MySQL InnoDB HA Cluster 8.0

Date: April 2023

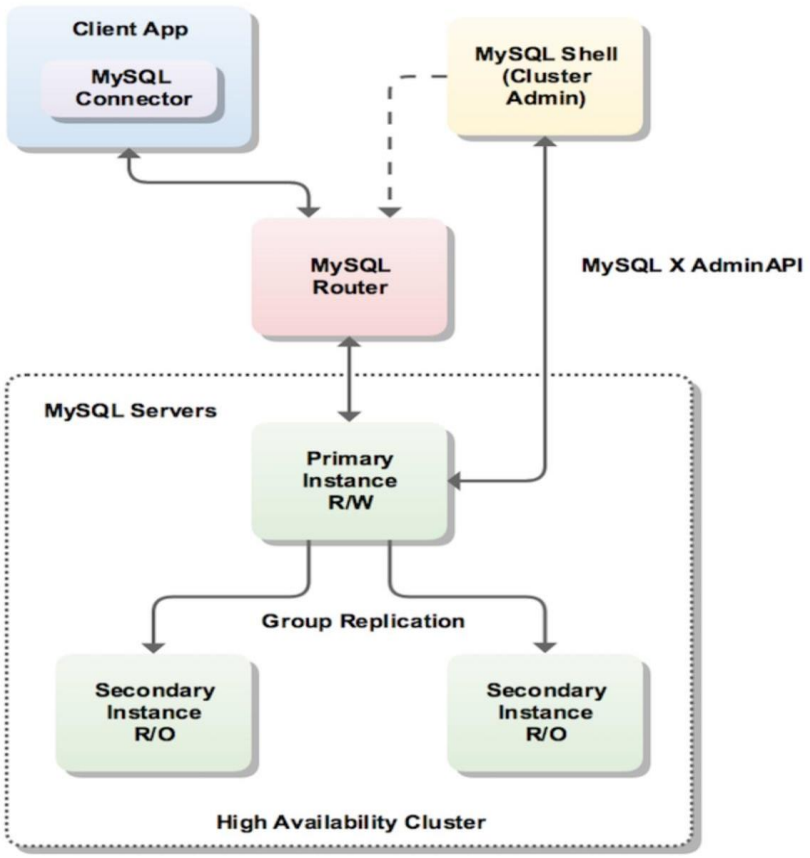
Authors: Mukesh Sharma

Setup INNODB cluster on CENTOS8

Requirement

- **Three mysql cluster nodes and at least one mysql router node**
- **Each node will have the domains of the all four servers in /etc/hosts file – innodb1, innodb2, indodb3,haproxy(mysql router node).**
- **CentOS 8 firewall tuning to unblock the nodes traffic between them.**
- **Disable mysql package system module to use the official MySQL repository.**
- **Three MySQL 8.0.28 server nodes will be installed**
- **To create and manage the cluster MySQL Shell 8.0 and dba object in it are used**
- **SELinux tuning to allow MySQL process to connect the network.**
- **Group replication with one primary (i.e. master) and two secondary nodes (i.e. slaves)**

Below a typical InnoDB cluster setup:



Step 1 : install MySQL 8 server (One on server Node: example innodb1)

a. Download the RPM repository file from MySQL web site – https://dev.mysql.com/downloads/repo/yum/
b. Install mysql80-community
c. Disable default mysql
d. Install mysql-community-server

```
[root@innodb1 ~]# wget https://repo.mysql.com/mysql80-community-release-el8-3.noarch.rpm
--2023-04-03 15:58:08--  https://repo.mysql.com/mysql80-community-release-el8-3.noarch.rpm
Resolving repo.mysql.com (repo.mysql.com)... 104.123.77.43
Connecting to repo.mysql.com (repo.mysql.com)|104.123.77.43|:443... connected.
HTTP request sent, awaiting response... 200 OK
```

```
Length: 14100 (14K) [application/x-redhat-package-manager]

Saving to: ‘mysql80-community-release-el8-3.noarch.rpm’

mysql80-community-release-el8-3.noarch.rpm
100%[=======================================================================================
=============================================================>]  13.77K  --.-KB/s    in 0s

2023-04-03 15:58:08 (285 MB/s) - ‘mysql80-community-release-el8-3.noarch.rpm’ saved
[14100/14100]

[root@innodb1 ~]# dnf install -y ./mysql80-community-release-el8-3.noarch.rpm

Last metadata expiration check: 0:59:32 ago on Mon 03 Apr 2023 02:59:01 PM EDT.

Dependencies resolved.

=======================================================================================
=======================================================================================
=======================================================================

 Package                                                                 Architecture
Version                                                 Repository
Size

=======================================================================================
=======================================================================================
=======================================================================

Installing:

 mysql80-community-release                                               noarch
el8-3                                                   @commandline
14 k

Transaction Summary

=======================================================================================
=======================================================================================
=======================================================================

Install  1 Package

Total size: 14 k

Installed size: 7.5 k

Downloading Packages:

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

  Preparing        :
1/1

  Installing       : mysql80-community-release-el8-3.noarch
1/1

  Verifying        : mysql80-community-release-el8-3.noarch
1/1

Installed:
```

```
  Mysql80-community-release-el8-3.noarch

Complete!

[root@innodb1 ~]# dnf module disable mysql

MySQL 8.0 Community Server
4.7 MB/s | 2.7 MB     00:00

MySQL Connectors Community
299 kB/s |  88 kB     00:00

MySQL Tools Community
1.5 MB/s | 650 kB     00:00

Dependencies resolved.

================================================================================
================================================================================
================================================================================

 Package                                                    Architecture
Version                                                     Repository
Size

================================================================================
================================================================================
================================================================================

Disabling modules:

 Mysql

Transaction Summary

================================================================================
================================================================================
================================================================================

Is this ok [y/N]: y

Complete!

[root@innodb1 ~]# dnf install -y mysql-community-server

Last metadata expiration check: 0:00:12 ago on Mon 03 Apr 2023 04:00:59 PM EDT.

Dependencies resolved.

================================================================================
================================================================================
================================================================================

 Package                                                                Architecture
Version                                                 Repository
Size

================================================================================
================================================================================
================================================================================

Installing:

 mysql-community-server                                                 x86 64
8.0.32-1.el8                                            mysql80-community
64 M

     replacing  mariadb-connector-c-config.noarch 3.1.11-2.el8_3

Installing dependencies:
```

```
 mysql-community-client                                                  x86 64
8.0.32-1.el8                                        mysql80-community
16 M

 mysql-community-client-plugins                                          x86 64
8.0.32-1.el8                                        mysql80-community
2.5 M

 mysql-community-common                                                  x86 64
8.0.32-1.el8                                        mysql80-community
656 k

 mysql-community-icu-data-files                                          x86 64
8.0.32-1.el8                                        mysql80-community
2.1 M

 mysql-community-libs                                                    x86 64
8.0.32-1.el8                                        mysql80-community
1.5 M



Transaction Summary

================================================================================
================================================================================
================================================================================

Install  6 Packages



Total download size: 87 M

Downloading Packages:

(1/6): mysql-community-client-plugins-8.0.32-1.el8.x86 64.rpm
4.7 MB/s | 2.5 MB     00:00

(2/6): mysql-community-icu-data-files-8.0.32-1.el8.x86 64.rpm
6.7 MB/s | 2.1 MB     00:00

(3/6): mysql-community-common-8.0.32-1.el8.x86 64.rpm
633 kB/s | 656 kB     00:01

(4/6): mysql-community-libs-8.0.32-1.el8.x86 64.rpm
7.2 MB/s | 1.5 MB     00:00

(5/6): mysql-community-client-8.0.32-1.el8.x86 64.rpm
6.3 MB/s |  16 MB     00:02

(6/6): mysql-community-server-8.0.32-1.el8.x86 64.rpm
12 MB/s |  64 MB     00:05

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

Total
14 MB/s |  87 MB     00:06

MySQL 8.0 Community Server
3.0 MB/s | 3.1 kB     00:00

Importing GPG key 0x3A79BD29:

 Userid     : "MySQL Release Engineering <mysql-build@oss.oracle.com>"

 Fingerprint: 859B E8D7 C586 F538 430B 19C2 467B 942D 3A79 BD29

 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022
```

```
Key imported successfully

MySQL 8.0 Community Server
1.9 MB/s | 1.9 kB     00:00

Importing GPG key 0x5072E1F5:

 Userid     : "MySQL Release Engineering <mysql-build@oss.oracle.com>"

 Fingerprint: A4A9 4068 76FC BD3C 4567 70C8 8C71 8D3B 5072 E1F5

 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

Key imported successfully

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

  Preparing        :
1/1

  Installing       : mysql-community-common-8.0.32-1.el8.x86 64
1/7

  Installing       : mysql-community-client-plugins-8.0.32-1.el8.x86 64
2/7

  Installing       : mysql-community-libs-8.0.32-1.el8.x86 64
3/7

  Running scriptlet: mysql-community-libs-8.0.32-1.el8.x86 64
3/7

  Installing       : mysql-community-client-8.0.32-1.el8.x86 64
4/7

  Installing       : mysql-community-icu-data-files-8.0.32-1.el8.x86 64
5/7

  Running scriptlet: mysql-community-server-8.0.32-1.el8.x86 64
6/7

  Installing       : mysql-community-server-8.0.32-1.el8.x86 64
6/7

  Running scriptlet: mysql-community-server-8.0.32-1.el8.x86 64
6/7

  Obsoleting       : mariadb-connector-c-config-3.1.11-2.el8 3.noarch
7/7

  Running scriptlet: mariadb-connector-c-config-3.1.11-2.el8 3.noarch
7/7

  Verifying        : mysql-community-client-8.0.32-1.el8.x86 64
1/7

  Verifying        : mysql-community-client-plugins-8.0.32-1.el8.x86 64
2/7

  Verifying        : mysql-community-common-8.0.32-1.el8.x86 64
3/7
```

```
  Verifying        : mysql-community-icu-data-files-8.0.32-1.el8.x86_64
4/7

  Verifying        : mysql-community-libs-8.0.32-1.el8.x86_64
5/7

  Verifying        : mysql-community-server-8.0.32-1.el8.x86 64
6/7

  Verifying        : mariadb-connector-c-config-3.1.11-2.el8_3.noarch
7/7



Installed:

  mysql-community-client-8.0.32-1.el8.x86_64       mysql-community-client-plugins-8.0.32-
1.el8.x86 64       mysql-community-common-8.0.32-1.el8.x86 64       mysql-community-icu-data-
files-8.0.32-1.el8.x86_64       mysql-community-libs-8.0.32-1.el8.x86_64

  mysql-community-server-8.0.32-1.el8.x86_64



Complete!

[root@innodb1 ~]# systemctl start mysqld

[root@innodb1 ~]# systemctl enable mysqld

[root@innodb1 ~]# systemctl status mysqld
```

Step 2. Install Mysql shell (One on server Node: example innodb1)

```
root@innodb1 ~]# dnf install -y mysql-shell

Last metadata expiration check: 0:02:55 ago on Mon 03 Apr 2023 04:00:59 PM EDT.

Dependencies resolved.

================================================================================================
================================================================================================
==========================================================================

 Package                                                            Architecture
Version                                                                            Repository
Size

================================================================================================
================================================================================================
==========================================================================

Installing:

 mysql-shell                                                        x86 64
8.0.32-1.el8                                                                       mysql-tools-
community                                                 23 M

Installing dependencies:

 python39-libs                                                      x86 64
3.9.16-1.module el8.8.0+1243+5f5a1e61                                              appstream
8.2 M

 python39-pip-wheel                                                 noarch
20.2.4-7.module el8.7.0+1213+291b6551                                              appstream
1.1 M
```

```
 python39-setuptools-wheel                                                  noarch
50 3.2-4.module el8.6.0+930+10acc06f                                                  appstream
497 k

Installing weak dependencies:

 python39                                                                   x86 64
3.9 16-1.module el8.8.0+1243+5f5a1e61                                                 appstream
33 k

 python39-pip                                                               noarch
20 2.4-7.module el8.7.0+1213+291b6551                                                 appstream
1.9 M

127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4

 python39-setuptools                                                        noarch
50 3.2-4.module el8.6.0+930+10acc06f                                                  appstream
871 k

Enabling module streams:

 python39
3.9


Transaction Summary

================================================================================================================
================================================================================================================
================================================================================

Install  7 Packages


Total download size: 35 M

Installed size: 203 M

Downloading Packages:

(1/7): python39-3.9.16-1.module el8.8.0+1243+5f5a1e61.x86 64.rpm
67 kB/s |  33 kB     00:00

(2/7): python39-pip-20.2.4-7.module el8.7.0+1213+291b6551.noarch.rpm
1.2 MB/s | 1.9 MB     00:01

(3/7): python39-libs-3.9.16-1.module el8.8.0+1243+5f5a1e61.x86 64.rpm
5.1 MB/s | 8.2 MB     00:01

(4/7): python39-setuptools-50.3.2-4.module el8.6.0+930+10acc06f.noarch.rpm
1.0 MB/s | 871 kB     00:00

(5/7): mysql-shell-8.0.32-1.el8.x86 64.rpm
11 MB/s |  23 MB     00:02

(6/7): python39-setuptools-wheel-50.3.2-4.module el8.6.0+930+10acc06f.noarch.rpm
148 kB/s | 497 kB     00:03

(7/7): python39-pip-wheel-20.2.4-7.module el8.7.0+1213+291b6551.noarch.rpm
202 kB/s | 1.1 MB     00:05

----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------

Total
5.6 MB/s |  35 MB     00:06

CentOS Stream 8 - AppStream
1.3 MB/s | 1.6 kB     00:00

Importing GPG key 0x8483C65D:

 Userid     : "CentOS (CentOS Official Signing Key) <security@centos.org>"

 Fingerprint: 99DB 70FA E1D7 CE22 7FB6 4882 05B5 55B3 8483 C65D

# For advice on how to change settings please see

 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial

Key imported successfully

Running transaction check
```

```
Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

  1/1 Preparing        :

  1/7 Installing       : python39-setuptools-wheel-50.3.2-4.module el8.6.0+930+10acc06f.noarch

  2/7 Installing       : python39-pip-wheel-20.2.4-7.module el8.7.0+1213+291b6551.noarch

  3/7 Installing       : python39-libs-3.9.16-1.module el8.8.0+1243+5f5a1e61.x86 64

  4/7 Installing       : python39-3.9.16-1.module el8.8.0+1243+5f5a1e61.x86 64

  4/7 Running scriptlet: python39-3.9.16-1.module el8.8.0+1243+5f5a1e61.x86 64

  5/7 Installing       : python39-setuptools-50.3.2-4.module el8.6.0+930+10acc06f.noarch

  5/7 Running scriptlet: python39-setuptools-50.3.2-4.module el8.6.0+930+10acc06f.noarch

  6/7 Installing       : python39-pip-20.2.4-7.module el8.7.0+1213+291b6551.noarch

  6/7 Running scriptlet: python39-pip-20.2.4-7.module el8.7.0+1213+291b6551.noarch

  7/7 Installing       : mysql-shell-8.0.32-1.el8.x86 64

  7/7 Running scriptlet: mysql-shell-8.0.32-1.el8.x86 64

  1/7 Verifying        : python39-3.9.16-1.module el8.8.0+1243+5f5a1e61.x86 64

  2/7 Verifying        : python39-libs-3.9.16-1.module el8.8.0+1243+5f5a1e61.x86 64

  3/7 Verifying        : python39-pip-20.2.4-7.module el8.7.0+1213+291b6551.noarch

  4/7 Verifying        : python39-pip-wheel-20.2.4-7.module el8.7.0+1213+291b6551.noarch

  5/7 Verifying        : python39-setuptools-50.3.2-4.module el8.6.0+930+10acc06f.noarch

  6/7 Verifying        : python39-setuptools-wheel-50.3.2-4.module el8.6.0+930+10acc06f.noarch

  7/7 Verifying        : mysql-shell-8.0.32-1.el8.x86 64


[client]

Installed:

  mysql-shell-8.0.32-1.el8.x86_64                                python39-3.9.16-
1.module_el8.8.0+1243+5f5a1e61.x86_64            python39-libs-3.9.16-
7.module_el8.7.0+1213+291b6551.noarch             python39-pip-20.2.4-

  python39-pip-wheel-20.2.4-7.module_el8.7.0+1213+291b6551.noarch  python39-setuptools-50.3.2-
4.module_el8.6.0+930+10acc06f.noarch  python39-setuptools-wheel-50.3.2-


Complete!
```

Step 3 Add the domains and IPs in the /etc/hosts

This step is not mandatory, the domains could be resolvable through the DNS, but domains in the /etc/hosts are faster resolved with only local for the system lookup. In fact, using /etc/hosts for the cluster domain names the IPs may be different in the different nodes, but this setup is more complex is beyond this article’s scope

```
192.168.87.187 innodb1
192.168.87.188 innodb2
192.168.87.189 innodb3
192.168.87.91 haproxy
```

Step 4: SELinux and firewall configuration

```
[root@innodb1 ~]# getenforce
[root@innodb1 ~]# setsebool -P mysql_connect_any 1
[root@innodb1 ~]# firewall-cmd --permanent --zone=trusted --add-source=192.168.0.0/24
[root@innodb1 ~]# firewall-cmd --reload
[root@innodb1 ~]# firewall-cmd --permanent --zone=trusted --add-source=192.168.87.187
[root@innodb1 ~]# firewall-cmd --permanent --zone=trusted --add-source=192.168.87.188
[root@innodb1 ~]# firewall-cmd --permanent --zone=trusted --add-source=192.168.87.189
[root@innodb1 ~]# firewall-cmd --reload
[root@innodb1 ~]# iptables -A OUTPUT -p tcp --sport 3306 -j ACCEPT
[root@innodb1 ~]# iptables -I INPUT -i ens160 -p tcp --destination-port 3306 -j ACCEPT
[root@innodb1 ~]# sudo firewall-cmd --zone=public --permanent --add-service=mysql
[root@innodb1 ~]# sudo systemctl restart firewalld
```

Step 5: Tune MySQL configuration.

Tune the MySQL generic and some InnoDB configuration variables for the purpose of the setup. For example, add to the bottom of file /etc/my.cnf (under section “[mysqld]”):

```
key_buffer_size                 = 256M
max_allowed_packet              = 256M
table_open_cache                = 500
open_files_limit                = 5000
sort_buffer_size                = 2048K
net_buffer_length               = 8K
read_buffer_size                = 256K
read_rnd_buffer_size            = 1024K
myisam_sort_buffer_size         = 64M
max_connections                 = 150
max_heap_table_size             = 512M
innodb_buffer_pool_size         = 2048M
innodb_log_buffer_size          = 32M
innodb_log_files_in_group       = 2
innodb_flush_log_at_trx_commit  = 0
innodb_lock_wait_timeout        = 50
innodb_file_per_table
innodb_thread_concurrency       = 0
innodb_flush_method             = O_DIRECT
```

Step 6: Set mysql root password and remote login for admin user

```
[root@innodb1 ~]# grep -i "temporary password" /var/log/mysqld.log
2023-04-03T20:02:10.192602Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: kgm68&cdfhJ0
[root@innodb1 ~]#  mysql -uroot -p"kgm68&cdfhJ0"
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'XXXXXXXX';
Query OK, 0 rows affected (0.01 sec)
mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> create USER 'admin'@'%' IDENTIFIED BY 'XXXXX';
mysql> GRANT ALL PRIVILEGES ON *.* TO 'admin'@'%';
mysql> create USER 'admin'@'%' IDENTIFIED BY 'XXXXX';
```

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE,
REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE,
REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE,
CREATE USER, EVENT, TRIGGER, CREATE TABLESPACE, CREATE ROLE, DROP ROLE ON *.* TO `admin`@`%`
WITH GRANT
OPTION;
grant
APPLICATION_PASSWORD_ADMIN,AUDIT_ADMIN,BACKUP_ADMIN,BINLOG_ADMIN,BINLOG_ENCRYPTION_ADMIN,CLONE_
ADMIN,CONNECTION_ADMIN,ENCRYPTION_KEY_ADMIN,FLUSH_OPTIMIZER_COSTS,FLUSH_STATUS,FLUSH_TABLES,FLU
SH_USER_RESOURCES,GROUP_REPLICATION_ADMIN,INNODB_REDO_LOG_ARCHIVE,INNODB_REDO_LOG_ENABLE,PERSIS
T_RO_VARIABLES_ADMIN,REPLICATION_APPLIER,REPLICATION_SLAVE_ADMIN,RESOURCE_GROUP_ADMIN,RESOURCE_
GROUP_USER,ROLE_ADMIN,SERVICE_CONNECTION_ADMIN,SESSION_VARIABLES_ADMIN,SET_USER_ID,SHOW_ROUTINE
,SYSTEM_USER,SYSTEM_VARIABLES_ADMIN,TABLE_ENCRYPTION_ADMIN,XA_RECOVER_ADMIN ON *.* TO
`admin`@`%` WITH GRANT OPTION;
```

Step 7: Check the configuration for MySQL 8.0 InnoDB Cluster and create a cluster administrative account.

The MySQL Shell command "mysqlsh" is used to configure and manage the MySQL 8.0 InnoDB Cluster using JavaScript syntax. The user accesses MySQL InnoDB Cluster functionality with the "dba" object through "mysqlsh"

```
[root@innodb1 ~]# mysqlsh
MySQL Shell 8.0.32

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
Creating a Classic session to 'root@localhost'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 14
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  localhost  JS > show databases;
SyntaxError: Unexpected identifier
```

```
 MySQL  localhost  JS > dba.configureInstance("root@localhost")
Please provide the password for 'root@localhost': ********
Save password for 'root@localhost'? [Y]es/[N]o/Ne[v]er (default No): No
Configuring local MySQL instance listening at port 3306 for use in an InnoDB cluster...

This instance reports its own address as innodb1:3306
Clients and other cluster members will communicate with it through this address by default. If
this is not correct, the report_host MySQL system variable should be changed.

ERROR: User 'root' can only connect from 'localhost'. New account(s) with proper source address
specification to allow remote connection from all instances must be created to manage the
cluster.

1) Create remotely usable account for 'root' with same grants and password
2) Create a new admin account for InnoDB cluster with minimal required grants
3) Ignore and continue
4) Cancel

Please select an option [1]: 2
Please provide an account name (e.g: icroot@%) to have it created with the necessary
privileges or leave empty and press Enter to cancel.
Account Name: clusteradmin@%
Password for new account: ********
Confirm password: ********

applierWorkerThreads will be set to the default value of 4.

NOTE: Some configuration options need to be fixed:
+----------------------------------------+---------------+----------------+--------------------
--------------------------------+
| Variable                               | Current Value | Required Value | Note
|
+----------------------------------------+---------------+----------------+--------------------
--------------------------------+
| binlog transaction dependency tracking | COMMIT ORDER  | WRITESET       | Update the server
variable                       |
| enforce gtid consistency               | OFF           | ON             | Update read-only
variable and restart the server |
| gtid mode                              | OFF           | ON             | Update read-only
variable and restart the server |
| server id                              | 1             | <unique ID>    | Update read-only
variable and restart the server |
+----------------------------------------+---------------+----------------+--------------------
--------------------------------+

Some variables need to be changed, but cannot be done dynamically on the server.
Do you want to perform the required configuration changes? [y/n]: y
```

```
Do you want to restart the instance after configuring it? [y/n]: y


Cluster admin user 'clusteradmin'@'%' created.
Configuring instance...
The instance 'innodb1:3306' was configured to be used in an InnoDB cluster.
Restarting MySQL...
NOTE: MySQL server at innodb1:3306 was restarted.
 MySQL  localhost  JS > \q
Bye!
```

Using the newly created admin user is easy to check whether the dba.configureInstance fixed the issues reported above:

```
[root@innodb1 ~]# mysqlsh
MySQL Shell 8.0.32

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
Creating a Classic session to 'root@localhost'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 10
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  localhost  JS > dba.checkInstanceConfiguration("clusteradmin@innodb1:3306")
Please provide the password for 'clusteradmin@innodb1:3306': ********
Save password for 'clusteradmin@innodb1:3306'? [Y]es/[N]o/Ne[v]er (default No): Y
Validating local MySQL instance listening at port 3306 for use in an InnoDB cluster...

This instance reports its own address as innodb1:3306
Clients and other cluster members will communicate with it through this address by default. If
this is not correct, the report_host MySQL system variable should be changed.
```

```
Checking whether existing tables comply with Group Replication requirements...
No incompatible tables detected


Checking instance configuration...
Instance configuration is compatible with InnoDB cluster


The instance 'innodb1:3306' is valid to be used in an InnoDB cluster.


{
    "status": "ok"
}
 MySQL  localhost  JS > \q
Bye!
```

Step 8:Create the MySQL Cluster.

It is a simple operation and it will create a group replication with the first server master, i.e. primary in the group. Note, first connect to the MySQL with the cluster administrator account and then issue a create command with the logical name of the cluster.

```
[root@innodb1 ~]# mysqlsh
MySQL Shell 8.0.32

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
Creating a Classic session to 'root@localhost'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 13
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  localhost  JS > \connect clusteradmin@innodb1
Creating a session to 'clusteradmin@innodb1'
```

```
Please provide the password for 'clusteradmin@innodb1': ********
Save password for 'clusteradmin@innodb1'? [Y]es/[N]o/Ne[v]er (default No): Y
Fetching schema names for auto-completion... Press ^C to stop.
Closing old connection...
Your MySQL connection id is 14 (X protocol)
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  innodb1:33060+ ssl  JS > dba.createCluster("mycluster1")
A new InnoDB Cluster will be created on instance 'innodb1:3306'.

Validating instance configuration at innodb1:3306...

This instance reports its own address as innodb1:3306

Instance configuration is suitable.
NOTE: Group Replication will communicate with other members using 'innodb1:3306'. Use the
localAddress option to override.

Creating InnoDB Cluster 'mycluster1' on 'innodb1:3306'...

Adding Seed Instance...
Cluster successfully created. Use Cluster.addInstance() to add MySQL instances.
At least 3 instances are needed for the cluster to be able to withstand up to
one server failure.

<Cluster:mycluster1>
 MySQL  innodb1:33060+ ssl  JS > var cluster = dba.getCluster()
 MySQL  innodb1:33060+ ssl  JS > cluster.status()
{
    "clusterName": "mycluster1",
    "defaultReplicaSet": {
        "name": "default",
        "primary": "innodb1:3306",
        "ssl": "REQUIRED",
        "status": "OK_NO_TOLERANCE",
        "statusText": "Cluster is NOT tolerant to any failures.",
        "topology": {
            "innodb1:3306": {
                "address": "innodb1:3306",
                "memberRole": "PRIMARY",
                "mode": "R/W",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            }
        },
        "topologyMode": "Single-Primary"
    },
    "groupInformationSourceMember": "innodb1:3306"
```

```
}
 MySQL  innodb1:33060+ ssl  JS > \q
Bye!
```

STEP 9) Install and configure the rest two cluster nodes.

Repeat the above steps from 1 to 7 on rest for two nodes i.e. innodb2,innodb2)

STEP 10) Add the two MySQL nodes in the cluster.

```
[root@innodb2 ~]# ssh innodb1
root@innodb1's password:
[root@innodb1 ~]# mysqlsh
MySQL Shell 8.0.32

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
Creating a Classic session to 'root@localhost'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 39
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  localhost  JS >  \connect clusteradmin@innodb1
```

```
Creating a session to 'clusteradmin@innodb1'

Fetching schema names for auto-completion... Press ^C to stop.

Closing old connection...

Your MySQL connection id is 40 (X protocol)

Server version: 8.0.32 MySQL Community Server - GPL

No default schema selected; type \use <schema> to set one.

 MySQL  innodb1:33060+ ssl  JS > var cluster = dba.getCluster()

 MySQL  innodb1:33060+ ssl  JS > cluster.status()

{

    "clusterName": "mycluster1",

    "defaultReplicaSet": {

        "name": "default",

        "primary": "innodb1:3306",

        "ssl": "REQUIRED",

        "status": "OK_NO_TOLERANCE",

        "statusText": "Cluster is NOT tolerant to any failures.",

        "topology": {

            "innodb1:3306": {

                "address": "innodb1:3306",

                "memberRole": "PRIMARY",

                "mode": "R/W",

                "readReplicas": {},

                "replicationLag": "applier_queue_applied",

                "role": "HA",

                "status": "ONLINE",

                "version": "8.0.32"

            }

        },

        "topologyMode": "Single-Primary"

    },

    "groupInformationSourceMember": "innodb1:3306"

}

 MySQL  innodb1:33060+ ssl  JS > cluster.addInstance('clusteradmin@innodb2:3306')


NOTE: The target instance 'innodb2:3306' has not been pre-provisioned (GTID set is empty). The
Shell is unable to decide whether incremental state recovery can correctly provision it.

The safest and most convenient way to provision a new instance is through automatic clone
provisioning, which will completely overwrite the state of 'innodb2:3306' with a physical
snapshot from an existing cluster member. To use this method by default, set the
'recoveryMethod' option to 'clone'.
```

```
The incremental state recovery may be safely used if you are sure all updates ever executed in
the cluster were done with GTIDs enabled, there are no purged transactions and the new instance
contains the same GTID set as the cluster or a subset of it. To use this method by default, set
the 'recoveryMethod' option to 'incremental'.



Please select a recovery method [C]lone/[I]ncremental recovery/[A]bort (default Clone): C
Validating instance configuration at innodb2:3306...

This instance reports its own address as innodb2:3306

Instance configuration is suitable.
NOTE: Group Replication will communicate with other members using 'innodb2:3306'. Use the
localAddress option to override.

A new instance will be added to the InnoDB Cluster. Depending on the amount of
data on the cluster this might take from a few seconds to several hours.

Adding instance to the cluster...

Monitoring recovery process of the new cluster member. Press ^C to stop monitoring and let it
continue in background.
Clone based state recovery is now in progress.

NOTE: A server restart is expected to happen as part of the clone process. If the
server does not support the RESTART command or does not come back after a
while, you may need to manually start it back.

* Waiting for clone to finish...
NOTE: innodb2:3306 is being cloned from innodb1:3306
** Stage DROP DATA: Completed
** Clone Transfer
    FILE COPY  ############################################################  100%  Completed
    PAGE COPY  ############################################################  100%  Completed
    REDO COPY  ############################################################  100%  Completed

NOTE: innodb2:3306 is shutting down...

* Waiting for server restart... ready
* innodb2:3306 has restarted, waiting for clone to finish...
** Stage RESTART: Completed
* Clone process has finished: 73.65 MB transferred in about 1 second (~73.65 MB/s)
```

```
State recovery already finished for 'innodb2:3306'

The instance 'innodb2:3306' was successfully added to the cluster.

 MySQL  innodb1:33060+ ssl  JS > cluster.status()
{
    "clusterName": "mycluster1",
    "defaultReplicaSet": {
        "name": "default",
        "primary": "innodb1:3306",
        "ssl": "REQUIRED",
        "status": "OK_NO_TOLERANCE",
        "statusText": "Cluster is NOT tolerant to any failures.",
        "topology": {
            "innodb1:3306": {
                "address": "innodb1:3306",
                "memberRole": "PRIMARY",
                "mode": "R/W",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            },
            "innodb2:3306": {
                "address": "innodb2:3306",
                "memberRole": "SECONDARY",
                "mode": "R/O",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            }
        },
        "topologyMode": "Single-Primary"
    },
    "groupInformationSourceMember": "innodb1:3306"
```

```
}
 MySQL  innodb1:33060+ ssl  JS > \q
Bye!
```

Repeat above from node 3 aa below

```
[root@innodb1 ~]# mysqlsh
MySQL Shell 8.0.32

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
Creating a Classic session to 'root@localhost'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 66
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  localhost  JS > \connect clusteradmin@innodb1
Creating a session to 'clusteradmin@innodb1'
Fetching schema names for auto-completion... Press ^C to stop.
Closing old connection...
Your MySQL connection id is 67 (X protocol)
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  innodb1:33060+ ssl  JS > var cluster = dba.getCluster()
 MySQL  innodb1:33060+ ssl  JS > cluster.status()
{
    "clusterName": "mycluster1",
    "defaultReplicaSet": {
        "name": "default",
        "primary": "innodb1:3306",
        "ssl": "REQUIRED",
        "status": "OK_NO_TOLERANCE",
        "statusText": "Cluster is NOT tolerant to any failures.",
        "topology": {
```

```
            "innodb1:3306": {
                "address": "innodb1:3306",
                "memberRole": "PRIMARY",
                "mode": "R/W",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            },
            "innodb2:3306": {
                "address": "innodb2:3306",
                "memberRole": "SECONDARY",
                "mode": "R/O",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            }
        },
        "topologyMode": "Single-Primary"
    },
    "groupInformationSourceMember": "innodb1:3306"
}

 MySQL  innodb1:33060+ ssl  JS > cluster.addInstance('clusteradmin@innodb3:3306')

NOTE: The target instance 'innodb3:3306' has not been pre-provisioned (GTID set is empty). The
Shell is unable to decide whether incremental state recovery can correctly provision it.

The safest and most convenient way to provision a new instance is through automatic clone
provisioning, which will completely overwrite the state of 'innodb3:3306' with a physical
snapshot from an existing cluster member. To use this method by default, set the
'recoveryMethod' option to 'clone'.

The incremental state recovery may be safely used if you are sure all updates ever executed in
the cluster were done with GTIDs enabled, there are no purged transactions and the new instance
contains the same GTID set as the cluster or a subset of it. To use this method by default, set
the 'recoveryMethod' option to 'incremental'.

Please select a recovery method [C]lone/[I]ncremental recovery/[A]bort (default Clone): C
Validating instance configuration at innodb3:3306...

This instance reports its own address as innodb3:3306
```

```
Instance configuration is suitable.
NOTE: Group Replication will communicate with other members using 'innodb3:3306'. Use the
localAddress option to override.

A new instance will be added to the InnoDB Cluster. Depending on the amount of
data on the cluster this might take from a few seconds to several hours.

Adding instance to the cluster...

Monitoring recovery process of the new cluster member. Press ^C to stop monitoring and let it
continue in background.
Clone based state recovery is now in progress.

NOTE: A server restart is expected to happen as part of the clone process. If the
server does not support the RESTART command or does not come back after a
while, you may need to manually start it back.

* Waiting for clone to finish...
NOTE: innodb3:3306 is being cloned from innodb1:3306
** Stage DROP DATA: Completed
** Clone Transfer
    FILE COPY  ############################################################  100%  Completed
    PAGE COPY  ############################################################  100%  Completed
    REDO COPY  ############################################################  100%  Completed

NOTE: innodb3:3306 is shutting down...

* Waiting for server restart... ready
* innodb3:3306 has restarted, waiting for clone to finish...
** Stage RESTART: Completed
* Clone process has finished: 73.65 MB transferred in about 1 second (~73.65 MB/s)

State recovery already finished for 'innodb3:3306'
The instance 'innodb3:3306' was successfully added to the cluster.

 MySQL  innodb1:33060+ ssl  JS > \connect clusteradmin@innodb1
Creating a session to 'clusteradmin@innodb1'
Fetching schema names for auto-completion... Press ^C to stop.
Closing old connection...
```

```
Your MySQL connection id is 92 (X protocol)
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  innodb1:33060+ ssl  JS > cluster.status()
{
    "clusterName": "mycluster1",
    "defaultReplicaSet": {
        "name": "default",
        "primary": "innodb1:3306",
        "ssl": "REQUIRED",
        "status": "OK",
        "statusText": "Cluster is ONLINE and can tolerate up to ONE failure.",
        "topology": {
            "innodb1:3306": {
                "address": "innodb1:3306",
                "memberRole": "PRIMARY",
                "mode": "R/W",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            },
            "innodb2:3306": {
                "address": "innodb2:3306",
                "memberRole": "SECONDARY",
                "mode": "R/O",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            },
            "innodb3:3306": {
                "address": "innodb3:3306",
                "memberRole": "SECONDARY",
                "mode": "R/O",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
```

```
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            }
        },
        "topologyMode": "Single-Primary"
    },
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
    "groupInformationSourceMember": "innodb1:3306"
}
```

Step 11: Deploy MySQL Router

In order for applications to handle failover, they need to be aware of the topology of the InnoDB cluster. They also need to know, at any time, which of the instances is the PRIMARY. While it’s possible for applications to implement that logic by themselves, MySQL Router can do that for you, with minimal work and no code changes in applications.

The recommended deployment of MySQL Router is on the same host as the application. During bootstrap, MySQL Router needs to connect to the cluster and have privileges to query the performance_schema, mysql_innodb_cluster_metadata and create a restricted, read-only account to be used by itself during normal operation.

```
[root@innodb1 ~]# ssh haproxy
The authenticity of host 'haproxy (192.168.87.91)' can't be established.
ECDSA key fingerprint is SHA256:Xs36DwXgm739xXJ5N7wotPL0gBZm2LCKNTSFyCP+ZO4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'haproxy,192.168.87.91' (ECDSA) to the list of known hosts.
root@haproxy's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Mon Apr  3 10:24:13 2023 from 192.168.87.69
```

```
[root@haproxy ~]# dnf install -y mysql-router-community
Last metadata expiration check: 6:06:56 ago on Mon 03 Apr 2023 10:24:28 AM PDT.
Dependencies resolved.
================================================================================
 Package                                 Repository                  Architecture
================================================================================
Installing:
 mysql-router-community                  mysql-tools-community       x86_64

Transaction Summary
================================================================================
Install  1 Package

Total download size: 5.0 M
Installed size: 22 M
Downloading Packages:
mysql-router-community-8.0.32-1.el8.x86_64.rpm
--------------------------------------------------------------------------------
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
  Running scriptlet: mysql-router-community-8.0.32-1.el8.x86_64
  Installing       : mysql-router-community-8.0.32-1.el8.x86_64
  Running scriptlet: mysql-router-community-8.0.32-1.el8.x86_64
  Verifying        : mysql-router-community-8.0.32-1.el8.x86_64

Installed:
  mysql-router-community-8.0.32-1.el8.x86_64
```

Second, bootstrap the configuration to create the initial configuration and the needed user(s).

```
[root@haproxy ~]# mysqlrouter --bootstrap clusteradmin@innodb1:3306 --user mysqlrouter --conf-use-sockets --account routerfriend1 --account-create if-not-exists
```

```
Please enter MySQL password for clusteradmin:

# Bootstrapping system MySQL Router instance...


Please enter MySQL password for routerfriend1:
- Creating account(s) (only those that are needed, if any)
- Verifying account (using it to run SQL queries that would be run by Router)
- Storing account in keyring
- Adjusting permissions of generated files
- Creating configuration /etc/mysqlrouter/mysqlrouter.conf

Existing configuration backed up to '/etc/mysqlrouter/mysqlrouter.conf.bak'

# MySQL Router configured for the InnoDB Cluster 'mycluster1'

After this MySQL Router has been started with the generated configuration

    $ /etc/init.d/mysqlrouter restart
or
    $ systemctl start mysqlrouter
or
    $ mysqlrouter -c /etc/mysqlrouter/mysqlrouter.conf

InnoDB Cluster 'mycluster1' can be reached by connecting to:

## MySQL Classic protocol

- Read/Write Connections: localhost:6446, /tmp/mysql.sock
- Read/Only Connections:  localhost:6447, /tmp/mysqlro.sock

## MySQL X protocol

- Read/Write Connections: localhost:6448, /tmp/mysqlx.sock
- Read/Only Connections:  localhost:6449, /tmp/mysqlxro.sock

[root@haproxy ~]# systemctl start mysqlrouter
[root@haproxy ~]# systemctl status mysqlrouter
● mysqlrouter.service - MySQL Router
```

```
   Loaded: loaded (/usr/lib/systemd/system/mysqlrouter.service; disabled; vendor preset:
disabled)
   Active: active (running) since Mon 2023-04-03 16:41:36 PDT; 43s ago
 Main PID: 5145 (mysqlrouter)
   Status: "running"
    Tasks: 24 (limit: 23008)
   Memory: 12.1M
   CGroup: /system.slice/mysqlrouter.service
           └─5145 /usr/bin/mysqlrouter

Apr 03 16:41:36 haproxy systemd[1]: Starting MySQL Router...
Apr 03 16:41:36 haproxy systemd[1]: Started MySQL Router.
[root@haproxy ~]# systemctl enable mysqlrouter
Created symlink /etc/systemd/system/multi-user.target.wants/mysqlrouter.service →
/usr/lib/systemd/system/mysqlrouter.service.
[root@haproxy ~]# firewall-cmd --permanent --zone=public --add-port=6446/tcp
success
[root@haproxy ~]# firewall-cmd --permanent --zone=public --add-port=6447/tcp
success
[root@haproxy ~]# firewall-cmd --permanent --zone=public --add-port=6448/tcp
success
[root@haproxy ~]# firewall-cmd --permanent --zone=public --add-port=6449/tcp
success
[root@haproxy ~]# firewall-cmd --reload
success
```

Port 6446 is for **Read/Write** operations and the **6447** is only for **Read operations** of the classic MySQL protocol, the other two ports are for the MySQL X protocol. Note that, by executing the above firewall rules, all IPs would be allowed to contact the specified ports.

Step 12:Connection using mysql router read/write node—> port 6446

```
[root@haproxy ~]# mysqlsh --uri clusteradmin@localhost:6446
Please provide the password for 'clusteradmin@localhost:6446': ********
Save password for 'clusteradmin@localhost:6446'? [Y]es/[N]o/Ne[v]er (default No): y
MySQL Shell 8.0.32

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
Creating a session to 'clusteradmin@localhost:6446'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 1048
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  localhost:6446 ssl  JS > var cluster = dba.getCluster()
 MySQL  localhost:6446 ssl  JS > cluster.status()
{
    "clusterName": "mycluster1",
    "defaultReplicaSet": {
        "name": "default",
        "primary": "innodb1:3306",
        "ssl": "REQUIRED",
        "status": "OK",
        "statusText": "Cluster is ONLINE and can tolerate up to ONE failure.",
        "topology": {
            "innodb1:3306": {
                "address": "innodb1:3306",
                "memberRole": "PRIMARY",
                "mode": "R/W",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            },
            "innodb2:3306": {
                "address": "innodb2:3306",
                "memberRole": "SECONDARY",
                "mode": "R/O",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            },
            "innodb3:3306": {
                "address": "innodb3:3306",
                "memberRole": "SECONDARY",
```

```
                "mode": "R/O",
                "readReplicas": {},
                "replicationLag": "applier_queue_applied",
                "role": "HA",
                "status": "ONLINE",
                "version": "8.0.32"
            }
        },
        "topologyMode": "Single-Primary"
    },
    "groupInformationSourceMember": "innodb1:3306"
```

Connection using admin user

```
[root@haproxy ~]# mysqlsh --uri admin@localhost:6446
Please provide the password for 'admin@localhost:6446': ********
Save password for 'admin@localhost:6446'? [Y]es/[N]o/Ne[v]er (default No): yes
MySQL Shell 8.0.32

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
Creating a session to 'admin@localhost:6446'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 2838
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  localhost:6446 ssl  JS > \sql
Switching to SQL mode... Commands end with ;
Fetching global names for auto-completion... Press ^C to stop.
 MySQL  localhost:6446 ssl  SQL > show databases;
+--------------------------------+
| Database                       |
+--------------------------------+
| information_schema             |
| mysql                          |
```

```
| mysql_innodb_cluster_metadata |
| performance_schema            |
| sys                           |
+-------------------------------+
6 rows in set (0.0016 sec)
 MySQL  localhost:6446 ssl  SQL > create database mukeshdb;
 MySQL  localhost:6446 ssl  SQL > use mukeshdb;
Default schema set to `mukeshdb`.
Fetching global names, object names from `mukeshdb` for auto-completion... Press ^C to stop.
 MySQL  localhost:6446 ssl  mukeshdb  SQL > CREATE TABLE Persons (
                                        ->     PersonID int,
                                        ->     LastName varchar(255),
                                        ->     FirstName varchar(255),
                                        ->     Address varchar(255),
                                        ->     City varchar(255)
                                        -> );
Query OK, 0 rows affected (0.0228 sec)
 MySQL  localhost:6446 ssl  mukeshdb  SQL > show tables;
+--------------------+
| Tables_in_mukeshdb |
+--------------------+
| Persons            |
+--------------------+
1 row in set (0.0033 sec)
 MySQL  localhost:6446 ssl  mukeshdb  SQL > \q
Bye!
```

Connect to read only node using mysql router → port →6447

```
[root@haproxy ~]# mysqlsh --uri root@localhost:6447
Please provide the password for 'root@localhost:6447': ********
Save password for 'root@localhost:6447'? [Y]es/[N]o/Ne[v]er (default No): y
MySQL Shell 8.0.32
```

```
Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
Creating a session to 'root@localhost:6447'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 42
Server version: 8.0.32 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
 MySQL  localhost:6447 ssl  JS > \sql
Switching to SQL mode... Commands end with ;
Fetching global names for auto-completion... Press ^C to stop.
 MySQL  localhost:6447 ssl  SQL > show databases;
+--------------------------------+
| Database                       |
+--------------------------------+
| information_schema             |
| mukeshdb                       |
| mysql                          |
| mysql_innodb_cluster_metadata  |
| performance_schema             |
| sys                            |
+--------------------------------
```