

Лабораторная работа № 6

Установка и настройка системы управления базами данных MariaDB

Абд эль хай мохамад

Содержание

Цель работы.....	2
Задание.....	2
Выполнение работы	2
1. Установка MariaDB.....	2
2. Конфигурация кодировки символов	6
3. Создание базы данных.....	8
Вхожу в базу данных с правами администратора: mysql -u root -p.....	8
4. Резервные копии	10
5. Внесение изменений в настройки внутреннего окружения виртуальной машины	11
Вывод.....	12
Ответы на контрольные вопросы.....	12

Цель работы

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

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 внести изменения в Vagrantfile.

Выполнение работы

1. Установка MariaDB

Загружаю операционную систему и перехожу в рабочий каталог с проектом:

Запускаю виртуальную машину server: make server

На виртуальной машине server вхожу под своим пользователем и открываю терминал.

Перехожу в режим суперпользователя: sudo -i

Устанавливаю необходимые для работы с базами данных пакеты:

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

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

Содержание файла /etc/my.cnf.d/client.cnf:

```
#      Эти две группы читаются клиентской библиотекой
```

```
#      Используйте его для параметров, которые влияют на всех клиентов, но
```

```
#      не на сервер
```

```
#
```

```
[client]
```

```
#      Эта группа не читается клиентской библиотекой mysql,
```

Если вы используете один и тот же .cnf файл для MySQL и MariaDB,
используйте его только для параметров клиента MariaDB

[client-mariadb]

[group] - группа представляет собой имя программы или группы, для которой необходимо задать параметры. Любые параметры или строки, задающие значения переменных, которые указаны после строки группы, будут относиться к названной группе, пока не закончится файл параметров или же не будет указана другая строка группы. Группа client обеспечивает возможность задавать параметры, относящиеся ко всем клиентам MySQL (кроме самого mysqld).

Содержание файла /etc/my.cnf.d/mysql-clients.cnf:

#

Эти группы читаются средствами командной строки MariaDB

Используйте его для параметров, которые влияют только на одну утилиту

#

[mysql] - инструмент командной строки MySQL

[mysqlupgrade] – утилита проверяет все таблицы во всех базах данных на предмет несовместимости с текущей версией MySQL Server, также обновляет системные таблицы, чтобы вы могли использовать новые привилегии или возможности, которые могли быть добавлены.

[mysqladmin] - клиент для выполнения административных операций.

[mysqlbinlog] - утилита для обработки бинарных файлов журнала

[mysqlcheck] - программа выполняет обслуживание таблиц: проверяет, ремонт, оптимизирует, или анализирует их.

[mysqldump] - программа резервного копирования базы данных

[mysqlimport] - программа резервного копирования базы данных

[mysqlshow] – программа отображает информацию о базе данных, таблице и столбце

[mysqlslap] - это диагностическая программа, предназначенная для эмуляции нагрузки клиента для сервера MySQL и для отчетности о сроках каждого этапа.

Содержание файла /etc/my.cnf.d/server.cnf:

```
# Эти группы читаются сервером MariaDB.

# Используйте его для параметров, которые должен видеть сервер (но не
# клиенты).

#

# См. примеры файлов my.cnf сервера в каталоге /usr/share/mysql/

# это считывается автономным демоном и встроенными серверами [сервер]
# это только для автономного демона mysqld [mysqld]
# это только для встроенного сервера [embedded]
# Эта группа считывается только серверами MariaDB-5.5.
# Если вы используете тот же .cnf-файл для MariaDB разных версий,
# используйте эту группу для параметров, которыми старые серверы не
# смогут воспользоваться
```

[mysqld-5.5] - [MySQL Server](#), является основной программой, которая

[выполняет большую часть работы в установке MySQL.](#)

```
# Эти две группы читаются только серверами MariaDB, а не MySQL.
# Если вы используете тот же .cnf-файл для MySQL и MariaDB,
# вы можете использовать опции MariaDB только здесь [mariadb]
```

[mariadb-5.5] - [ответвление от системы управления базами данных MySQL](#)

Содержание файла /etc/my.cnf:

[mysqld] – [параметры для mysql](#)

`datadir=/var/lib/mysql socket=/var/lib/mysql/mysql.sock`

`datadir` – [путь к каталогу с файлами баз данных MariaDB](#)

`socket` – [путь к файлу сокета MySQL](#)

```
# Отключение символических ссылок рекомендуется для усиления
# безопасности
```

`risks symbolic-links=0`

```
# Параметры пользователя и группы игнорируются при использовании
# systemd.
```

```
# Если вам нужно запустить mysqld под другим пользователем или
# группой, настроить файл systemd для mariadb в соответствии с
# инструкцией в http://fedoraproject.org/wiki/systemd
```

[mysqld_safe] - [рекомендуемый способ запуска сервера mysqld в Unix](#),

[добавляет некоторые функции безопасности, такие как перезапуск сервера при возникновении ошибки и протоколирование информации о времени выполнения в журнал ошибок.](#)

log-error=/var/log/mariadb/mariadb.log – путь к журналу ошибок

pid-file=/var/run/mariadb/mariadb.pid - путь, который mysqld должен использовать для файла идентификатора процесса.

```
#
```

```
# включить все файлы из каталога конфигурации
```

```
#
```

```
!includedir /etc/my.cnf.d
```

Для запуска и включения программного обеспечения mariadb ввожу:

```
systemctl start mariadb
```

```
systemctl enable mariadb
```

Запускаю скрипт конфигурации безопасности mariadb, используя:

```
mysql_secure_installation
```

С помощью запустившегося диалога и путём выбора [Y/n] устанавливаю пароль для пользователя root базы данных, отключаю удалённый root доступ и удаляю тестовую базу данных и анонимных пользователей.

Для входа в базу данных с правами администратора базы данных ввожу:

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

Из приглашения интерактивной оболочки MariaDB для отображения доступных в настоящее время баз данных ввожу MySQL-запрос SHOW DATABASES;

[В системе находятся следующие базы данных: information_schema, mysql, performance_schema.](#)

Для выхода из интерфейса интерактивной оболочки MariaDB ввожу exit;

The screenshot shows a terminal window with two tabs, both titled 'root@server:~'. The active tab displays the following content:

```
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 charsets.
warnings     (\W) Show warnings after every statement.
nowarning    (\w) Don't show warnings after every statement.

For server side help, type 'help contents'

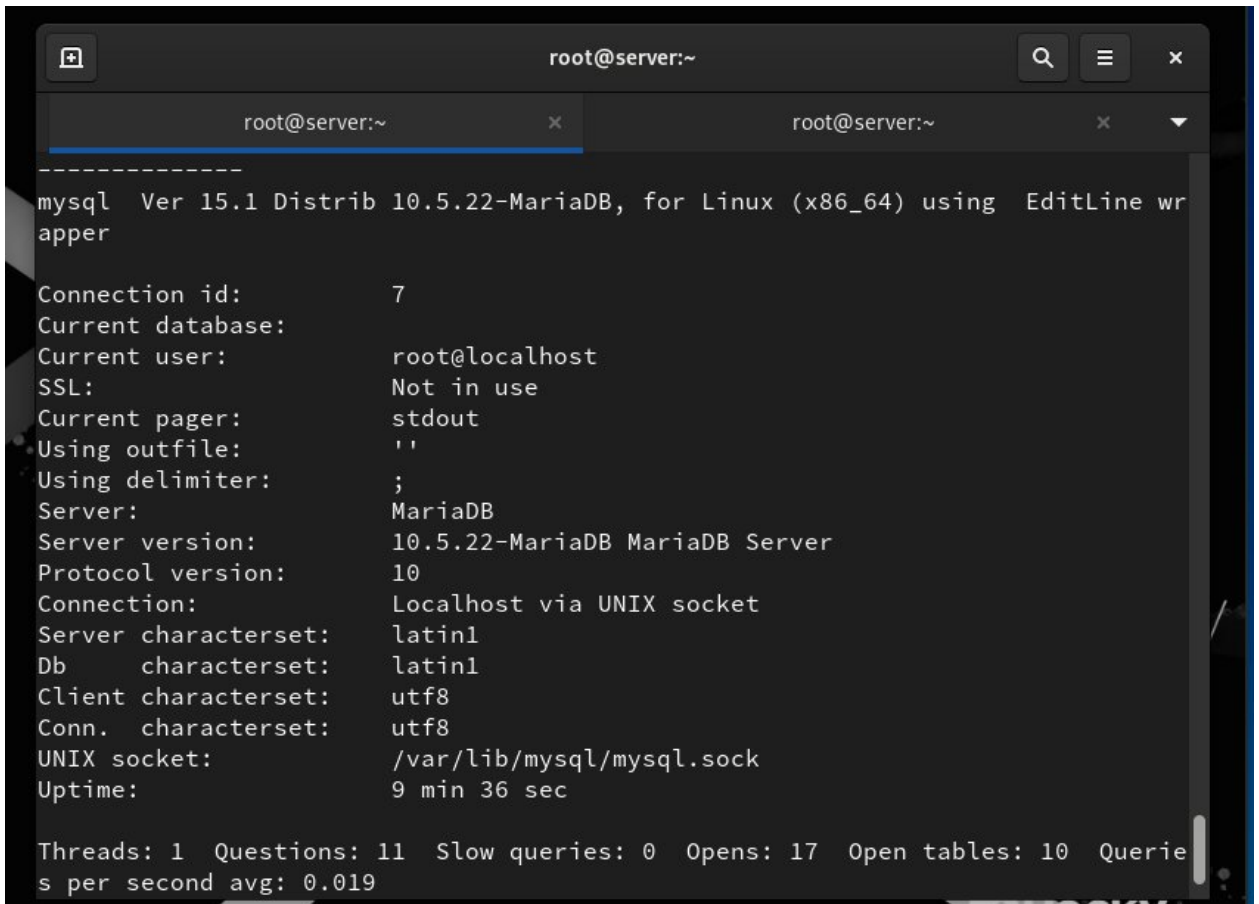
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database                |
+-----+
| information_schema       |
| mysql                    |
| performance_schema       |
+-----+
3 rows in set (0.027 sec)

MariaDB [(none)]>
```

The terminal window has a dark theme and a taskbar at the bottom with various application icons and the text 'Right Ctrl'.

2. Конфигурация кодировки символов

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

A screenshot of a terminal window titled 'root@server:~'. The terminal shows the output of the 'mysql' command, displaying connection details for MariaDB. The output includes connection ID, current database, user, SSL status, pager, outfile, delimiter, server version, protocol version, connection type, character sets, UNIX socket, and uptime. At the bottom, it shows statistics like threads, questions, slow queries, opens, open tables, and queries per second.

```
-----
mysql Ver 15.1 Distrib 10.5.22-MariaDB, for Linux (x86_64) using EditLine wrapper

Connection id:          7
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 36 sec

Threads: 1  Questions: 11  Slow queries: 0  Opens: 17  Open tables: 10  Queries
s per second avg: 0.019
```

В каталоге /etc/my.cnf.d создаю файл utf8.cnf:

```
cd /etc/my.cnf.d
```

```
touch utf8.cnf
```

Открываю его на редактирование и указываю в нём следующую конфигурацию:

```
[client]
```

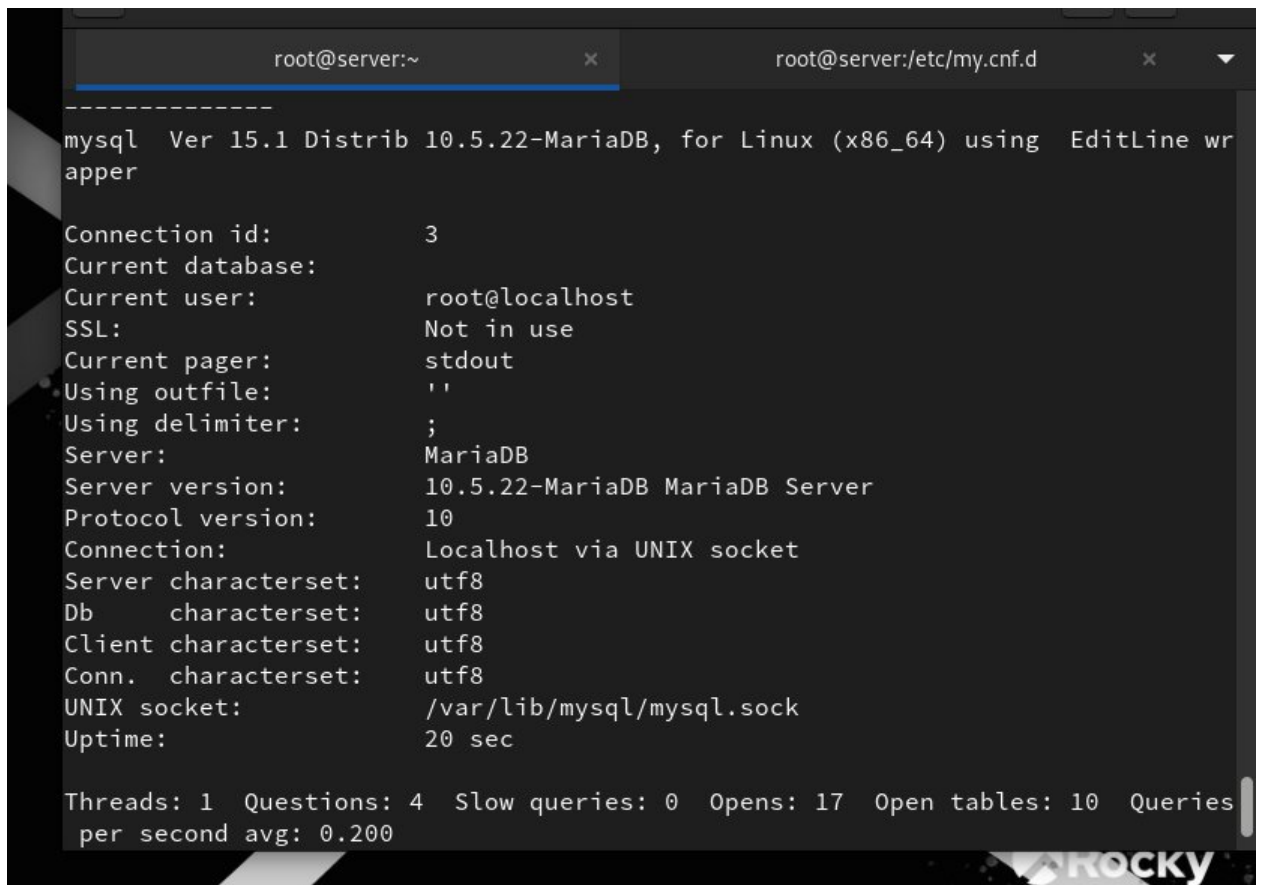
```
default-character-set = utf8
```

```
[mysqld]
```

```
character-set-server = utf8
```

Перезапускаю MariaDB: `systemctl restart mariadb`

Вхожу в базу данных с правами администратора и проверяю статус MariaDB. данные сервера и базы данных теперь кодируются utf8.

A screenshot of a terminal window with two tabs. The first tab is titled 'root@server:~' and the second is 'root@server:/etc/my.cnf.d'. The terminal displays the output of the 'mysql' command, showing connection details for a MariaDB server. The output includes connection ID, current database, user, SSL status, pager, outfile, delimiter, server version, protocol version, connection type, character sets, UNIX socket, and uptime. At the bottom, it shows statistics: 1 thread, 4 questions, 0 slow queries, 17 opens, 10 open tables, and a query rate of 0.200 per second. A 'ROCKY' logo is visible in the bottom right corner of the terminal window.

```
-----
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:                 20 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 17  Open tables: 10  Queries
per second avg: 0.200
```

3.Создание базы данных

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

Создаю базу данных с именем addressbook:

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

Перехожу к базе данных addressbook

```
USE addressbook;
```

Отображаю имеющиеся в базе данных addressbook таблицы:

```
SHOW TABLES;
```

Создаю таблицу city с полями name и city:

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

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

Сделайте следующий MySQL-запрос:

```
SELECT * FROM city;
```

Команда позволяет узнать, какие записи доступны в таблице.

```
MariaDB [addressbook]> INSERT INTO city(name,city) VALUES ( 'Aya' , 'Lebanon' );
Query OK, 1 row affected (0.027 sec)

MariaDB [addressbook]> SELECT * FROM city;
+-----+-----+
| name      | city      |
+-----+-----+
| Иванов    | Москва    |
| Maria     | Poland    |
| liam      | Irlande   |
| sean      | Irlande   |
| Cilian    | UK        |
| Aya       | Lebanon   |
+-----+-----+
6 rows in set (0.000 sec)

MariaDB [addressbook]>
```

Создаю пользователя для работы с базой данных addressbook и задаю для него пароль:
CREATE USER maabeldelhay@'%' IDENTIFIED BY '123456';
Предоставляю права доступа созданному пользователю maabeldelhay на действия с базой
данных addressbook (просмотр, добавление, обновление, удаление данных):
GRANT SELECT, INSERT, UPDATE, DELETE ON addressbook.* TO →
maabeldelhay @'%';
Обновляю привилегии (права доступа) базы данных addressbook:
FLUSH PRIVILEGES;

Просматриваю общую информацию о таблице city базы данных addressbook:
DESCRIBE city;
Выхожу из окружения MariaDB: quit

```
MariaDB [addressbook]> CREATE USER maabeldelhay@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.003 sec)

MariaDB [addressbook]> GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO m
aabeldelhay@'%';
Query OK, 0 rows affected (0.006 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    |       |
+-----+-----+-----+-----+-----+-----+

```

Просматриваю список баз данных: mysqlshow -u root -p

Просматриваю список таблиц базы данных addressbook :
mysqlshow -u root -p addressbook или mysqlshow -u user -p addressbook

```
[root@server.maabelhay.net ~]# mysqlshow -u root -p addressbook
Enter password:
Database: addressbook
+-----+
| Tables |
+-----+
| city   |
+-----+
[root@server.maabelhay.net ~]# mysqlshow -u maabelhay -p addressbook
Enter password:
Database: addressbook
+-----+
| Tables |
+-----+
| city   |
+-----+
[root@server.maabelhay.net ~]#
```

4.Резервные копии

На виртуальной машине server создаю каталог для резервных копий:
`mkdir -p /var/backup`

Создаю резервную копию базы данных addressbook:

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

Создаю сжатую резервную копию базы данных addressbook:

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

Создаю сжатую резервную копию базы данных addressbook с указанием даты создания копии:
`mysqldump -u root -p addressbook | gzip >`

`$(date+var/backup/addressbook.→%Y%m%d.%H%M%S.sql.gz)`

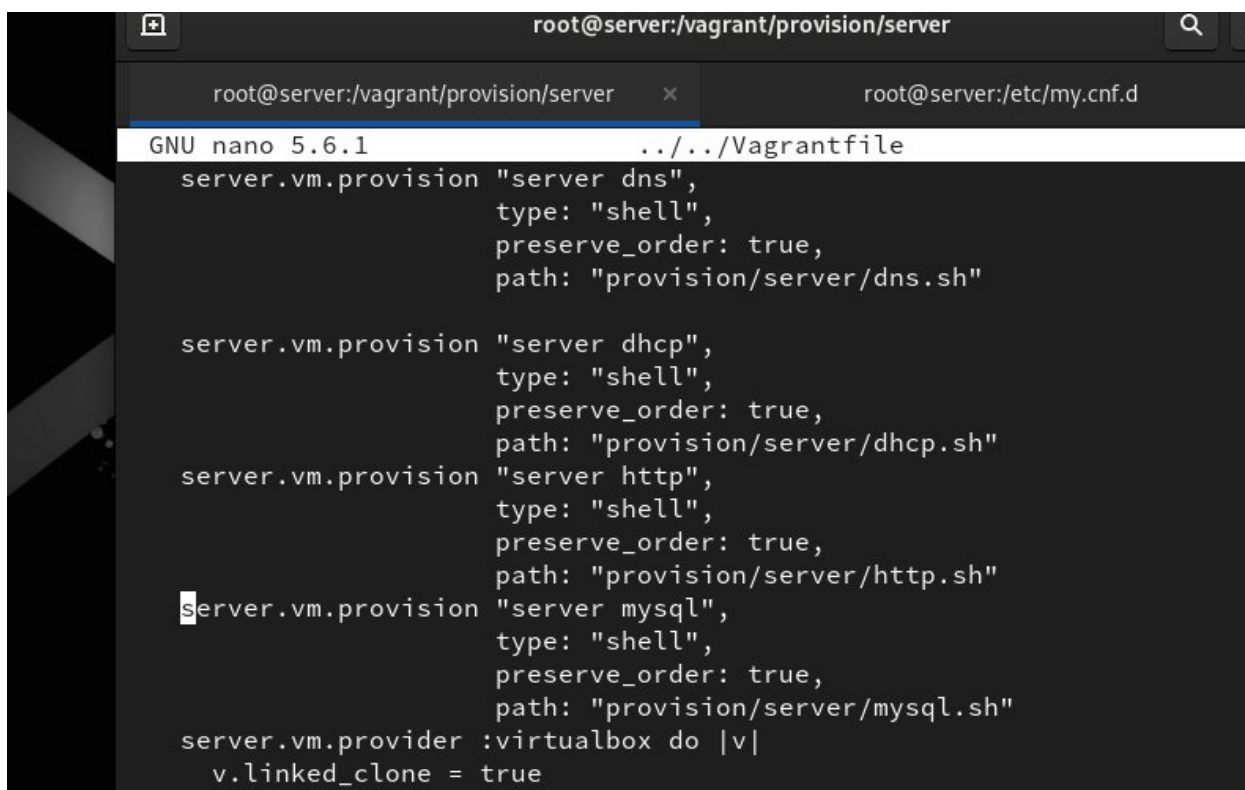
Восстанавливаю базу данных addressbook из резервной копии:

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

Восстанавливаю базу данных addressbook из сжатой резервной копии:

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

5. Внесение изменений в настройки внутреннего окружения виртуальной машины



```
root@server:/vagrant/provision/server
root@server:/etc/my.cnf.d
GNU nano 5.6.1      ../../Vagrantfile
server.vm.provision "server dns",
                    type: "shell",
                    preserve_order: true,
                    path: "provision/server/dns.sh"

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

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

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

server.vm.provider :virtualbox do |v|
  v.linked_clone = true
```

На виртуальной машине server перехожу в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создаю в нём каталог `mysql`, в который перемещаю 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/
```

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

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

Для отработки созданного скрипта во время загрузки виртуальных машин в конфигурационном файле `Vagrantfile` добавляю в конфигурации сервера следующую запись:

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

path: "provision/server/mysql.sh"

Вывод

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

Ответы на контрольные вопросы

1.Какая команда отвечает за настройки безопасности в MariaDB?

`mysql_secure_installation`

2.Как настроить MariaDB для доступа через сеть?

После команды `mysql_secure_installation` настроить MariaDB для удаленного клиентского доступа.

Чтобы включить MariaDB для прослушивания удаленных подключений, вам необходимо отредактировать файл `my.cnf` или взять пример файла `my.cnf` сервера в каталоге `/usr/share/mysql/`

После того, как вы установили файл по умолчанию, используйте текстовый редактор, чтобы открыть файл и попытаться найти строки, подобные этому в разделе `[mysqld]`:

```
[mysqld]
```

```
...
```

```
skip-networking
```

```
...
```

```
bind-address = <some ip-address>
```

```
...
```

Убедитесь, что они закомментированы символами hash (#)

```
[mysqld]
```

```
...
```

```
#skip-networking
```

```
...
```

```
#bind-address = <some ip-address>
```

```
...
```

Сохраните файл и перезапустите демон или службу `mysqld` или `mariadb`.

Теперь, когда ваша установка сервера MariaDB настроена на прием соединений с удаленных хостов, мы должны добавить пользователя, которому разрешено подключаться от чего-то другого, кроме «localhost»

Чтобы создать нового пользователя:

- войдите в `mysql` в командной строке;

- если вы заинтересованы в просмотре любых существующих удаленных пользователей, введите следующую команду SQL в таблице mysql.user:

```
SELECT User, Host FROM mysql.user WHERE Host <> 'localhost';
```

Далее, чтобы создать пользователя будет использован оператор GRANT. Он позволяет создавать учетные записи пользователей MariaDB и предоставлять привилегии для учетных записей.

В синтаксисе GRANT указывается следующая информация:

- список разрешенных привилегий
- к какой базе данных / таблицам эти привилегии применяются
- имя пользователя
- хост, к которому этот пользователь может подключиться
- При необходимости пароль

Пример: GRANT ALL PRIVILEGES ON *.* TO 'root'@'192.168.100.%' IDENTIFIED BY 'password' WITH GRANT OPTION;

Создается пользователь root со всеми правами, который может подключаться из любого места к локальной сети 192.168.100.0/24 и подключается посредством пароля password.

Настраивается firewall на разрешение входящих запросов от удалённых клиентов:

```
firewall-cmd --add-port=3306/tcp
```

```
firewall-cmd --permanent --add-port=3306/tcp
```

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?

В случае, когда пользователь находится в среде оболочки MariaDB используется команда SHOW TABLES;

Для просмотра таблиц базы данных из консоли используется команда mysqlshow -u root -p <название базы данных>

5.Какая команда позволяет узнать, какие поля доступны в таблице?

```
DESCRIBE FROM <название таблицы>;
```

или

```
SHOW FIELDS FROM <название таблицы>;
```

6.Какая команда позволяет узнать, какие записи доступны в таблице?

```
SELECT * FROM <название таблицы>;
```

7.Как удалить запись из таблицы?

```
DELETE FROM <название таблицы> WHERE `id`=<идентификатор записи>
```

8.Где расположены файлы конфигурации MariaDB? Что можно настроить с их помощью?

Конфигурационные файлы MariaDB находятся в каталоге /etc/my.cnf.d а также таковым является файл /etc/my.cnf.(Пункт 5 «Установка MariaDB»).

9.Где располагаются файлы с базами данных MariaDB?

```
/var/lib/mysql
```

10.Как сделать резервную копию базы данных и затем её восстановить?

Данные действия описаны в Ходе работы - Пункт 4. Резервные копии.