

Отчёт о лабораторной работе

Лабораторная работа 6

Мошаров Денис Максимович

Содержание

Цель работы

Приобретение практических навыков по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

Выполнение лабораторной работы

Запустим наш сервер через vagrant (рис. [-@fig:001]).

```
C:\Users\denis>cd C:\work_asp\dmmosharov\vagrant
C:\work_asp\dmmosharov\vagrant>vagrant up server
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".1" or ":1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: You assigned a static IP ending in ".1" or ":1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: Clearing any previously set forwarded ports...
==> server: Clearing any previously set network interfaces...
==> server: Preparing network interfaces based on configuration...
      server: Adapter 1: nat
      server: Adapter 2: intnet
==> server: Forwarding ports...
      server: 22 (guest) => 2222 (host) (adapter 1)
==> server: Running 'pre-boot' VM customizations...
==> server: Booting VM...
==> server: Waiting for machine to boot. This may take a few minutes...
      server: SSH address: 127.0.0.1:2222
      server: SSH username: vagrant
      server: SSH auth method: password
```

Запуск сервера

Далее установим пакет MariaDB (рис. [-@fig:002]).

```
[root@server.dmmosharov.net ~]# dnf -y install mariadb mariadb-server
Last metadata expiration check: 2:50:54 ago on Tue 27 Jan 2026 02:03:08 PM UTC.
Dependencies resolved.
=====
Package          Arch    Version        Repository      Size
=====
mariadb          x86_64  3:10.11.15-1.el10_1   appstream     1.6 M
mariadb-server   x86_64  3:10.11.15-1.el10_1   appstream     10 M
=====
Installing:
mariadb-common  noarch   3:10.11.15-1.el10_1   appstream     35 k
mariadb-errmsg   noarch   3:10.11.15-1.el10_1   appstream     262 k
mysql-selinux    noarch   1.0.14-1.el10_0    appstream     37 k
perl-DBD-MariaDB x86_64  1.23-10.el10       appstream     154 k
perl-Sys-Hostname x86_64  1.25-512.2.el10_0  appstream     17 k
Installing dependencies:
mariadb-backup   x86_64  3:10.11.15-1.el10_1   appstream     6.5 M
mariadb-client-utils x86_64  3:10.11.15-1.el10_1   appstream     39 k
mariadb-gssapi-server x86_64  3:10.11.15-1.el10_1   appstream     17 k
mariadb-server-utils x86_64  3:10.11.15-1.el10_1   appstream     261 k
=====
Transaction Summary
=====
Install 11 Packages

Total download size: 19 M
Installed size: 124 M
Downloading Packages:
[1-3/11]: mariadb-10.11.1 0% [1 --- B/s | 0 B --:-- ETA
```

Установка пакета

Теперь рассмотрим основные конфигурационный файлы
Файл: /etc/my.cnf

Это главный конфигурационный файл MariaDB.
client-server: Секция для настроек, которые применяются и к серверу, и к клиентам.
!includedir /etc/my.cnf.d: Директива, которая говорит MariaDB загрузить все .cnf файлы из директории /etc/my.cnf.d. Это основной механизм подключения всех остальных файлов. (рис. [-@fig:003]).

```
GNU nano 8.1                               /etc/my.cnf
#
# This group is read both by the client and the server
# use it for options that affect everything
#
[client-server]

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
```

/etc/my.cnf

Файл: auth_gssapi.cnf

Файл для подключения плагина аутентификации GSSAPI (Kerberos).

mariadb: Секция для настроек MariaDB.

plugin-load-add=auth_gssapi.so: Закомментированная строка для загрузки плагина аутентификации GSSAPI. (рис. [-@fig:004]).

```
GNU nano 8.1                               auth_gssapi.cnf
[mariadb]
#plugin-load-add=auth_gssapi.so
```

auth_gssapi.cnf

Файл: mariadb-server.cnf

Основной файл конфигурации для самого сервера MariaDB.

server, mysqld: Секции для настроек серверного процесса mysqld.
 datadir=/var/lib/mysql: Указывает директорию для хранения данных баз.
 socket=/var/lib/mysql/mysql.sock: Путь к файлу сокета для локальных подключений.
 log-error=/var/log/mariadb/mariadb.log: Файл для записи ошибок сервера.
 pid-file=/run/mariadb/mariadb.pid: Файл для хранения ID процесса сервера.
 galera: Секция для настройки Galera Cluster (система репликации).
 wsrep_on=ON: Включает поддержку репликации Galera.
 wsrep_provider=, wsrep_cluster_address=: Пустые параметры для пути к библиотеке Galera и адреса кластера.
 binlog_format=row, default_storage_engine=InnoDB, innodb_autoinc_lock_mode=2:
 Обязательные настройки для корректной работы Galera.
 bind-address=0.0.0.0: Закомментированная строка, которая позволила бы серверу принимать подключения со всех сетевых интерфейсов.
 wsrep_slave_threads=1, innodb_flush_log_at_trx_commit=0: Настройки производительности для репликации (рис. [-@fig:005]).

```
GNU nano 8.1                               mariadb-server.cnf
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql/
#
# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mysqld/mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mariadb/mariadb.log
pid-file=/run/mariadb/mariadb.pid

#
# * Galera-related settings
#
[galera]
# Mandatory settings
#wsrep_on=ON
#wsrep_provider=
#wsrep_cluster_address=
#binlog_format=row
#default_storage_engine=InnoDB
#innodb_autoinc_lock_mode=2
#
# Allow server to accept connections on all interfaces.
#
#bind-address=0.0.0.0
#
# Optional setting
#wsrep_slave_threads=1
#innodb_flush_log_at_trx_commit=0

# this is only for embedded server

mariadb-server.cnf
```

Файл: provider_lz4.cnf
 server: Секция для настроек сервера.
 plugin_load_add=provider_lz4: Загружает плагин, предоставляющий алгоритм сжатия LZ4.
 provider_lz4=force_plus_permanent: Настраивает плагин так, чтобы он загружался принудительно и не мог быть выгружен. (рис. [-@fig:006]).

```
GNU nano 8.1                                     provider_lz4.cnf
[server]
plugin_load_add=provider_lz4
provider_lz4=force_plus_permanent
```

provider_lz4.cnf

Файл: spider.cnf

Файл для подключения движка хранилища Spider.

mariadb: Секция для настроек MariaDB.

plugin-load-add = ha_spider: Закомментированная строка для загрузки плагина Spider, который используется для шардинга (разделения данных по разным серверам) (рис. [-@fig:007]).

```
GNU nano 8.1                                     spider.cnf
[mariadb]
#
# Uncomment line to enable
#
#plugin-load-add = ha_spider

# Read more at https://mariadb.com/kb/en/spider/
```

spider.cnf

Файл: client.cnf

Конфигурация для клиентских приложений.

client: Настройки для всех клиентских утилит (MySQL и MariaDB).

client-mariadb: Настройки, которые применяются только к клиентам MariaDB (рис. [-@fig:008]).

```
GNU nano 8.1                                     client.cnf
#
# These two groups are read by the client library
# Use it for options that affect all clients, but not the server
#

[client]

# This group is not read by mysql client library,
# If you use the same .cnf file for MySQL and MariaDB,
# use it for MariaDB-only client options
[client-mariadb]
```

client.cnf

Файл: mysql-clients.cnf

Файл-шаблон для настройки отдельных клиентских утилит MariaDB.

Содержит пустые секции, такие как mysql, mysql_upgrade, mysqldump, mysqladmin и другие. В них можно добавлять специфичные параметры для каждой утилиты. (рис. [-@fig:009]).

```
GNU nano 8.1                               mysql-clients.cnf
#
# These groups are read by MariaDB command-line tools
# Use it for options that affect only one utility
#
[mysql]
[mysql_upgrade]
[mysqladmin]
[mysqlbinlog]
[mysqlcheck]
[mysqldump]
[mysqlimport]
[mysqlshow]
[mysqlslap]
```

mysql-clients.cnf

Файл: provider_lzo.cnf

server: Секция для настроек сервера.

plugin_load_add=provider_lzo: Загружает плагин, предоставляющий алгоритм сжатия LZO.

provider_lzo=force_plus_permanent: Настраивает плагин так, чтобы он загружался принудительно и не мог быть выгружен. (рис. [-@fig:010]).

```
GNU nano 8.1                               provider_lzo.cnf
[server]
plugin_load_add=provider_lzo
provider_lzo=force_plus_permanent
```

provider_lzo.cnf

Файл: enable_encryption.preset

Файл-пресет для включения шифрования данных “в состоянии покоя”.

mariadb: Секция для настроек MariaDB.

aria-encrypt-tables, encrypt-binlog, encrypt-tmp-disk-tables, encrypt-tmp-files, loose-innodb-encrypt-log, loose-innodb-encrypt-tables: Директивы, включающие шифрование для таблиц Aria и InnoDB, бинарного лога и временных файлов. (рис. [-@fig:011]).

```
GNU nano 8.1          enable_encryption.preset
#
# !include this file into your my.cnf (or any of *.cnf files in /etc/my.cnf.d)
# and it will enable data at rest encryption. This is a simple way to
# ensure that everything that can be encrypted will be and your
# data will not leak unencrypted.
#
# DO NOT EDIT THIS FILE! On MariaDB upgrades it might be replaced with a
# newer version and your edits will be lost. Instead, add your edits
# to the .cnf file after the !include directive.
#
# NOTE that you also need to install an encryption plugin for the encryption
# to work. See https://mariadb.com/kb/en/mariadb/data-at-rest-encryption/#encryption-key
#
[mariadb]
aria-encrypt-tables
encrypt-binlog
encrypt-tmp-disk-tables
encrypt-tmp-files
loose-innodb-encrypt-log
loose-innodb-encrypt-tables
```

enable_encryption.preset

Файл: provider_bzip2.cnf

server: Секция для настроек сервера.

plugin_load_add=provider_bzip2: Загружает плагин, предоставляющий алгоритм сжатия bzip2.

provider_bzip2=force_plus_permanent: Настраивает плагин так, чтобы он загружался принудительно и не мог быть выгружен. (рис. [-@fig:012]).

```
GNU nano 8.1          provider_bzip2.cnf
[server]
plugin_load_add=provider_bzip2
provider_bzip2=force_plus_permanent
```

provider_bzip2.cnf

Файл: provider_snappy.cnf

server: Секция для настроек сервера.

plugin_load_add=provider_snappy: Загружает плагин, предоставляющий алгоритм сжатия Snappy.

provider_snappy=force_plus_permanent: Настраивает плагин так, чтобы он загружался принудительно и не мог быть выгружен. (рис. [-@fig:013]).

```
GNU nano 8.1          provider_snappy.cnf
[server]
plugin_load_add=provider_snappy
provider_snappy=force_plus_permanent
```

provider_snappy.cnf

Запустим службу mariadb и добавим её в автозагрузку. Также с помощью ss убедимся, что mariadb прослушивает 3306 порт (рис. [-@fig:014]).

```
[root@server.dmmosharov.net my.cnf.d]# systemctl start mariadb
[root@server.dmmosharov.net my.cnf.d]# systemctl enable mariadb
Created symlink '/etc/systemd/system/mysql.service' → '/usr/lib/systemd/system/mariadb.service'.
Created symlink '/etc/systemd/system/mysqld.service' → '/usr/lib/systemd/system/mariadb.service'.
Created symlink '/etc/systemd/system/multi-user.target.wants/mariadb.service' → '/usr/lib/systemd/system/mariadb.service'.
[root@server.dmmosharov.net my.cnf.d]# ss -tulpen | grep mysql
[root@server.dmmosharov.net my.cnf.d]# ss -tulpen | grep maria
tcp    LISTEN  0      80          0.0.0.0:3306      0.0.0.0:*
users:(("mariadb",pid=111716,fd=20))
                                         uid:27  ino:1435582  sk:2025  cgroup:/system.slice/mariadb.service
ce/mariadb.service <->
tcp    LISTEN  0      80          :::3306      :::*[::]:*
users:(("mariadb",pid=111716,fd=21))
                                         uid:27  ino:1435583  sk:202e  cgroup:/system.slice/mariadb.service
ce/mariadb.service v6only:1 <->
[root@server.dmmosharov.net my.cnf.d]#
```

Запуск mariadb

Запустим команду mysql_secure_installation и настроим базу данных (рис. [-@fig:015]).

```
[root@server.dmmosharov.net my.cnf.d]# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
... Success!

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!
```

Настройка БД

Теперь подключимся к базе данных и выведем справку. Как видим, всё работает (рис. [-@fig:016]).

```
[root@server.dmmosharov.net my.cnf.d]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.11.15-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> \h

General information about MariaDB can be found at
http://mariadb.org

List of all client commands:
Note that all text commands must be first on line and end with ';'
?          (?) Synonym for 'help'.
charset   (\C) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
clear     (\c) Clear the current input statement.
connect   (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter.
edit      (\e) Edit command with $EDITOR.
ego       (\G) Send command to MariaDB server, display result vertically.
exit      (\q) Exit mysql. Same as quit.
go        (\g) Send command to MariaDB server.
help      (\h) Display this help.
nopager  (\n) Disable pager, print to stdout.
noteed   (\t) Don't write into outfile.
nowarning (\w) Don't show warnings after every statement.
pager    (\P) Set PAGER [to_pager]. Print the query results via PAGER.
print    (\p) Print current command.
prompt   (\R) Change your mysql prompt.
quit     (\q) Quit mysql.
rehash   (\#) Rebuild completion hash.
sandbox  (\-) Disallow commands that access the file system (except \P without an argument and \e).
source   (\.) Execute an SQL script file. Takes a file name as an argument.
status   (\s) Get status information from the server.
system   (\!) Execute a system shell command.
tee      (\T) Set outfile [to_outfile]. Append everything into given outfile.
use     (\u) Use another database. Takes database name as argument.
```

Подключение к БД

Посмотрим на списки баз данных. На текущем этапе их всего 4 (рис. [-@fig:017]).

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]> exit;
Bye
[root@server.dmmosharov.net my.cnf.d]# █
```

Списки БД

Получим статус работы БД. Далее, построчно проанализируем его:

Информация о клиенте

mysql Ver 15.1 Distrib 10.11.15-MariaDB, for Linux (x86_64) using EditLine wrapper
Эта строка описывает программу, с помощью которой вы подключились. Это клиент
mysql, который является частью дистрибутива MariaDB версии 10.11.15.

Детали текущего сеанса

Connection id: 14

Уникальный номер вашего текущего подключения к серверу. Current database:

Название базы данных, которая выбрана в данный момент. Здесь она не выбрана.

Current user: root@localhost
Имя пользователя и хост, с которого выполнено подключение.
SSL: Not in use
Показывает, используется ли шифрование SSL/TLS для этого соединения. В данном случае — нет.
Current pager: stdout
Программа для постраничного вывода. stdout означает прямой вывод в консоль.
Using outfile: ''
Указывает, перенаправляется ли вывод в файл. В данном случае — нет.
Using delimiter: ;
Символ, который используется для завершения SQL-команд.

Информация о сервере
Server: MariaDB
Тип сервера, к которому вы подключены.
Server version: 10.11.15-MariaDB MariaDB Server
Полная версия программного обеспечения сервера.
Protocol version: 10
Версия протокола, используемого для связи клиента и сервера.
Connection: Localhost via UNIX socket
Способ подключения. Здесь — локальное соединение через UNIX сокет, что очень эффективно.
UNIX socket: /var/lib/mysql/mysql.sock
Точный путь к файлу сокета, который используется для этого соединения.

Настройки кодировок
Server characterset: latin1
Кодировка символов по умолчанию для всего сервера.
Db characterset: latin1
Кодировка символов по умолчанию для текущей базы данных. Client characterset: utf8mb3
Кодировка, в которой клиент отправляет запросы на сервер.
Conn. characterset: utf8mb3
Кодировка, установленная для текущего соединения.

Статистика и время работы
Uptime: 4 min 57 sec
Общее время работы сервера с момента последнего запуска.
Threads: 1 Questions: 27 Slow queries: 0 Opens: 20 Open tables: 13 Queries per second avg: 0.090
Краткая статистика производительности:
1 активное подключение (поток).
27 запросов было выполнено с момента запуска.
0 медленных запросов.
20 раз сервер открывал файлы таблиц.
13 таблиц открыто в данный момент.
0.090 — среднее количество запросов в секунду (рис. [-@fig:018]).

```
[root@server.dmmosharov.net my.cnf.d]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.11.15-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> status
-----
mysql Ver 15.1 Distrib 10.11.15-MariaDB, for Linux (x86_64) using EditLine wrapper

Connection id: 14
Current database:
Current user: root@localhost
SSL: Not in use
Current pager: stdout
Using outfile:
Using delimiter:
Server: MariaDB
Server version: 10.11.15-MariaDB MariaDB Server
Protocol version: 10
Connection: Localhost via UNIX socket
Server characterset: latin1
Db characterset: latin1
Client characterset: utf8mb3
Conn. characterset: utf8mb3
UNIX socket: /var/lib/mysql/mysql.sock
Uptime: 4 min 57 sec

Threads: 1 Questions: 27 Slow queries: 0 Opens: 20 Open tables: 13 Queries per second avg: 0.090
-----
MariaDB [(none)]> exit;
Bye
[root@server.dmmosharov.net my.cnf.d]#
```

Статус БД

Перейдём в каталог /etc/my.cnf.d и создадим файл utf8.cnf (рис. [-@fig:019]).

```
[root@server.dmmosharov.net my.cnf.d]# cd /etc/my.cnf.d
[root@server.dmmosharov.net my.cnf.d]# touch utf8.cnf
[root@server.dmmosharov.net my.cnf.d]# nano uth8.cnf
[root@server.dmmosharov.net my.cnf.d]# █
```

Создание файла конфигурации

Поместим в него следующие строки. Тут мы указываем, что хотим видеть кодировку utf8 по умолчанию (рис. [-@fig:020]).

```
GNU nano 8.1                                     uth8.cnf
[client]
default-character-set = utf8
[mysqld]
character-set-server = utf8
```

Содержание файла utf8

Перезапустим mariadb и войдём снова в БД. Теперь посмотрим ещё раз статус БД и увидим, что кодировка сменилась с latin1 на utf8 (рис. [-@fig:021]).

```
[root@server.dmmosharov.net my.cnf.d]# systemctl restart mariadb
[root@server.dmmosharov.net my.cnf.d]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.11.15-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> status
-----
mysql Ver 15.1 Distrib 10.11.15-MariaDB, for Linux (x86_64) using EditLine wrapper

Connection id: 3
Current database:
Current user: root@localhost
SSL: Not in use
Current pager: stdout
Using outfile: ''
Using delimiter: ;
Server: MariaDB
Server version: 10.11.15-MariaDB MariaDB Server
Protocol version: 10
Connection: Localhost via UNIX socket
Server characterset: utf8mb3
Db characterset: utf8mb3
Client characterset: utf8mb3
Conn. characterset: utf8mb3
UNIX socket: /var/lib/mysql/mysql.sock
Uptime: 20 sec

Threads: 1 Questions: 4 Slow queries: 0 Opens: 17 Open tables: 10 Queries per second
avg: 0.200
-----
MariaDB [(none)]> exit;
```

Успешная смена кодировки

Теперь создадим базу данных addressbook , перейдём в неё и убедимся, что она пустая. После этого создадим в ней таблицу city и добавим в неё три записи: Иванов, Петров, Сидоров. Выведем эту таблицу (рис. [-@fig:022]).

```
MariaDB [(none)]> CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> USE addressbook;
Database changed
MariaDB [addressbook]> HOW TABLES;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'HOW TABLES' at line 1
MariaDB [addressbook]> SHOW TABLES;
Empty set (0.000 sec)

MariaDB [addressbook]> CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
Query OK, 0 rows affected (0.004 sec)

MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Иванов','Москва');
Query OK, 1 row affected (0.001 sec)

MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Петров','Сочи');
Query OK, 1 row affected (0.001 sec)

MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Сидоров','Дубна');
Query OK, 1 row affected (0.001 sec)

MariaDB [addressbook]> SELECT * FROM city;
+-----+-----+
| name | city |
+-----+-----+
| Иванов | Москва |
| Петров | Сочи |
| Сидоров | Дубна |
+-----+-----+
3 rows in set (0.000 sec)
```

Наполнение таблицы

Теперь создадим пользователя dmmosharov и предоставим ему доступ к этой таблице, после чего убедимся, что количество баз данных увеличилось с 4 до 5. После этого попробуем показать базу данных addressbook от имени рута и от имени созданного нами пользователя. Как видим, операция успешна, а значит, пользователь имеет доступ к БД (рис. [-@fig:023]).

```

Query OK, 0 rows affected (0.000 sec)

MariaDB [addressbook]> DESCRIBE city;
+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| name  | varchar(40) | YES  |     | NULL    |       |
| city   | varchar(40) | YES  |     | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [addressbook]> quit
Bye
[root@server.dmmosharov.net my.cnf.d]# mysqlshow -u root -p
Enter password:
+-----+
| Databases      |
+-----+
| addressbook   |
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
[root@server.dmmosharov.net my.cnf.d]# mysqlshow -u root -p addressbook
Enter password:
Database: addressbook
+-----+
| Tables        |
+-----+
| city          |
+-----+
[root@server.dmmosharov.net my.cnf.d]# mysqlshow -u user -p addressbook
Enter password:
mysqlshow: Access denied for user 'user'@'localhost' (using password: YES)
[root@server.dmmosharov.net my.cnf.d]# mysqlshow -u user -p addressbook
Enter password:
mysqlshow: Access denied for user 'user'@'localhost' (using password: YES)
[root@server.dmmosharov.net my.cnf.d]# mysqlshow -u root -p dmmosharov -p addressbook
Enter password:
Database: addressbook
+-----+
| Tables        |
+-----+
| city          |
+-----+
[root@server.dmmosharov.net my.cnf.d]#

```

Проверка прав доступа для нового пользователя

Теперь попробуем теперь сделать бэкапы и откатиться на них. Мы создадим обычный бэкап, сжатый и с таймстемпом. После этого сохраним все конфигурационные файлы в vagrant (рис. [-@fig:024]).

```

[root@server.dmmosharov.net my.cnf.d]# mkdir -p /var/backup
[root@server.dmmosharov.net my.cnf.d]# mysqldump -u root -p addressbook > /var/backup/addressbook.sql
Enter password:
[root@server.dmmosharov.net my.cnf.d]# mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz
Enter password:
[root@server.dmmosharov.net my.cnf.d]# mysqldump -u root -p addressbook | gzip > $(date+/
var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz)
-bash: date+/var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz: No such file or directory
-bash: $(date+/var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz): ambiguous redirect
Enter password:
mysqldump: Got error 32 on write
[root@server.dmmosharov.net my.cnf.d]# mysql -u root -p addressbook < /var/backup/addressbook.sql
Enter password:
[root@server.dmmosharov.net my.cnf.d]# zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook
Enter password:
[root@server.dmmosharov.net my.cnf.d]# cd /vagrant/provision/server
[root@server.dmmosharov.net server]# mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d
[root@server.dmmosharov.net server]# mkdir -p /vagrant/provision/server/mysql/var/backup
[root@server.dmmosharov.net server]# cp -R /etc/my.cnf.d/utf8.cnf /vagrant/provision/server/mysql/etc/my.cnf.d
[root@server.dmmosharov.net server]# cp -R /var/backup/* /vagrant/provision/server/mysql/var/backup/
[root@server.dmmosharov.net server]# cd /vagrant/provision/server
[root@server.dmmosharov.net server]# touch mysql.sh
[root@server.dmmosharov.net server]# chmod +x mysql.sh
[root@server.dmmosharov.net server]# nano mysql.sh

```

Бэкапы и сохранение vagrant

В созданном скрипте mysql.sh пропишем следующие строки (рис. [-@fig:025]).

```

GNU nano 8.1                               mysql.sh
cd /vagrant/provision/server
touch mysql.sh
chmod +x mysql.sh
Открыв его на редактирование, пропишите в нём следующий скрипт:
#!/bin/bash
echo "Provisioning script $0"
systemctl restart named
echo "Install needed packages"
dnf -y install mariadb mariadb-server
echo "Copy configuration files"
cp -R /vagrant/provision/server/mysql/etc/* /etc
mkdir -p /var/backup
cp -R /vagrant/provision/server/mysql/var/backup/* /var/backup
echo "Start mysql service"
systemctl enable mariadb
systemctl start mariadb
if [[ ! -d /var/lib/mysql/mysql ]]
then
echo "Securing mariadb"
mysql_secure_installation <<EOF
y
123456
123456
y
y
y
y
EOF
echo "Create database"
mysql -u root -p123456 <<EOF
CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
EOF
mysql -u root -p123456 addressbook < /var/backup/addressbook.sql
fi

```

mysql.sh

И запишем в vagrantfile автозагрузку этого скрипта (рис. [-@fig:026]).

```

server.vm.provision "shell",
  type: "shell",
  preserve_order: true,
  path: "provision/server/mysql.sh"
end

```

Vagrantfile

Выводы

В результате выполнения лабораторной работы были получены навыки работы с базами данных и их настройкой