DEPLOYING - InnoDB ClusterSets

Introduction

InnoDB ClusterSets: Objective: deploying MySQL sandboxes and then creating an InnoDB ClusterSets

*This lab walks you through creating MySQL Sandboxes, deploying InnoDB ClusterSets, bootstrapping MySQL Router and testing failovers

Estimated Time: 15 minutes

Objectives

In this lab, you will do the followings:

- Connect to MySQL Shell
- Create MySQL Sandboxes
- Create InnoDB ClusterSets

Prerequisites

This lab assumes you have:

- An Oracle account
- All previous labs successfully completed

MySQL Instances ports

- "Portland": 3310, 3320, 3330
- "Seattle": 3410, 3420, 3430
- Lab standard
    - 🟩 shell> the command must be executed in the Operating System shell
    - 🟦 mysql> the command must be executed in a client like MySQL, MySQL Workbench
    - 🟧 mysqlsh> the command must be executed in MySQL shell

Notes:

- Open a notepad file and your linux Private IP on student###-serverA
- serverA PRIVATE ip: (client_ip)

Task 1: Connect to mysql-enterprise on Server

1. Connect to your MySQL Shell

    🟩 **shell>**

    ```
    ```

```
<copy>mysqlsh</copy>
```

2. Create 3 Additional MySQL Sandboxes

   a. 🟧 **mysqlsh>**

```
<copy>dba.deploySandboxInstance(3410, {password: "password"})
dba.deploySandboxInstance(3420, {password: "password"})
dba.deploySandboxInstance(3430, {password: "password"})</copy>
```

Task 2: Create an InnoDB ClusterSet

1. Using the MySQL Shell Connection, connect the Shell to Sandbox on Port 3310 and create ClusterSet starting with 3410 Instance

   a. 🟧 **mysqlsh>**

```
<copy>\connect root@localhost:3310</copy>
```

   b. 🟧 **mysqlsh>**

```
<copy>var PortlandCluster = dba.getCluster()</copy>
```

   c. 🟧 **mysqlsh>**

```
<copy>PortlandCluster.status()</copy>
```

   d. 🟧 **mysqlsh>**

```
<copy>NWClusterSet = PortlandCluster.createClusterSet("NWCluster")
</copy>
```

   e. 🟧 **mysqlsh>**

```
<copy>NWClusterSet.status()</copy>
```

   f. 🟧 **mysqlsh>**

```
<copy>SeattleCluster =
NWClusterSet.createReplicaCluster("127.0.0.1:3410","SeattleCluster")
</copy>
```

g. 🟧 **mysqlsh>**

```
<copy>SeattleCluster.status()</copy>
```

h. 🟧 **mysqlsh>**

```
<copy>NWClusterSet.status()</copy>
```

2. Add 2 instances to Secondary (Replica) InnoDB Cluster

a. 🟧 **mysqlsh>**

```
<copy>SeattleCluster.addInstance('root@localhost:3420')</copy>
```

b. 🟧 **mysqlsh>**

```
<copy>SeattleCluster.addInstance('root@localhost:3430')</copy>
```

c. 🟧 **mysqlsh>**

```
<copy>SeattleCluster.status()</copy>
```

d. 🟧 **mysqlsh>**

```
<copy>\connect root@localhost:3410</copy>
```

e. 🟧 **mysqlsh>**

```
<copy>\sql</copy>
```

f. 🟧 **mysqlsh>**

```
<copy>SHOW DATABASES;</copy>
```

g. 🟧 **mysqlsh>**

```
<copy>USE world;</copy>
```

h. 🟧 **mysqlsh>**

```
<copy>SHOW TABLES;</copy>
```

i. 🟧 **mysqlsh>**

```
<copy>\js</copy>
```

j. 🟧 **mysqlsh>**

```
<copy>\connect root@localhost:3310</copy>
```

Task 3: Test failovers

1. Test changing the Primary. This is good for instances where you want to safely failover to a new Replica

   a. Failover to 3320 instance

   🟧 **mysqlsh>**

   ```
   <copy>PortlandCluster.setPrimaryInstance("root@localhost:3320")</copy>
   ```

   b. 🟧 **mysqlsh>**

   ```
   <copy>\connect root@localhost:3320</copy>
   ```

   c. Check status

   🟧 **mysqlsh>**

   ```
   <copy>PortlandCluster.status()</copy>
   ```

d. 🟧 **mysqlsh>**

```
<copy>NWClusterSet = dba.getClusterSet()</copy>
```

e. 🟧 **mysqlsh>**

```
<copy>NWClusterSet.status()</copy>
```

f. Failover back to 3310 instance

🟧 **mysqlsh>**

```
<copy>PortlandCluster.setPrimaryInstance("root@localhost:3310")</copy>
```

g. 🟧 **mysqlsh>**

```
<copy>\connect root@localhost:3310</copy>
```

h. Check status (**Note** You can see extended details by passing the {extended: [1|2} })

🟧 **mysqlsh>**

```
<copy>PortlandCluster.status()</copy>
```

i. 🟧 **mysqlsh>**

```
<copy>NWClusterSet = dba.getClusterSet()</copy>
```

j. 🟧 **mysqlsh>**

```
<copy>NWClusterSet.status()</copy>
```

Task 4: Deploy MySQL Router

1. Create a new SSH Shell window to your Compute Instance and create a directory for MySQL Router configuration and data

🟩 **shell>**

```
<copy>cd ~/mysqlrouter</copy>
```

2. Bootstrap MySQL Router and Deploy Router against 3310 Instance (Which is now the Source)

🟩 **shell>**

```
<copy>mysqlrouter --bootstrap root@localhost:3310 -d
/home/opc/mysqlrouter --name='Portland'</copy>
```

🟩 **shell>**

```
<copy>./start.sh &</copy>
```

🟩 **shell>**

```
<copy>ps -ef | grep mysqlrouter</copy>
```

🟩 **shell>**

```
<copy>mysql -P6446 --protocol=tcp -uroot -ppassword</copy>
```

🟦 **mysql>**

```
<copy>SELECT @@port;</copy>
```

🟧 **mysqlsh>**

```
<copy>NWClusterSet.listRouters()</copy>
```

🟧 **mysqlsh>**

```
<copy>NWClusterSet.routingOptions()</copy>
```

🟧 **mysqlsh>**

```
<copy>NWClusterSet.describe()</copy>
```

3. Failover the Source and check if the Router follows

   🟧 **mysqlsh>**

   ```
   <copy>PortlandCluster.setPrimaryInstance('root@localhost:3320')</copy>
   ```

   🟦 **mysql>**

   ```
   <copy>SELECT @@port;</copy>
   ```

   🟧 **mysqlsh>**

   ```
   <copy>PortlandCluster.setPrimaryInstance('root@localhost:3310')</copy>
   ```

4. Failover to the Replica Cluster

   🟧 **mysqlsh>**

   ```
   <copy>NWClusterSet.setPrimaryCluster('SeattleCluster')</copy>
   ```

   🟦 **mysql>**

   ```
   <copy>SELECT @@port;</copy>
   ```

   🟧 **mysqlsh>**

   ```
   <copy>\connect root@localhost:3410</copy>
   ```

   🟧 **mysqlsh>**

   ```
   <copy>NWClusterSet.status()</copy>
   ```

   🟦 **mysql>**

```
<copy>SELECT @@port;</copy>
```

5. Fail back to Portland Cluster

   🟧 **mysqlsh>**

   ```
   <copy>NWClusterSet.setPrimaryCluster('PortlandCluster')</copy>
   ```

   🟧 **mysqlsh>**

   ```
   <copy>\connect root@localhost:3310</copy>
   ```

   🟧 **mysqlsh>**

   ```
   <copy>NWClusterSet.status()</copy>
   ```

   🟦 **mysql>**

   ```
   <copy>SELECT @@port;</copy>
   ```

Learn More

- CREATE USER
- MySQL Access Control Lists

Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering