# Как импортировать и экспортировать базы данных в MySQL или MariaDB

Управление базами данных — важная часть в администрировании Linux-серверов и веб-приложений. Импорт и экспорт баз данных являются незыблемыми столпами в DevOps и системном администрировании.

Как минимум, любой разработчик должен уметь создавать резервные копии и переносить их между серверами. В этой статье будет рассмотрен **импорт и экспорт баз данных MySQL и MariaDB**.

Для того, чтобы выполнить дальнейшие шаги, вам нужен будет Linux-сервер (в инструкции все действия выполняются на Ubuntu). Арендовать виртуальный сервер вы можете у нас по [этой ссылке](https://timeweb.cloud/my/servers/create).

## Работа с базой данных

MySQL и MariaDB — это популярные реляционные СУБД, которые используются для хранения данных больших приложений.

MariaDB – это модифицированная версия MySQL. Сервис MariaDB был создан первоначальной командой разработчиков MySQL из-за проблем с лицензированием и распространением после того, как базу данных MySQL приобрела корпорация Oracle. С момента приобретения MySQL и MariaDB развивались по-разному. Несмотря на это, MariaDB использует файлы данных и определений таблиц MySQL, а также идентичные клиентские протоколы, клиентские API, порты и сокеты.

Процесс создания базы данных:

1. Подключитесь к MySQL или MariaDB с правами супер-администратора:

* Для MySQL: mysql -u root -p
* Для MariaDB: mariadb -u root -p

1. Используйте SQL-запрос для создания базы данных, если она не существует:

CREATE DATABASE IF NOT EXISTS <имя\_вашей\_базы>;

Таким образом, основные этапы работы с базами данных остаются неизменными как в MySQL, так и MariaDB.

Для просмотра существующих баз данных можно использовать команду SHOW DATABASES;.

Для переключения на нужную базу данных выполните USE <имя базы данных>;. После вы можете просмотреть существующие таблицы через SHOW TABLES;.

Некоторые полезные команды вы можете увидеть ниже:

* CREATE TABLE IF NOT EXISTS users (user\_id INT NOT NULL PRIMARY KEY AUTO\_INCREMENT, username VARCHAR(100) NOT NULL); — создает таблицу users с полями user\_id и username.
* INSERT INTO users (username) VALUES ('John Doe'); — вставляет в таблицу users новую колонку.
* SELECT \* FROM users; — выборка всех записей из таблицы users.

Статистику по работе сервера можно получить через SHOW GLOBAL STATUS;.

## Экспорт

**Экспорт данных MySQL и MariaDB** осуществляется при помощи утилиты командной строки mysqldump. Она экспортирует данные из базы в SQL-файл, с которым можно легко работать: импортировать или переместить.

Для проведения данной операции вам понадобится база данных, а также учетные данные пользователя (имеющий хотя-бы право на чтение базы данных).

Сама команда указана ниже:

mysqldump -u <имя пользователя> -p<пароль> <имя базы данных> > data-dump.sql

Где:

* -p<пароль> — пароль от пользователя (в ручном виде), или можете использовать просто -p (автоматически запросит пароль);
* data-dump.sql — файл дампа базы данных;
* <имя пользователя> — имя привилегированного пользователя (права на чтение);
* <имя базы данных> — имя базы данных.

Также можно создавать дампы удаленно, с флагом -h:

mysqldump -h <ip-адрес> -u <имя пользователя> -p<пароль> <имя базы данных> > data-dump.sql # если MySQL сервер на стандартном порту  
# ИЛИ  
mysqldump -h <ip-адрес> -P <порт> -u <имя пользователя> -p<пароль> <имя базы данных> > data-dump.sql # если MySQL на нестандартном порту

Флаг -h используется для указания адреса удаленного сервера MySQL, с который вы хотите создать дамп базы данных. Также можно использовать флаг -P для указания порта сервера.

Аналогично с импортом, выводятся только ошибки. Вы можете проверить корректность экспортируемых данных, чтобы убедиться в том, что процедура прошла успешно. В итоге вы успешно проведете **экспорт таблиц MySQL**.

В SQL-файле будет указано название СУБД (MySQL или MariaDB), имя базы данных и другая информация.

## Импорт данных

Для **импорта дампа базы данных в MySQL** уже не надо использовать mysqldump, можно обойтись прямым обращением к mysql.

Для этого введите команду ниже:

mysql -u <имя пользователя> -p<пароль> <имя новой базы данных> < data-dump.sql

Где:

* -p<пароль> — пароль от пользователя (в ручном виде), или можете использовать просто -p (будет запрошен пароль автоматически);
* data-dump.sql — файл дампа базы данных;
* <имя пользователя> — имя привилегированного пользователя (права на запись);
* <имя новой базы данных> — имя новой базы данных в которую мы импортируем дамп.

Если процесс импорта выполнен корректно, то вы ничего не увидите. В ином же случае могут отобразиться сообщения об ошибках или предупреждениях. Проверить правильность данных можно через следующие команды MySQL:

SHOW DATABASES; # покажет базы данных  
USE <БАЗА ДАННЫХ>; # переходим в базу данных  
SHOW TABLES; # покажет таблицы в базе данных

## systemD-служба для бекапа

Допустим, вы хотите автоматизировать процесс бекапа (экспорта) базы данных. Мы будем использовать службу-таймер, которая будет запускать скрипт для бекапа.

Timer — это механизм для планирования выполнения конкретной службы в заданное время или через определенные интервалы.

Для этого подключитесь к серверу и выполните следующие команды:

mkdir -p /usr/bin/backup\_scripts # создаем директорию, где будут храниться скрипты для бекапа

После создайте и откройте файл /usr/bin/backup\_scripts/mysql\_backup.sh через любой редактор (например nano) и запишите в нем следующее:

TIMESTAMP=$(date +'%F') # получаем текущую дату  
BACKUP\_DIR='<путь до директории, где будут сохраняться дампы>'  
MYSQL\_USER='<имя пользователя>'  
MYSQL\_PASSWORD='<пароль пользователя>'  
DATABASE\_NAME='<название базы данных>'  
  
mkdir -p '$BACKUP\_DIR/$TIMESTAMP' # создаем директорию  
mysqldump -u $MYSQL\_USER -p$MYSQL\_PASSWORD $DATABASE\_NAME > '$BACKUP\_DIR/$TIMESTAMP/$DATABASE\_NAME-$TIMESTAMP.sql' # создаем дамп  
find '$BACKUP\_DIR;' -type d -mtime +28 -exec rm -rf {} \; # удаление бекапов старше 4 недель

Укажите в нем директорию для бекапа, имя пользователя и пароль, и также название базы данных.

Далее выдадим этому скрипту права на исполнение:

chmod +x /usr/bin/backup\_scripts/mysql\_backup.sh

Следующим шагом будет создание службы и таймера systemD:

1. Создайте файл службы (/etc/systemd/system/mysql-backup.service)

[Unit]  
Description=MySQL Database Backup Service  
  
[Service]  
Type=oneshot  
ExecStart=/usr/bin/backup\_scripts/mysql\_backup.sh  
  
[Install]  
WantedBy=multi-user.target

1. Создайте файл таймера (/etc/systemd/system/mysql-backup.timer):

[Unit]  
Descripton=Run MySQL Backup Service Weekly  
  
[Timer]  
OnCalendar=weekly  
Persistent=true  
  
[Install]  
WantedBy=timers.target

Конечным действием является активация службы и таймера:

systemctl daemon-reload # перезагрузка конфигурации systemD  
systemctl enable mysql-backup.timer # включаем таймер в автозагрузку  
systemctl start mysql-backup.timer # запуск таймера

После проверьте статус службы:

systemctl status mysql-backup.timer  
systemctl list-timers

Готово! Теперь каждую неделю будет создаваться дамп нужной вам базы данных.

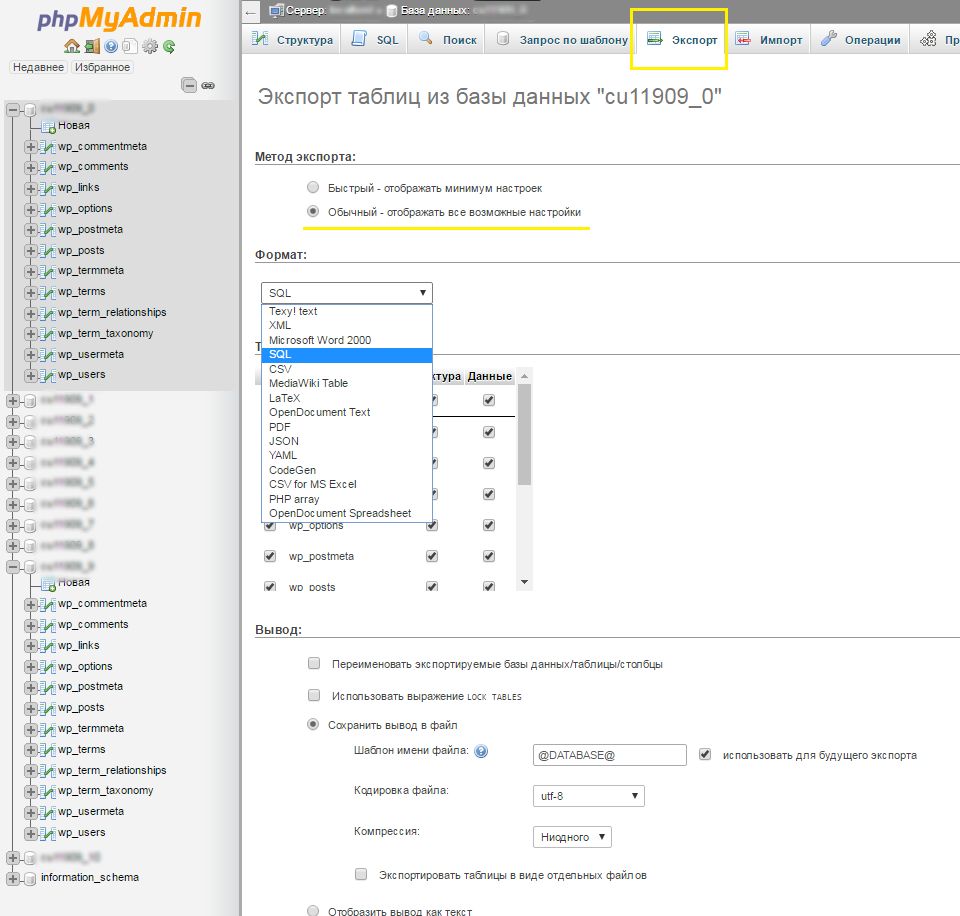
## Экспорт и импорт через phpMyAdmin

Если вы пользователь phpMyAdmin, совершать импорт и экспорт можно не только через консольные утилиты, а через веб-интерфейс самого phpMyAdmin. Чаще всего этот способ удобнее, если размер дампа небольшой (не более 1 гигабайта).

В этой секции будет рассматриваться поверхностный импорт и экспорт базы данных, больше вы можете найти [на нашем официальном сайте](https://timeweb.com/ru/docs/virtualnyj-hosting/bazy-dannyh/osnovy-raboty-v-phpmyadmin/).

### Экспорт

Для процесса экспорта перейдите в нужную базу данных и нажмите на вкладку «Экспорт», а после выберите нужный вам метод экспорта: быстрый или расширенный (он подойдет, если в вашем случае процесс экспорта специфичен и требует настройки).



При экспортировании данных в быстром режиме:

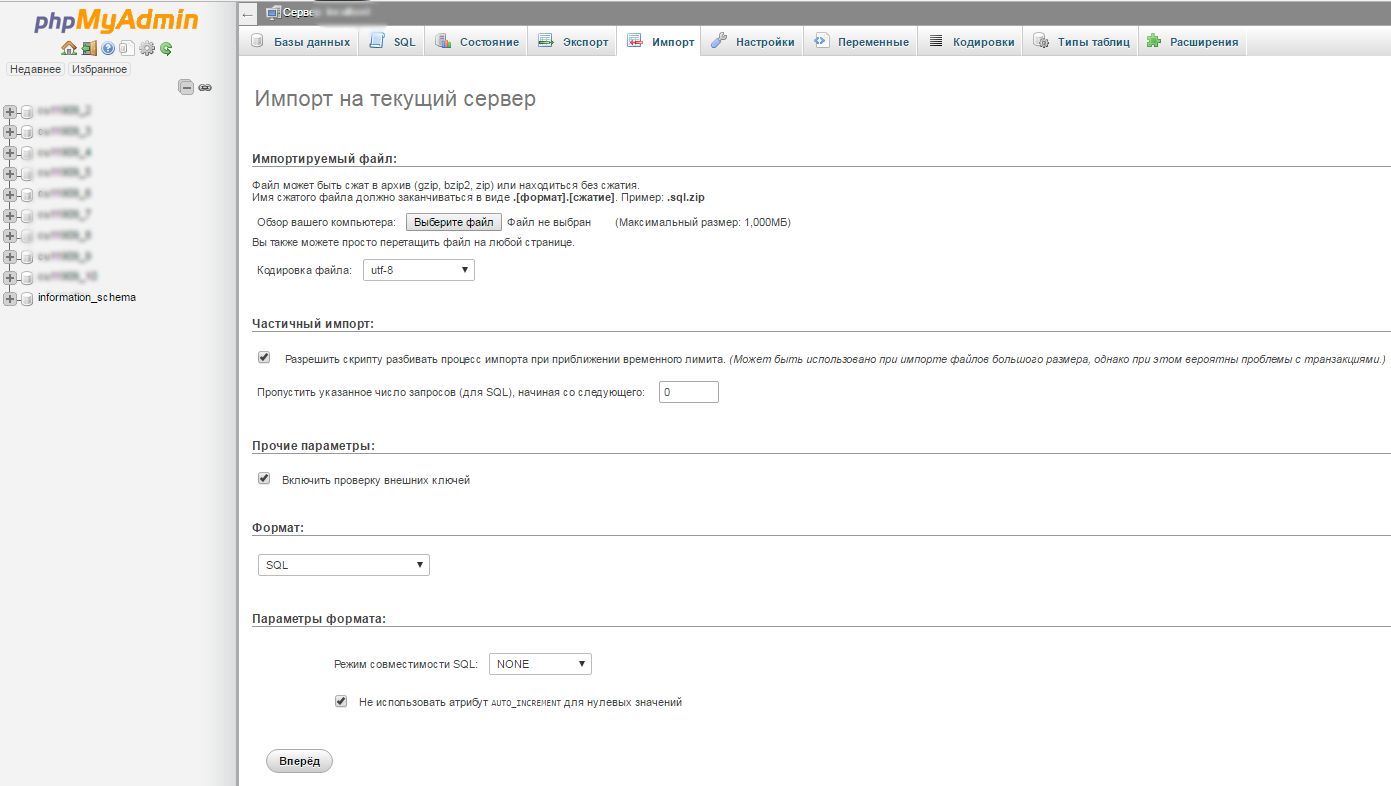
1. Опять же, выберите нужную базу из левой панели.
2. Перейдите к разделу «Экспорт» через меню PMA.
3. Инициируйте процесс сохранения дампа на локальный ПК с помощью кнопки «Экспорт».

Для экспорта конкретных таблиц:

1. Выберите базу данных в левой области, открыв список её таблиц.
2. Отметьте необходимые для выгрузки таблицы галочками.
3. В нижней части экрана из списка действий с отмеченными элементами выберите «Экспорт».
4. На следующем шаге убедитесь в правильном формате и нажмите «Вперед» для сохранения дампа.

### Импорт

**Импорт базы данных** аналогичен экспорту. Войдите в phpMyAdmin и перейдите в интересующую вас базу данных, а после перейдите на вкладку «Импорт»:



Чтобы осуществить **импорт базы данных**, необходимо выполнить следующие действия:

1. Для выбора нужной базы в левом меню следует раскрыть список доступных таблиц.
2. В интерфейсе PhpMyAdmin перейдите на вкладку «Импорт».
3. Загрузите файл дампа для импорта, выбрав его через кнопку «Выбрать файл» и загрузив с вашего устройства.
4. Установите параметры: определите кодировку, формат файла (можно использовать стандартные настройки).
5. Подтвердите процесс импортом путем нажатия соответствующей кнопки.

В списке слева вы можете выбрать базу данных, с которой работали, и посмотреть имеющиеся файлы, а также их содержимое (и изменить их).

# Заключение

Выбор более удобного метода остается за вами. Если вы хотите вручную делать бекапы или быстро получить доступ к базе данных — вам следует использовать phpMyAdmin. В случае, если вам нужно автоматизировать рутинные задачи, или вы переносите инфраструктуру, более удобным может показаться ручной вариант через утилиты командной строки.

Это руководство содержало в себе базовые знания и команды для грамотного **импорта и экспорта данных**.