

Лабораторная работа №6

Администрирование сетевых подсистем

Машков Илья Евгеньевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Установка MariaDB	7
3.2	Создание базы данных	11
3.3	Резервные копии	14
3.4	Внесение изменений в настройки внутреннего окружения виртуальной машины	15
4	Выводы	18
	Список литературы	19

Список иллюстраций

3.1	Установка пакетов	7
3.2	Запуск и включение mariadb	8
3.3	Проверка работы mariadb	8
3.4	Скрипт безопасности mariadb	9
3.5	Запуск базы данных	10
3.6	Список команд	10
3.7	Просмотр баз данных	11
3.8	Создание БД	11
3.9	Просмотр таблиц в БД	12
3.10	Заполнение строк таблицы и обращение к ней	12
3.11	Настройка пользователя для работы с БД и просмотр общей информации	13
3.12	Список всех ДБ	13
3.13	Список таблиц	14
3.14	Работа с резервными копиями	15
3.15	Создание поддиректорий для хранения конфигов и резервных копий	15
3.16	Копирование конфигов и резервных копий	16
3.17	mysql.sh	16
3.18	Изменения Vagrantfile	17

Список таблиц

1 Цель работы

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

2 Задание

1. Установите необходимые для работы MariaDB пакеты.
2. Настройте в качестве кодировки символов по умолчанию utf8 в базах данных.
3. В базе данных MariaDB создайте тестовую базу addressbook, содержащую таблицу city с полями name и city, т.е., например, для некоторого сотрудника указан город, в котором он работает.
4. Создайте резервную копию базы данных addressbook и восстановите из неё данные.
5. Напишите 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.

3 Выполнение лабораторной работы

3.1 Установка MariaDB

Устанавливаю необходимые пакеты для работы с MariaDB (рис. [3.1]).

```
[user@server ~]$ sudo -i
[sudo] password for user:
[root@server ~]# dnf -y install mariadb mariadb-server
Extra Packages for Enterprise Linux 9 - x86_64 9.7 kB/s | 35 kB      00:03
Extra Packages for Enterprise Linux 9 - x86_64 2.3 MB/s | 20 MB      00:08
Rocky Linux 9 - BaseOS                        735 B/s | 4.3 kB      00:06
Rocky Linux 9 - BaseOS                        5.1 MB/s | 6.3 MB      00:01
Rocky Linux 9 - AppStream                     15 kB/s | 4.8 kB      00:00
Rocky Linux 9 - AppStream                     6.7 MB/s | 11 MB      00:01
Rocky Linux 9 - Extras                        10 kB/s | 3.1 kB      00:00
Rocky Linux 9 - Extras                        45 kB/s | 16 kB      00:00
Dependencies resolved.
=====
Package                                Arch      Version                                Repository      Size
=====
Installing:
mariadb                                x86_64    3:10.5.27-1.el9_5.0.2                appstream       1.6 M
mariadb-server                          x86_64    3:10.5.27-1.el9_5.0.2                appstream       9.7 M
Upgrading:
selinux-policy                         noarch    38.1.65-1.el9                        baseos          42 k
selinux-policy-targeted                noarch    38.1.65-1.el9                        baseos          6.5 M
Installing dependencies:
mariadb-common                          x86_64    3:10.5.27-1.el9_5.0.2                appstream       27 k
mariadb-errmsg                          x86_64    3:10.5.27-1.el9_5.0.2                appstream       211 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
Installing weak dependencies:
mariadb-backup                          x86_64    3:10.5.27-1.el9_5.0.2                appstream       6.5 M
mariadb-gssapi-server                  x86_64    3:10.5.27-1.el9_5.0.2                appstream       14 k
mariadb-server-utils                   x86_64    3:10.5.27-1.el9_5.0.2                appstream       210 k

Transaction Summary
=====
Install  9 Packages
Upgrade  2 Packages
```

Рис. 3.1: Установка пакетов

Запускаю и включаю mariadb (рис. [3.2]).

```
[root@server my.cnf.d]# systemctl start mariadb
[root@server my.cnf.d]# 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 my.cnf.d]#
```

Рис. 3.2: Запуск и включение mariadb

Затем убеждаюсь в том, что mariadb прослушивает порт 3306, но на скриншоте этого нет, т.к. это первая его версия, после парочки изменений всё заработало (рис. [3.3]).

```
[root@server my.cnf.d]# ss -tulpen | grep mysql
[root@server my.cnf.d]#
```

Рис. 3.3: Проверка работы mariadb

Запускаю скрипт конфигурации безопасности mariadb (рис. [3.4]).


```

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.

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!
[root@server my.cnf.d]#

```

Рис. 3.4: Скрипт безопасности mariadb

В диалоговом окне было выбрано следующее: устанавливаю пароль для пользователя root базы данных, отключая удалённый корневой доступ и удаляю тестовую базу данных и любых анонимных пользователей.

Запускаю базу данных с правами администратора (рис. [3.5]).

```

[root@server my.cnf.d]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.5.27-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. 3.5: Запуск базы данных

Просматриваю список команд(рис. [3.6]).

```

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 mult
i-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_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 argume
nt 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'

MariaDB [(none)]>

```

Рис. 3.6: Список команд

Затем просматриваю все доступные базы данных (рис. [3.7]).

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

MariaDB [(none)]> 
```

Рис. 3.7: Просмотр баз данных

На тот момент мне были доступны три базы данных: `information_schema`, `mysql`, `performance_schema`.

3.2 Создание базы данных

Создаю БД с именем `addressbook` (рис. [3.8]).

```
MariaDB [(none)]> CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.000 sec)
```

Рис. 3.8: Создание БД

Перехожу к нашей БД и отображаю таблицы, которых там, ожидаемо, нет(рис. [3.9]).

```

MariaDB [(none)]> USE addressbook;
Database changed
MariaDB [addressbook]> SHOW TABLES;
Empty set (0.001 sec)

MariaDB [addressbook]>

```

Рис. 3.9: Просмотр таблиц в БД

Создаю таблицу с названием **“city”** и заполняю строки **“name”**(Иванов, Петров, Сидоров) и **“city”**(Москва, Сочи, Дубна). Затем обращаюсь к таблице через **“*”**, что означает вывод всей информации (рис. [3.10]).

```

MariaDB [addressbook]> CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
Query OK, 0 rows affected (0.009 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]> SELECT * FROM city;
+-----+-----+
| name      | city    |
+-----+-----+
| Иванов    | Москва  |
| Петров    | Сочи    |
| Сидоров   | Дубна   |
+-----+-----+
3 rows in set (0.001 sec)

MariaDB [addressbook]>

```

Рис. 3.10: Заполнение строк таблицы и обращение к ней

Затем создаю пользователя для работы с БД, задаю ему пароль, предоставляю права доступа для пользователя и обновляю привелегии, а также просматриваю общую информацию о ней и выхожу (рис. [3.11]).

```

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

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

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. 3.11: Настройка пользователя для работы с БД и просмотр общей информации

Просматриваю список всех БД, где вижу созданную мной БД addressbook(рис. [3.12]).

```

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

```

Рис. 3.12: Список всех ДБ

Просматриваю список таблиц в addressbook двумя способами (рис. [3.13]).

```

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

```

Рис. 3.13: Список таблиц

3.3 Резервные копии

Создаю каталог для хранения резервных копий, делаю резервную копию addressbook, сжатую резервную копию, восстанавливаю БД из резервной копии и из сжатой резервной копии (рис. [3.14]).

```

[root@server my.cnf.d]# cd ~
[root@server ~]# mkdir -p /var/backup
[root@server ~]#
[root@server ~]# mysqldump -u root -p addressbook > /var/backup/addressbook.sql
Enter password:
[root@server ~]# mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz
Enter password:
[root@server ~]# 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 ~]# mysqldump -u root -p addressbook | gzip > $(date
+var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz)
-bash: +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 ~]# mysqldump -u root -p addressbook | gzip > $(date
+var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz)
-bash: +var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz: No such file or directory
Enter password: -bash: $(date
+var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz): ambiguous redirect

mysqldump: Got errno 32 on write
[root@server ~]# mysqldump -u root -p addressbook | gzip > $(date +var/backup/addressboo
k.%Y%m%d.%H%M%S.sql.gz)
Enter password:
[root@server ~]# mysqldump -u root -p addressbook | gzip > $(date +var/backup/addressboo
k.%Y%m%d.%H%M%S.sql.gz)
Enter password:
[root@server ~]# mysqldump -u root -p addressbook | gzip > $(date +var/backup/addressboo
k.%Y%m%d.%H%M%S.sql.gz)
Enter password:
[root@server ~]# mysql -u root -p addressbook < /var/backup/addressbook.sql
Enter password:
[root@server ~]# zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook
Enter password:
[root@server ~]# cd /var/backup
[root@server backup]# ls -l
total 20
-rw-r--r--. 1 root root 797 Dec 15 19:25 addressbook.20251215.192509.sql.gz
-rw-r--r--. 1 root root 797 Dec 15 19:25 addressbook.20251215.192519.sql.gz
-rw-r--r--. 1 root root 797 Dec 15 19:25 addressbook.20251215.192535.sql.gz
-rw-r--r--. 1 root root 1997 Dec 15 19:22 addressbook.sql
-rw-r--r--. 1 root root 797 Dec 15 19:22 addressbook.sql.gz
[root@server backup]#

```

Рис. 3.14: Работа с резервными копиями

3.4 Внесение изменений в настройки внутреннего окружения виртуальной машины

Создаю поддиректории в каталоге внутреннего окружения машины server для хранения конфигурационных файлов mariadb и резервных копий (рис. [3.15]).

```

[root@server ~]# cd /vagrant/provision/server
[root@server server]# mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d
[root@server server]# mkdir -p /vagrant/provision/server/mysql/var/backup

```

Рис. 3.15: Создание поддиректорий для хранения конфигов и резервных копий

Копирую конфигурационные файлы и резервные копии в соответствующие

директории (рис. [3.16]).

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

Рис. 3.16: Копирование конфигов и резервных копий

Создаю скрипт `mysql.sh`, который будет воспроизводить все проделанные мной действия (рис. [3.17]).

```
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  
2005  
2005  
y  
y  
y  
y  
EOF  
  
echo "Create database"  
mysql -u root -p2005 <<EOF  
CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;  
EOF  
mysql -u root -p2005 addressbook < /var/backup/addressbook.sql  
  
fi
```

Рис. 3.17: `mysql.sh`

Для отработки скрипта добавляю соответствующую запись в Vagrantfile (рис. [3.18]).


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

Рис. 3.18: Изменения Vagrantfile

4 Выводы

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

Список литературы

Администрирование сетевых подсистем