

# **Отчёт по лабораторной работе №6**

**Администрирование сетевых подсистем**

Ищенко Ирина НПИбд-02-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>18</b>
<b>4</b>	<b>Контрольные вопросы</b>	<b>19</b>

# Список иллюстраций

2.1	Установка пакетов . . . . .	6
2.2	/etc/my.cnf . . . . .	7
2.3	Запуск ПО . . . . .	7
2.4	Прослушивание порта . . . . .	7
2.5	Скрипт конфигурирования безопасности . . . . .	8
2.6	Скрипт конфигурирования безопасности . . . . .	8
2.7	Отображение БД . . . . .	9
2.8	Статус . . . . .	9
2.9	Отображение БД . . . . .	12
2.10	Статус . . . . .	12
2.11	Создание и заполнение таблицы . . . . .	13
2.12	Вывод таблицы . . . . .	13
2.13	Создание пользователя, права доступа . . . . .	14
2.14	Список БД . . . . .	14
2.15	Список таблиц БД . . . . .	15
2.16	Копии БД . . . . .	15
2.17	Копия БД с датой . . . . .	15
2.18	Восстановление БД . . . . .	15
2.19	Настройки внутреннего окружения . . . . .	16
2.20	mysql.sh . . . . .	16
2.21	Vagrantfile . . . . .	17

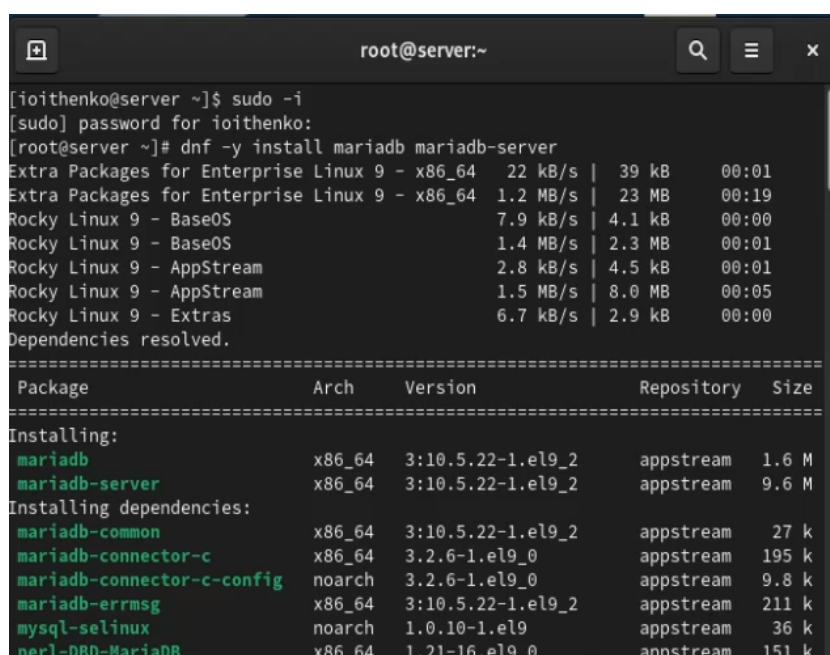
## **Список таблиц**

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Установим необходимые для работы с базами данных пакеты (рис. 2.1).



```
root@server:~  
[ioithenko@server ~]$ sudo -i  
[sudo] password for ioithenko:  
[root@server ~]# dnf -y install mariadb mariadb-server  
Extra Packages for Enterprise Linux 9 - x86_64 22 kB/s | 39 kB 00:01  
Extra Packages for Enterprise Linux 9 - x86_64 1.2 MB/s | 23 MB 00:19  
Rocky Linux 9 - BaseOS 7.9 kB/s | 4.1 kB 00:00  
Rocky Linux 9 - BaseOS 1.4 MB/s | 2.3 MB 00:01  
Rocky Linux 9 - AppStream 2.8 kB/s | 4.5 kB 00:01  
Rocky Linux 9 - AppStream 1.5 MB/s | 8.0 MB 00:05  
Rocky Linux 9 - Extras 6.7 kB/s | 2.9 kB 00:00  
Dependencies resolved.  
=====
```

Package	Arch	Version	Repository	Size
Installing:				
mariadb	x86_64	3:10.5.22-1.el9_2	appstream	1.6 M
mariadb-server	x86_64	3:10.5.22-1.el9_2	appstream	9.6 M
Installing dependencies:				
mariadb-common	x86_64	3:10.5.22-1.el9_2	appstream	27 k
mariadb-connector-c	x86_64	3.2.6-1.el9_0	appstream	195 k
mariadb-connector-c-config	noarch	3.2.6-1.el9_0	appstream	9.8 k
mariadb-errmsg	x86_64	3:10.5.22-1.el9_2	appstream	211 k
mysql-selinux	noarch	1.0.10-1.el9	appstream	36 k
perl-DBD-MariaDB	x86_64	1.21-16.el9_0	appstream	151 k

Рис. 2.1: Установка пакетов

Просмотрим конфигурационные файлы mariadb в каталоге /etc/my.cnf.d и в файле /etc/my.cnf (рис. 2.2).

```
[root@server ~]# ls /etc/my.cnf.d
auth_gssapi.cnf  enable_encryption.preset  mysql-clients.cnf
client.cnf      mariadb-server.cnf       spider.cnf
[root@server ~]# ls /etc/my.cnf
/etc/my.cnf
[root@server ~]# cat /etc/my.cnf
#
# This group is read both 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 ~]#
```

Рис. 2.2: /etc/my.cnf

[client-server] - секция, которая обозначает, что следующие параметры будут применяться как к клиенту, так и к серверу. Использование квадратных скобок указывает на начало нового раздела в конфигурационном файле. !includedir /etc/my.cnf.d - Здесь указывается конкретный путь к директории, из которой будут включены файлы конфигурации. /etc/my.cnf.d — это стандартное место для хранения дополнительных конфигурационных файлов.

Для запуска и включения программного обеспечения mariadb используем (рис. 2.3).

```
[root@server ~]# systemctl start mariadb
[root@server ~]# systemctl enable mariadb
```

Рис. 2.3: Запуск ПО

Убедимся, что mariadb прослушивает порт (рис. 2.4).

```
/etc/passwd:root:x:0:0:root:/bin:/usr/sbin/passwd
[root@server ~]# ss -tulpen | grep 3306
tcp LISTEN 0      80          *:3306      *:~         users:((("mariadb",pid=10047,fd=19)) uid:27 ino:43042 sk:14 cgroup:/system.slice/mariadb.servic
e v6only:0 <->
```

Рис. 2.4: Прослушивание порта

Запустим скрипт конфигурации безопасности mariadb, используя: mysql\_secure\_installation. Установим пароль для пользователя root базы данных, отключим удалённый кор-

невой доступ и удалим тестовую базу данных и любых анонимных пользователей (рис. 2.5) и (рис. 2.6).

```
[root@server ~]# 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'.
```

Рис. 2.5: Скрипт конфигурирования безопасности

```
root@server:~
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!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n]
```

Рис. 2.6: Скрипт конфигурирования безопасности

Войдем в базу данных с правами администратора базы данных. Просмотрим список команд MySQL, введя \h. Из приглашения интерактивной оболочки



MariaDB для отображения доступных в настоящее время баз данных введем MySQL-запрос (рис. 2.7). Базы данных в системе: information\_schema, mysql, performance\_schema. Выйдем из интерфейса интерактивной оболочки MariaDB.

```
For server side help, type 'help contents'

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.002 sec)

MariaDB [(none)]> exit;
Bye
```

Рис. 2.7: Отображение БД

Войдем в базу данных с правами администратора. Просмотрим статус MariaDB (рис. 2.8).

```
root@server:~
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> status;
-----
mysql Ver 15.1 Distrib 10.5.22-MariaDB, for Linux (x86_64) using EditLine wrapper

Connection id:          11
Current database:
Current user:            root@localhost
SSL:                    Not in use
Current pager:           stdout
Using outfile:           ''
Using delimiter:         ;
Server:                  MariaDB
Server version:          10.5.22-MariaDB MariaDB Server
Protocol version:        10
Connection:              Localhost via UNIX socket
Server characterset:     latin1
Db characterset:         latin1
Client characterset:     utf8
Conn. characterset:      utf8
UNIX socket:             /var/lib/mysql/mysql.sock
Uptime:                  11 min 49 sec
```

Рис. 2.8: Статус

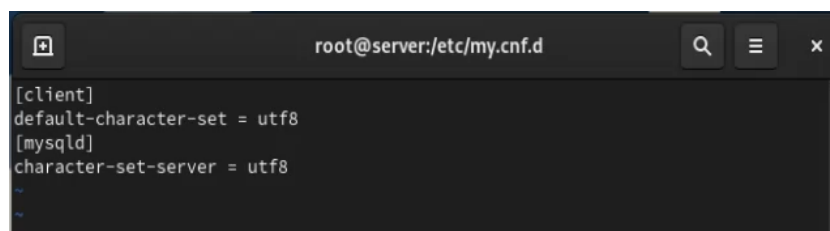
1. mysql Ver 15.1 Distrib 10.5.22-MariaDB, for Linux (x86\_64) using EditLine wrapper Эта строка сообщает о версии клиента MySQL, который использу-

ется для подключения к базе данных. В данном случае это версия 10.5.22 MariaDB, работающая на 64-разрядной версии Linux с использованием обертки EditLine для обработки ввода.

2. Connection id: 11 Здесь указывается идентификатор соединения. Уникальный номер, который помогает отслеживать активные соединения с сервером.
3. Current database: Эта строка показывает, что в данный момент не выбрана ни одна база данных. Это может означать, что нужно выполнить команду USE , чтобы выбрать базу данных для работы.
4. Current user: root@localhost Указывает на текущего пользователя, подключенного к серверу. В данном случае это пользователь root, который подключается с локального хоста.
5. ssl: Not in use Эта строка говорит о том, что соединение не защищено с помощью SSL (Secure Sockets Layer). Это может быть важным аспектом безопасности, особенно при удаленных подключениях.
6. Current pager: stdout Указывает, что вывод команд отображается в стандартный поток (stdout). Это означает, что результаты запросов будут выводиться непосредственно в терминал.
7. Using outfile: " Здесь указано, что не задан файл для вывода результатов запросов. Если бы он был задан, то результаты могли бы быть записаны в файл.
8. Using delimiter: ; Эта строка показывает, что в качестве разделителя команд используется точка с запятой (;). Это стандартный разделитель для SQL-запросов.
9. Server: MariaDB Указывает на то, что сервер базы данных — это MariaDB, которая является ответвлением MySQL.

10. `Server version: 10.5.22-MariaDB MariaDB ServerProtocol version: 10` Здесь указана версия сервера MariaDB и версия протокола сервера, используемая для связи с клиентом.
11. `Connection: Localhost via UNIX socket` Эта строка говорит о том, что соединение осуществляется локально через UNIX-сокеты, что обычно быстрее и безопаснее, чем использование TCP/IP.
12. `Server characterset: latin1` Указывает на набор символов (charset), используемый сервером — в данном случае это latin1. Это важно для корректного отображения и хранения текстовой информации.
13. `Db characterset: latin1` Эта строка также указывает на набор символов, используемый для текущей базы данных.
14. `Client characterset: utf8` Указывает на набор символов, используемый клиентом для общения с сервером — в данном случае это utf8, который поддерживает большее количество символов и языков.
15. `Conn. characterset: utf8` Эта строка подтверждает, что набор символов, используемый для текущего соединения, также utf8. Это важно для корректной передачи данных между клиентом и сервером.
16. `UNIX socket: /var/Lib/mysql/mysql.sock` Здесь указан путь к файлу сокета UNIX, который используется для соединения с сервером базы данных. Это стандартное место для хранения сокетов в некоторых дистрибутивах Linux.

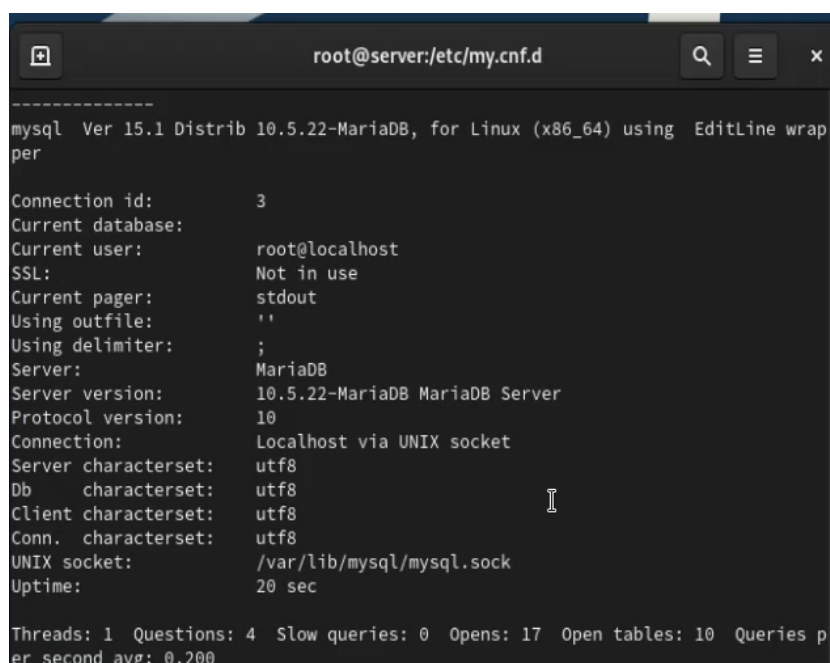
В каталоге `/etc/my.cnf.d` создадим файл `utf8.cnf`. Заполним конфигурацию (рис. 2.9):

A terminal window titled 'root@server:/etc/my.cnf.d' with search, menu, and close icons. It displays the following configuration:

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

Рис. 2.9: Отображение БД

Перезапустим MariaDB. Войдем в базу данных с правами администратора и посмотрим статус MariaDB (рис. 2.10). Были изменены наборы символов, используемых сервером и текущей БД на utf8.

A terminal window titled 'root@server:/etc/my.cnf.d' with search, menu, and close icons. It displays the status of MariaDB:

```
mysql Ver 15.1 Distrib 10.5.22-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:          10.5.22-MariaDB MariaDB Server
Protocol version:        10
Connection:              Localhost via UNIX socket
Server characterset:     utf8
Db characterset:         utf8
Client characterset:     utf8
Conn. characterset:      utf8
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
```

Рис. 2.10: Статус

Войдем в базу данных с правами администратора. Создадим базу данных с именем addressbook. Перейдем к базе данных addressbook. Отобразим имеющиеся в базе данных addressbook таблицы. Создадим таблицу city с полями name и city. Заполним несколько строк таблицы некоторыми данными (рис. 2.11).

```
root@server:/etc/my.cnf.d
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]> USE addressbook;
Database changed
MariaDB [addressbook]> SHOW TABLES;
Empty set (0.001 sec)

MariaDB [addressbook]> CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
Query OK, 0 rows affected (0.007 sec)

MariaDB [addressbook]> INSERT INTO city(name, city) VALUES('Иванов', 'Москва');
Query OK, 1 row affected (0.008 sec)

MariaDB [addressbook]> INSERT INTO city(name, city) VALUES('Петров', 'Сочи');
Query OK, 1 row affected (0.003 sec)

MariaDB [addressbook]> INSERT INTO city(name, city) VALUES('Сидоров', 'Дубна');
Query OK, 1 row affected (0.003 sec)

MariaDB [addressbook]>
```

Рис. 2.11: Создание и заполнение таблицы

Сделаем MySQL-запрос: `SELECT * FROM city`; Выводятся все данные, содержащиеся в таблице `city` (рис. 2.12).

```
MariaDB [addressbook]> SELECT * FROM city;
+-----+-----+
| name   | city   |
+-----+-----+
| Иванов | Москва |
| Петров | Сочи   |
| Сидоров | Дубна  |
+-----+-----+
3 rows in set (0.001 sec)
```

Рис. 2.12: Вывод таблицы

Создадим пользователя для работы с базой данных `addressbook` и зададим для него пароль. Предоставим права доступа созданному пользователю на действия с базой данных `addressbook` (просмотр, добавление, обновление, удаление данных) Обновим привилегии (права доступа) базы данных `addressbook`. Посмотрим общую информацию о таблице `city` базы данных `addressbook` (рис. 2.13). Выйдем из окружения.

```
root@server:/etc/my.cnf.d
MariaDB [addressbook]> CREATE USER ioithenko@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.005 sec)

MariaDB [addressbook]> GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO ioi
thenko@'%';
Query OK, 0 rows affected (0.003 sec)

MariaDB [addressbook]> FLUSH PRIVILEGES;
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 'PRI
VILIGES' at line 1
MariaDB [addressbook]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 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]>
```

Рис. 2.13: Создание пользователя, права доступа

Посмотрим список баз данных (рис. 2.14).

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

Рис. 2.14: Список БД

Посмотрим список таблиц базы данных addressbook (рис. 2.15).

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

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

Рис. 2.15: Список таблиц БД

Создадим каталог для резервных копий /var/backup. Сделаем резервную копию базы данных addressbook, сжатую резервную копию базы данных addressbook (рис. 2.16), сжатую резервную копию базы данных addressbook с указанием даты создания копии (рис. 2.18):

```
[root@server my.cnf.d]# mkdir -p /var/backup
[root@server my.cnf.d]# mysqldump -u root -p addressbook > /var/backup/addressbook.sql
Enter password:
[root@server my.cnf.d]# mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz
Enter password:
```

Рис. 2.16: Копии БД

```
[root@server my.cnf.d]# mysqldump -u root -p addressbook | gzip > $(date +%Y%m%d.%H%M%S).sql.gz
Enter password:
[root@server my.cnf.d]# ls /var/backup
addressbook.20241007.163359.sql.gz  addressbook.sql  addressbook.sql.gz
[root@server my.cnf.d]#
```

Рис. 2.17: Копия БД с датой

Восстановим базу данных addressbook из сжатой резервной копии (рис. 2.18).

```
Enter password:
[root@server my.cnf.d]# zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook
Enter password:
[root@server my.cnf.d]#
```

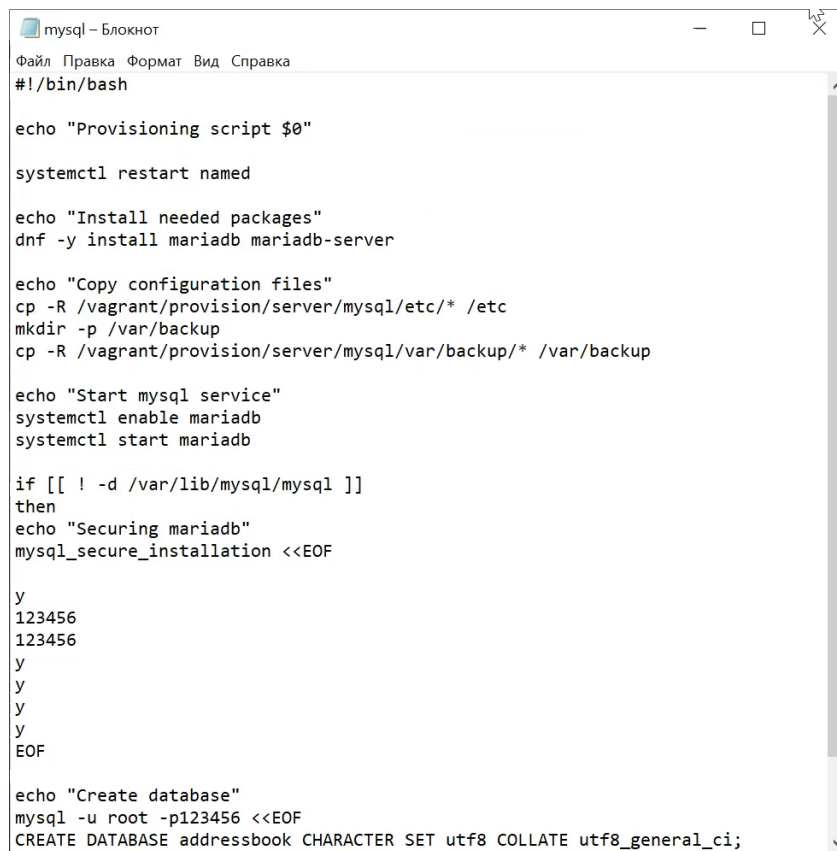
Рис. 2.18: Восстановление БД

Внесем изменения в настройки внутреннего окружения ВМ (рис. 2.19).

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

Рис. 2.19: Настройки внутреннего окружения

Добавим скрипт mysql.sh (рис. 2.20).



```
mysql - Блокнот
Файл Правка Формат Вид Справка
#!/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

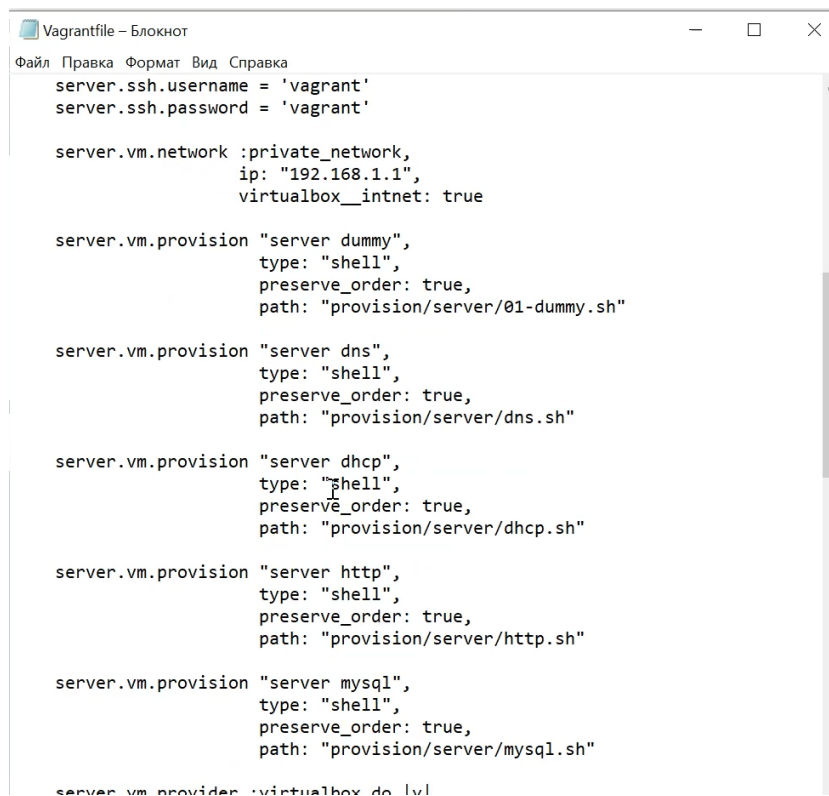
if [[ ! -d /var/lib/mysql/mysql ]]
then
echo "Securing mariadb"
mysql_secure_installation <<EOF
Y
123456
123456
Y
Y
Y
Y
EOF

echo "Create database"
mysql -u root -p123456 <<EOF
CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Рис. 2.20: mysql.sh

Внесем изменения в Vagrantfile (рис. 2.21).



A screenshot of a Notepad window titled "Vagrantfile - Блокнот". The window contains a Vagrantfile with the following configuration: 

```
server.ssh.username = 'vagrant'
server.ssh.password = 'vagrant'

server.vm.network :private_network,
  ip: "192.168.1.1",
  virtualbox____intnet: true

server.vm.provision "server dummy",
  type: "shell",
  preserve_order: true,
  path: "provision/server/01-dummy.sh"

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

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

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

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

server.vm.provider :virtualbox do |v|
```

Рис. 2.21: Vagrantfile

## **3 Выводы**

В ходе лабораторной работы я приобрела практических навыков по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

## 4 Контрольные вопросы

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?

```
SHOW DATABASES;
```

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

```
SHOW TABLES;
```

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

```
DESCRIBE table_name;
```

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

```
SELECT * FROM table_name;
```

7. Как удалить запись из таблицы?

`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`.