

Лабораторная работа № 6. Установка и настройка системы управления базами данных MariaDB

6.1. Цель работы

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

6.2. Предварительные сведения

Система управления базами данных MariaDB представляет собой ответвление от MySQL компании Oracle. MariaDB имеет высокую совместимость с MySQL. Документацию по работе с MariaDB и основы синтаксиса SQL см. в [1–3].

6.3. Задание

1. Установите необходимые для работы MariaDB пакеты (см. раздел 6.4.1).
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 (см. раздел 6.4.1).
4. Создайте резервную копию базы данных addressbook и восстановите из неё данные (см. раздел 6.4.1).
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 (см. раздел 6.4.5).

6.4. Последовательность выполнения работы

6.4.1. Установка MariaDB

В упражнении выполняется базовая установка MariaDB. Также отключается доступ к базе данных по сети и применяются параметры безопасности. Затем проверяется наличие доступных системных баз данных по умолчанию.

1. Загрузите вашу операционную систему и перейдите в рабочий каталог с проектом:


```
cd /var/tmp/user_name/vagrant
```

 Здесь user_name — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.
2. Запустите виртуальную машину server:


```
make server-up
```

 (или, если вы работаете под ОС Windows, то `vagrant up server`).
3. На виртуальной машине server войдите под вашим пользователем и откройте терминал. Перейдите в режим суперпользователя:


```
sudo -i
```
4. Установите необходимые для работы с базами данных пакеты:


```
dnf -y install mariadb mariadb-server
```
5. Просмотрите конфигурационные файлы mariadb в каталоге `/etc/my.cnf.d` и в файле `/etc/my.cnf`. В отчёте прокомментируйте построчно их содержание.
6. Для запуска и включения программного обеспечения mariadb используйте:


```
systemctl start mariadb
systemctl enable mariadb
```
7. Убедитесь, что mariadb прослушивает порт, используя


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

Вы должны увидеть процесс `mysqld`, прослушивающий порт 3306.

8. Запустите скрипт конфигурации безопасности `mariadb`, используя:
`mysql_secure_installation`

С помощью запустившегося диалога и путём выбора [Y/n] установите пароль для пользователя `root` базы данных (обратите внимание, что это не пользователь `root` операционной системы), отключите удалённый корневой доступ и удалите тестовую базу данных и любых анонимных пользователей.

9. Для входа в базу данных с правами администратора базы данных введите
`mysql -u root -p`
10. Просмотрите список команд MySQL, введя `\h`.
11. Из приглашения интерактивной оболочки MariaDB для отображения доступных в настоящее время баз данных введите MySQL-запрос
`SHOW DATABASES;`

В отчёте укажите, какие базы данных есть в системе.

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

6.4.2. Конфигурация кодировки символов

1. Войдите в базу данных с правами администратора:
`mysql -u root -p`
2. Для отображения статуса MariaDB введите из приглашения интерактивной оболочки MariaDB:
`status`

В отчёте построчно поясните выведенную на экран информацию.

3. В каталоге `/etc/my.cnf.d` создайте файл `utf8.cnf`:
`cd /etc/my.cnf.d`
`touch utf8.cnf`

Откройте его на редактирование и укажите в нём следующую конфигурацию:

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

4. Перезапустите MariaDB:
`systemctl restart mariadb`
5. Войдите в базу данных с правами администратора и посмотрите статус MariaDB. В отчёте поясните, что изменилось.

6.4.3. Создание базы данных

1. Войдите в базу данных с правами администратора:
`mysql -u root -p`
2. Создайте базу данных с именем `addressbook`:
`CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE utf8_general_ci;`
3. Перейдите к базе данных `addressbook`
`USE addressbook;`
4. Отобразите имеющиеся в базе данных `addressbook` таблицы:
`SHOW TABLES;`
5. Создайте таблицу `city` с полями `name` и `city`:
`CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));`
6. Заполните несколько строк таблицы некоторыми данными по аналогии с соответствиями с синтаксисом MySQL:
`INSERT INTO city(name,city) VALUES ('Иванов','Москва');`

В частности, добавь в базу сведения о Петрове и Сидорове:
Петров, Сочи
Сидоров, Дубна

7. Сделайте следующий MySQL-запрос:
`SELECT * FROM city;`
 и в отчёте поясните результат его выполнения.
8. Создайте пользователя для работы с базой данных addressbook (вместо user до знака @ используйте ваш логин) и задайте для него пароль:
`CREATE USER user@'%' IDENTIFIED BY 'password';`
9. Предоставьте 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@'%' ;`
10. Обновите привилегии (права доступа) базы данных addressbook:
`FLUSH PRIVILEGES;`
11. Посмотрите общую информацию о таблице city базы данных addressbook:
`DESCRIBE city;`
12. Выйдете из окружения MariaDB:
`quit`
13. Просмотрите список баз данных:
`mysqlshow -u root -p`
14. Просмотрите список таблиц базы данных addressbook:
`mysqlshow -u root -p addressbook`
 или
`mysqlshow -u user -p addressbook`

6.4.4. Резервные копии

1. На виртуальной машине server создайте каталог для резервных копий:
`mkdir -p /var/backup`
2. Сделайте резервную копию базы данных addressbook:
`mysqldump -u root -p addressbook > /var/backup/addressbook.sql`
3. Сделайте сжатую резервную копию базы данных addressbook:
`mysqldump -u root -p addressbook | gzip > /var/backup/addressbook.sql.gz`
4. Сделайте 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)`
5. Восстановите базу данных addressbook из резервной копии:
`mysql -u root -p addressbook < /var/backup/addressbook.sql`
6. Восстановите базу данных addressbook из сжатой резервной копии:
`zcat /var/backup/addressbook.sql.gz | mysql -u root -p addressbook`

6.4.5. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине server перейдите в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создайте в нём каталог mysql, в который поместите 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/`
2. В каталоге /vagrant/provision/server создайте исполняемый файл mysql.sh:

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

Открыв его на редактирование, пропишите 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 вами действия по установке и настройке сервера баз данных.

3. 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"
```

6.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.

2. Формулировка задания работы.
3. Описание результатов выполнения задания:
 - скриншоты (снимки экрана), фиксирующие выполнение работы;
 - подробное описание настроек служб в соответствии с заданием;
 - полные тексты конфигурационных файлов настраиваемых в работе служб;
 - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

6.6. Контрольные вопросы

1. Какая команда отвечает за настройки безопасности в MariaDB?
2. Как настроить MariaDB для доступа через сеть?
3. Какая команда позволяет получить обзор доступных баз данных после входа в среду оболочки MariaDB?
4. Какая команда позволяет узнать, какие таблицы доступны в базе данных?
5. Какая команда позволяет узнать, какие поля доступны в таблице?
6. Какая команда позволяет узнать, какие записи доступны в таблице?
7. Как удалить запись из таблицы?
8. Где расположены файлы конфигурации MariaDB? Что можно настроить с их помощью?
9. Где располагаются файлы с базами данных MariaDB?
10. Как сделать резервную копию базы данных и затем её восстановить?

Список литературы

1. MariaDB Foundation. — URL: <https://mariadb.org> (дата обр. 13.09.2021).
2. Документация по MariaDB. — URL: <https://mariadb.com/kb/ru/5306/>.
3. Основы языка SQL. — URL: <http://citforum.ru/programming/32less/les44.shtml> (дата обр. 13.09.2021).