

MySQL (Русский)



Эта страница нуждается в сопроводителе



Статья не гарантирует актуальность информации. Помогите русскоязычному сообществу поддержкой подобных страниц. См. [Команда переводчиков ArchWiki](#)

MySQL - это широко применяемая свободная многопоточная многопользовательская система управления реляционными базами данных. MySQL использует один из диалектов языка [SQL](#) для управления базами данных. Более подробную информацию об особенностях MySQL можно посмотреть на [официальном сайте](#).

Примечание: На сегодняшний день в Archlinux к использованию по умолчанию предлагается MariaDB - официальный форк сервера MySQL. Настоятельно рекомендуется всем пользователям обновиться до [MariaDB](#). Oracle MySQL был перемещен из официальных репозиториях Archlinux в AUR. Для получения более подробной информации прочтите [объявление](#).

Contents

[hide]

- 1 [Установка](#)
 - 1.1 [Автоматический запуск при загрузке системы](#)
 - 1.2 [Апгрейд с Oracle MySQL до MariaDB](#)
 - 1.3 [После обновления](#)
- 2 [Настройка](#)
 - 2.1 [Добавление пользователя](#)
 - 2.2 [Включение удалённого доступа](#)
 - 2.3 [Отключение удаленного доступа](#)
 - 2.4 [Режим автодополнения](#)
 - 2.5 [Использование кодировки UTF-8](#)
 - 2.6 [Использование файловой системы TMPFS для каталога tmpdir](#)
- 3 [Бэкап](#)
- 4 [Поиск и устранение неисправностей](#)
 - 4.1 [Не запускается демон MySQL](#)
 - 4.2 [Не могу запустить mysql_upgrade из-за невозможности запуска MySQL](#)
 - 4.3 [Сброс пароля для root](#)
 - 4.4 [Проверка и восстановление всех таблиц](#)
 - 4.5 [Оптимизация всех таблиц](#)
- 5 [Смотрите также](#)

Установка

MariaDB - так именуется форк популярного сервера баз данных MySQL, выбранный сообществом Archlinux. [Установите](#) пакет [mariadb](#) из [официальных репозиториях](#). В качестве альтернативы вы можете установить, например:

- **Oracle MySQL** — Официальный релиз от компании Oracle.
<https://www.mysql.com/> || [mysql](#)^{AUR}
- **Percona Server** — Альтернатива, которая обещает высокую производительность и разные широкие возможности.
<http://www.percona.com/software/percona-server/> || [percona-server](#)

Tip: Если база данных, размещаемая в каталоге `/var/lib/mysql` и расположена в разделе с файловой системой [btrfs](#), то необходимо отключить механизм [Copy-on-Write](#) для этого каталога перед тем как создавать новую базу данных:

```
# chattr +C /var/lib/mysql
```

Перед запуском демона `mysqld` обязательно необходимо выполнить команду для создания таблиц по умолчанию:

```
# mysql_install_db --user=mysql --basedir=/usr --datadir=/var/lib/mysql
```

Следующая команда в интерактивном режиме проведет Вас по ряду рекомендуемых мер безопасности для базы данных:

```
# mysql_secure_installation
```

При MariaDB по стандарту запустится только так:

Warning: For security reasons, the Systemd service file for this package will prevent MariaDB from using a `datadir` under the `$HOME` hierarchy

Now the `mariadb.service` can be started and/or enabled with [systemd](#).

Если есть необходимость во фронендах к MariaDB, то можете использовать [mysql-gui-tools](#) AUR[[ссылка недействительна](#): сохранено в [aur-mirror](#)] или [mysql-workbench](#).

Автоматический запуск при загрузке системы

Для автоматического запуска сервера баз данных MySQL при загрузке операционной системы, добавьте демон `mysqld` в систему инициализации [systemd](#).

```
# systemctl enable mysqld
# systemctl start mysqld
```

Апгрейд с Oracle MySQL до MariaDB

Примечание: Перед перезапуском демона `mysqld` на следующем шаге, возможно, понадобится удалить следующие файлы из каталога `/var/lib/mysql`, а именно: `ib_logfile0`, `ib_logfile1` и `aria_log_control`.

Пользователи, желающие произвести переход, должны остановить процесс `mysqld`, установить `mariadb`, `libmariadbclient` или `mariadb-clients`, после чего перезапустить `mysqld` и выполнить:

```
# mysql_upgrade -p
```

После обновления

После обновления MySQL можно запустить полезную утилиту `mysql_upgrade` для автоматической проверки и обновления MySQL-таблиц на предмет совместимости структур данных с текущей версией MySQL.

```
# mysql_upgrade -u root -p
```

Настройка

После установки и запуска MySQL необходимо настроить учетную запись `root` для администрирования MySQL. Задайте пароль для пользователя `root`. Это можно осуществить вручную или автоматически на стадии выполнения предыдущего

скрипта. Ручная установка пароля для root возможна при помощи утилиты `mysqladmin`

```
mysqladmin -u root password newpass
```

MySQL построен по клиент-серверной архитектуре. Это значит, что систему управления базами данных MySQL можно трактовать как сервер, обменивающийся сообщениями с MySQL-клиентами. Консольный клиент, запускаемый в терминале при помощи команды `mysql`, позволяет подключиться к серверу баз данных MySQL. Осуществим такое подключение от имени пользователя root:

```
$ mysql -p -u root
```

Добавление пользователя

Создание нового пользователя состоит из двух действий: создание пользователя; установка прав. Пример: создадим пользователя *monty* с паролем *some_pass*.

```
$ mysql -u root -p

MariaDB> CREATE USER 'monty'@'localhost' IDENTIFIED BY 'some_pass';
MariaDB> GRANT ALL PRIVILEGES ON *.* TO 'monty'@'localhost'
-> WITH GRANT OPTION;
MariaDB> quit
```

Включение удалённого доступа

Если Вы хотите подключаться к Вашему серверу MySQL удалённо с других хостов, то Вам необходимо раскомментировать (убрать символ #) следующие строки в файле `/etc/mysql/my.cnf`: `[mysqld]`

```
...
#skip-networking
bind-address = <some ip-address>
...
```

Таким образом Вы позволите удалённо подключаться любому пользователю (в том числе и root): Подключить к базе с правами пользователя root:

```
$ mysql -u root -p -h IPorHOSTNAME
```

Проверка прав пользователей на удаленное подключение:

```
SELECT User, Host FROM mysql.user WHERE Host <> 'localhost';
```

Предоставить права удаленного доступа пользователю (в данном случае root):

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'192.168.1.%' IDENTIFIED BY  
'my_optional_remote_password' WITH GRANT OPTION;
```

При необходимости шаблон '%' можно заменить на конкретное значение хоста.
Пароль пользователя в базе может отличаться от пароля пользователя в системе.

Отключение удаленного доступа

MySQL-сервер по умолчанию доступен из внешней сети. Если вы используете MySQL локально, то можно усилить безопасность, отключив прослушивание TCP-порта 3306. Для отключения удаленного доступа раскомментируйте в файле `/etc/mysql/my.cnf` строку:

```
skip-networking
```

Теперь вы сможете соединиться с MySQL-сервером только через localhost.

Режим автодополнения

Примечание: Активация этого режима может увеличить время инициализации MySQL-клиента.

По умолчанию механизм автодополнения команд и имен в клиенте mysql отключен. Для включения автоматического дополнения отредактируйте общий конфигурационный файл `/etc/mysql/my.cnf`, заменив параметр `no-auto-rehash` на параметр `auto-rehash`. Автодополнение вступит в силу после очередного запуска клиента mysql.

Использование кодировки UTF-8

В файле `/etc/mysql/my.cnf` добавьте к настройкам секции `mysqld` следующие параметры:

```
[mysqld]  
init_connect          = 'SET collation_connection =  
utf8_general_ci,NAMES utf8'  
collation_server      = utf8_general_ci  
character_set_client  = utf8  
character_set_server  = utf8
```

Использование файловой системы TMPFS для каталога tmpdir

MySQL использует специальный каталог `tmpdir` для хранения временных файлов (например, для хранения временных таблиц). На самом деле `tmpdir` является внутренним алиасом MySQL, под маской которого может скрываться произвольный каталог.

Создаем этот каталог с соответствующими правами доступа:

```
# mkdir -pv /var/lib/mysqltmp  
# chown mysql:mysql /var/lib/mysqltmp
```

Определим UID и GID для пользователя `mysql`:

```
$ id mysql
```

```
uid=27(mysql) gid=27(mysql) groups=27(mysql)
```

Добавим в файл `/etc/fstab` строку вида:

```
tmpfs    /var/lib/mysqltmp    tmpfs
rw,gid=27,uid=27,size=100m,mode=0750,noatime    0 0
```

Не забудем в файле `/etc/mysql/my.cnf` связать алиас `tmpdir` с реальным каталогом. Добавьте к группе настроек секции `mysqld` строчку:

```
tmpdir    = /var/lib/mysqltmp
```

Перезагрузитесь или (остановите сервер MySQL, примонтируйте `tmpdir`, запустите сервер MySQL).

Бэкап

Легкий бэкап баз данных можно осуществить с помощью утилиты `mysqldump`. Ниже следующий скрипт помещает дамп всех баз данных в файл `db_backup.gz`, который будет расположен в той же папке, что и сам скрипт, и удаляет все бинарные логи старше одной недели:

```
#!/bin/bash

THISDIR=$(dirname $(readlink -f "$0"))

mysqldump --single-transaction --flush-logs --master-data=2 --all-
databases -u root -p | gzip > $THISDIR/db_backup.gz
mysql -u root -p -e 'purge master logs before date_sub(now(), interval
7 day);'
```

Смотрите также страницу официального [руководства по работе с mysqldump](#).

Поиск и устранение неисправностей

Не запускается демон MySQL

Если MySQL не запускается, а в логах отсутствуют записи, то можно проверить содержимое каталогов `/var/lib/mysql` и `/var/lib/mysql/mysql` на предмет прав доступа. Если владельцами файлов в этих директориях является не пользователь `mysql` из одноименной группы, то сделайте следующее:

```
# chown mysql:mysql /var/lib/mysql -R
```

Если у вас остаются проблемы, связанные с доступом, несмотря на вышеозначенную рекомендацию, попробуйте скопировать `my.cnf` в `/etc/`. Для этого выполните команду:

```
# cp /etc/mysql/my.cnf /etc/my.cnf
```

Теперь попробуйте [запустить](#) [broken link: invalid section] `mysqld`.

Ежели вы получите в файле `/var/lib/mysql/hostname.err` такие строки:

```
[ERROR] Can't start server : Bind on unix socket: Permission denied
[ERROR] Do you already have another mysqld server running on socket:
/var/run/mysqld/mysqld.sock ?
[ERROR] Aborting
```

Виной этому могут быть некорректные права доступа на `/var/run/mysqld`.
Выполните:

```
# chown mysql:mysql /var/run/mysqld -R
```

Если вы запустили `mysqld`, но получили следующее сообщение об ошибке:

```
Fatal error: Can't open and lock privilege tables: Table 'mysql.host'
doesn't exist
```

Выполните следующую команду из каталога `/usr` для создания таблиц по умолчанию:

```
# cd /usr
# mysql_install_db --user=mysql --ldata=/var/lib/mysql/
```

Не могу запустить `mysql_upgrade` из-за невозможности запуска MySQL

Попытайтесь запустить MySQL в безопасном режиме:

```
# mysqld_safe --datadir=/var/lib/mysql/
```

После этого выполните:

```
# mysql_upgrade -u root -p
```

Сброс пароля для root

[Остановите](#) [broken link: invalid section] демона `mysqld` и выполните:

```
# mysqld_safe --skip-grant-tables &
```

Подсоединитесь к серверу MySQL:

```
# mysql -u root mysql
```

Измените пароль для пользователя root:

```
mysql> UPDATE mysql.user SET Password=PASSWORD('MyNewPass') WHERE
User='root';
mysql> FLUSH PRIVILEGES;
mysql> exit
```

Запустите [\[broken link: invalid section\]](#) mysqld.

Проверка и восстановление всех таблиц

Можно выполнить автоматическую проверку и восстановление всех таблиц во всех базах данных. Более подробно смотрите [здесь](#).

```
# mysqlcheck -A --auto-repair -u root -p
```

Оптимизация всех таблиц

Для принудительной оптимизации всех таблиц с автоматической фиксацией возникающих ошибок, выполните:

```
# mysqlcheck -A --auto-repair -f -o -u root -p
```

Смотрите также

- [LAMP](#) - статья на ArchWiki, описывающая установку и базовую настройку LAMP-сервера (Linux, Apache, MySQL, PHP).
- [PhpMyAdmin](#) - статья на ArchWiki, описывающая веб-ориентированный фронтенд для управления базами данных MySQL.
- [PHP](#) - статья на ArchWiki, посвященная установке и настройке интерпретатора PHP.
- [MySQL Performance Tuning Scripts and Know-How](#) - англоязычная статья про MySQL.

Categories:

- [Русский](#)
- [Database management systems \(Русский\)](#)