Configuration & Manage MariaDB Server Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Session - 42 Agenda:

Manage a MariaDB database server using Ansible:

EX 358 Exam Exclusive Content:

1. Install and configure a basic MariaDB service
2. Restrict access to a MariaDB server to specific network addresses
3. Create a MariaDB database
4. Manage MariaDB database users and access rights
5. Add records to an existing MariaDB database
6. Issue simple SQL queries against a MariaDB database
7. Create a MariaDB backup
8. Import a MariaDB database from a backup

Install & Configure MariaDB Manually:

Install the MariaDB server packages.

[root@ansible-server ~]# dnf install -y mariadb-server

Start & enable the MariaDB service.

[root@ansible-server ~]# systemctl enable --now mariadb

Verify the service status.

[root@ansible-server ~]# systemctl status mariadb

Secure the database and set the root password.

[root@ansible-server ~]# mysql_secure_installation

Login into MariaDB from root account (use the root password which you set earlier).

[root@ansible-server ~]# mysql -u root -p

Check available databases:

MariaDB [(none)]> SHOW DATABASES;

Create a new database as nehraclasses.

MariaDB [(none)]> CREATE DATABASE nehraclasses;

Create & grant full access on this nehraclasses database to nehraclasses_user.

MariaDB [(none)]> GRANT ALL ON nehraclasses.* TO nehraclasses_user@localhost IDENTIFIED BY 'redhat';

Again, check available databases:

MariaDB [(none)]> SHOW DATABASES;

Exit from MariaDB prompt.

MariaDB [(none)]> quit;

Now, check if you're able to login into database from nehraclasses_user which you created earlier.

[root@ansible-server ~]# mysql -u nehraclasses_user -p

Check available databases:

MariaDB [(none)]> SHOW DATABASES;

Check for the errors.

MariaDB [(none)]> SHOW ERRORS;

Exit from MariaDB prompt.

MariaDB [(none)]> quit;

MariaDB configuration file location.

[root@ansible-server ~]# cat /etc/my.cnf

Configuration & Manage MariaDB Server Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Grant Access to a User from a Remote System in the MariaDB database:
In this section, we will create a new database named wpdb and a user named wpuser, and grant access to the remote system to connect to the database wpdb as user wpuser.

First, log in to the MariaDB shell with the following command:
 # mysql -u root -p

Provide your admin (root) password as shown in the Webdock backend and when you get the prompt create a database and user with the following command:
MariaDB [(none)]> CREATE DATABASE wpdb;
MariaDB [(none)]> CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'password';

Next, you will need to grant permissions to the remote system with IP address 192.168.229.129 to connect to the database named wpdb as user wpuser. You can do it with the following command:
MariaDB [(none)]> GRANT ALL ON wpdb.* to 'wpuser'@'192.168.229.129' IDENTIFIED BY 'password' WITH GRANT OPTION;

Next, flush the privileges and exit from the MariaDB shell with the following command:
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> EXIT;

A brief explanation of each parameter is shown below:
wpdb: It is the name of the MariaDB database that the user wants to connect to.
wpuser: It is the name of the MariaDB database user.
192.168.229.129: It is the IP address of the remote system from which the user wants to connect.
password: It is the password of the database user.

If you want to grant remote access on all databases for wpuser, run the following command:
MariaDB [(none)]> GRANT ALL ON *.* to 'wpuser'@'192.168.229.129' IDENTIFIED BY 'password' WITH GRANT OPTION;

If you want to grant access to all remote IP addresses on wpdb as wpuser, use % instead of IP address (192.168.229.129) as shown below:
MariaDB [(none)]> GRANT ALL ON wpdb.* to 'wpuser'@'%' IDENTIFIED BY 'password' WITH GRANT OPTION;

If you want to grant access to all IP addresses in the subnet 192.168.229.0/24 on wpdb as user wpuser, run the following command:
MariaDB [(none)]> GRANT ALL ON wpdb.* to 'wpuser'@'192.168.229.%' IDENTIFIED BY 'password' WITH GRANT OPTION;

Configure the Firewall:
Allow traffic from TCP port 3306 in the firewall.
[root@ansible-server ~]# firewall-cmd --permanent --add-port=3306/tcp
[root@ansible-server ~]# firewall-cmd --reload
[root@ansible-server ~]# firewall-cmd --list-ports

Test Connection from Remote System:
At this point, the MariaDB server is configured to allow connection from the remote system with IP address 192.168.229.0/24. Now, it's time to test the connection from the client system to the MariaDB server.
[root@ansible-server ~]# ssh node1

First, you will need to install the MariaDB Client package in the remote system. You can install it with the following command:
[root@node1 ~]# dnf install -y mariadb

Once the installation is completed, connect to the MariaDB server by running the following command on the remote system:
[root@node1 ~]# mysql -u wpuser -h 192.168.229.128 -p

You can now list the databases using the following command:

Configuration & Manage MariaDB Server Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

MariaDB [(none)]> show databases;
MariaDB [(none)]> quit;

If you will try to login from root user remotely, server will not allow you to login since we have disabled this option while securing the database.
[root@node1 ~]# mysql -u root -h 192.168.229.128 -p
[root@node1 ~]# exit

Backup and Restore MariaDB Database:
Login into database and identify/create the database you want to backup.

To create a table in MariaDB, first you will need to open a terminal interface and log in to the MariaDB shell. You can do it using the following command.
[root@ansible-server ~]# mysql -u root -p

Next, create a database named books using the CREATE DATABASE statement:
MariaDB [(none)]> CREATE DATABASE books;

Please verify your database using the following command.
MariaDB [(none)]> SHOW DATABASES;

Next, switch the database to books and create a table named books using the CREATE TABLE statement.
MariaDB [(none)]> use books;
MariaDB [books]> CREATE TABLE books(title VARCHAR(50) NOT NULL,author VARCHAR(30) NOT NULL,published_year INT NOT NULL,PRIMARY KEY(title));

Print the table structure using the following command.
MariaDB [books]> DESCRIBE books;

Following that is to verify the created table using the following command.
MariaDB [information_schema]> SHOW TABLES;

You can use the INSERT statement to insert the data in the first row of the table.
MariaDB [books]> INSERT INTO books VALUE ("Ulysses", "James Joyse", 1923);

Next, repeat the same command to insert the remaining row of table.
MariaDB [books]> INSERT INTO books VALUE ("Catch-22", "Joseph Heller", 1990);
MariaDB [books]> INSERT INTO books VALUE ("Robinson Crusoe", "Daniel Defoe", 1719);
MariaDB [books]> INSERT INTO books VALUE ("Tom Jones", "Henry Fielding", 1749);
MariaDB [books]> INSERT INTO books VALUE ("Emma", "Jane Austen", 1816);

Now, verify all inserted data using the following statement.
MariaDB [books]> SELECT * FROM books;
MariaDB [books]> quit;

Create a backup of any database using below mysqldump command from the shell prompt.
[root@ansible-server ~]# mysqldump -u root -p books > books.sql

Verify the existence of the file you have created using the mysqldump command.
[root@ansible-server ~]# ls -lh

Check and verify the size of the file if you want.
[root@ansible-server ~]# du -sh books.sql

Now login into database using root account and create a new database to import the data from this database file.
[root@ansible-server ~]# mysql -u root -p
MariaDB [(none)]> create database newdb;

Verify the presence of the newdb database.

Configuration & Manage MariaDB Server Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

MariaDB [(none)]> show databases;

Exit from the MariaDB
MariaDB [(none)]> quit;

Now, import the database from the dump file into this newly created database newdb.
[root@ansible-server ~]# mysql -u root -p newdb < books.sql

Login into MariaDB to verify the changes.
[root@ansible-server ~]# mysql -u root -p
MariaDB [(none)]> show databases;
MariaDB [(none)]> use newdb;
MariaDB [newdb]> show tables;
MariaDB [newdb]> SELECT * FROM books;
MariaDB [newdb]> quit;

Configure & Manage MariaDB Using Ansible:
Let's create an ansible playbook to configure the MariaDB server at the managed node(s) say node1.
[root@ansible-server ~]$ vim mariadb-server.yml

```
---
- name: MariaDB
  hosts: node1
  gather_facts: true
  become: true
  tasks:
    - name: install mariadb
      yum:
        name:
          - mariadb-server
          - python3-PyMySQL
        state: latest
    - name: start mariadb
      service:
        name: mariadb
        enabled: true
        state: started
  ...
```

Execute the ansible playbook, it will show an error because of missing mysql_user module.
[vikasnehra@ansible-server ~]$ ansible-playbook mariadb-server.yml

Install the communit.mysql collection to fix the issue.
[vikasnehra@ansible-server ~]$ ansible-galaxy collection install community.mysql

Now, we can execute the playbook.
[vikasnehra@ansible-server ~]$ ansible-playbook mariadb-server.yml

Login into the node1 machine or use ansible ad-hoc commands to secure the mysql installation and set the root password to login into the database.
[vikasnehra@ansible-server ~]$ ssh node1
[vikasnehra@ansible-server ~]$ sudo su -
[root@node1 ~]# mysql_secure_installation
[root@node1 ~]# exit
[root@node1 ~]# exit

Configuration & Manage MariaDB Server Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Reset the MariaDB database root password and deny root login remotely.

[vikasnehra@ansible-server ~]$ vim mariadb-server2.yml

```
---
- name: MariaDB
  hosts: node1
  gather_facts: true
  become: true
  vars:
    mysql_root_password: "P@$$w0rd"
  tasks:
    - name: install mariadb
      yum:
        name:
          - mariadb-server
          - python3-PyMySQL
        state: latest
    - name: start mariadb
      service:
        name: mariadb
        enabled: true
        state: started
    - name: mysql_root_password
      mysql_user:
        login_user: root
        login_password: redhat
        user: root
        check_implicit_admin: true
        password: "{{ mysql_root_password }}"
        host: localhost
    - name: remove remote root
      mysql_user:
        check_implicit_admin: true
        login_user: root
        login_password: "{{ mysql_root_password }}"
        user: root
        host: "{{ ansible_fqdn }}"
        state: absent
...
```

Now, we can execute the playbook.

[root@ansible-server ~]$ ansible-playbook mariadb-server2.yml

Create a new database using ansible playbook.

[vikasnehra@ansible-server ~]$ vim mariadb-db.yml

```
---
- name: MariaDB
  hosts: node1
  gather_facts: true
  become: true
  vars:
    mysql_root_password: "P@$$w0rd"
  tasks:
    - name: Create a new database with name 'bobdata'
      mysql_db:
        name: bobdata
        state: present
        login_user: root
        login_password: "{{ mysql_root_password }}"
...
```

Configuration & Manage MariaDB Server Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Execute the ansible playbook.

```
[vikasnehra@ansible-server ~]$ ansible-playbook mariadb-db.yml
```

Verify the same by going on the shell promt of the node1 machine or using ansible ad-hoc command.

```
[vikasnehra@ansible-server ~]$ ssh node1
[root@node1 ~]# mysql -u root -p -e "show databases" | grep bob
```

Create few more databases.

```
[vikasnehra@ansible-server ~]$ vim mariadb-db2.yml
---
- name: MariaDB
  hosts: node1
  gather_facts: true
  become: true
  vars:
    mysql_root_password: "P@$$w0rd"
  tasks:
     - name: Create new databases with names 'foo' and 'bar'
       mysql_db:
         name:
           - foo
           - bar
         state: present
         login_user: root
         login_password: "{{ mysql_root_password }}"
...
```

Execute the ansible playbook.

```
[vikasnehra@ansible-server ~]$ ansible-playbook mariadb-db2.yml
```

Verify the same from node1.

```
[vikasnehra@node1 ~]$ mysql -u root -e "show databases" -p
```

Copy database dump file to remote host.

```
[vikasnehra@ansible-server ~]$ vim mariadb-db3.yml
---
- name: MariaDB
  hosts: node1
  gather_facts: true
  become: true
  vars:
    mysql_root_password: "P@$$w0rd"
  tasks:
      - name: Dump database
        mysql_db:
           name: foo
           state: dump
           target: /tmp/foo.sql
           login_user: root
           login_password: "{{ mysql_root_password }}"
...
```

Execute the ansible playbook.

```
[vikasnehra@ansible-server ~]$ ansible-playbook mariadb-db3.yml
```

Verify the same.

```
[vikasnehra@ansible-server ~]$ ansible node1 -m command -a 'ls -lh /tmp/foo.sql'
```

Restore the database from dump.

```
[vikasnehra@ansible-server ~]$ vim mariadb-db4.yml
```

Configuration & Manage MariaDB Server Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
---
- name: MariaDB
  hosts: node1
  gather_facts: true
  become: true
  vars:
    mysql_root_password: "P@$$w0rd"
  tasks:
      - name: Restore database
        mysql_db:
          name: my_db
          state: import
          target: /tmp/foo.sql
          login_user: root
          login_password: "{{ mysql_root_password }}"
...
```

Execute the ansible playbook.

```
[vikasnehra@ansible-server ~]$ ansible-playbook mariadb-db4.yml
```

Verify the same from node1.

```
[vikasnehra@node1 ~]$ mysql -u root -e "show databases" -p
```

Remove the database(s).

```
[vikasnehra@ansible-server ~]$ vim mariadb-db5.yml
---
- name: MariaDB
  hosts: node1
  gather_facts: true
  become: true
  vars:
    mysql_root_password: "P@$$w0rd"
  tasks:
      - name: Make sure there is neither a database with name 'foo', nor one with name 'bar'
        mysql_db:
          name:
           - foo
           - bar
          state: absent
          login_user: root
          login_password: "{{ mysql_root_password }}"
...
```

Execute the ansible playbook.

```
[vikasnehra@ansible-server ~]$ ansible-playbook mariadb-db5.yml
```

Verify the tasks from node1.

```
[vikasnehra@node1 ~]$ mysql -u root -e "show databases" -p
```

Thank You