MySQL Logical Backup

Introduction

The mysqldump client utility performs logical backups, producing a set of SQL statements that can be executed to reproduce the original database object definitions and table data.

MySQL Shell is an advanced client and code editor for MySQL. In addition to the provided SQL functionality, similar to mysql, MySQL Shell provides scripting capabilities for JavaScript and Python and includes APIs for working with MySQL.

In this lab you will see how mysqldump and Mysql Shell dump & load, and compare both.

Estimated Lab Time: 15 minutes

Objectives

In this lab, you will:

- explore mysqldump
- explore MySQL Shell dump&load

> **Note:**

- Server: mysql1

Task 1: mysqldump

If not already connected to app-srv and mysql1 then do the following

- a. Connect with your SSH client using the public IP and the provided ssh Example of connections from Linux, MAC, Windows Powershell

  ```
  <span style="color:green">shell></span> <copy> ssh -i id_rsa_app-srv
  opc@<public_ip></copy>
  ```

- b. Connect to shell-mysql1

  ```
  <span style="color:green">shell-app-srv$</span> <copy> ssh -i
  $HOME/sshkeys/id_rsa_mysql1 opc@mysql1 </copy>
  ```

1. Create the export folder

   ```
   <span style="color:green">shell></span> <copy>sudo mkdir -p
   /mysql/exports</copy>
   ```

```
<span style="color:green">shell></span> <copy>sudo chown mysqluser:mysqlgrp /mysql/exports/</copy>
```

```
<span style="color:green">shell></span> <copy>sudo chmod 770 /mysql/exports/</copy>
```

2. Export all the data with mysqldump

```
<span style="color:green">shell></span> <copy>mysqldump -uadmin -p -hmysql1 -P3307 --single-transaction --events --routines --flush-logs --all-databases > /mysql/exports/full.sql</copy>
```

3. View the content of the file /mysql/exports/full.

```
<span style="color:green">shell></span> <copy>nano /mysql/exports/full.sql</copy>
```

4. Export employees database

```
<span style="color:green">shell></span> <copy>mysqldump -uadmin -p -hmysql1 -P3307 --single-transaction --set-gtid-purged=OFF --databases employees > /mysql/exports/employees.sql</copy>
```

```
<span style="color:green">shell></span> <copy>ls -l /mysql/exports/employees.sql</copy>
```

5. Drop employees database, using the mysql client

```
<span style="color:green">shell></span> <copy>mysql -uadmin -p -hmysql1 -P3307</copy>
```

```
<span style="color:blue">mysql></span> <copy>show databases;</copy>
```

```
<span style="color:blue">mysql></span> <copy>DROP DATABASE employees;</copy>
```

```
<span style="color:blue">mysql></span> <copy>show databases;</copy>
```

6. Import the employees database. It can be done in shell (as when we loaded first example data) or from within the mysql client.

```
<span style="color:blue">mysql></span> <copy>SOURCE
/mysql/exports/employees.sql</copy>
```

```
<span style="color:blue">mysql></span> <copy>show databases;</copy>
```

```
<span style="color:blue">mysql></span> <copy>show tables in employees;
</copy>
```

7. Exit from mysql client

```
<span style="color:blue">mysql></span> <copy>exit</copy>
```

Task 2: MySQL Shell

1. Connect with MySQL Shell

```
<span style="color:green">shell></span> <copy>mysqlsh
admin@mysql1:3307</copy>
```

2. Export employees database using javascript command mode.
   Please **note the time required** with the default 4 threads.

```
<span style="color:blue">My</span><span style="color: orange">SQL </span>
<span style="background-color:orange">SQL</span>><copy>\js</copy>
```

```
<span style="color:blue">My</span><span style="color: orange">SQL </span>
<span style="background-color:yellow">JS</span>>
<copy>util.dumpSchemas(['employees'],'/mysql/exports/employees')</copy>
```

3. Check the content of the directory /mysql/exports/employees

```
<span style="color:blue">My</span><span style="color: orange">SQL </span>
<span style="background-color:yellow">JS</span>><copy>\q</copy>
```

```
<span style="color:green">shell></span> <copy>ls -l
/mysql/exports/employees</copy>
```

4. Reconnect mysqlsh and drop employees database

```
<span style="color:green">shell></span> <copy>mysqlsh
admin@mysql1:3307</copy>
```

```
<span style="color:blue">My</span><span style="color: orange">SQL </span>
<span style="background-color:orange">SQL</span>><copy> drop database
employees;</copy>
```

```
<span style="color:blue">My</span><span style="color: orange">SQL </span>
<span style="background-color:orange">SQL</span>><copy> show databases;
</copy>
```

5. Load files is disabled by default, so enable it now to let MysQL Shell execute the load

```
<span style="color:blue">My</span><span style="color: orange">SQL </span>
<span style="background-color:orange">SQL</span>><copy> set persist
local_infile=ON;</copy>
```

6. Switch now in javascript command mode and re import the employees database.
   Please **note the time required** with the default 4 threads.

```
<span style="color:blue">My</span><span style="color: orange">SQL </span>
<span style="background-color:orange">SQL</span>><copy> \js</copy>
```

```
<span style="color:blue">My</span><span style="color: orange">SQL </span>
<span style="background-color:yellow">JS</span>>
<copy>util.loadDump('/mysql/exports/employees')</copy>
```

```
<span style="color:blue">My</span><span style="color: orange">SQL </span>
<span style="background-color:yellow">JS</span>><copy>\sql show databases;
</copy>
```

```
<span style="color:blue">My</span><span style="color: orange">SQL </span>
<span style="background-color:yellow">JS</span>><copy>\q</copy>
```

7. We can also calculate the time with mysqldump. To use mysqldump in a more secure way we store the credential in login-path file (discussed in the module dedicated to security)

    - Let's create the login-path

```
<span style="color:green">shell></span> <copy>mysql_config_editor set --
login-path=mysql1 --host=mysql1 --port=3307 --user=admin --password</copy>
```

    - Let's test the login-path

```
<span style="color:green">shell></span> <copy>mysql --login-path=mysql1 -e
"SHOW DATABASES"</copy>
```

8. Now we can test the export.
   Please **compare the mysqldump time** with the MysQL Shell dump.

```
<span style="color:green">shell></span> <copy>time mysqldump --login-
path=mysql1 --single-transaction --set-gtid-purged=OFF --databases employees
> /mysql/exports/employees_time_test.sql</copy>
```

9. And now drop the database employees.

```
<span style="color:green">shell></span> <copy>mysql --login-path=mysql1 -e
"drop database employees"</copy>
```

```
<span style="color:green">shell></span> <copy>mysql --login-path=mysql1 -e
"show databases"</copy>
```

10. And we can check the import time.
    Please **compare the mysql import time** with the MysQL load.

```
<span style="color:blue">mysql></span> <copy>time mysql --login-path=mysql1
< /mysql/exports/employees_time_test.sql</copy>
```

```
<span style="color:green">shell></span> <copy>mysql --login-path=mysql1 -e
"show databases"</copy>
```

Learn More

- https://dev.mysql.com/doc/refman/8.4/en/mysqldump.html
- To use MySQL Shell at command line read: https://dev.mysql.com/doc/mysql-shell/8.4/en/command-line-integration-overview.html
- https://dev.mysql.com/doc/mysql-shell/8.4/en/mysql-shell-utilities-dump-instance-schema.html
- https://dev.mysql.com/doc/mysql-shell/8.4/en/mysql-shell-utilities-load-dump.html

Acknowledgements

- **Author** - Marco Carlessi, Principal Sales Consultant
- **Contributors** - Perside Foster, Principal Sales Consultant, Selena Sánchez, MySQL Solutions Engineer
- **Last Updated By/Date** - Perside Foster, Partner Solutions Engineer, November 2024