

# Отчёт по лабораторной работе №6

Дисциплина: Администрирование сетевых подсистем

true

## Содержание

### Цель работы

Целью данной работы является приобретение практических навыков по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

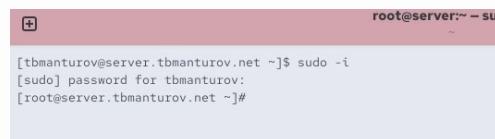
### Задание

1. Установить необходимые для работы MariaDB пакеты
2. Настроить в качестве кодировки символов по умолчанию utf8 в базах данных.
3. В базе данных MariaDB создать тестовую базу addressbook, содержащую таблицу city с полями name и city, т.е., например, для некоторого сотрудника указан город, в котором он работает
4. Создать резервную копию базы данных addressbook и восстановить из неё данные
5. Написать скрипт для Vagrant, фиксирующий действия по установке и настройке базы данных MariaDB во внутреннем окружении виртуальной машины server.  
Соответствующим образом следует внести изменения в Vagrantfile

## Выполнение лабораторной работы

### Установка MariaDB

Загрузили нашу операционную систему и перешли в рабочий каталог с проектом: cd /var/tmp/tbmanturov/vagrant ([рис. @fig-001])



```
[tbmanturov@server.tbmanturov.net ~]$ sudo -i
[sudo] password for tbmanturov:
[root@server.tbmanturov.net ~]#
```

Переход в режим суперпользователя

Далее установили необходимые для работы с базами данных пакеты: dnf -y install mariadb mariadb-server ([рис. @fig-004])



```
[root@server.tbmanturov.net ~]$ dnf -y install mariadb mariadb-server
Extra Packages for Enterprise Linux 18 - x86_64
41.40 kB | 32 kB   00:00
3.1 kB/s | 5.5 kB   00:01
```

Установка необходимых пакетов для работы с базами данных

Далее посмотрели конфигурационные файлы mariadb в каталоге /etc/my.cnf.d и файл /etc/my.cnf ([рис. @fig-005])

Пояснения к файлам:

- Файл */etc/my.cnf*:
  - Основной конфигурационный файл MySQL/MariaDB
  - Содержит секцию [client-server] для общих настроек клиента и сервера
  - директива includedir /etc/my.cnf.d - это означает, что файлы из каталога /etc/my.cnf.d подключаются к основной конфигурации
- Файлы в каталоге */etc/my.cnf.d/*:
  - auth\_gssapi.cnf - настройки аутентификации GSSAPI
  - client.cnf - параметры для клиентских подключений
  - enable\_encryption.preset - предустановки для включения шифрования
  - mariadb-server.cnf - основные настройки сервера MariaDB
  - mysql-clients.cnf - настройки для клиентов MySQL
  - provider\_\*.cnf - конфигурации различных провайдеров сжатия (bzip2, lz4, lzo, snappy)
  - spider.cnf - настройки для движка Spider

```
[root@server.tbmanturov.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.tbmanturov.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
```

Файлы в каталоге /etc/my.cnf.d и файл /etc/my.cnf

Далее для запуска и включения программного обеспечения mariadb использовали: `systemctl start mariadb` и `systemctl enable mariadb` ([рис. @fig-006])

```
[root@server.tbmanturov.net ~]# systemctl start mariadb
[root@server.tbmanturov.net ~]# systemctl enable mariadb
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.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.tbmanturov.net ~]#
```

Запуск и включение mariadb

Далее убедились, что mariadb прослушивает порт 3306: `ss -tulpen | grep mariadb` ([рис. @fig-007])

```
[root@server.tbmanturov.net ~]# ss -tulpen | grep mariadb
Failed to open cgroup by ID
tcp LISTEN 0 0 0.0.0.0:3306 0.0.0.0:* users:(("mariadb",pid:8198,fd:18))
  license:mariadb.service <-->
  tcp LISTEN 0 0 :::3306 ::*: users:(("mariadb",pid:8198,fd:19))
  license:mariadb.service vhostonly:1 <-->
[root@server.tbmanturov.net ~]#
```

Порт который прослушивает mariadb

Далее запустили скрипт конфигурации безопасности mariadb: `mysql_secure_installation`. Установили пароль для пользователя root базы данных, отключили удалённый корневой доступ и удалили тестовую базу данных и любых анонимных пользователей ([рис. @fig-008]), ([рис. @fig-009]), ([рис. @fig-010])

```
[root@server.tbmanturov.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] n
... skipping.

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

Change the root password? [Y/n] n
... skipping.

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.
```

Запуск скрипта конфигурации безопасности mariadb (1)

Для входа в базу данных с правами администратора базы данных ввели mysql -u root -p ([рис. @fig-011])

```
[root@server.tbmanturov.net ~]# mysql -u root -p
Enter password: [REDACTED]
```

Вход в базу данных с правами администратора (1)

Псомотреди список команд MySQL, введя \h ([рис. @fig-012])

```
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 ';'
?
    (\?) Syonym for 'help'.
charsets  (\CC) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
clear    (\C) Clear the current input statement.
connect   (\r) Connect to the server. Optional arguments are db and host.
delimiter  (\d) Set statement delimiter.
edit     (\e) Edit command with $EDITOR.
ego      (\O) Send command to MariaDB server, display result vertically.
exit     (\q) Exit from MariaDB.
quit     (\q) Exit from MariaDB.
Send command to MariaDB server.
help     (\H) Display this help.
nopager  (\N) Disable pager, print to stdout.
noresetlog (\L) Don't reset log 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.
rehash   (\R) Rehash completion hash.
rehashdb (\R) Rehash completion hash.
source   (\.) Execute an SQL script file. Takes a file name as an argument.
status   (\s) Get status information from the server.
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'.
```

Список команд MySQL

Из приглашения интерактивной оболочки MariaDB для отображения доступных в настоящее время баз данных ввели MySQL-запрос SHOW DATABASES; ([рис. @fig-013])

В системе присутствуют вот такие базы данных:

- information\_schema - системная БД, содержащая метаданные о всех других базах данных
- mysql - системная БД, хранящая информацию о пользователях, привилегиях и других системных настройках
- performance\_schema - системная БД для мониторинга производительности сервера
- sys - системная БД, предоставляющая удобные представления для анализа производительности

Это стандартный набор системных баз данных, которые создаются при первоначальной установке MariaDB/MySQL. Пользовательские базы данных в системе пока что отсутствуют.

```
MariaDB [(none)]> SHOW DATABASES
    ->
    ->
    ->
    ->
```

Доступные базы данных в настоящее время

Для выхода из интерфейса интерактивной оболочки MariaDB ввели `exit`; ([рис. @fig-014])

```
-> ``C
MariaDB [(none)]> exit
Bye
```

Выход из интерфейса интерактивной оболочки MariaDB

## Конфигурация кодировки символов

Опять вошли в базу данных с правами администратора `mysql -u root -p` ([рис. @fig-015])

```
[root@server.tbmanturov.net ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
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.
```

Вход в базу данных с правами администратора (2)

Далее посмотрели статус MariaDB введя `status` ([рис. @fig-016])

Пояснения к статусу MariaDB:

- Информация о клиенте:
  - mysql Ver 15.1 Distrib 10.11.11-MariaDB - версия клиента и сервера MariaDB
  - for Linux (x86\_64) - платформа
- Текущее соединение:
  - Connection id: 14 - идентификатор соединения
  - Current database: - нет выбранной БД
  - Current user: root@localhost - пользователь и хост
  - SSL: Not in use - SSL не используется
- Серверная информация:
  - Server: MariaDB - тип СУБД
  - Server version: 10.11.11-MariaDB - версия сервера
  - Connection: Localhost via UNIX socket - тип подключения
  - UNIX socket: /var/lib/mysql/mysql.sock - сокет-файл
- Кодировки:
  - Server characterset: latin1 - кодировка сервера
  - Client characterset: utf8mb3 - кодировка клиента
- Статистика сервера:
  - Uptime: 15 min 12 sec - время работы сервера
  - Threads: 1 - количество активных потоков

- Questions: 27 - выполнено запросов
- Queries per second avg: 0.029 - среднее количество запросов в секунду

```
MariaDB [(none)]> status
+-----+
| mysql Ver 15.1 Distrib 10.11.11-MariaDB, for Linux (x86_64) using Editline wrapper |
+-----+
Current connection: 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: latin1
Db characterset: latin1
Client characterset: utf8mb3
Conn. characterset: utf8mb3
UNIX socket: /var/lib/mysql/mysql.sock
Uptime: 3 min 52 sec
Threads: 1 Questions: 16 Slow queries: 0 Opens: 20 Open tables: 13 Queries per second avg: 0.008
+-----+
```

Статус MariaDB

Далее в каталоге `/etc/my.cnf.d` создали файл `utf8.cnf` ([рис. @fig-017]):

```
cd /etc/my.cnf.d
touch utf8.cnf
```

```
[root@server.tbmanturov.net ~]# cd /etc/my.cnf.d
[root@server.tbmanturov.net my.cnf.d]# touch utf8.cnf
[root@server.tbmanturov.net my.cnf.d]#
```

Создание файла utf8.cnf

Открыв этот файл на редактирование пропишем в нём следующие строки ([рис. @fig-018]):

```
[client]
default-character-set = utf8
[mysqld]
character-set-server = utf8
```

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

Редактирование файла utf8.cnf

Далее перезапустили MariaDB с помощью `systemctl restart mariadb` ([рис. @fig-019])

```
[root@server.tbmanturov.net my.cnf.d]# systemctl restart mariadb
[root@server.tbmanturov.net my.cnf.d]#
```

Презапуск MariaDB

Далее опять вошли в базу данных с правами администратора, посмотрели статус MariaDB и поняли что сервер и базы данных теперь используют utf8mb3 вместо latin1, что обеспечивает поддержку Unicode символов и корректное отображение кириллицы и других языков ([рис. @fig-020])

```
MariaDB [(none)]> status
+-----+
| mysql Ver 15.1 Distrib 10.11.11-MariaDB, for Linux (x86_64) using Editline wrapper |
+-----+
Current connection: 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: 15 sec
Threads: 1 Questions: 4 Slow queries: 0 Opens: 17 Open tables: 10 Queries per second avg: 0.266
+-----+
```

Статус MariaDB после конфигурации кодировки символов

## Создание базы данных

Опять зашли в базу данных с правами администратора mysql -u root -p ([рис. @fig-021])

```
[root@server.tbmanturov.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
```

Вход в базу данных с правами администратора (3)

Создали базу данных с именем addressbook: CREATE DATABASE addressbook  
CHARACTER SET utf8 COLLATE utf8\_general\_ci; ([рис. @fig-022])

```
MariaDB [(none)]> CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> USE addressbook;
```

Создание базы данных с именем addressbook

Перешли к базе данных addressbook: USE addressbook; ([рис. @fig-023])

```
MariaDB [(none)]> USE addressbook;
Database changed
```

Переход к базе данных addressbook

Отобразили имеющиеся в базе данных addressbook таблицы: SHOW TABLES; ([рис. @fig-024])

```
MariaDB [addressbook]> SHOW TABLES;
Empty set (0.002 sec)
```

Таблицы имеющиеся в базе данных addressbook

Создали таблицу city с полями name и city: CREATE TABLE city(name VARCHAR(40),  
city VARCHAR(40)); ([рис. @fig-025])

```
MariaDB [addressbook]> CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
Query OK, 0 rows affected (0.017 sec)
```

Создание таблицы city

Заполнили несколько строк таблицы некоторыми данными по аналогии в соответствии  
с синтаксисом MySQL ([рис. @fig-026]):

```
INSERT INTO city(name,city) VALUES ('Иванов','Москва');

INSERT INTO city(name,city) VALUES ('Петров','Сочи');

INSERT INTO city(name,city) VALUES ('Сидоров','Дубна');
```

```
MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Иванов','Москва')
-> ^C
MariaDB [addressbook]> SHOW TABLES;
+-----+
| Tables_in_addressbook |
+-----+
| city |
+-----+
1 row in set (0.001 sec)

MariaDB [addressbook]> SHOW city
-> 
-> 
-> ^C
MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Петров','Сочи')
-> 
-> 
-> ^C
MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Сидоров','Дубна')
-> ^C
MariaDB [addressbook]>
```

Заполнение строк таблицы city

Сделали следующий MySQL-запрос: SELECT \* FROM city; ([рис. @fig-027])

Таблица содержит 3 записи с двумя полями name и city:

- Иванов | Москва
- Петров | Сочи

- Сидоров | Дубна

```
MariaDB [addressbook]> SELECT * FROM city;
+-----+-----+
| name | city |
+-----+-----+
| Sidorov | Dubna |
| Petrov | Sochi |
| Иванов | Москва |
+-----+
3 rows in set (0.000 sec)
```

Запрос на выборку всех данных из таблицы city

Создали пользователя для работы с базой данных addressbook и задали для него пароль: ``CREATE USER tbmanturov@'%' IDENTIFIED BY 'password';1234' ([рис. @fig-028])

```
MariaDB [addressbook]> CREATE USER tbmanturov@'%' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0.002 sec)
```

Создание пользователя для работы с базой данных addressbook

Далее предоставили права доступа созданному пользователю tbmanturov на действия с базой данных addressbook (просмотр, добавление, обновление, удаление данных):  
GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.\* TO tbmanturov@'%'; ([рис. @fig-029])

```
MariaDB [addressbook]> GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO tbmanturov@'%';
Query OK, 0 rows affected (0.002 sec)
```

Предоставление прав доступа созданному пользователю на действия с базой данных addressbook

Обновили привилегии (права доступа) базы данных addressbook: FLUSH PRIVILEGES; ([рис. @fig-030])

```
MariaDB [addressbook]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)
```

Обновление прав доступа базы данных addressbook

Далее посмотрели общую информацию о таблице city базы данных addressbook:  
DESCRIBE city; ([рис. @fig-031])

```
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)
```

Общая информация о таблице city базы данных addressbook

Далее вышли из окружения MariaDB: quit ([рис. @fig-032])

```
MariaDB [addressbook]> quit
Bye
```

Выход из окружения MariaDB

Посмотрели список баз данных: mysqlshow -u root -p ([рис. @fig-033])

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

Список баз данных

Посмотрели список таблиц базы данных addressbook: mysqlshow -u root -p addressbook и mysqlshow -u tbmanturov -p addressbook ([рис. @fig-034]), ([рис.

@fig-035])

```
[root@server.tbmanturov.net my.cnf.d]# mysqlshow -u root -p addressbook
Enter password:
Database: addressbook
+-----+
| Tables |
+-----+
| city   |
+-----+
```

Список таблиц базы данных addressbook от администратора

## Резервные копии

На виртуальной машине server создали каталог для резервных копий: `mkdir -p /var/backup` ([рис. @fig-036])

```
[root@server.tbmanturov.net my.cnf.d]# mkdir -p /var/backup
```

Создание каталога для резервных копий

Далее сделали резервную копию базы данных addressbook, сжатую резервную копию и сжатую резервную копию с указанием даты ([рис. @fig-037]):

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

```
[root@server.tbmanturov.net ~]# mysqldump -u root -p addressbook > /var/backup/addressbook.sql
Enter password:
[root@server.tbmanturov.net ~]# mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz
Enter password:
[root@server.tbmanturov.net ~]# 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
```

Создание резервных копий базы данных addressbook

Далее восстановили базу данных addressbook из резервной копии и из сжатой резервной копии ([рис. @fig-038]):

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

```
[root@server.tbmanturov.net ~]# mysql -u root -p addressbook < /var/backup/addressbook.sql
[bash]: /var/backup/addressbook.sql: No such file or directory
[root@server.tbmanturov.net ~]# mysql -u root -p addressbook < /var/backup/addressbook.sql
Enter password:
[root@server.tbmanturov.net ~]# mysql -u root -p addressbook < /var/backup/addressbook.sql
Enter password:
[root@server.tbmanturov.net ~]# zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook
Enter password:
```

Восстановление базы данных addressbook из резервных копий

## Внесение изменений в настройки внутреннего окружения виртуальной машины

На виртуальной машине server перешли в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создали в нём каталог `mysql`, в который поместили в соответствующие подкаталоги конфигурационные файлы MariaDB и резервную копию базы данных addressbook ([рис. @fig-039]):

```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d
mkdir -p /vagrant/provision/server/mysql/var/backup
cp -R /etc/my.cnf.d(utf8.cnf /vagrant/provision/server/mysql/etc/my.cnf.d/
cp -R /var/backup/* /vagrant/provision/server/mysql/var/backup/
```

```
[root@server tbmanturov.net]# cd /vagrant/provision/server
[root@server tbmanturov.net server]# mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d
[root@server tbmanturov.net server]# cp -R /etc/my.cnf.d/*utf8.cnf
/vagrant/provision/server/mysql/etc/my.cnf.d
```

Копирование конфигурационных файлов и резервных копий базы данных в каталог mysql

В каталоге /vagrant/provision/server создали исполняемый файл mysql.sh ([рис. @fig-040]):

```
cd /vagrant/provision/server
```

```
touch mysql.sh
```

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

```
[root@server tbmanturov.net server]# touch mysql.sh
[root@server tbmanturov.net server]# chmod +x mysql.sh
```

Создание исполняемого файла mysql.sh

Открыв его на редактирование прописали в нём следующие строки ([рис. @fig-041]):

```
#!/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
```

```

GNU nano 8.1                               mysql.sh
#!/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
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

```

Редактирование файла mysql.sh

Этот скрипт повторяет произведённые вами действия по установке и настройке сервера баз данных

Для отработки созданного скрипта во время загрузки виртуальных машин в конфигурационном файле Vagrantfile необходимо добавить в конфигурации сервера следующую запись ([рис. @fig-042]):

```

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

# Упрощенные provision скрипты для теста
server.vm.provision "fix network",
  type: "shell",
  inline: <<-SHELL
    # Фиксим сетевые интерфейсы
    sudo nmcli con mod "System eth0" ipv4.method auto
    sudo nmcli con up "System eth0"
    sudo systemctl restart network
  SHELL
server.vm.provision "server mysql", I
  type: "shell",
  preserve_order: true,
  path: "provision/server/mysql.sh"
# Уберите сложные настройки iptables для теста
# server.vm.provision "server internet", ...

```

Редактирование Vagrantfile

## Контрольные вопросы + ответы

1. Какая команда отвечает за настройки безопасности в MariaDB?

Настройки безопасности в MariaDB обычно управляются с помощью команды `mysql_secure_installation`. Эта команда выполняет несколько шагов, включая установку пароля для пользователя `root`, удаление анонимных учетных записей, отключение удаленного входа для пользователя `root` и удаление тестовых баз данных.

2. Как настроить MariaDB для доступа через сеть?

Для настройки MariaDB для доступа через сеть, вы можете отредактировать файл конфигурации MariaDB (обычно называемый `my.cnf`) и убедиться, что параметр `bind-address` установлен на IP-адрес, доступный в вашей сети. Также, убедитесь, что пользователь имеет права доступа извне, например, с использованием команды `GRANT`.

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

Обзор доступных баз данных после входа в среду оболочки MariaDB можно получить

с помощью команды SHOW DATABASES;.

4. Какая команда позволяет узнать, какие таблицы доступны в базе данных?

Для просмотра доступных таблиц в базе данных используйте команду SHOW TABLES;.

5. Какая команда позволяет узнать, какие поля доступны в таблице?

Чтобы узнать, какие поля доступны в таблице, используйте команду DESCRIBE table\_name; или SHOW COLUMNS FROM table\_name;.

6. Какая команда позволяет узнать, какие записи доступны в таблице?

Для просмотра записей в таблице можно использовать команду SELECT \* FROM table\_name;.

7. Как удалить запись из таблицы?

Для удаления записи из таблицы используйте команду DELETE FROM table\_name WHERE condition;, где condition - условие, определяющее, какие записи следует удалить

8. Где расположены файлы конфигурации MariaDB? Что можно настроить с их помощью?

Файлы конфигурации MariaDB обычно располагаются в различных местах в зависимости от системы, но основной файл - my.cnf. Он может быть в /etc/my.cnf, /etc/mysql/my.cnf или /usr/etc/my.cnf. С помощью этих файлов можно настроить различные параметры, такие как порт, пути к файлам данных, параметры безопасности и другие.

9. Где располагаются файлы с базами данных MariaDB?

Файлы с базами данных MariaDB располагаются в директории данных. Обычно это /var/lib/mysql/ на Linux-системах.

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

Для создания резервной копии базы данных используйте команду mysqldump. Например, mysqldump -u username -p dbname > backup.sql. Для восстановления базы данных из резервной копии используйте команду mysql -u username -p dbname < backup.sql.

## Выводы

В ходе выполнения лабораторной работы №6 мы приобрели практические навыки по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

## Список литературы

1. Лабораторная работа №6