Лекция
19.02.2019

DBA1. p7.

Управление пользователями

Ильшат Каразбаев
руководитель группы DBA
АО ТК Центр

Немного обо мне

Вместе со своей командой администрирую:

СУБД MySQL, Mariadb, galeracluster, Postgres

Главный по базам в ТК Центр

Повестка дня:

1. Управление пользователями
2. Управление ролями
3. Автоматизация
4. Демонстрация
5. Задания
6. Литература

Управление пользователями. Создание

Необходимые привилегии:

1. CREATE USER
2. INSERT в базу mysql

Создание:

1. CREATE USER ‘foo’@’ip’ IDENTIFIED BY ‘password’;

Если пользователь уже существует:

1. CREATE OR REPLACE
2. DROP USER IF EXISTS … потом CREATE USER

Управление пользователями. Пароли

В открытом виде:

1. CREATE USER ‘foo’@’ip’ IDENTIFIED BY ‘unencrypted_pass’;

С помощью хеша:

1. SELECT PASSWORD(‘unencrypted_pass’);
2. CREATE USER ‘foo’@’ip’ IDENTIFIED BY PASSWORD ‘password_hash’;

С помощью плагинов:

1. SHOW PLUGINS;
2. CREATE USER ‘foo’@’ip’ IDENTIFIED VIA pam;

Управление пользователями. Шифрование трафика

Пример:

```
CREATE USER 'someone'@'localhost'
   REQUIRE SUBJECT '/CN=www.mydom.com/O=My Dom, Inc./C=US/ST=Oregon/L=Portland'
   AND ISSUER '/C=FI/ST=Somewhere/L=City/ O=Some Company/CN=Peter Parker/emailAddress=p.parker@marvel.com'
   AND CIPHER 'SHA-DES-CBC3-EDH-RSA';
```

Option	Description
`REQUIRE NONE`	TLS is not required for this account, but can still be used.
`REQUIRE SSL`	The account must use TLS, but no valid X509 certificate is required.
`REQUIRE X509`	The account must use TLS and must have a valid X509 certificate.
`REQUIRE ISSUER 'issuer'`	The account must use TLS and must have a valid X509 certificate. Also, the Certificate Authority must be the one specified via the string `issuer`.
`REQUIRE SUBJECT 'subject'`	The account must use TLS and must have a valid X509 certificate. Also, the certificate's Subject must be the one specified via the string `subject`.
`REQUIRE CIPHER 'cipher'`	The account must use TLS and must have a valid X509 certificate. Also, the encryption used for the connection must use one of the methods specified in the string `cipher`.

Управление пользователями. Ограничение ресурсов

Начиная с версии MariaDB 10.2

```
CREATE USER 'someone'@'localhost' WITH
   MAX_USER_CONNECTIONS 10
   MAX_QUERIES_PER_HOUR 200;
```

Limit Type	Decription
MAX_QUERIES_PER_HOUR	Number of statements that the account can issue per hour (including updates)
MAX_UPDATES_PER_HOUR	Number of updates (not queries) that the account can issue per hour
MAX_CONNECTIONS_PER_HOUR	Number of connections that the account can start per hour
MAX_USER_CONNECTIONS	Number of simultaneous connections that can be accepted from the same account; if it is 0, `max_connections` will be used instead; if `max_connections` is 0, there is no limit for this account's simultaneous connections.
MAX_STATEMENT_TIME	Timeout, in seconds, for statements executed by the user. See also Aborting Statements that Exceed a Certain Time to Execute.

Управление пользователями. Именование

Имена пользователей состоят из имени и хоста, например:

```
'user_name'@'host_name'
```

Кавычки обязательны, можно использовать одинарные, двойные или бекстики:

```
1.   'user_name'@'host_name'
2.   "user_name"@"host_name"
3.   `user_name`@`host_name`
```

Управление пользователями. Hostname часть

1. localhost - клиент будет коннектиться по сокету локально
2. Hostname часть - чувствительна к регистру
3. Можно использовать ip адрес в качестве хостнейма (ipv6 тоже поддерживается)
4. Можно использовать вайлдакрд в ip адресе, в этом случае вайлдкард аналогичен маске подсети (поддерживается только ipv4)

Пример в следующем слайде!

Управление пользователями. Username часть

Можно иметь несколько пользователей с вайлдкард частью, обработка в следующем порядке:

1. Полное совпадение и имени пользователя и хостнейма
2. Вайлдкард у хостнейма без имени пользователя более точен, чем у хостнейма с именем пользователя
3. При одинаковом вайлдкарде у хостнейма с именем пользователя и без, будет обработан первый

```
+---------+-------------+
| User    | Host        |
+---------+-------------+
| joffrey | 192.168.0.3 |
|         | 192.168.0.% |
| joffrey | 192.168.%   |
|         | 192.168.%   |
+---------+-------------+
```

Управление пользователями. Изменение пользователя.

1. ALTER USER - работает точно так же, как и CREATE, но уже с существующим пользователем. Например, поменять пароль можно точно так же, как и назначить при создании, или поменять другие опции.
2. SET PASSWORD - изменение пароля пользователя.
3. RENAME USER - изменить имя пользователя
4. DROP USER - удаление пользователя. Нужно быть очень осторожным при удалении пользователя, так как все процедуры, триггеры и функции, созданные этим пользователем и доступные другим, перестанут работать. Чтобы исправить - нужно их пересоздать от нового пользователя или изменить дефайнеры в системных таблицах

Роли

Функционал доступен с MariaDB 10.0.5

Роли используются для унификации управления привилегиями в группе пользователей. Например, выделение одинаковых привилегий разработчикам.

Можно, конечно, создать пользователей отдельно и назначить привилегии каждому в отдельности. Но, сложно управлять.

Можно вообще, создать целевых пользователей, но потом тяжело отслеживать кем были внесены изменения

Роли. Роли по умолчанию, переключения между ролями

Узнать роль:

SELECT CURRENT_ROLE;

Переключиться на роль:

SET ROLE developer;

Назначить роль по умолчанию (при коннекте у пользователя сразу появятся права роли):

SET DEAFULT ROLE developer FOR user@host;

Роли. Роли и хранимые процедуры, триггеры, функции

1. Когда пользователю доступны привилегии роли, то у него появляется два набора привилегий, в то время как DEFINER у сущностей базы может быть только один.
2. По умолчанию используется DEFINER пользователя
3. Можно явно задать DEFINER=CURRENT_ROLE или CURRENT_USER в зависимости от потребности
4. Нужно с осторожностью создавать сущности базы с DEFINER=CURRENT_ROLE, так как они могут не работать из-за недостатка привилегий.

Роли. Создание

```
CREATE [OR REPLACE] ROLE [IF NOT EXISTS] role
  [WITH ADMIN
    {CURRENT_USER | CURRENT_ROLE | user | role}]
```

Создание пользователя влечет создание строк в таблицах mysql.user и mysql.roles_mapping

WITH ADMIN - делегирование привилегии назначения привилегий роли другому пользователю/роли (по умолчанию CURRENT USER)

Роли. Другие операции

DROP ROLE [IF EXISTS] role_name [,role_name ...] - удалить роль

CURRENT_ROLE, CURRENT_ROLE() - выбрать роль, которая сейчас используется

SET ROLE { role | NONE } - использовать роль и ее привилегии

SET DEFAULT ROLE { role | NONE } [FOR user@host] - назначение роли по умолчанию

GRANT/REVOKE - назначение и удаление привилегий

Роли. Системные таблицы

1. Information Schema APPLICABLE_ROLES Table

Показывает роли, которые может использовать пользователь

2. Information Schema ENABLED_ROLES Table

Показывает роли, которые задействованы во время сессии

Автоматизация с помощью ansible

модуль mysql_user

Пример:

```
- mysql_user:
    name: bob
    password: '*EE0D72C1085C46C5278932678FBE2C6A782821B4'
    encrypted: yes
    priv: '*.*:ALL'
    state: present
```

Демонстарция

Задания

1. Создайте пользователя с лимитом по количеству сессий в час
2. Обновите пользователя, убрав ограничение по количеству сессий
3. Воссоздайте ситуацию, описанную в https://mariadb.com/kb/en/library/roles_overview/ , секция “**Roles and views (and stored routines)”**

Вопросы?

telegram: karazbaev

vk, instagram: barazbay

twitter: karazbay

Литература

1. https://mariadb.com/kb/en/library/user-account-management/
2. https://mariadb.com/kb/en/library/roles_overview/
3. https://mariadb.com/kb/en/library/set-default-role/
4. https://docs.ansible.com/ansible/latest/modules/mysql_user_module.html#mysql-user-module