

Отчёт по лабораторной работе 6

**Установка и настройка системы управления базами данных
MariaDB**

Элсаиед Адел

Содержание

1 Цель работы	5
2 Выполнение	6
2.1 Установка MariaDB и первичная настройка безопасности	6
2.2 Конфигурация кодировки символов	10
2.3 Создание базы данных и управление доступом	12
2.4 Резервное копирование и восстановление базы данных	16
2.5 Внесение изменений в настройки внутреннего окружения виртуальной машины	17
3 Вывод	19
4 Контрольные вопросы	20

Список иллюстраций

2.1 Установка пакетов MariaDB и просмотр конфигурации	6
2.2 Запуск MariaDB и включение автозагрузки	8
2.3 Проверка прослушивания порта 3306	8
2.4 Настройка безопасности MariaDB	9
2.5 Справка и список баз данных	10
2.6 Статус MariaDB до настройки UTF-8	11
2.7 Файл utf8.cnf	11
2.8 Статус MariaDB после настройки UTF-8	12
2.9 Содержимое таблицы city	14
2.10 Описание структуры таблицы city	15
2.11 Просмотр баз данных и таблиц	15
2.12 Резервное копирование и восстановление базы данных	16
2.13 Копирование файлов	17
2.14 Скрипт автоматической настройки MariaDB	18

Список таблиц

1 Цель работы

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

2 Выполнение

2.1 Установка MariaDB и первичная настройка безопасности

- На виртуальной машине **server** выполнен переход в режим суперпользователя и установка пакетов MariaDB:

- `dnf -y install mariadb mariadb-server`

В результате были установлены сервер MariaDB, клиентские утилиты и сопутствующие компоненты (включая `mariadb-server`, `mariadb-client-utils`, `mariadb-common`, `mariadb-backup` и др.). Установка завершилась успешно со статусом **Complete!**.

```
Installed:
  mariadb-3:10.11.11-1.el10.x86_64
  mariadb-client-utils-3:10.11.11-1.el10.x86_64
  mariadb-errmsg-3:10.11.11-1.el10.noarch
  mariadb-server-3:10.11.11-1.el10.x86_64
  mysql-selinux-1.0.14-1.el10_0.noarch
  perl-Sys-Hostname-1.25-512.2.el10_0.x86_64

Complete!
[root@server.elsaiedadel.net www.elsaiedadel.net]# [root@server.elsaiedadel.net www.elsaiedadel.net]# cat /etc/my.cnf.d/
cat: /etc/my.cnf.d/: Is a directory
[root@server.elsaiedadel.net www.elsaiedadel.net]# ls /etc/my.cnf.d/
auth_gssapi.cnf  enable_encryption.preset  mysql-clients.cnf  provider_lz4.cnf  provider_snappy.cnf
client.cnf       mariadb-server.cnf      provider_bzip2.cnf provider_lzo.cnf  spider.cnf
[root@server.elsaiedadel.net www.elsaiedadel.net]# cat /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

[root@server.elsaiedadel.net www.elsaiedadel.net]#
```

Рис. 2.1: Установка пакетов MariaDB и просмотр конфигурации

2. Выполнена проверка конфигурационных файлов MariaDB:

- При попытке просмотра `/etc/my.cnf.d/` командой `cat /etc/my.cnf.d/` получено сообщение *Is a directory*, что подтверждает: `/etc/my.cnf.d` – каталог с дополнительными фрагментами конфигурации.
- Командой `ls /etc/my.cnf.d/` определён набор конфигурационных файлов:
 - `auth_gssapi.cnf` – параметры аутентификации через GSSAPI (Kerberos).
 - `client.cnf` – настройки клиентских программ.
 - `mysql-clients.cnf` – общие параметры клиентских утилит.
 - `mariadb-server.cnf` – основной файл конфигурации серверной части.
 - `enable-encryption.preset` – предустановки параметров шифрования.
 - `provider_bzip2.cnf, provider_lz4.cnf, provider_lzo.cnf, provider_snappy.cnf` – настройки модулей сжатия.
 - `spider.cnf` – параметры движка Spider.
- Файл `/etc/my.cnf` является основным конфигурационным файлом и подключает все файлы из каталога `/etc/my.cnf.d`.

Построчный комментарий содержимого `/etc/my.cnf`:

- `#` – комментарий.
- `[client-server]` – секция параметров, применяемых и к клиенту, и к серверу.
- `!includedir /etc/my.cnf.d` – директива подключения всех файлов конфигурации из каталога.

3. Выполнен запуск MariaDB и добавление службы в автозагрузку:

```
systemctl start mariadb  
systemctl enable mariadb
```

В результате созданы символические ссылки systemd, подтверждающие включение сервиса при старте ОС.

```
[root@server.elsaiedadel.net www.elsaiedadel.net]# systemctl start mariadb
[root@server.elsaiedadel.net www.elsaiedadel.net]# 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.elsaiedadel.net www.elsaiedadel.net]# ss -tulpen | grep 3306
tcp    LISTEN  0          80                  0.0.0.0:3306          0.0.0.0:*      users:(("mariadb",pid=24689,fd=18))
e/mariadb.service <->
tcp    LISTEN  0          80                  [::]:3306           [::]:*      users:(("mariadb",pid=24689,fd=19))
e/mariadb.service v6only:1 <->
[root@server.elsaiedadel.net www.elsaiedadel.net]#
```

Рис. 2.2: Запуск MariaDB и включение автозагрузки

4. Проверено, что сервер MariaDB прослушивает порт **3306**:

```
ss -tulpen | grep 3306
```

В выводе отображается процесс `mariadb`, слушающий порт 3306 по IPv4 и IPv6.

```
[root@server.elsaiedadel.net www.elsaiedadel.net]# systemctl start mariadb
[root@server.elsaiedadel.net www.elsaiedadel.net]# 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.elsaiedadel.net www.elsaiedadel.net]# ss -tulpen | grep 3306
tcp    LISTEN  0          80                  0.0.0.0:3306          0.0.0.0:*      users:(("mariadb",pid=24689,fd=18))
e/mariadb.service <->
tcp    LISTEN  0          80                  [::]:3306           [::]:*      users:(("mariadb",pid=24689,fd=19))
e/mariadb.service v6only:1 <->
[root@server.elsaiedadel.net www.elsaiedadel.net]#
```

Рис. 2.3: Проверка прослушивания порта 3306

5. Выполнена настройка безопасности MariaDB:

```
mysql_secure_installation
```

В ходе диалога:

- включена аутентификация через `unix_socket`;
- установлен пароль пользователя `root` СУБД;
- применены обновления таблиц привилегий.

```
[root@server.elsaiedadel.net www.elsaiedadel.net]# 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]
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]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
    ... Success!
```

Рис. 2.4: Настройка безопасности MariaDB

6. Выполнен вход в MariaDB и просмотр доступных команд и баз данных:

```
\h
SHOW DATABASES;
```

В системе присутствуют базы данных:

- `information_schema`
- `mysql`
- `performance_schema`
- `sys`

```

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 charset
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.
notee    (\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.
warnings (\W) Show warnings after every statement.

For server side help, type 'help contents'

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+

```

Рис. 2.5: Справка и список баз данных

2.2 Конфигурация кодировки символов

1. Выполнена проверка статуса MariaDB до изменения кодировки:

`status`

Пояснение вывода:

- версия клиента и сервера MariaDB;
- параметры подключения (user, socket, protocol);
- текущие кодировки сервера, базы данных, клиента и соединения;
- статистика работы сервера (uptime, queries, threads).

```

MariaDB [(none)]>
MariaDB [(none)]> status
-----
mysql Ver 15.1 Distrib 10.11.11-MariaDB, for Linux (x86_64) using EditLine wrapper

Connection id:          13
Current database:
Current user:           root@localhost
SSL:                   Not in use
Current pager:          stdout
Using outfile:
Using delimiter:         ;
Server:                 MariaDB
Server version:         10.11.11-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:                 2 min 24 sec

Threads: 1 Questions: 25 Slow queries: 0 Opens: 20 Open tables: 13 Queries per second avg: 0.173
-----
MariaDB [(none)]>

```

Рис. 2.6: Статус MariaDB до настройки UTF-8

2. В каталоге `/etc/my.cnf.d` создан файл `utf8.cnf` со следующим содержимым:

[client]

`default-character-set = utf8`

[mysqld]

`character-set-server = utf8`

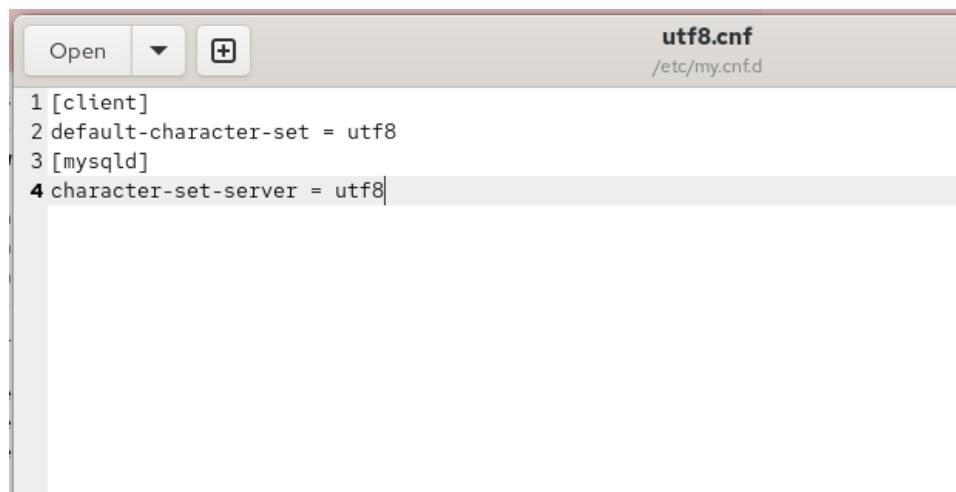


Рис. 2.7: Файл utf8.cnf

3. Выполнен перезапуск MariaDB и повторная проверка статуса:

```
systemctl restart mariadb
```

```
status
```

Результат:

- Server characterset изменён с latin1 на utf8mb3;
- Db characterset изменён с latin1 на utf8mb3.

Это подтверждает успешное применение конфигурации кодировки UTF-8.

```
[root@server.elsaiedadel.net my.cnf.d]#
[root@server.elsaiedadel.net my.cnf.d]# systemctl restart mariadb
[root@server.elsaiedadel.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: 10.11.11-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 10.11.11-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:          10.11.11-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:                 12 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second avg: 0.333
-----
MariaDB [(none)]> █
```

Рис. 2.8: Статус MariaDB после настройки UTF-8

2.3 Создание базы данных и управление доступом

1. Выполнен вход в систему управления базами данных MariaDB с правами администратора:

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

2. Создана база данных addressbook с указанием кодировки utf8 и правил сортировки utf8_general_ci, что обеспечивает корректную работу с кириллическими символами:

```
CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Команда выполнена успешно, что подтверждается сообщением *Query OK, 1 row affected.*

3. Выполнен переход к работе с созданной базой данных:

```
USE addressbook;
```

4. Проверено наличие таблиц в базе данных addressbook:

```
SHOW TABLES;
```

Результат — пустой набор, так как таблицы на данном этапе отсутствуют.

5. Создана таблица city с двумя полями строкового типа:

- name — имя человека;
- city — название города проживания.

```
CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
```

6. Таблица city заполнена тестовыми данными:

```
INSERT INTO city(name, city) VALUES ('Иванов', 'Москва');  
INSERT INTO city(name, city) VALUES ('Петров', 'Сочи');  
INSERT INTO city(name, city) VALUES ('Сидоров', 'Дубна');
```

7. Выполнен запрос на выборку всех данных из таблицы:

```
SELECT * FROM city;
```

Результат выполнения запроса: Отображены все строки таблицы city, содержащие три записи. Это подтверждает корректное создание таблицы и успешную вставку данных.

```

MariaDB [(none)]>
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]> 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 (0.004 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)

MariaDB [addressbook]> █

```

Рис. 2.9: Содержимое таблицы city

8. Создан пользователь MariaDB для работы с базой данных addressbook с доступом с любых хостов:

```
CREATE USER elsaiedadel@'%' IDENTIFIED BY '123456';
```

9. Пользователю предоставлены права на основные операции с данными (чтение, добавление, изменение, удаление):

```
GRANT SELECT, INSERT, UPDATE, DELETE ON addressbook.* TO elsaiedadel@'%';
```

10. Выполнено обновление таблиц привилегий для применения изменений:

```
FLUSH PRIVILEGES;
```

11. Просмотрена структура таблицы city:

```
DESCRIBE city;
```

В результате отображены поля таблицы, их типы данных и допустимость значений NULL.

```

MariaDB [addressbook]>
MariaDB [addressbook]> CREATE USER elsaiedadel@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.001 sec)

MariaDB [addressbook]> GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO elsaiedadel@'%';
Query OK, 0 rows affected (0.001 sec)

MariaDB [addressbook]> FLUSH PRIVILEGES;
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 (0.000 sec)

MariaDB [addressbook]>

```

Рис. 2.10: Описание структуры таблицы city

12. Выполнен выход из интерактивной оболочки MariaDB:

```
quit
```

13. Проверен список всех баз данных в системе:

```
mysqlshow -u root -p
```

14. Проверен список таблиц базы данных addressbook:

```
mysqlshow -u root -p addressbook
```

```

[root@server.elsaiedadel.net my.cnf.d]# mysqlshow -u root -p
Enter password:
+-----+
| Databases      |
+-----+
| addressbook   |
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
[root@server.elsaiedadel.net my.cnf.d]# mysqlshow -u root -p addressbook
Enter password:
Database: addressbook
+-----+
| Tables      |
+-----+
| city      |
+-----+
[root@server.elsaiedadel.net my.cnf.d]#

```

Рис. 2.11: Просмотр баз данных и таблиц

2.4 Резервное копирование и восстановление базы данных

- На виртуальной машине server создан каталог для хранения резервных копий:

```
mkdir -p /var/backup
```

- Выполнено резервное копирование базы данных addressbook в SQL-файл:

```
mysqldump -u root -p addressbook > /var/backup/addressbook.sql
```

- Создана сжатая резервная копия базы данных с использованием gzip:

```
mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz
```

- Создана сжатая резервная копия с указанием даты и времени создания в имени файла:

```
mysqldump -u root -p addressbook | gzip > $(date +/var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz)
```

- Выполнено восстановление базы данных из обычной резервной копии:

```
mysql -u root -p addressbook < /var/backup/addressbook.sql
```

- Выполнено восстановление базы данных из сжатой резервной копии:

```
zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook
```

```
[root@server.elsaiedadel.net my.cnf.d]# 
[root@server.elsaiedadel.net my.cnf.d]# mkdir -p /var/backup
[root@server.elsaiedadel.net my.cnf.d]# mysqldump -u root -p addressbook > /var/backup/addressbook.sql
Enter password:
[root@server.elsaiedadel.net my.cnf.d]# mysqldump -u root -p addressbook > gzip > /var/backup/addressbook.sql.gz
Enter password:
[root@server.elsaiedadel.net my.cnf.d]# mysqldump -u root -p addressbook > $(date +/var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz)
Enter password:
[root@server.elsaiedadel.net my.cnf.d]# ls /var/backup/
addressbook.20260103.100154.sql.gz addressbook.sql addressbook.sql.gz
[root@server.elsaiedadel.net my.cnf.d]# mysql -u root -p addressbook < /var/backup/addressbook.sql
Enter password:
[root@server.elsaiedadel.net my.cnf.d]# zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook
```

Рис. 2.12: Резервное копирование и восстановление базы данных

2.5 Внесение изменений в настройки внутреннего окружения виртуальной машины

1. В каталоге `/vagrant/provision/server` создана структура для хранения конфигурационных файлов MariaDB и резервных копий базы данных:

```
mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d  
mkdir -p /vagrant/provision/server/mysql/var/backup
```

В созданные каталоги скопированы:

- файл конфигурации кодировки `utf8.cnf`;
- резервные копии базы данных `addressbook`.

```
[root@server.elsaiedadel.net backup]#  
[root@server.elsaiedadel.net backup]# cd /vagrant/provision/server/  
[root@server.elsaiedadel.net server]# mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d  
[root@server.elsaiedadel.net server]# mkdir -p /vagrant/provision/server/mysql/var/backup  
[root@server.elsaiedadel.net server]# cp -R /etc/my.cnf.d/utf8.cnf /vagrant/provision/server/mysql/etc/my.cnf.d/  
[root@server.elsaiedadel.net server]# cp -R /var/backup/* /vagrant/provision/server/mysql/var/backup/  
[root@server.elsaiedadel.net server]# cd /vagrant/provision/server/  
[root@server.elsaiedadel.net server]# touch mysql.sh  
[root@server.elsaiedadel.net server]# chmod +x mysql.sh  
[root@server.elsaiedadel.net server]# █
```

Рис. 2.13: Копирование файлов

2. В каталоге `/vagrant/provision/server` создан исполняемый скрипт `mysql.sh`:

```
touch mysql.sh  
chmod +x mysql.sh
```

Скрипт автоматизирует:

- установку MariaDB;
- копирование конфигурационных файлов;
- восстановление резервной копии базы данных;
- выполнение базовой настройки безопасности;
- создание базы данных `addressbook`.

```
#!/bin/bash
# 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
```

Рис. 2.14: Скрипт автоматической настройки MariaDB

3. Для автоматического выполнения скрипта при запуске виртуальной машины в файл `Vagrantfile` добавлена конфигурация provisioner'a:

3 Вывод

В ходе выполнения лабораторной работы была произведена установка и настройка сервера баз данных MariaDB на виртуальной машине. Выполнены действия по первичной конфигурации и усилению безопасности СУБД, настройке кодировки символов для корректной работы с кириллическими данными, созданию пользовательской базы данных и таблиц, а также управлению доступом пользователей. Были отработаны операции резервного копирования и восстановления базы данных, что является важным элементом администрирования. Дополнительно реализована автоматизация установки и настройки MariaDB с помощью provisioning-скрипта Vagrant, что обеспечивает воспроизводимость конфигурации и упрощает развертывание серверного окружения.

4 Контрольные вопросы

1. Какая команда отвечает за настройки безопасности в MariaDB?

За базовую настройку безопасности MariaDB отвечает команда `mysql_secure_installation`. Она запускает интерактивный скрипт, с помощью которого можно задать пароль пользователю `root` СУБД, отключить удалённый доступ для `root`, удалить анонимных пользователей и тестовую базу данных, а также обновить таблицы привилегий.

2. Как настроить MariaDB для доступа через сеть?

Для доступа к MariaDB по сети необходимо: - в конфигурационных файлах (например, `/etc/my.cnf.d/mariadb-server.cnf`) указать параметр `bind-address` (например, `0.0.0.0` для приёма подключений с любых интерфейсов); - создать пользователя с указанием хоста (`user@'%'` или конкретный IP); - предоставить этому пользователю необходимые привилегии; - убедиться, что порт 3306 открыт в настройках брандмауэра.

3. Какая команда позволяет получить обзор доступных баз данных после входа в среду оболочки MariaDB?

Для просмотра списка доступных баз данных используется SQL-команда:

```
SHOW DATABASES;
```

4. Какая команда позволяет узнать, какие таблицы доступны в базе данных? После выбора базы данных командой `USE имя_базы;` список таблиц выводится командой:

```
SHOW TABLES;
```

5. Какая команда позволяет узнать, какие поля доступны в таблице? Для просмотра структуры таблицы и списка её полей используется команда:

```
DESCRIBE имя_таблицы;
```

или

```
SHOW COLUMNS FROM имя_таблицы;
```

6. Какая команда позволяет узнать, какие записи доступны в таблице? Для просмотра содержимого таблицы применяется SQL-запрос:

```
SELECT * FROM имя_таблицы;
```

Он выводит все строки и все столбцы указанной таблицы.

7. Как удалить запись из таблицы? Удаление записи выполняется командой DELETE с указанием условия:

```
DELETE FROM имя_таблицы WHERE условие;
```

Например, удаление записи по значению поля:

```
DELETE FROM city WHERE name = 'Иванов';
```

8. Где расположены файлы конфигурации MariaDB? Что можно настроить с их помощью? Основные файлы конфигурации MariaDB располагаются в:

- /etc/my.cnf;
- /etc/my.cnf.d/.

С их помощью можно настраивать параметры сервера и клиента: сетевые настройки, кодировки символов, параметры хранения данных, пути к файлам, настройки безопасности, журналирование, производительность и поведение сервиса при запуске.

9. Где располагаются файлы с базами данных MariaDB? Файлы баз данных MariaDB по умолчанию хранятся в каталоге:

/var/lib/mysql/

Внутри этого каталога для каждой базы данных создаётся отдельный подкаталог с файлами таблиц и служебной информацией.

10. Как сделать резервную копию базы данных и затем её восстановить?

Резервное копирование выполняется с помощью утилиты mysqldump, например:

```
mysqldump -u root -p имя_базы > backup.sql
```

Восстановление базы данных из резервной копии выполняется командой:

```
mysql -u root -p имя_базы < backup.sql
```

Для сжатых резервных копий можно использовать gzip и zcat:

```
mysqldump -u root -p имя_базы | gzip > backup.sql.gz
zcat backup.sql.gz | mysql -u root -p имя_базы
```