

Отчет по лабораторной работе № 6.
Установка и настройка системы управления базами
данных MariaDB

Данила Стариков
НПИбд-02-22

2024

Содержание

1	Цель работы	3
2	Выполнение работы	4
2.1	Установка MariaDB	4
2.2	Конфигурация кодировки символов	7
2.3	Создание базы данных	9
2.4	Резервные копии	13
2.5	Внесение изменений в настройки внутреннего окружения виртуальной машины	14
3	Выводы	15

1 Цель работы

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

2 Выполнение работы

2.1 Установка MariaDB

1. Загрузили операционную систему и перешли в рабочий каталог с проектом:

```
cd /var/tmp/dastarikov/vagrant
```

2. Запустили виртуальную машину `server`:

```
make server-up
```

3. На виртуальной машине `server` вошли под своим пользователем и открыли терминал. Перешли в режим суперпользователя:

```
sudo -i
```

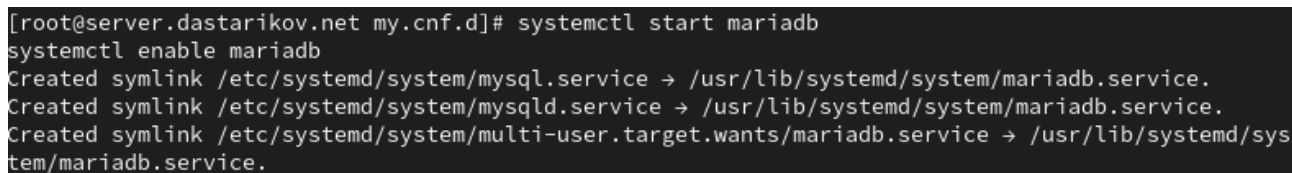
4. Установили необходимые для работы с базами данных пакеты:

```
dnf -y install mariadb mariadb-server
```

5. Просмотрели конфигурационные файлы `mariadb` в каталоге `/etc/my.cnf.d` и в файле `/etc/my.cnf`.

6. Для запуска и включения программного обеспечения `mariadb` использовали (Рис. 1):

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

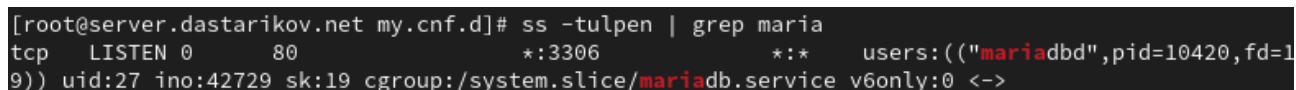


```
[root@server.dastarikov.net my.cnf.d]# systemctl start mariadb  
systemctl enable mariadb  
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.  
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
```

Рис. 1: Запуск ПО `mariadb`.

7. Убедились, что `mariadb` прослушивает порт (Рис. 2), используя

```
ss -tulpen | grep mysql
```



```
[root@server.dastarikov.net my.cnf.d]# ss -tulpen | grep maria  
tcp LISTEN 0      80          *:3306      *:*        users:(("mariadb",pid=10420,fd=19)) uid:27 ino:42729 sk:19 cgroup:/system.slice/mariadb.service v6only:0 <->
```

Рис. 2: Проверка прослушивания порта.

8. Запустили скрипт конфигурации безопасности `mariadb`, используя:

```
mysql_secure_installation
```

С помощью запустившегося диалога и путём выбора [Y/n] установили пароль для пользователя `root` базы данных, отключили удалённый корневой доступ и удалили тестовую базу данных и любых анонимных пользователей (Рис. 3).

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

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

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

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

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

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

Switch to unix_socket authentication [Y/n] 123456
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!
```

Рис. 3: Настройка конфигурации безопасности `mariadb`.

9. Для входа в базу данных с правами администратора базы данных ввели (Рис. 4)

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

```
[root@server.dastarikov.net my.cnf.d]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.5.22-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)]> █
```

Рис. 4: Вход в базу данных с правами администратора.

10. Просмотрели список команд MySQL, введя \h (Рис. 5).

```
Note that all text commands must be first on line and end with ';'
?          (?) Synonym for 'help'.
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.
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.
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.
charset    (C) Switch to another charset. Might be needed for processing binlog with multi-byte ch
arsets.
warnings   (W) Show warnings after every statement.
nowarning  (w) Don't show warnings after every statement.

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

Рис. 5: Просмотр списка команд MySQL.

11. Из приглашения интерактивной оболочки MariaDB для отображения доступных в настоящее время баз данных ввели MySQL-запрос (Рис. 6)т

```
SHOW DATABASES;
```

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.001 sec)
```

Рис. 6: Просмотр доступных баз данных.

Доступно три базы данных: `information_schema`, `mysql`, `performance_schema`.

12. Для выхода из интерфейса интерактивной оболочки MariaDB ввели (Рис. 7)
`exit;`

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

Рис. 7: Выход из интерактивной оболочки MariaDB.

2.2 Конфигурация кодировки символов

1. Вошли в базу данных с правами администратора:

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

2. Для отображения статуса MariaDB ввели из приглашения интерактивной оболочки MariaDB (Рис. 8):

```
status
```

```

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

Connection id:          15
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:                 9 min 9 sec

Threads: 1  Questions: 31  Slow queries: 0  Opens: 20  Open tables: 13  Queries per second avg: 0.056
-----

```

Рис. 8: Просмотр статуса MariaDB.

3. В каталоге `/etc/my.cnf.d` создали файл `utf8.cnf`:

```

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

```

Открыли его на редактирование и указали в нём следующую конфигурацию:

```

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

```

4. Перезапустили MariaDB:

```
systemctl restart mariadb
```

5. Вошли в базу данных с правами администратора и посмотрели статус MariaDB (Рис. 9).


```

MariaDB [(none)]> status
-----
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:                 11 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second avg: 0.363
-----

```

Рис. 9: Просмотр статуса MariaDB после изменения конфигурации.

Теперь в базах данных и в выводе запросов будет использоваться кодировка UTF-8.

2.3 Создание базы данных

1. Вошли в базу данных с правами администратора (Рис. 10):

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

2. Создали базу данных с именем addressbook (Рис. 10):

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

3. Перешли к базе данных addressbook (Рис. 10)

```
USE addressbook;
```

4. Отобразили имеющиеся в базе данных addressbook таблицы (Рис. 10):

```
SHOW TABLES;
```

```

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

MariaDB [(none)]> USE addressbook;
Database changed
MariaDB [addressbook]> SHOW TABLES;
Empty set (0.000 sec)

```

Рис. 10: Создание базы данных addressbook.

5. Создали таблицу `city` с полями `name` и `city` (Рис. 11):

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

6. Заполнили несколько строк таблицы некоторыми данными по аналогии в соответствии с синтаксисом MySQL (Рис. 11):

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

В частности, добавили в базу сведения о Петрове и Сидорове (Рис. 11):

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

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

```
MariaDB [addressbook]> CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
Query OK, 0 rows affected (0.030 sec)

MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Иванов','Москва');
Query OK, 1 row affected (0.006 sec)

MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Петров','Сочи');
Query OK, 1 row affected (0.005 sec)

MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Сидоров','Дубна');
Query OK, 1 row affected (0.005 sec)
```

Рис. 11: Создание и заполнение таблицы.

7. Сделали следующий MySQL-запрос (Рис. 12):

```
SELECT * FROM city;
```

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

Рис. 12: Просмотр вхождений таблицы.

Запрос вывел все вхождения таблицы, которые были добавлены.

8. Создали пользователя для работы с базой данных `addressbook` и задали для него пароль (Рис. 13):

```
CREATE USER dastarikov@'%' IDENTIFIED BY 'password';
```

9. Предоставили права доступа созданному пользователю `dastarikov` на действия с базой данных `addressbook` (просмотр, добавление, обновление, удаление данных) (Рис. 13):

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

Обновили привилегии (права доступа) базы данных `addressbook` (Рис. 13):

```
FLUSH PRIVILEGES;
```

```
MariaDB [addressbook]> CREATE USER dastarikov@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.005 sec)

MariaDB [addressbook]> GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO dastarikov@'%';
Query OK, 0 rows affected (0.005 sec)

MariaDB [addressbook]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)
```

Рис. 13: Создание нового пользователя для работы с таблицей.

10. Посмотрели общую информацию о таблице `city` базы данных `addressbook` (Рис. 14):

```
DESCRIBE city;
```

```
MariaDB [addressbook]> DESCRIBE city;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name  | varchar(40)   | YES  |     | NULL    |       |
| city  | varchar(40)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.002 sec)
```

Рис. 14: Общая информация о таблице.

11. Вышли из окружения MariaDB:

```
quit
```

12. Просмотрели список баз данных (Рис. 15):

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

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

Рис. 15: Просмотр списка баз данных.

13. Просмотрели список таблиц базы данных `addressbook` (Рис. 16):

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

или (Рис. 17)

```
mysqlshow -u dastarikov -p addressbook
```

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

Рис. 16: Просмотр таблиц базы данных `addressbook` пользователем `root`.

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

Рис. 17: Просмотр таблиц базы данных `addressbook` пользователем `dastarikov`.

2.4 Резервные копии

1. На виртуальной машине `server` создали каталог для резервных копий (Рис. 18):

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

2. Сделали резервную копию базы данных `addressbook` (Рис. 18):

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

3. Сделали сжатую резервную копию базы данных `addressbook` (Рис. 18):

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

4. Сделали сжатую резервную копию базы данных `addressbook` с указанием даты создания копии (Рис. 18):

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

5. Восстановили базу данных `addressbook` из резервной копии (Рис. 18):

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

6. Восстановили базу данных `addressbook` из сжатой резервной копии (Рис. 18):

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

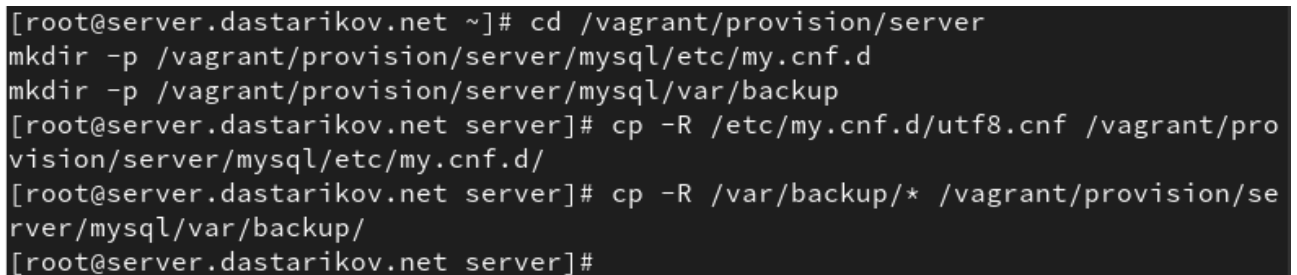
```
[root@server.dastarikov.net ~]# mkdir -p /var/backup  
[root@server.dastarikov.net ~]# mysqldump -u root -p addressbook > /var/backup/  
/addressbook.sql  
Enter password:  
[root@server.dastarikov.net ~]# mysqldump -u root -p addressbook | gzip > /var/  
/backup/addressbook.sql.gz  
Enter password:  
[root@server.dastarikov.net ~]# ^[[200~mysqldump -u root -p addressbook ~^C  
[root@server.dastarikov.net ~]# mysqldump -u root -p addressbook | gzip > $(da  
te +/var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz)  
Enter password:  
[root@server.dastarikov.net ~]# mysql -u root -p addressbook < /var/backup/add  
ressbook.sql  
Enter password:  
[root@server.dastarikov.net ~]# zcat /var/backup/addressbook.sql.gz | mysql -u  
root -p addressbook  
Enter password:  
[root@server.dastarikov.net ~]#
```

Рис. 18: Создание и восстановление резервной копии базы данных.

2.5 Внесение изменений в настройки внутреннего окружения виртуальной машины

1. 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. 19):

```
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.dastarikov.net ~]# cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d
mkdir -p /vagrant/provision/server/mysql/var/backup
[root@server.dastarikov.net server]# cp -R /etc/my.cnf.d/utf8.cnf /vagrant/pro
vision/server/mysql/etc/my.cnf.d/
[root@server.dastarikov.net server]# cp -R /var/backup/* /vagrant/provision/se
rver/mysql/var/backup/
[root@server.dastarikov.net server]#
```

Рис. 19: Создание каталога для настроек внутреннего окружения.

2. В каталоге `/vagrant/provision/server` создали исполняемый файл `mysql.sh`:

```
cd /vagrant/provision/server
touch mysql.sh
chmod +x mysql.sh
```

Открыв его на редактирование, прописали в нём следующий скрипт:

```
#!/bin/bash
echo "Provisioning script $0"
systemctl restart named
echo "Install needed packages"
dnf -y install mariadb mariadb-server
echo "Copy configuration files"
cp -R /vagrant/provision/server/mysql/etc/* /etc
mkdir -p /var/backup
cp -R /vagrant/provision/server/mysql/var/backup/* /var/backup
echo "Start mysql service"
systemctl enable mariadb
systemctl start mariadb
if [[ ! -d /var/lib/mysql/mysql ]]
then
echo "Securing mariadb"
mysql_secure_installation <<EOF
```

```
y
123456
123456
y
y
y
y
EOF
echo "Create database"
mysql -u root -p123456 <<EOF
CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE
↪ utf8_general_ci;
EOF
mysql -u root -p123456 addressbook < /var/backup/addressbook.sql
fi
```

Этот скрипт, по сути, повторяет произведённые в работе действия по установке и настройке сервера баз данных.

3. Для отработки созданного скрипта во время загрузки виртуальных машин в конфигурационном файле Vagrantfile добавили в конфигурации сервера следующую запись:

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

3 Выводы