLEAF_NODES_FLUSHED_TO_DISK_CHECKPOINT_SECONDS		
Tokudb_LEAF_NODE_COMPRESSION_RATIO		
Tokudb_NONLEAF_NODE_COMPRESSION_RATIO		
Tokudb_OVERALL_NODE_COMPRESSION_RATIO		
Tokudb_NONLEAF_NODE_PARTIAL_EVICTIONS		
Tokudb_NONLEAF_NODE_PARTIAL_EVICTIONS_BYTES		
Tokudb_LEAF_NODE_PARTIAL_EVICTIONS		
Tokudb_LEAF_NODE_PARTIAL_EVICTIONS_BYTES		
Tokudb_LEAF_NODE_FULL_EVICTIONS		
Tokudb_LEAF_NODE_FULL_EVICTIONS_BYTES		
Tokudb_NONLEAF_NODE_FULL_EVICTIONS		
Tokudb_NONLEAF_NODE_FULL_EVICTIONS_BYTES		

Имя	Тип	Область видимости
Tokudb_LEAF_NODES_CREATED		
Tokudb_NONLEAF_NODES_CREATED		
Tokudb_LEAF_NODES_DESTROYED		
Tokudb_NONLEAF_NODES_DESTROYED		
Tokudb_MESSAGES_INJECTED_AT_ROOT_BYTES		
Tokudb_MESSAGES_FLUSHED_FROM_H1_TO_LEAVES_BYTES		
Tokudb_MESSAGES_IN_TREES_ESTIMATE_BYTES		
Tokudb_MESSAGES_INJECTED_AT_ROOT		
Tokudb_BROADCASE_MESSAGES_INJECTED_AT_ROOT		
Tokudb_BASEMENTS_DECOMPRESSED_TARGET_QUERY		
Tokudb_BASEMENTS_DECOMPRESSED_PRELOCKED_RANGE		
Tokudb_BASEMENTS_DECOMPRESSED_PREFETCH		
Tokudb_BASEMENTS_DECOMPRESSED_FOR_WRITE		
Tokudb_BUFFERS_DECOMPRESSED_TARGET_QUERY		
Tokudb_BUFFERS_DECOMPRESSED_PRELOCKED_RANGE		
Tokudb_BUFFERS_DECOMPRESSED_PREFETCH		
Tokudb_BUFFERS_DECOMPRESSED_FOR_WRITE		
Tokudb_PIVOTS_FETCHED_FOR_QUERY		
Tokudb_PIVOTS_FETCHED_FOR_QUERY_BYTES		
Tokudb_PIVOTS_FETCHED_FOR_QUERY_SECONDS		
Tokudb_PIVOTS_FETCHED_FOR_PREFETCH		
Tokudb_PIVOTS_FETCHED_FOR_PREFETCH_BYTES		
Tokudb_PIVOTS_FETCHED_FOR_PREFETCH_SECONDS		
Tokudb_PIVOTS_FETCHED_FOR_WRITE		
Tokudb_PIVOTS_FETCHED_FOR_WRITE_BYTES		
Tokudb_PIVOTS_FETCHED_FOR_WRITE_SECONDS		
Tokudb_BASEMENTS_FETCHED_TARGET_QUERY		
Tokudb_BASEMENTS_FETCHED_TARGET_QUERY_BYTES		
Tokudb_BASEMENTS_FETCHED_TARGET_QUERY_SECONDS		
Tokudb_BASEMENTS_FETCHED_PRELOCKED_RANGE		
Tokudb_BASEMENTS_FETCHED_PRELOCKED_RANGE_BYTES		
Tokudb_BASEMENTS_FETCHED_PRELOCKED_RANGE_SECONDS		

Имя	Тип	Область видимости
Tokudb_BASEMENTS_FETCHED_PREFETCH		
Tokudb_BASEMENTS_FETCHED_PREFETCH_BYTES		
Tokudb_BASEMENTS_FETCHED_PREFETCH_SECONDS		
Tokudb_BASEMENTS_FETCHED_FOR_WRITE		
Tokudb_BASEMENTS_FETCHED_FOR_WRITE_BYTES		
Tokudb_BASEMENTS_FETCHED_FOR_WRITE_SECONDS		
Tokudb_BUFFERS_FETCHED_TARGET_QUERY		
Tokudb_BUFFERS_FETCHED_TARGET_QUERY_BYTES		
Tokudb_BUFFERS_FETCHED_TARGET_QUERY_SECONDS		
Tokudb_BUFFERS_FETCHED_PRELOCKED_RANGE		
Tokudb_BUFFERS_FETCHED_PRELOCKED_RANGE_BYTES		
Tokudb_BUFFERS_FETCHED_PRELOCKED_RANGE_SECONDS		
Tokudb_BUFFERS_FETCHED_PREFETCH		
Tokudb_BUFFERS_FETCHED_PREFETCH_BYTES		
Tokudb_BUFFERS_FETCHED_PREFETCH_SECONDS		
Tokudb_BUFFERS_FETCHED_FOR_WRITE		
Tokudb_BUFFERS_FETCHED_FOR_WRITE_BYTES		
Tokudb_BUFFERS_FETCHED_FOR_WRITE_SECONDS		
Tokudb_LEAF_COMPRESSION_TO_MEMORY_SECONDS		
Tokudb_LEAF_SERIALIZATION_TO_MEMORY_SECONDS		
Tokudb_LEAF_DECOMPRESSION_TO_MEMORY_SECONDS		
Tokudb_LEAF_DESERIALIZATION_TO_MEMORY_SECONDS		
Tokudb_NONLEAF_COMPRESSION_TO_MEMORY_SECONDS		
Tokudb_NONLEAF_SERIALIZATION_TO_MEMORY_SECONDS		
Tokudb_NONLEAF_DECOMPRESSION_TO_MEMORY_SECONDS		
Tokudb_NONLEAF_DESERIALIZATION_TO_MEMORY_SECONDS		
Tokudb_PROMOTION_ROOTS_SPLIT		
Tokudb_PROMOTION_LEAF_ROOTS_INJECTED_INTO		
Tokudb_PROMOTION_H1_ROOTS_INJECTED_INTO		
Tokudb_PROMOTION_INJECTIONS_AT_DEPTH_0		
Tokudb_PROMOTION_INJECTIONS_AT_DEPTH_1		
Tokudb_PROMOTION_INJECTIONS_AT_DEPTH_2		

Имя	Тип	Область видимости
Tokudb_PROMOTION_INJECTIONS_AT_DEPTH_3		
Tokudb_PROMOTION_INJECTIONS_LOWER_THAN_DEPTH_3		
Tokudb_PROMOTION_STOPPED_NONEMPTY_BUFFER		
Tokudb_PROMOTION_STOPPED_AT_HEIGHT_1		
Tokudb_PROMOTION_STOPPED_CHILD_LOCKED_OR_NOT_IN_MEMORY		
Tokudb_PROMOTION_STOPPED_CHILD_NOT_FULLY_IN_MEMORY		
Tokudb_PROMOTION_STOPPED_AFTER_LOCKING_CHILD		
Tokudb_BASEMENT_DESERIALIZATION_FIXED_KEY		
Tokudb_BASEMENT_DESERIALIZATION_VARIABLE_KEY		
Tokudb_PRO_RIGHTMOST_LEAF_SHORTCUT_SUCCESS		
Tokudb_PRO_RIGHTMOST_LEAF_SHORTCUT_FAIL_POS		
Tokudb_RIGHTMOST_LEAF_SHORTCUT_FAIL_REACTIVE		
Tokudb_CURSOR_SKIP_DELETED_LEAF_ENTRY		
Tokudb_FLUSHER_CLEANER_TOTAL_NODES		
Tokudb_FLUSHER_CLEANER_H1_NODES		
Tokudb_FLUSHER_CLEANER_HGT1_NODES		
Tokudb_FLUSHER_CLEANER_EMPTY_NODES		
Tokudb_FLUSHER_CLEANER_NODES_DIRTIED		
Tokudb_FLUSHER_CLEANER_MAX_BUFFER_SIZE		
Tokudb_FLUSHER_CLEANER_MIN_BUFFER_SIZE		
Tokudb_FLUSHER_CLEANER_TOTAL_BUFFER_SIZE		
Tokudb_FLUSHER_CLEANER_MAX_BUFFER_WORKDONE		
Tokudb_FLUSHER_CLEANER_MIN_BUFFER_WORKDONE		
Tokudb_FLUSHER_CLEANER_TOTAL_BUFFER_WORKDONE		
Tokudb_FLUSHER_CLEANER_NUM_LEAF_MERGES_STARTED		
Tokudb_FLUSHER_CLEANER_NUM_LEAF_MERGES_RUNNING		
Tokudb_FLUSHER_CLEANER_NUM_LEAF_MERGES_COMPLETED		
Tokudb_FLUSHER_CLEANER_NUM_DIRTIED_FOR_LEAF_MERGE		
Tokudb_FLUSHER_FLUSH_TOTAL		
Tokudb_FLUSHER_FLUSH_IN_MEMORY		
Tokudb_FLUSHER_FLUSH_NEEDED_IO		
Tokudb_FLUSHER_FLUSH_CASCADES		

Имя	Тип	Область видимости
Tokudb_FLUSHER_FLUSH_CASCADES_1		
Tokudb_FLUSHER_FLUSH_CASCADES_2		
Tokudb_FLUSHER_FLUSH_CASCADES_3		
Tokudb_FLUSHER_FLUSH_CASCADES_4		
Tokudb_FLUSHER_FLUSH_CASCADES_5		
Tokudb_FLUSHER_FLUSH_CASCADES_GT_5		
Tokudb_FLUSHER_SPLIT_LEAF		
Tokudb_FLUSHER_SPLIT_NONLEAF		
Tokudb_FLUSHER_MERGE_LEAF		
Tokudb_FLUSHER_MERGE_NONLEAF		
Tokudb_FLUSHER_BALANCE_LEAF		
Tokudb_HOT_NUM_STARTED		
Tokudb_HOT_NUM_COMPLETED		
Tokudb_HOT_NUM_ABORTED		
Tokudb_HOT_MAX_ROOT_FLUSH_COUNT		
Tokudb_TXN_BEGIN		
Tokudb_TXN_BEGIN_READ_ONLY		
Tokudb_TXN_COMMITS		
Tokudb_TXN_ABORTS		
Tokudb_LOGGER_NEXT_LSN		
Tokudb_LOGGER_WRITES		
Tokudb_LOGGER_WRITES_BYTES		
Tokudb_LOGGER_WRITES_UNCOMPRESSED_BYTES		
Tokudb_LOGGER_WRITES_SECONDS		
Tokudb_LOGGER_WAIT_LONG		
Tokudb_LOADER_NUM_CREATED		
Tokudb_LOADER_NUM_CURRENT		
Tokudb_LOADER_NUM_MAX		
Tokudb_MEMORY_MALLOC_COUNT		
Tokudb_MEMORY_FREE_COUNT		
Tokudb_MEMORY_REALLOC_COUNT		
Tokudb_MEMORY_MALLOC_FAIL		

Имя	Тип	Область видимости
Tokudb_MEMORY_REALLOC_FAIL		
Tokudb_MEMORY_REQUESTED		
Tokudb_MEMORY_USED		
Tokudb_MEMORY_FREED		
Tokudb_MEMORY_MAX_REQUESTED_SIZE		
Tokudb_MEMORY_LAST_FAILED_SIZE		
Tokudb_MEM_ESTIMATED_MAXIMUM_MEMORY_FOOTPRINT		
Tokudb_MEMORY_MALLOCATOR_VERSION		
Tokudb_MEMORY_MMAP_THRESHOLD		
Tokudb_FILESYSTEM_THREADS_BLOCKED_BY_FULL_DISK		
Tokudb_FILESYSTEM_FSYNC_TIME		
Tokudb_FILESYSTEM_FSYNC_NUM		
Tokudb_FILESYSTEM_LONG_FSYNC_TIME		
Tokudb_FILESYSTEM_LONG_FSYNC_NUM		

Последнее обновление: 2024-04-30

8.3 Сравнение функций

MyDB Сервер для *MySQL* — это свободно доступная, распространяемая с открытым исходным кодом, полностью совместимая и расширенная замена любой базы данных MySQL. Продукт обеспечивает превосходную и оптимизированную производительность, более высокую масштабируемость и доступность, а также улучшенные функции резервного копирования, мониторинга и управления.

Предприятия доверяют MyDB Сервер для MySQL, поскольку он обеспечивает лучшую производительность, надёжность и масштабируемость для самых требовательных рабочих нагрузок.

Мы обеспечиваем эти преимущества, постоянно улучшая *MyDB* Сервер для *MySQL* как по сравнению с базовым продуктом *Percona Server for MySQL* от компании Percona, так и по сравнению со стандартным сервером базы данных *MySQL*:

Особенности	**MyDB Сервер для MySQL 8.4**	**MySQL 8.4**
Открытый исходный код	Да	Да
Соответствие требованиям ACID	Да	Да
Многоверсионное управление параллелизмом	Да	Да
Блокировка на уровне строк	Да	Да
Автоматическое восстановление после сбоя	Да	Да
Разделение таблиц	Да	Да
Представления	Да	Да
Подзапросы	Да	Да
Триггеры	Да	Да
Хранимые процедуры	Да	Да
Внешние ключи	Да	Да
Оконные функции	Да	Да
Общие табличные выражения	Да	Да
Геопространственные объекты (ГИС, SRS)	Да	Да
Репликация GTID	Да	Да
Групповая репликация	Да	Да
Механизм хранения MyRocks	Да	Нет

Улучшения для разработчиков	**MyDB Сервер для MySQL 8.4**	**MySQL 8.4**
NoSQL-интерфейс уровня сокетов	Да	Да
Поддержка X API	Да	Да
Функции JSON	Да	Да
Улучшения полнотекстового поиска InnoDB	Да	Нет
Дополнительные функции хеширования/ дайджеста	Да	Нет

Возможности мониторинга и устранения неполадок	**MyDB Сервер для MySQL 8.4**	**MySQL 8.4**
Таблицы INFORMATION_SCHEMA	95	65
Глобальные счетчики производительности и состояния	853	434
Гистограммы оптимизатора	Да	Да
Счетчики производительности для каждой таблицы	Да	Нет
Счетчики производительности по индексам	Да	Нет
Счетчики производительности для каждого пользователя	Да	Нет
Счетчики производительности для каждого клиента	Да	Нет
Счетчики производительности на поток	Да	Нет
Глобальная статистика времени ответа на запрос	Да	Нет
Расширенный SHOW ENGINE INNODB STATUS	Да	Нет
Информация о сегментах Undo	Да	Нет
Информация о временных таблицах	Да	Нет
Расширенный журнал медленных запросов	Да	Нет
Статистика пользователя	Да	Нет

Характеристики производительности и масштабируемости	**MyDB Сервер для MySQL 8.4**	**MySQL 8.4**
Группы ресурсов InnoDB	Да	Да
Настраиваемые размеры страниц	Да	Да
Планирование транзакций с учетом конфликтов	Да	Да
Улучшенная масштабируемость за счет разделения мьютексов	Да	Нет
Улучшенный механизм хранения MEMORY	Да	Нет
Улучшенный сброс данных на диск	Да	Нет
Параллельный буфер двойной записи	Да	Да
Настраиваемое быстрое создание индекса)	Да	Нет
Сжатие по столбцам для VARCHAR/BLOB и JSON	Да	Нет
Сжатые столбцы со словарями	Да	Нет

Функции безопасности	MyDB Сервер для MySQL 8.4	MySQL 8.4
Роли SQL	Да	Да
Хеширование паролей на основе SHA-2	Да	Да
Политика ротации паролей	Да	Да
Плагин аутентификации PAM	Да	Только в версии Enterprise
Плагин ведения журнала аудита	Да	Только в версии Enterprise

Функции шифрования	MyDB Сервер для MySQL 8.4	MySQL 8.4
Хранение связки ключей в файле	Да	Да
Хранение связки ключей в хранилище Hashicorp	Да	Только в версии Enterprise
Шифрование данных InnoDB	Да	Да
Шифрование журналов InnoDB	Да	Да
Шифрование встроенных табличных пространств InnoDB (Общее, Системное, Undo, Временное)	Да	Да
Шифрование бинарных журналов	Да	Нет
Шифрование временных файлов	Да	Нет
Принудительное шифрование	Да	Нет

Операционные улучшения	MyDB Сервер для MySQL 8.4	MySQL 8.4
Атомарные DDL	Да	Да
Словарь транзакционных данных	Да	Да
Мгновенный DDL	Да	Да
SET PERSIST	Да	Да
Невидимые индексы	Да	Да
Пул потоков	Да	Только в версии Enterprise
Резервные блокировки	Да	Нет
Расширенная команда SHOW GRANTS	Да	Нет
Улучшенная обработка поврежденных таблиц	Да	Нет
Возможность остановки простаивающих транзакций	Да	Нет
Улучшенная команда START TRANSACTION WITH CONSISTENT SNAPSHOT	Да	Нет

Возможности базы данных как сервиса (DBaaS)	MyDB Сервер для MySQL 8.4	MySQL 8.4
Принудительное использование определенного механизма хранения	Да	Да

Последнее обновление: 2024-10-21

8.4 Схема нумерации версий

Номер версии идентифицирует выпуск продукта. Продукт содержит последние общедоступные функции (GA) на момент выпуска.

8.4.0	-1	.2
Базовая версия Oracle	Дополнительная версия сборки Percona	Дополнительная версия сборки MyDB

MyDB использует семантическую нумерацию версий, которая соответствует шаблону базовой версии, дополнительной версии сборки базового продукта *Percona Server for MySQL* и дополнительной версии сборки *MyDB* Сервер для *MySQL*. MyDB, так же, как и компания Percona, назначает уникальные неотрицательные целые числа в порядке возрастания для каждой дополнительной версии сборки. Номер версии объединяет базовый номер версии MySQL 8.4, дополнительную версию сборки Percona и дополнительную версию сборки MyDB.

Например, номер версии MyDB Сервер для MySQL 8.4.0-1.2 определяют следующую информацию:

- Базовая версия — самые левые цифры обозначают версию MySQL 8.0, используемую в качестве базовой.
- Дополнительная версия сборки Percona — средний номер, который увеличивается на единицу при каждом выпуске базового продукта *Percona Server for MySQL*.
- Дополнительная версия сборки MyDB — самый правый номер, который увеличивается на единицу при каждом выпуске *MyDB* Сервер для *MySQL*.

MyDB Сервер для MySQL 8.4.0-1.2 и 8.4.0-1.3 основаны на Percona Server for MySQL 8.4.0-1, который в свою очередь основан на MySQL 8.4.0.

Последнее обновление: 2024-04-30

8.5 Разработка MyDB Сервер для MySQL

MyDB Сервер для *MySQL* — это проект с открытым исходным кодом, предназначенный для создания дистрибутива сервера *MySQL* с улучшенной производительностью, масштабируемостью и диагностикой.

8.5.1 Отправка изменений

Мы основную ветвь исходного кода в постоянном состоянии стабильности, чтобы обеспечить возможность выпуска в любое время и минимизировать потерю времени разработчиков из-за неработающего кода.

Обзор рабочего процесса

В MyDB мы используем Git для контроля версий, Gitflic для хостинга кода и для управления выпусками.

Мы изменяем наше программное обеспечение, чтобы реализовать новые функции и/или исправить ошибки. Рефакторинг можно классифицировать либо как новую функцию, либо как исправление ошибки в зависимости от объема работы.

Передать код можно в виде запроов на слияние на Gitflic.

Для *MyDB* Сервер для *MySQL* у нас есть несколько веток Git, в которых происходит разработка. В настоящий момент это 8.0, и 8.4. Поскольку *MyDB* Сервер для *MySQL* не является традиционным проектом, а представляет вместо этого набор исправлений для существующего продукта, эти ветки не согласованы. Другими словами, мы не переносим изменения из одной ветки в другую автоматически. Чтобы включить ваши изменения в несколько веток, вы должны предложить к слиянию несколько веток.

Внесение изменений в проект

- `git branch https://gitflic.ru/project/mydb-ru/mydb-server/featureX` (где 'featureX' — это некоторое название для поставленной задачи)
- (разработчик вносит изменения в featureX, тестируя локально)
- Разработчик отправляет локальные изменения в `https://gitflic.ru/username/mydb-server/featureX`
- Разработчик может отправить запрос на слияние в https://gitflic.ru/project/mydb-ru/mydb-server,
- Код проходит проверку
- Как только код пройдёт проверку разработчиками и будет принят, его можно будет объединить.

Если изменение также применимо к стабильной версии (например, 8.0), то изменения должны быть сделаны на ветке 8.0 и предложены к слиянию в основную ветку (например, 8.4). В этом случае должны быть две ветки, проходящие через сборку параметров, и два

предложения по слиянию (одно для стабильной версии и версии с изменениями, добавленными в основную ветку). Это предотвращает ситуацию, когда кто-то другой должен догадаться, как объединить ваши изменения с каждой из поддерживаемых веток репозитория.

Последнее обновление: 2024-04-30

docs@mydb.ru

8.6 Политика в отношении товарных знаков

Настоящая Политика в отношении товарных знаков призвана гарантировать, что пользователи продуктов или услуг под брендом MyDB знают, что то, что они получают, действительно было разработано, одобрено, протестировано и поддерживается компанией MyDB. Товарные знаки помогают предотвратить путаницу на рынке, отличая продукты и услуги одной компании или человека от продуктов и услуг другой.

MyDB владеет рядом товарных знаков, включая, помимо прочего, MyDB и MyDB Сервер, а также отличительные визуальные значки и логотипы, связанные с этими знаками. Как незарегистрированные, так и зарегистрированные знаки MyDB защищены.

Использование любого товарного знака MyDB в имени, URL-адресе или другой идентифицирующей характеристике любого продукта, услуги, веб-сайта или иного использования не допускается без письменного разрешения MyDB, за следующими тремя ограниченными исключениями.

Во-первых, вы можете использовать соответствующий знак MyDB при упоминании добросовестного использования подлинного продукта MyDB.

Во-вторых, когда MyDB выпускает продукт под лицензией GNU General Public License («GPL»), вы можете использовать соответствующий знак MyDB при распространении дословной копии этого продукта в соответствии с условиями GPL.

В-третьих, вы можете использовать соответствующий знак MyDB для обозначения дистрибутива программного обеспечения MyDB, выпущенного под лицензией GPL, который был модифицирован с незначительными изменениями с единственной целью — позволить этому программному обеспечению работать на операционной системе или аппаратной платформе, для которой MyDB еще не выпустила программное обеспечение при условии, что эти изменения третьих лиц не влияют на поведение, функциональность, функции, дизайн или производительность программного обеспечения. Пользователи, приобретающие это программное обеспечение под брендом MyDB, получают практически точную реализацию программного обеспечения MyDB.

MyDB оставляет за собой право отозвать это разрешение в любое время по своему усмотрению. Например, если MyDB считает, что ваше изменение выходит за рамки ограниченной лицензии, предоставленной в настоящей Политике, или что использование вами знака MyDB наносит ущерб компании MyDB, MyDB отзовет это разрешение. После отзыва вы должны немедленно прекратить использование соответствующего знака MyDB. Если вы немедленно не прекратите использование знака MyDB после отзыва, компания MyDB может принять меры для защиты своих прав и интересов в отношении знака MyDB. Компания MyDB не предоставляет никаких лицензий на использование каких-либо знаков MyDB для любых других модифицированных версий программного обеспечения MyDB; такое использование потребует нашего предварительного письменного разрешения.

Ни закон о товарных знаках, ни какие-либо исключения, изложенные в настоящей Политике в отношении товарных знаков, не разрешают вам сокращать, изменять или иным образом использовать любой знак MyDB как часть вашего собственного бренда. Например, если компания XYZ создает модифицированную версию *MyDB* Сервер для *MySQL* , XYZ не может

маркировать эту модификацию как «XYZ MyDB Сервер» или «MyDB XYZ Server», даже если эта модификация в остальном соответствует третьему исключению, указанному выше.

Во всех случаях вы должны соблюдать действующее законодательство, основную лицензию и настоящую Политику в отношении товарных знаков, в которую время от времени вносятся поправки. Например, любое упоминание товарных знаков MyDB должно включать полное название товарного знака с правильным написанием и заглавными буквами, а также указание права собственности на ООО «Моя База Данных».

В случае сомнений относительно любого из условий или исключений, изложенных в настоящей Политике в отношении товарных знаков, свяжитесь с нами по адресу info@mydb.ru для получения помощи, и мы сделаем все возможное, чтобы вам помочь.

Последнее обновление: 2024-04-30

8.7 Указатель таблиц INFORMATION_SCHEMA

Это список таблиц `INFORMATION_SCHEMA`, которые существуют в *MyDB* Сервер для *MySQL*. Запись для каждой таблицы указывает на страницу документации, на которой описан элемент.

- INFORMATION_SCHEMA.CLIENT_STATISTICS
- INFORMATION_SCHEMA.GLOBAL_TEMPORARY_TABLES
- INFORMATION_SCHEMA.INDEX_STATISTICS
- PROCFS
- INFORMATION_SCHEMA.TABLE_STATISTICS
- INFORMATION_SCHEMA.TEMPORARY_TABLES
- THREAD_STATISTICS
- INFORMATION_SCHEMA.USER_STATISTICS
- INFORMATION_SCHEMA.XTRADB_ZIP_DICT
- INFORMATION_SCHEMA.XTRADB_ZIP_DICT_COLS

Последнее обновление: 2024-04-30

8.8 Часто задаваемые вопросы

8.8.1 Вопрос: Придется ли нам использовать лицензию *GPL* для нашего приложения, если мы будем использовать *MyDB* Сервер для *MySQL*?

Ответ: Это распространенное заблуждение относительно *GPL*. Мы предлагаем прочитать отличный справочный материал от Free Software Foundation по лицензии GPL версии 2, под которой распространяется *MySQL* и, следовательно, *MyDB* Сервер для *MySQL*. Этот документ содержит ссылки на многие другие документы, которые должны ответить на ваши вопросы. *MyDB* не может дать юридическую консультацию по поводу *GPL*.

8.8.2 Вопрос: Нужно ли мне устанавливать клиентские библиотеки *MyDB*?

Ответ: Нет, на клиентах ничего менять не нужно. *MyDB* Сервер для *MySQL* 100% совместим на уровне протокола и диалекта SQL со всеми существующими клиентскими библиотеками, коннекторами и приложениями для MySQL и Percona Server.

Последнее обновление: 2024-04-30

8.9 Информация об авторских правах и лицензировании

8.9.1 Лицензирование документации

Документация MyDB Сервер для MySQL является переведённой и адаптированной версией документации к продукту Percona Server for MySQL от компании Percona.

8.9.2 Лицензия на программное обеспечение

MyDB Сервер для *MySQL* создан на базе продукта Percona Server for MySQL от компании Percona, который в свою очередь создан на базе продукта MySQL от компании Oracle. Наряду с созданием собственных модификации, мы объединяем изменения из других источников, таких как вклады сообщества сторонних разработчиков и изменения из продуктов компании MariaDB.

MyDB не требует передачи авторских прав.

См. файлы COPYING, прилагаемые к каждому дистрибутиву программного обеспечения MyDB.

Последнее обновление: 2024-04-30

8.10 Глоссарий

8.10.1 ACID

Набор свойств, гарантирующих надежную обработку транзакций базы данных. Означает *Atomicity* (Атомарность), *Consistency* (Согласованность), *Isolation* (Изоляция), и *Durability* (Долговечность.)

8.10.2 Атомарность

Атомарность означает, что операции с базой данных применяются по правилу «все или ничего». Транзакция либо применяется полностью, либо не применяется вообще.

8.10.3 Согласованность

Согласованность означает, что каждая транзакция, изменяющая базу данных, переводит ее из одного согласованного состояния в другое.

8.10.4 Долговечность

Как только транзакция будет зафиксирована, она останется на долговечном носителе и не будет потеряна даже в случае отказа сервера.

8.10.5 Внешний ключ

Ссылочное ограничение между двумя таблицами. Пример: заказ на покупку в таблице `Purchase_orders` должен быть сделан клиентом, который существует в таблице клиентов.

8.10.6 Общая доступность (GA)

Завершённая версия продукта, доступная для широкой публики. Это заключительный этап цикла выпуска программного обеспечения.

8.10.7 Изоляция

Требование изоляции означает, что ни одна транзакция не может мешать другой.

8.10.8 InnoDB

Механизм хранения для MySQL и производных (MyDB Сервер для MySQL, Percona Server и MariaDB) первоначально написанный Innobase Oy, а затем приобретенный Oracle. Он предоставляет с ACID-совместимое управление данными с поддержкой внешних ключей. Начиная с MySQL версии 5.5, InnoDB стал механизмом хранения по умолчанию на всех платформах.

8.10.9 LSN

Порядковый номер журнала (Log Sequence Number, LSN) представляет собой 8-байтовое число. Каждое изменение данных добавляет запись в журнал REDO и генерирует номер LSN. Сервер увеличивает LSN при каждом изменении.

8.10.10 MariaDB

Ответвление MySQL, поддерживаемое в основном компаниями MariaDB Foundation и MariaDB Corporation. Их цель — добавлять функции и исправлять ошибки, предоставляя альтернативу для сообщества MySQL.

8.10.11 `my.cnf`

Имя файла конфигурации MySQL по умолчанию.

8.10.12 MyDB Сервер для MySQL

Ответвление Percona Server от компании MyDB с целью поддержки российской части MySQL-экосистемы.

8.10.13 MyISAM

Устаревший механизм хранения для MySQL, который использовался по умолчанию до MySQL 5.5.

8.10.14 MyRocks

Альтернативный ACID-совместимый механизм хранения на основе RocksDB. Благодаря использованию LSM-деревьев MyRocks обеспечивает более эффективное использование доступного пространства и пропускной способности дисковой системы, а также ускоренную репликацию и загрузку данных. MyRocks доступен в производных MySQL таких как MyDB Сервер для MySQL, Percona Server и MariaDB.

8.10.15 MySQL

База данных с открытым исходным кодом, породившая несколько дистрибутивов и ответвлений. Компания MySQL AB была основным сопровождающим и дистрибьютором MySQL до тех пор, пока ее не купила компания Sun Microsystems, которая затем была приобретена Oracle. Поскольку Oracle владеет товарным знаком MySQL, термин MySQL часто используется для обозначения дистрибутива MySQL от Oracle в отличие от его замен, таких как MyDB Сервер для MySQL, Percona Server и MariaDB.

8.10.16 NUMA

Non-Uniform Memory Access (NUMA) — это архитектура компьютерной памяти, используемая в многопроцессорной обработке, где время доступа к памяти зависит от местоположения памяти относительно процессора. В рамках NUMA процессор может получить доступ к своей

локальной памяти быстрее, чем к нелокальной памяти, то есть к памяти, локальной для другого процессора, или к памяти, совместно используемой между процессорами. Вся система по-прежнему может работать как единое целое, и вся память практически доступна отовсюду, но с потенциально более высокой задержкой и меньшей производительностью.

8.10.17 Percona Server for MySQL

Ответвление MySQL от компании Percona с улучшениями производительности и управления.

8.10.18 Механизм хранения

Механизм хранения (storage engine) — это часть программного обеспечения, которая реализует детали хранения и поиска данных для системы базы данных. Этот термин в основном используется в экосистеме MySQL, поскольку это первая широко используемая реляционная база данных, реализовавшая уровень абстракции вокруг хранилища. Это аналог уровня виртуальной файловой системы в операционной системе. Уровень VFS позволяет операционной системе читать и записывать несколько файловых систем (например, FAT, NTFS, XFS, ext3), а уровень Storage Engine позволяет серверу базы данных получать доступ к таблицам, хранящимся в разных механизмах (например, MyISAM, InnoDB или MyRocks).

8.10.19 Техническая предварительная версия

Элементом технической предварительной версии может быть функция, переменная или значение переменной. Перед использованием этой функциональности в рабочей среде мы рекомендуем протестировать восстановление из резервных копий в вашей среде, а также использовать альтернативный метод резервного копирования для обеспечения избыточности. Элементы технической предварительной версии включаются в выпуски для того, чтобы позволить пользователям протестировать функциональность и оставить отзыв. Такая функциональность либо обновляется и выпускается в качестве "общая доступность (GA)", либо удаляется, если она не оказалась полезной. Функциональность может измениться между стадиями технической предварительной версии и общей доступности.

8.10.20 XtraDB

Улучшенная версия InnoDB от компании Percona и включённая в ответвлениях Percona Server и MyDB Сервер для MySQL.

Последнее обновление: 2024-04-30