

# **Отчёт по лабораторной работе №6**

**Дисциплина: Администрирование сетевых подсистем**

Ибрахим Мухсейн Алькамаль

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Выполнение лабораторной работы</b>	<b>6</b>
2.1 Установка MariaDB . . . . .	6
2.2 Конфигурация кодировки символов . . . . .	10
2.3 Создание базы данных . . . . .	12
2.4 Резервные копии . . . . .	15
2.5 Внесение изменений в настройки внутреннего окружения виртуальной машины . . . . .	16
<b>3 Выводы</b>	<b>19</b>
<b>4 Ответы на контрольные вопросы</b>	<b>20</b>

# Список иллюстраций

2.1	Установка пакетов mariadb и mariadb-server с помощью dnf . . . . .	6
2.2	Запуск и включение службы mariadb через systemctl . . . . .	7
2.3	Проверка прослушивания порта 3306 процессом mariadb . . . . .	7
2.4	Выполнение mysql_secure_installation и настройка параметров безопасности . . . . .	8
2.5	Выполнение mysql_secure_installation и настройка параметров безопасности . . . . .	8
2.6	Вход в MariaDB под пользователем root . . . . .	9
2.7	Список клиентских команд MariaDB . . . . .	9
2.8	Отображение списка системных баз данных (SHOW DATABASES) . .	10
2.9	Вход в MariaDB и выполнение команды status (до изменения кодировки) . . . . .	10
2.10	Создание файла utf8.cnf в каталоге /etc/my.cnf.d . . . . .	11
2.11	Содержимое файла utf8.cnf с параметрами кодировки UTF-8 . . . .	11
2.12	Проверка статуса MariaDB после изменения кодировки . . . . .	12
2.13	Вход в MariaDB под пользователем root . . . . .	13
2.14	Создание базы данных addressbook и проверка списка таблиц . .	13
2.15	Создание таблицы city, вставка данных и результат SELECT * FROM city . . . . .	14
2.16	Создание пользователя, назначение прав и описание структуры таблицы city . . . . .	14
2.17	Просмотр баз данных и таблиц addressbook с помощью mysqlshow . .	15
2.18	резервного копирования . . . . .	16
2.19	Создание каталогов mysql и копирование конфигурации и резервных копий . . . . .	16
2.20	Содержимое скрипта mysql.sh для автоматической настройки MariaDB	17
2.21	Добавление конфигурации server mysql в Vagrantfile . . . . .	18

# **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

### 2.1 Установка MariaDB

На виртуальной машине `server` выполнен переход в режим суперпользователя и произведена установка пакетов `mariadb` и `mariadb-server` с использованием менеджера пакетов `dnf` (рис. 2.1). В процессе установки были автоматически подключены зависимости и выполнена загрузка необходимых компонентов.

```
[alkamal@server.alkamal.net ~]$ sudo -i
[sudo] password for alkamal:
[root@server.alkamal.net ~]# dnf -y install mariadb mariadb-server
Last metadata expiration check: 2:17:21 ago on Sun 08 Feb 2026 01:48:25 PM UTC.
Dependencies resolved.
=====
 Package           Architecture Version      Repository  Size
=====
Installing:
 mariadb           x86_64       3:10.5.29-3.el9_7    appstream   1.6 M
 mariadb-server    x86_64       3:10.5.29-3.el9_7    appstream   9.7 M
Installing dependencies:
 mariadb-common    x86_64       3:10.5.29-3.el9_7    appstream   27 k
 mariadb-errmsg    x86_64       3:10.5.29-3.el9_7    appstream   210 k
 mysql-selinux     noarch      1.0.14-1.el9_6      appstream   36 k
 perl-DBD-MariaDB x86_64       1.21-17.el9      appstream   149 k
 perl-Sys-Hostname x86_64       1.23-481.1.el9_6  appstream   15 k
Installing weak dependencies:
 mariadb-backup    x86_64       3:10.5.29-3.el9_7    appstream   6.5 M
 mariadb-gssapi-server x86_64     3:10.5.29-3.el9_7    appstream   14 k
 mariadb-server-utils x86_64     3:10.5.29-3.el9_7    appstream   210 k
=====
Transaction Summary
=====
Install 10 Packages

Total download size: 19 M
Installed size: 113 M
Downloading Packages:
(1/10): mariadb-common-10.5.29-3.el9_7.x86_64.rpm      101 kB/s | 27 kB   00:00
(2/10): mariadb-errmsg-10.5.29-3.el9_7.x86_64.rpm      350 kB/s | 210 kB   00:00
(3/10): mariadb-gssapi-server-10.5.29-3.el9_7.x86_64.rpm 55 kB/s | 14 kB   00:00
(4/10): mariadb-10.5.29-3.el9_7.x86_64.rpm            564 kB/s | 1.6 MB   00:02
(5/10): mariadb-server-utils-10.5.29-3.el9_7.x86_64.rpm 701 kB/s | 210 kB   00:00
(6/10): mysql-selinux-1.0.14-1.el9_6.noarch.rpm        226 kB/s | 36 kB   00:00
(7-8/10): mariadb-backup- 32% [=====] 1.2 MB/s | 6.1 MB   00:10 ETA
```

Рисунок 2.1: Установка пакетов `mariadb` и `mariadb-server` с помощью `dnf`

После установки выполнен запуск службы mariadb и добавление её в автозагрузку с помощью `systemctl start mariadb` и `systemctl enable mariadb` (рис. 2.3). Созданы символические ссылки в каталогах `systemd`, что подтверждает успешную регистрацию службы.

```
[root@server.alkamal.net ~]# systemctl start mariadb
[root@server.alkamal.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.alkamal.net ~]# ss -tulpen | grep mysql
[root@server.alkamal.net ~]# ss -tulpen | grep 3306
tcp   LISTEN  0      80          *:3306           *:*    users:(("mariadb",pid=1
4932,fd=19))
              uid:27  ino:51822 sk:12 cgroup:/system.slice/mariadb.service v6only:0 <->
```

Рисунок 2.2: Запуск и включение службы mariadb через systemctl

Далее проверено, что процесс `mariadb` прослушивает порт 3306, с использованием команды `ss -tulpen | grep 3306` (рис. 2.3). В выводе отображён процесс `mariadb`, находящийся в состоянии LISTEN на порту 3306, что подтверждает корректный запуск сервера базы данных.

```
[root@server.alkamal.net ~]# systemctl start mariadb
[root@server.alkamal.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.alkamal.net ~]# ss -tulpen | grep mysql
[root@server.alkamal.net ~]# ss -tulpen | grep 3306
tcp   LISTEN  0      80          *:3306           *:*    users:(("mariadb",pid=1
4932,fd=19))
              uid:27  ino:51822 sk:12 cgroup:/system.slice/mariadb.service v6only:0 <->
```

Рисунок 2.3: Проверка прослушивания порта 3306 процессом mariadb

Выполнен скрипт первичной настройки безопасности `mysql_secure_installation`. В ходе диалога включена аутентификация через `unix_socket`, установлен пароль для пользователя `root` СУБД, удалены анонимные пользователи (рис. 2.4), запрещён удалённый доступ для `root` и удалена тестовая база данных. После применения изменений произведена перезагрузка таблиц привилегий. (рис. 2.5)

```
[root@server.alkamal.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] 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.
```

Рисунок 2.4: Выполнение mysql\_secure\_installation и настройка параметров безопасности

```
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] y
  ... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Рисунок 2.5: Выполнение mysql\_secure\_installation и настройка параметров безопасности

После завершения настройки выполнен вход в систему управления базами данных от имени администратора с помощью mysql -u root -p (рис. 2.6).

Отображена информация о версии сервера MariaDB и параметрах подключения.

```
[root@server.alkamal.net ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.5.29-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.
```

Рисунок 2.6: Вход в MariaDB под пользователем root

В интерактивной оболочке MariaDB выполнена команда для просмотра списка доступных клиентских команд (рис. 2.7). В выводе представлен перечень управляющих команд консольного клиента.

```
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.
notee     (\t) Don't write into outfile.
nowarning (\w) Don't show warnings after every statement.
pager     (\P) Set PAGER [to_page]. 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'
```

Рисунок 2.7: Список клиентских команд MariaDB

С использованием SQL-запроса `SHOW DATABASES;` получен список доступных системных баз данных (рис. 2.8). В системе присутствуют базы данных: `information_schema`, `mysql`, `performance_schema`.

```

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

MariaDB [(none)]> exit
Bye
[root@server.alkamal.net ~]# 

```

Рисунок 2.8: Отображение списка системных баз данных (SHOW DATABASES)

## 2.2 Конфигурация кодировки символов

Выполнен вход в MariaDB от имени администратора базы данных с использованием команды `mysql -u root -p`. После аутентификации выведена информация о версии сервера и установленном соединении (рис. 2.9).

```

[root@server.alkamal.net ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.5.29-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 10.5.29-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:         10.5.29-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:                 9 min 28 sec

Threads: 1  Questions: 27  Slow queries: 0  Opens: 20  Open tables: 13  Queries per second avg: 0.047
-----

MariaDB [(none)]> exit
Bye

```

Рисунок 2.9: Вход в MariaDB и выполнение команды `status` (до изменения кодировки)

Из интерактивной оболочки выполнена команда `status`, которая отобразила

текущее состояние сервера (рис. 2.9). В выводе указаны: идентификатор соединения (Connection id), текущий пользователь (root@localhost), способ подключения (Localhost via UNIX socket), версия сервера (10.5.29-MariaDB), а также параметры кодировки. До изменения конфигурации значения Server characterset и Db characterset установлены в latin1, а Client characterset и Conn. characterset – в utf8.

Далее в каталоге /etc/my.cnf.d создан файл utf8.cnf и открыт для редактирования (рис. 2.10).

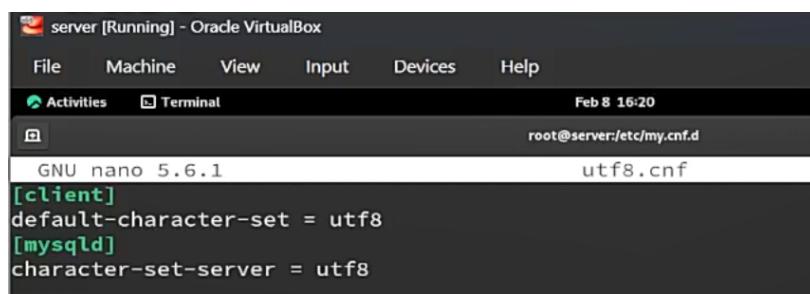
```
[root@server.alkamal.net ~]# cd /etc/my.cnf.d
[root@server.alkamal.net my.cnf.d]# touch utf8.cnf
[root@server.alkamal.net my.cnf.d]# nano utf8.cnf
```

Рисунок 2.10: Создание файла utf8.cnf в каталоге /etc/my.cnf.d

В файл добавлена конфигурация:

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

Данные параметры задают кодировку utf8 по умолчанию для клиентских подключений и для сервера MariaDB (рис. 2.11).



The screenshot shows a terminal window titled "server [Running] - Oracle VirtualBox". The window has a dark theme with white text. At the top, there's a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". Below the menu is a toolbar with "Activities" and "Terminal" icons. The status bar at the bottom right shows "root@server:/etc/my.cnf.d" and the date and time "Feb 8 16:20". The main area of the terminal shows the command "GNU nano 5.6.1" followed by the file content:

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

Рисунок 2.11: Содержимое файла utf8.cnf с параметрами кодировки UTF-8

После внесения изменений выполнен перезапуск службы MariaDB командой

`systemctl restart mariadb`, затем произведён повторный вход в систему и выполнена команда `status` (рис. 2.12).

```
[root@server.alkamal.net my.cnf.d]# systemctl restart mariadb
[root@server.alkamal.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: 10.5.29-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 10.5.29-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:         10.5.29-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:                 1 min 4 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second
avg: 0.062
-----
```

Рисунок 2.12: Проверка статуса MariaDB после изменения кодировки

В результате изменения конфигурации параметры `Server characterset` и `Db characterset` изменились с `latin1` на `utf8`. Параметры `Client characterset` и `Conn. characterset` также установлены в `utf8`. Это подтверждает корректное применение новой конфигурации кодировки после перезапуска сервера.

## 2.3 Создание базы данных

Выполнен вход в MariaDB под пользователем `root` с использованием команды `mysql -u root -p` (рис. 2.13). После аутентификации отображена информация о версии сервера и установленном соединении.

```
[root@server.alkamal.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 4
Server version: 10.5.29-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)]> █
```

Рисунок 2.13: Вход в MariaDB под пользователем root

Создана база данных addressbook с кодировкой utf8 и сопоставлением utf8\_general\_ci, затем выполнен переход к ней и проверено отсутствие таблиц командой SHOW TABLES ; (рис. 2.14). База данных успешно создана, на момент проверки таблицы отсутствуют.

Создана таблица city с полями name и city типа VARCHAR(40). В таблицу добавлены записи: Иванов – Москва, Петров – Сочи, Сидоров – Дубна.

```
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 (0.013 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 (0.009 sec)

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

Рисунок 2.14: Создание базы данных addressbook и проверка списка таблиц

Выполнен запрос SELECT \* FROM city ; (рис. 2.15). Результат запроса отображает три строки, соответствующие введённым данным, что подтверждает корректность создания таблицы и вставки записей.

```

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

MariaDB [addressbook]>

```

Рисунок 2.15: Создание таблицы city, вставка данных и результат SELECT \* FROM city

Создан пользователь alkamal@'%' с паролем и предоставлены права SELECT, INSERT, UPDATE, DELETE на базу данных addressbook. После выполнения FLUSH PRIVILEGES ; изменения прав вступили в силу. Командой DESCRIBE city; получена структура таблицы (рис. 2.16). Таблица содержит два поля: name и city, оба типа VARCHAR(40), допускающие значение NULL.

```

MariaDB [addressbook]> CREATE USER alkamal@'%' IDENTIFIED BY '654321';
Query OK, 0 rows affected (0.009 sec)

MariaDB [addressbook]> GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO alkamal@'%';
Query OK, 0 rows affected (0.010 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]> quit
Bye

```

Рисунок 2.16: Создание пользователя, назначение прав и описание структуры таблицы city

После выхода из MariaDB выполнена проверка списка баз данных с помощью mysqlshow -u root -p, а также просмотр таблиц базы addressbook под пользователями root и alkamal (рис. 2.17). В списке присутствует база addressbook, а в её составе отображается таблица city, что подтверждает корректность создания базы данных и настройки прав доступа.

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

Рисунок 2.17: Просмотр баз данных и таблиц addressbook с помощью mysqlshow

## 2.4 Резервные копии

На виртуальной машине `server` создан каталог для хранения резервных копий с использованием команды `mkdir -p /var/backup` (рис. 2.18). Каталог предназначен для размещения дампов базы данных.

Выполнено создание полной резервной копии базы данных `addressbook` с помощью команды `mysqldump -u root -p addressbook > /var/backup/addressbook.sql` (рис. 2.18). В результате сформирован SQL-файл, содержащий структуру таблиц и данные базы.

Создана сжатая резервная копия базы данных с использованием конвейера и утилиты `gzip`: `mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz` (рис. 2.18). В результате получен архивированный файл дампа.

Дополнительно выполнено создание сжатой резервной копии с указанием даты и времени создания файла: `mysqldump -u root -p addressbook | gzip > $(date +/var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz)` (рис. 2.18). Имя файла автоматически формируется с добавлением временной метки.

Произведено восстановление базы данных из обычной резервной копии командой `mysql -u root -p addressbook < /var/backup/addressbook.sql` (рис. 2.18). Данные базы импортированы из SQL-файла.

Выполнено восстановление базы данных из сжатой резервной копии с использованием `zcat`: `zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook` (рис. 2.18). Архив предварительно распакован в поток вывода и передан в клиент `mysql`, что подтверждает корректность процедуры восстановления.

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

Рисунок 2.18: резервного копирования

## 2.5 Внесение изменений в настройки внутреннего окружения виртуальной машины

На виртуальной машине `server` выполнен переход в каталог `/vagrant/provision/server`. Создан каталог `mysql` со структурой подкаталогов `etc/my.cnf.d` и `var/backup`. В созданные каталоги скопированы файл конфигурации `utf8.cnf` и резервные копии базы данных `addressbook` (рис. 2.19).

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

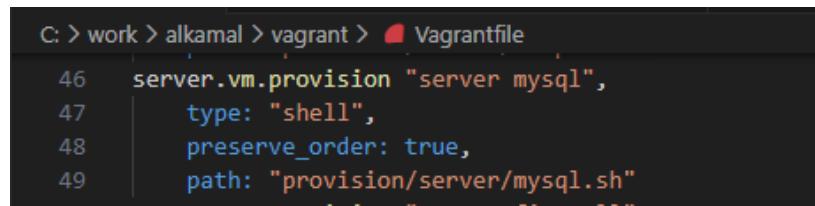
Рисунок 2.19: Создание каталогов mysql и копирование конфигурации и резервных копий

В каталоге `/vagrant/provision/server` создан исполняемый файл `mysql.sh`, которому назначены права на выполнение. В файл добавлен скрипт автоматической установки и настройки MariaDB (рис. 2.20). Скрипт выполняет установку пакетов `mariadb` и `mariadb-server`, копирование конфигурационных файлов, создание каталога `/var/backup`, запуск и включение службы `mariadb`, выполнение `mysql_secure_installation` в неинтерактивном режиме, создание базы данных `addressbook` и восстановление данных из резервной копии.

```
GNU nano 5.6.1                                     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
I
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
```

Рисунок 2.20: Содержимое скрипта `mysql.sh` для автоматической настройки MariaDB

В конфигурационный файл `Vagrantfile` добавлена секция провижининга для сервера, обеспечивающая автоматическое выполнение скрипта `mysql.sh` при запуске виртуальной машины (рис. 2.21). Запись использует тип `shell`, параметр `preserve_order: true` и путь к скрипту `provision/server/mysql.sh`.



```
C: > work > alkamal > vagrant > Vagrantfile
46   server.vm.provision "server mysql",
47     type: "shell",
48     preserve_order: true,
49     path: "provision/server/mysql.sh"
```

Рисунок 2.21: Добавление конфигурации `server mysql` в `Vagrantfile`

## 3 Выводы

В ходе работы выполнена установка и базовая настройка сервера баз данных MariaDB на виртуальной машине `server`. Проверена корректность запуска службы и прослушивания стандартного порта 3306.

Произведена настройка параметров безопасности с использованием `mysql_secure_installation`. Установлен пароль пользователя `root` СУБД, удалены анонимные пользователи, запрещён удалённый доступ администратора и удалена тестовая база данных.

Настроена кодировка `utf8` на уровне клиента и сервера путём изменения конфигурационных файлов и перезапуска службы. Корректность применения параметров подтверждена через команду `status`.

Создана база данных `addressbook`, таблица `city`, добавлены тестовые записи и проверена их выборка SQL-запросом. Настроен пользователь с ограниченными правами доступа к базе данных.

Освоены процедуры резервного копирования и восстановления базы данных с использованием `mysqldump`, включая создание сжатых архивов.

Автоматизирован процесс установки и настройки MariaDB посредством скрипта `mysql.sh`, подключённого к механизму `provisioning` в `Vagrantfile`, что обеспечивает воспроизводимость конфигурации при повторном развёртывании виртуальной машины.

## **4 Ответы на 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.