MySQL

Установка и настройка MySQL. Создание БД и пользователей. Резервное копирование и администрирование баз MySQL. Репликация GTID БД

[Введение](#_c713xnuw3xem)

[Установка MySQL](#_en7u82a8e9xn)

[Пример работы с БД](#_8f60egox1xiz)

[Создание бэкапа и восстановление](#_c0u8ycxaqua8)

[Настройка GTID-репликации](#_ws4g6dy9zil1)

[Удаленная работа с MySQL](#_ejcbdze5gyaz)

Практическое [задание](#_3tfrjxxltv85)

[Дополнительные материалы](#_jeeoh8oz8lfe)

[Используемая литература](#_uvp6qax5r1ok)

# 

# Введение

Чтобы реализовать полноценный LAMP или LNMP, мы научились устанавливать и настраивать Linux, Apache2/Nginx и PHP. Дополнительно нужен MySQL.

У MySQL есть часто используемые форки: **mariadb** и **perconadb**. Настройки и работа с этими БД преимущественно аналогичны друг другу.

Научимся настраивать полноценный веб-сервер с почтой и базами данных.

# Установка MySQL

Устанавливаем.

|  |
| --- |
| sudo apt-get install mysql-server mysql-client mysql-common php7.0-mysql |

Будет предложено создать пароль для рута (относящегося не к Linux, а к MySQL —это разные вещи).

Правим **/etc/mysql/my.cnf**.

|  |
| --- |
| bind-address=другой машины либо 0.0.0.0 |

Например, **bind-address=192.168.131.1**, если хотите с хостовой машины обращаться к MySQL. Или оставьте так.

|  |
| --- |
| bind-address=127.0.0.1 |

Этот вариант рекомендуется, если HTTP-сервер работает на одной машине с сервером СУБД и репликация не требуется. В этом случае только локальные процессы смогут обращаться к СУБД. Так как репликация будет невозможна, для нее необходимо использовать IP-адрес, отличный от **127.0.0.1**.

|  |
| --- |
| [mysqld]  skip-character-set-client-handshake  character-set-server=utf8  init-connect='SET NAMES utf8'  collation-server=utf8\_general\_ci  ...  [client]  default-character-set=utf8  ..  [mysqldump]  default-character-set=utf8  .. |

Очистим MySQL от мусора. Запустим.

|  |
| --- |
| mysql\_secure\_installation |

Зайдем рутом.

|  |
| --- |
| mysql -u root -p |

## Пример работы с БД

Создаем базу **mydatabase**.

|  |
| --- |
| CREATE DATABASE mydatabase; |

Добавляем пользователя **user** с паролем **1234** и даем ему все права на эту базу.

|  |
| --- |
| GRANT ALL ON mydatabase.\* TO user@localhost IDENTIFIED BY '1234';  Exit |

Заходим пользователем **user** и вводим пароль.

|  |
| --- |
| mysql -u user -p  1234 |

Указываем, какую базу будем использовать.

|  |
| --- |
| USE mydatabase;  Создаем таблицу  create table mytable (  id int not null auto\_increment,  txt varchar(100),  n int,  primary key (id)); |

Добавляем значения.

|  |
| --- |
| insert into mytable (txt,n) values('Wall Street',7);  insert into mytable (txt,n) values('5th Avenu',1);  ...  select \* from mytable; |

Получаем на выходе содержимое таблицы. Чтобы работать в веб-интерфейсе, можно установить **phpMyAdmin**. Можно подготовить команды в отдельный файл **cmds.sql**.

|  |
| --- |
| USE mydatabase;  insert into mytable (txt,n) values('Gorki park',7);  insert into mytable (txt,n) values('Kremlin',NULL); |

Выполняем данные из файла.

|  |
| --- |
| mysql -u user -p mydatabase <cmds.sql |

Можно зайти как **user** и, сделав **USE** и **SELECT**, оценить результат.

## Создание бэкапа и восстановление

Делаем бэкап.

|  |
| --- |
| mysqldump -u user -p mydatabase > backup.sql |

С помощью скриптов и крона можно выполнять его регулярно. Восстанавливаем (используем **mysqladmin**, а не **mysql**).

Создаем рутом базу **newbase**.

|  |
| --- |
| mysqladmin -u root -p create newbase |

Если **mysqladmin** не установлен, можно сделать так.

|  |
| --- |
| mysql -u root -p -e "create database newbase;" |

Импортируем бэкап в **newbase**.

|  |
| --- |
| mysql -u root -p --default-character-set=utf8 newbase < backup.sql  mysql -u root -p  USE newbase;  select \* from mytable;  exit |

Сравниваем таблицы и разбираемся, какие строки откуда.

## Настройка GTID-репликации

В MySQL есть два вида репликации: **GTID** и **binlog**. Первый предпочтительнее (появился в версии 5.6). Для GTID-репликации тоже нужны бинлоги. Их ведение и ежеминутное копирование с MySQL-сервера на бэкап-сервер позволяет восстановить данные из дампа, добавив изменения из бинлогов — фактически это бэкап «минута в минуту». Но у такой схемы есть и недостатки.

* Долгое восстановление из дампа.
* Применение всех транзакций из бинлогов от времени дампа до последней транзакции в крайнем бинлоге.

На этот случай и делается реплика. Сначала поменяем конфиг MySQL **/etc/mysql/mysql.conf.d/mysqld.cnf.**

|  |
| --- |
| [mysqld\_safe]  socket = /var/run/mysqld/mysqld.sock  nice = 0  [mysqld]  # \* Basic Settings  user = mysql  pid-file = /var/run/mysqld/mysqld.pid  socket = /var/run/mysqld/mysqld.sock  port = 3306  datadir = /var/lib/mysql  tmpdir = /tmp  bind-address = 0.0.0.0  # \* Fine Tuning  key\_buffer\_size = 16M  max\_allowed\_packet = 16M  thread\_stack = 192K  thread\_cache\_size = 8  max\_connections = 300  # \* Query Cache Configuration  query\_cache\_limit = 1M  query\_cache\_size = 16M  # Error log  log\_error = /var/log/mysql/error.log  # Replication GTID  gtid-mode = on  enforce-gtid-consistency = true  # id сервера при репликации (любое число); на реплике должен быть больше (неважно, насколько)  server-id = 100  # Binlogs  # Формат бинлогов, должен быть MIXED или ROW, иначе часть потом не восстановится  binlog\_format = MIXED  # Директория для бинлогов, может быть в любом месте (главное — верные права)  log\_bin = /var/lib/mysql/binlogs/mysql-bin.log  # Ротация бинлогов и размер  expire\_logs\_days = 10  max\_binlog\_size = 100M |

На сервере для репликации поменяем конфиг, но увеличим значение server-id. Создадим папку с бинлогами на мастере и дадим ей соответствующие права.

|  |
| --- |
| mkdir /var/lib/mysql/binlogs  chown mysql:mysql /var/lib/mysql/binlogs && chmod 0770 /var/lib/mysql/binlogs |

Самый простой способ сделать репликацию — остановить мастер, скопировать папку с БД и бинлогами на слейв, запустить мастер и настроить репликацию на слейве. Вариант не всегда наилучший, зато самый простой и надежный — с гарантией, что репликация не выйдет из строя через пару секунд. В противном случае требуется понимание **GTID** и бинлогов. По документации и мануалам настройка GTID-репликации может работать нестабильно.

Зайдем в MySQL на мастере и создадим юзера для репликации.

|  |
| --- |
| Master> CREATE USER 'repl'@'192.168.1.%' IDENTIFIED BY 'password';  Master> GRANT REPLICATION SLAVE ON \*.\* TO 'repl'@'192.168.1.%';  Master> FLUSH PRIVILEGES;  Master> \q  systemctl stop mysql |

Остановим MySQL на слейве и скопируем все с мастера через **rsync**. Для этого надо иметь возможность заходить юзером **root** на мастер.

|  |
| --- |
| systemctl stop mysql  rm -rf /var/lib/mysql  rsync -aAzpP root@master.lan:/var/lib/mysql/ /var/lib/mysql/  chown -R mysql:mysql /var/lib/mysql/ |

Генерируем новый **UUID** и заменяем его на слейве. Иначе репликация не запустится, так как оба MySQL будут с одинаковыми идентификаторами.

|  |
| --- |
| uuidgen  sed -i 's/^server-uuid=.\*/server-uuid=UUID/g' /var/lib/mysql/auto.cnf  # UUID это то, что получилось после запуска uuidgen  systemctl start mysql |

Возвращаем мастер к нормальной работе.

|  |
| --- |
| systemctl start mysql |

Настраиваем репликацию на слейве и проверяем, что она работает.

|  |
| --- |
| Slave> CHANGE MASTER TO  MASTER\_HOST="192.168.1.100",  MASTER\_USER="repl",  MASTER\_PASSWORD="password",  MASTER\_AUTO\_POSITION = 1;  Slave> show slave status \G; |

Если все верно, увидим.

|  |
| --- |
| Slave\_IO\_Running: Yes  Slave\_SQL\_Running: Yes |

## Удаленная работа с MySQL

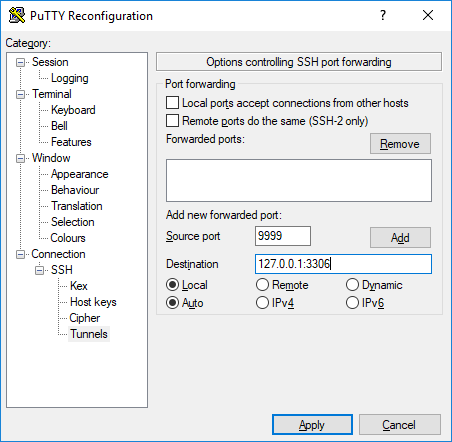
Даже если MySQL доступен только на локальном хосте, есть возможность работать удаленным клиентом, используя проброс портов.

Это делается так (на машине с клиентом).

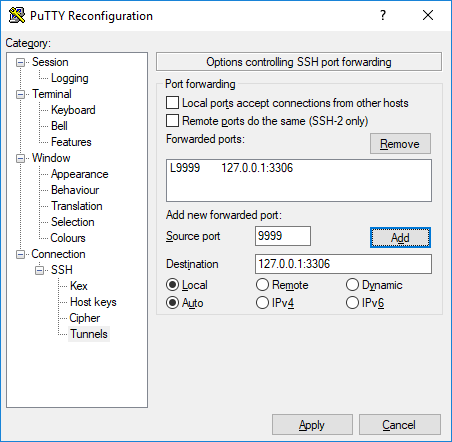
|  |
| --- |
| ssh -L 9999:127.1:3306 user@server |

После этого клиент сможет присоединиться к серверу, используя адрес **127.0.0.1:9999**.

То же самое с использованием **putty**



Нажимаем **Add**.



И **Apply** или **Connect**.

# Практическое задание

1. Установить MySQL.
2. Создать несложную таблицу, вручную выполнить бэкапирование, удалить таблицу, восстановить из бэкапа.
3. \* Настроить репликацию.

# Дополнительные материалы

1. <https://www.mysql.com/>
2. <https://mariadb.com/>

# Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы.

1. <https://www.mysql.com/>