

Лабораторная работа №6

Установка и настройка системы управления базами данных MariaDB

Сахно Никита НФИбд-02-23

Содержание

1	Цель работы	1
2	Задание	1
3	Выполнение лабораторной работы.....	1
4	Выводы.....	9

1 Цель работы

Приобрести практические навыки по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

2 Задание

1. Установить необходимые для работы MariaDB пакеты (см. раздел 6.4.1).
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.

3 Выполнение лабораторной работы

Установка MariaDB

Загрузим операционную систему и перейдем в рабочий каталог с проектом: `cd C:\Users\nikita\work\study\nvsakhno\vagrant`

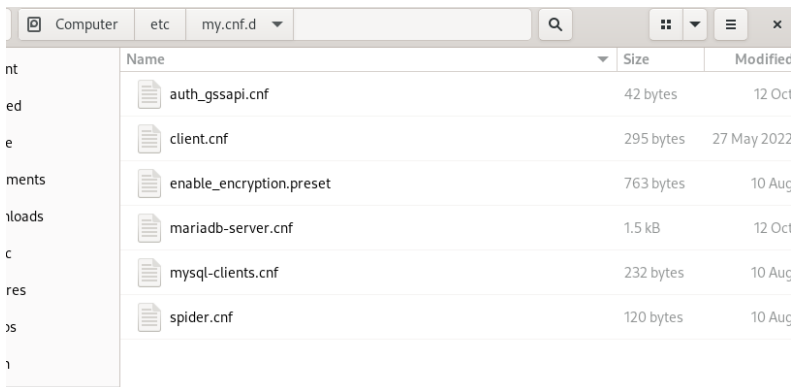
Запустим виртуальную машину server: `make server-up`

На виртуальной машине server войдем под своим пользователем и откроем терминал. Перейдем в режим суперпользователя: `sudo -i` Установим необходимые для работы с базами данных пакеты:

```
Rocky Linux 9 - BaseOS
Rocky Linux 9 - AppStream
Rocky Linux 9 - Extras
Dependencies resolved.
=====
Package                                     Architecture
-----
Installing:
mariadb                                     x86_64
mariadb-server                             x86_64
Installing dependencies:
mariadb-common                             x86_64
mariadb-connector-c                         x86_64
mariadb-connector-c-config                 noarch
```

Установка пакетов

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



Name	Size	Modified
auth_gssapi.cnf	42 bytes	12 Oct
client.cnf	295 bytes	27 May 2022
enable_encryption.preset	763 bytes	10 Aug
mariadb-server.cnf	1.5 kB	12 Oct
mysql-clients.cnf	232 bytes	10 Aug
spider.cnf	120 bytes	10 Aug

Каталог `/etc/my.cnf.d`



```
1 #
2 # This group is read both both by the client and the server
3 # use it for options that affect everything
4 #
5 [client-server]
6
7 #
8 # include all files from the config directory
9 #
10 !includedir /etc/my.cnf.d
11
```

Файл `/etc/my.cnf`

Для запуска и включения программного обеспечения mariadb используем следующие команды:

```
systemctl enable mariadb.service
systemctl start mariadb.service
mariadb.service → /usr/lib/systemd/system/mariadb.service.
```

Запуска и включения ПО mariadb

Убедимся, что mariadb прослушивает порт 3306:

```

~ 3306
mariadb LISTEN 10140/mariadb
+[*] users:({"mariadb",pid:10140,fd:19}) uid:27 fno:42070 sk:17 cgroup:/system.slice/mariadb.service

```

Поиск нужного процесса

Запустим скрипт конфигурации безопасности mariadb. С помощью запустившегося диалога и путём выбора [Y/n] установим пароль для пользователя root базы данных (обратите внимание, что это не пользователь root операционной системы), отключим удалённый корневой доступ и удалим тестовую базу данных и любых анонимных пользователей.

```

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'.

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

```

Запуск скрипта конфигурации безопасности mariadb

Для входа в базу данных с правами администратора базы данных введем: `mysql -u root -p` Просмотрим список команд MySQL, введя `\h`.

```

[root@server.dmbelicheva.net ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.5.22-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)]> \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'.
clear  (\c) Clear the current input statement.

```

Вход в базу данных

Из приглашения интерактивной оболочки MariaDB для отображения доступных в настоящее время баз данных введем MySQL-запрос `SHOW DATABASES`; Для выхода из интерфейса интерактивной оболочки MariaDB введите `exit`;

```

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

MariaDB [(none)]> exit;
Bye

```

Доступные базы данных

Видно, что на данный момент есть три базы данных (mysql, information_schema, performance_schema).

Конфигурация кодировки символов

Войдем в базу данных с правами администратора: `mysql -u root -p` Для отображения статуса MariaDB введем из приглашения интерактивной оболочки MariaDB: `status`

```

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
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)]> status
-----
mysql Ver 15.1 Distrib 10.5.22-MariaDB, for Linux (x86_64) using Editline wrapper

Connection id:          10
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:                 20 min 17 sec

Threads: 1  Questions: 19  Slow queries: 0  Opens: 20  Open tables: 13  Queries per second avg: 0.015
-----

```

Статус MariaDB

Вывелась информация, из которой мы можем узнать, например, текущую бд, текущего пользователя, сервер, версию сервера, версию протокола, набор символов сервера, базы данных и т.д.

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

```

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

```

Откроем его на редактирование и укажем в нём следующую конфигурацию:

```
root@server:/etc/my.cnf.d
GNU nano 5.6.1 utf8.cnf
[ctrl+w]
default-character-set = utf8
[mysql]
character-set-server = utf8
```

Редактирование файла

Перезапустим MariaDB: `systemctl restart mariadb` Войдем в базу данных с правами администратора и посмотрим статус MariaDB.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
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)]> 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:                  27 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second avg: 0.148
-----
```

Статус MariaDB

Увидим, что поменялся стандарт кодирования символов для сервера и базы данных на utf8.

Создание базы данных

Войдем в базу данных с правами администратора: `mysql -u root -p` 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.

```
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 (0.066 sec)

MariaDB [addressbook]> SHOW TABLES;
+-----+
| Tables_in_addressbook |
+-----+
| city                   |
+-----+
1 row in set (0.000 sec)
```

Создание базы данных и таблицы

Заполним несколько строк таблицы некоторыми данными по аналогии в соответствии с синтаксисом MySQL: Добавим в базу сведения о Петрове и Сидорове: Петров, Сочи Сидоров, Дубна

Сделаем следующий MySQL-запрос: `SELECT * FROM city;`. Увидим, что вывелась таблица с колонками город и имя с заполненными мною данными. Создадим пользователя для работы с базой данных addressbook и зададим для него пароль: `CREATE USER dmbelicheva@'%' IDENTIFIED BY 'password';` Предоставим права доступа созданному пользователю user на действия с базой данных addressbook (просмотр, добавление, обновление, удаление данных): `GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO user@'%;`

```
MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ('Иванов','Москва');
Query OK, 1 row affected (0.016 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 (0.010 sec)

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

MariaDB [addressbook]> CREATE USER dmbelicheva@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.027 sec)

MariaDB [addressbook]> GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO dmbelicheva@'%;
Query OK, 0 rows affected (0.011 sec)
```

Различные операции по работе с базой данных

Обновим привилегии (права доступа) базы данных addressbook: `FLUSH PRIVILEGES;` Посмотрим общую информацию о таблице city базы данных addressbook: `DESCRIBE city;` Выйдем из окружения MariaDB: `quit` Просмотрим список баз данных: `mysqlshow -u root -p` Просмотрим список таблиц базы данных addressbook: `mysqlshow -u user -p addressbook`

```
MariaDB [addressbook]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.047 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 (0.019 sec)

MariaDB [addressbook]> quit
Bye
```

Работа с базой данных

Резервные копии

На виртуальной машине server создадим каталог для резервных копий: `mkdir -p /var/backup` Сделаем резервную копию базы данных addressbook: `mysqldump -u root -p addressbook > /var/backup/addressbook.sql` Сделаем сжатую резервную копию базы

данных addressbook: `mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz` Сделаем сжатую резервную копию базы данных addressbook с указанием даты создания копии: `mysqldump -u root -p addressbook | gzip > $(date +%Y-%m-%d-%H%M%S).sql.gz` Восстановим базу данных addressbook из резервной копии: `mysql -u root -p addressbook < /var/backup/addressbook.sql` Восстановим базу данных addressbook из сжатой резервной копии: `zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook`

```
account backup crash empty games lib lock mail nis preserve sasl www  
adm cache db ftp kerberos local log named opt run ssh yp
```

Работа с резервными копиями

Внесение изменений в настройки внутреннего окружения виртуальной машины

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создадим в нём каталог `mysql`, в который поместим в соответствующие подкаталоги конфигурационные файлы MariaDB и резервную копию базы данных addressbook:

```
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/  
touch mysql.sh  
chmod +x mysql.sh
```

Внесения изменений в настройки внутреннего окружения

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

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

Открыв его на редактирование, пропишем в нём следующий скрипт:

```
root@server:/vagrant/provision/server

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

Редактирование файла

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

Для отработки созданного скрипта во время загрузки виртуальных машин 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"
```

```
35 |         ip: "192.168.1.1",
36 |         virtualbox__intnet: true
37 |
38 |     server.vm.provision "server dummy",
39 |         type: "shell",
40 |         preserve_order: true,
41 |         path: "provision/server/01-dummy.sh"
42 |
43 |     server.vm.provision "server dns",
44 |         type: "shell",
45 |         preserve_order: true,
46 |         path: "provision/server/dns.sh"
47 |
48 |     server.vm.provision "server dhcp",
49 |         type: "shell",
50 |         preserve_order: true,
51 |         path: "provision/server/dhcp.sh"
52 |
53 |     server.vm.provision "server http",
54 |         type: "shell",
55 |         preserve_order: true,
56 |         path: "provision/server/http.sh"
57 |
58 |     server.vm.provision "server mysql",
59 |         type: "shell",
60 |         preserve_order: true,
61 |         path: "provision/server/mysql.sh"
62 |
```

Редактирование файла Vagrantfile

4 Выводы

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