

# Лабораторная работа №6 (Установка и настройка системы управления базами данных MariaDB)

---

Шубина София Антоновна

2 октября 2025

Российский университет дружбы народов

Приобрести практические навыки 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 восстановите 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 внести изменения в Vagrantfile.

# Выполнение лабораторной работы

---

## Установим необходимые для работы с базами данных пакеты:

```
[sashubina@server.sashubina.net ~]$ sudo -i
[sudo] password for sashubina:
[root@server.sashubina.net ~]# dnf -y install mariadb mariadb-server
Last metadata expiration check: 0:53:14 ago on Fri 03 Oct 2025 06:49:28 PM UTC.
Package mariadb-3:10.11.11-1.el10.x86_64 is already installed.
Package mariadb-server-3:10.11.11-1.el10.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

**Рис. 1:** Установка пакетов

# Просмотрим конфигурационные файлы mariadb в каталоге /etc/my.cnf.d и в файле /etc/my.cnf.

```
[root@server.sashubina.net ~]# cat /etc/my.cnf.d/*.cnf
[mariadb]
#plugin-load-add=auth_gssapi.so
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql/
#
# this is read by the standalone daemon and embedded servers
[server]
#
# this is only for the mysqld standalone daemon
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mysqld/mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mariadb/mariadb.log
pid-file=/run/mariadb/mariadb.pid
#
# * Galera-related settings
#
[galera]
# Mandatory settings
wsrep_on=ON
```

Рис. 2: Каталог /etc/my.cnf.d

```
#  
# 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
```

Рис. 3: Файл /etc/my.cnf

## Убедимся, что mariadb прослушивает порт 3306:

```
[root@server.sashubina.net ~]# systemctl start mariadb
[root@server.sashubina.net ~]# systemctl enable mariadb
[root@server.sashubina.net ~]# ss -tulpen | grep mysq
[root@server.sashubina.net ~]# ss -tulpen | grep :3306
tcp    LISTEN 0      80          0.0.0.0:3306  0.0.0.0:*    users:((("mariabdb",pid=1361,fd=20))
                                             uid:27 ino:12397 sk:10 cgroup:/system.slice/mar
iadb.service <->
tcp    LISTEN 0      80          [::]:3306   [::]:*      users:((("mariabdb",pid=1361,fd=21))
                                             uid:27 ino:12398 sk:19 cgroup:/system.slice/mar
iadb.service v6only:1 <->
[root@server.sashubina.net ~]#
```

Рис. 4: Поиск нужного процесса



# Запустим скрипт конфигурации безопасности mariadb.

```
Remove anonymous users? [Y/n]
... 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]
... 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]
- 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]
... 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!
[root@server.sashubina.net ~]#
```

**Рис. 5:** Запуск скрипта конфигурации безопасности mariadb

# Просмотрим список команд MySQL, введя \h.

```
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 c
clear  (\c) Clear the current input statement.
connect (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter.
edit   (\e) Edit command with $EDITOR.
ego    (\G) Send command to MariaDB server, display result vertically.
exit   (\q) Exit mysql. Same as quit.
go      (\g) Send command to MariaDB server.
help   (\h) Display this help.
nopager (\n) Disable pager, print to stdout.
notee   (\t) Don't write into outfile.
nowarning (\w) Don't show warnings after every statement.
pager   (\P) Set PAGER [to_pager]. Print the query results via PAGER.
print   (\p) Print current command.
prompt (\R) Change your mysql prompt.
quit    (\q) Quit mysql.
rehash  (\#) Rebuild completion hash.
sandbox (\.) Disallow commands that access the file system (except \P without an argument and \
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'
```

Рис. 6: список команд MySQL

Для выхода из интерфейса интерактивной оболочки MariaDB введите `exit;`

```
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
4 rows in set (0.005 sec)  
  
MariaDB [(none)]>
```

Рис. 7: Доступные базы данных

# Для отображения статуса MariaDB введем из приглашения интерактивной оболочки MariaDB: status

```
MariaDB [(none)]> status
-----
mysql Ver 15.1 Distrib 10.11.11-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:         10.11.11-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:                 25 min 2 sec

Threads: 1  Questions: 27  Slow queries: 0  Opens: 20  Open tables: 13  Queries per se
-----
```

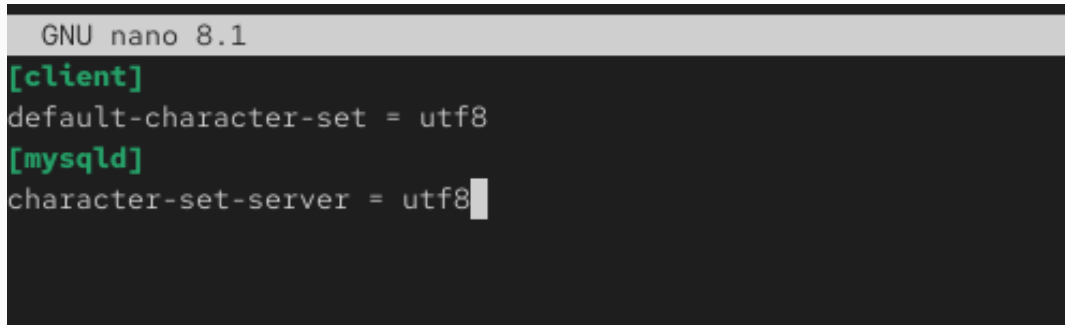
**Рис. 8:** Статус MariaDB

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

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

**Рис. 9:** создание файла

Откроем его на редактирование и укажем в нём следующую конфигурацию:



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

Рис. 10: Редактирование файла

# Войдем в базу данных с правами администратора и посмотрим статус MariaDB.

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

Connection id:          3
Current database:
Current user:           root@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server:                 MariaDB
Server version:         10.11.11-MariaDB MariaDB Server
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    latin1
Db characterset:        latin1
Client characterset:    utf8mb3
Conn. characterset:     utf8mb3
UNIX socket:            /var/lib/mysql/mysql.sock
Uptime:                 24 sec

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

Рис. 11: Статус MariaDB

**Создадим базу данных с именем  
addressbook. ;**

---



## Создадим базу данных с именем addressbook. ;

```
MariaDB [(none)]> CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;  
Query OK, 1 row affected (0.003 sec)
```

**Рис. 12:** Создание базы данных и таблицы

## USE addressbook;

```
MariaDB [addressbook]> USE addressbook;  
Database changed  
MariaDB [addressbook]> █
```

Рис. 13: USE addressbook

## Заполним несколько строк таблицы некоторыми данными по аналогии в соответствии с синтаксисом MySQL:

```
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 (0.006 sec)  
  
MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Иванов','Москва');  
Query OK, 1 row affected (0.002 sec)  
  
MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Петров','Сочи');  
Query OK, 1 row affected (0.002 sec)  
  
MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Сидоров','Дубна');  
Query OK, 1 row affected (0.002 sec)  
  
MariaDB [addressbook]>
```

Рис. 14: заполнение строк

Сделаем следующий MySQL-запрос: `SELECT * FROM city;`. Увидим, что вывелась таблица с колонками город и имя с заполненными мною данными.

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

Рис. 15: `SELECT * FROM city`

**Создадим пользователя для работы с базой данных addressbook и зададим для него пароль и предоставим права доступа созданному пользователю user**

```
MariaDB [addressbook]> CREATE USER sashubina@'%' IDENTIFIED BY 'password';  
Query OK, 0 rows affected (0.003 sec)  
  
MariaDB [addressbook]> GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO sashubina@'%';  
Query OK, 0 rows affected (0.002 sec)
```

**Рис. 16:** создание пользователя и предоставление прав доступа

## Обновим привилегии (права доступа) базы данных addressbook и посмотрим общую информацию о таблице city базы данных addressbook

```
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. 17: обновим права доступа и посмотрим инф

## Просмотрим список баз данных: `mysqlshow -u root -p`

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

**Рис. 18:** Работа с базой данных

Просмотрим список таблиц базы данных addressbook: `mysqlshow -u user -p addressbook`

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

Рис. 19: список таблиц



# Резервные копии

```
[root@server.sashubina.net my.cnf.d]# mkdir -p /var/backup
[root@server.sashubina.net my.cnf.d]# mysqldump -u root -p addressbook > /var/backup/addressbook.sql
Enter password:
[root@server.sashubina.net my.cnf.d]# mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz
Enter password:
[root@server.sashubina.net my.cnf.d]# 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
Enter password:
mysqldump: Got errno 32 on write
[root@server.sashubina.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.sashubina.net my.cnf.d]# mysql -u root -p addressbook < /var/backup/addressbook.sql
Enter password:
[root@server.sashubina.net my.cnf.d]# zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook
Enter password:
[root@server.sashubina.net my.cnf.d]#
```

Рис. 20: Работа с резервными копиями

# Внесение изменений в настройки внутреннего окружения виртуальной машины

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

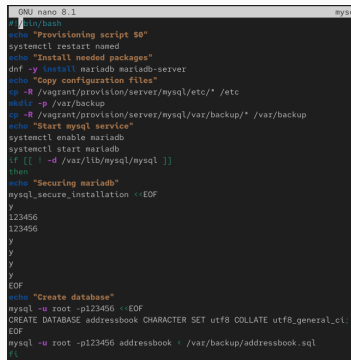
**Рис. 21:** Внесения изменений в настройки внутреннего окружения

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

```
[root@server.sashubina.net server]# cd /vagrant/provision/server  
[root@server.sashubina.net server]# touch mysql.sh  
[root@server.sashubina.net server]# chmod +x mysql.sh  
[root@server.sashubina.net server]#
```

Рис. 22: Создание файла

Открыв его на редактирование, пропишем в нём следующий скрипт:

A screenshot of a terminal window titled 'GNU nano 8.1' with a 'mysql' tab. The terminal displays a shell script for provisioning a MySQL database. The script includes commands for restarting named, installing mariadb and mariadb-server, copying configuration files, creating a backup directory, enabling and starting mariadb, securing it with a password '123456', and creating a database named 'addressbook' with a utf8 character set. The script ends with a command to run a SQL file from a backup directory.

```
GNU nano 8.1 mysql
#!/bin/bash
echo "Provisioning script 50"
systemctl restart named
echo "Install needed packages"
dnf -y install mariadb mariadb-server
echo "Copy configuration files"
cp -R /vagrant/provision/server/mysql/etc/* /etc
mkdir -p /var/backup
cp -R /vagrant/provision/server/mysql/var/backup/* /var/backup
echo "Start mysql service"
systemctl enable mariadb
systemctl start mariadb
if [[ ! -d /var/lib/mysql/mysql ]]
then
echo "Securing mariadb"
mysql_secure_installation <<EOF
y
123456
123456
y
y
y
y
EOF
echo "Create database"
mysql -u root -p123456 <<EOF
CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
EOF
mysql -u root -p123456 addressbook < /var/backup/addressbook.sql
fi
```

Рис. 23: Редактирование файла

в конфигурационном файле Vagrantfile необходимо добавить в конфигурации сервера следующую запись:

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

**Рис. 24:** Редактирование файла Vagrantfile

В процессе выполнения данной лабораторной работы я приобрела практические навыки по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.