Installation of MySQL Enterprise

Introduction

In this lab you will install MySQL 8 Enterprise on Linux

Estimated Time: 40 minutes

Objectives

In this lab, you will be guided trough the following tasks:

- Tarball installation of MySQL 8 Enterprise on Linux

> **Note:** Server: mysql1

- we call this instance mysql-advanced
- this is the most used instance in the labs

Task 1: Install MySQL Enterprise

1. If not already connected, connect to mysql1 server thought app-srv

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

2. Verify that you are connected to right server (**mysql1**).
If you are still connected to app-srv, please repeat previous step and recheck.

```
<span style="color:green">shell-mysql1></span> <copy>hostname</copy>
```

3. On Oracle Linux 8, RHEL 8, or CentOS 8, the ncurses-compat-libs package is required when installing MySQL Enterprise Edition using the tar package. This requirement does not apply when using RPM packages for installation.

```
<span style="color:green">shell-mysql1></span> <copy>sudo yum install -y
ncurses-compat-libs </copy>
```

4. Usually, to run mysql the user “mysql” is used, but because it is already available we create a new one.

Create a new user/group for your MySQL service (mysqluser/mysqlgrp) and a add ‘mysqlgrp’ group to opc to help with labs execution

```
<span style="color:green">shell-mysql1></span> <copy>sudo groupadd mysqlgrp
</copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo useradd -r -g
mysqlgrp -s /bin/false mysqluser </copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo usermod -a -G
mysqlgrp opc </copy>
```

5. To simplify the lab, add the mysql bin folder to the bash profile and customize the client prompt.

    1. Please insert these lines at the end of the file /home/opc/.bashrc You can edit the file with the editor that you prefer, here some examples:

        ```
        <span style="color:green">shell-mysql1></span> <copy>nano
        /home/opc/.bashrc </copy>
        ```

        ```
        <span style="color:green">shell-mysql1></span> <copy>vi
        /home/opc/.bashrc </copy>
        ```

    2. Add these lines at the end, then save and exit

        ```
        <copy>export PATH=$PATH:/mysql/mysql-latest/bin</copy>
        ```

        ```
        <copy>export MYSQL_PS1="\\u on \\h>\\_" </copy>
        ```

6. Close the ssh session and reopen it to activate the new privilege and settings for opc user

    ```
    <span style="color:green">shell-mysql1></span> <copy>exit</copy>
    ```

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

7. Create new directory structure:

```
<span style="color:green">shell-mysql1></span> <copy>sudo mkdir /mysql/ /mysql/etc /mysql/data /mysql/log /mysql/temp /mysql/binlog</copy>
```

8. We have the binary files inside our /workshop directory in a compressed tar

```
<span style="color:green">shell-mysql1></span> <copy>ls -l /workshop/linux/mysql-commercial-8.*-x86_64.tar.xz</copy>
```

9. Extract the tarball in your /mysql folder

```
<span style="color:green">shell-mysql1></span> <copy>cd /mysql/ </copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo tar xvf /workshop/linux/mysql-commercial-8.*-x86_64.tar.xz</copy>
```

10. Create a symbolic link to mysql binary installation

```
<span style="color:green">shell-mysql1></span> <copy>sudo ln -s mysql-commercial-8.*-x86_64 mysql-latest </copy>
```

11. Create a new configuration file my.cnf inside /mysql/etc To help you we created one with some variables, please copy it

```
<span style="color:green">shell-mysql1></span> <copy>sudo cp /workshop/support/my.cnf.mysql1 /mysql/etc/my.cnf </copy>
```

12. Check the content of the configuration file to have a look inside. Please note that, because the port 3306 is already in use by the community server previously installed , we use now port 3307.

```
<span style="color:green">shell-mysql1></span> <copy>cat /mysql/etc/my.cnf </copy>
```

13. For security reasons change ownership and permissions

```
<span style="color:green">shell-mysql1></span> <copy>sudo chown -R mysqluser:mysqlgrp /mysql </copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo chmod -R 750 /mysql </copy>
```

The following permission is for the Lab purpose so that opc account can make changes and copy files to overwrite the content `<span style="color:green">shell-mysql1></span> <copy>sudo chmod -R 770 /mysql/etc </copy>`

14. Save the changes.

```
<span style="color:green">shell-mysql1></span> <copy>sudo /mysql/mysql-latest/bin/mysqld --defaults-file=/mysql/etc/my.cnf --initialize --user=mysqluser </copy>
```

15. Log out.

```
<span style="color:green">shell-mysql1></span> <copy>exit</copy>
```

16. Log in again from the ssh for the changes to take effect on the user profile.

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

17. Start your new mysql instance

```
<span style="color:green">shell-mysql1></span> <copy>sudo /mysql/mysql-latest/bin/mysqld --defaults-file=/mysql/etc/my.cnf --user=mysqluser & </copy>
```

18. Verify that process is running

```
<span style="color:green">shell-mysql1></span> <copy>ps -ef | grep mysqld </copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>netstat -an | grep 3307 </copy>
```

19. Another way is searching the message “ready for connections” in error log as one of the last

```
<span style="color:green">shell-mysql1></span> <copy>grep -i ready /mysql/log/err_log.log </copy>
```

20. Retrieve root password for first login

    ```
    <span style="color:green">shell-mysql1></span> <copy>grep -i 'temporary password' /mysql/log/err_log.log</copy>
    ```

21. Login to your mysql-advanced using the temporary password and set the new one

    ```
    <span style="color:green">shell-mysql1></span> <copy>mysql -uroot -p -h 127.0.0.1 -P3307</copy>
    ```

    ```
    <span style="color:blue">mysql></span> <copy>SET PASSWORD='Welcome1!';
    </copy>
    ```

22. Check server status.

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

23. Shutdown the service

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

    - Using **-h 127.0.0.1** instead of **-hlocalhost** bypasses hostname resolution and connects directly via IP address
    - This direct IP approach is particularly useful when troubleshooting connection issues or when configuring precise network settings.

    ```
    <span style="color:green">shell-mysql1></span> <copy>mysqladmin -uroot -h127.0.0.1 -p -P3307 shutdown</copy>
    ```

24. Configure automatic startup and shutdown with system, adding a systemd service unit configuration file with details about the MySQL service.

    The file is named mysqld.service and is placed in /usr/lib/systemd/system. We created one for you (See addendum for the content)

```
<span style="color:green">shell-mysql1></span> <copy>sudo cp /workshop/support/mysqld-advanced.service /usr/lib/systemd/system/</copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo chmod 644 /usr/lib/systemd/system/mysqld-advanced.service</copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo systemctl enable mysqld-advanced.service</copy>
```

25. Test start, stop and restart

```
<span style="color:green">shell-mysql1></span> <copy>sudo systemctl start mysqld-advanced</copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo systemctl status mysqld-advanced</copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo systemctl stop mysqld-advanced</copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo systemctl status mysqld-advanced</copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo systemctl restart mysqld-advanced</copy>
```

```
<span style="color:green">shell-mysql1></span> <copy>sudo systemctl status mysqld-advanced</copy>
```

26. Create a new administrative user called 'admin' with remote access and full privileges

```
<span style="color:green">shell-mysql1></span> <copy>mysql -uroot -p -h 127.0.0.1 -P3307</copy>
```

```
<span style="color:blue">mysql></span> <copy>CREATE USER 'admin'@'%' IDENTIFIED BY 'Welcome1!';</copy>
```

```
<span style="color:blue">mysql></span> <copy>GRANT ALL PRIVILEGES ON *.* TO 'admin'@'%' WITH GRANT OPTION;</copy>
```

27. In the configuration file was specified to load the commercial Thread Pool Plugin, now let's check if it’s loaded and active.
    We can use different commands, please note that the last two are the same query with a different line terminator (“;” vs “\G” show output in a different format)

```
<span style="color:blue">mysql></span> <copy>SHOW PLUGINS;</copy>
```

```
<span style="color:blue">mysql></span> <copy>SELECT * FROM information_schema.plugins WHERE plugin_name LIKE 'thread%';</copy>
```

```
<span style="color:blue">mysql></span> <copy>SELECT * FROM information_schema.plugins WHERE plugin_name LIKE 'thread%'\G</copy>
```

28. Exit MySQL

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

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, March 2025