Requirements for MySQL and MariaDB

Since v5.2.3/15

MySQL/MariaDB/Aurora MySQL		
Capture	**Hub**	**Integrate**
✓	✓	✓

Contents

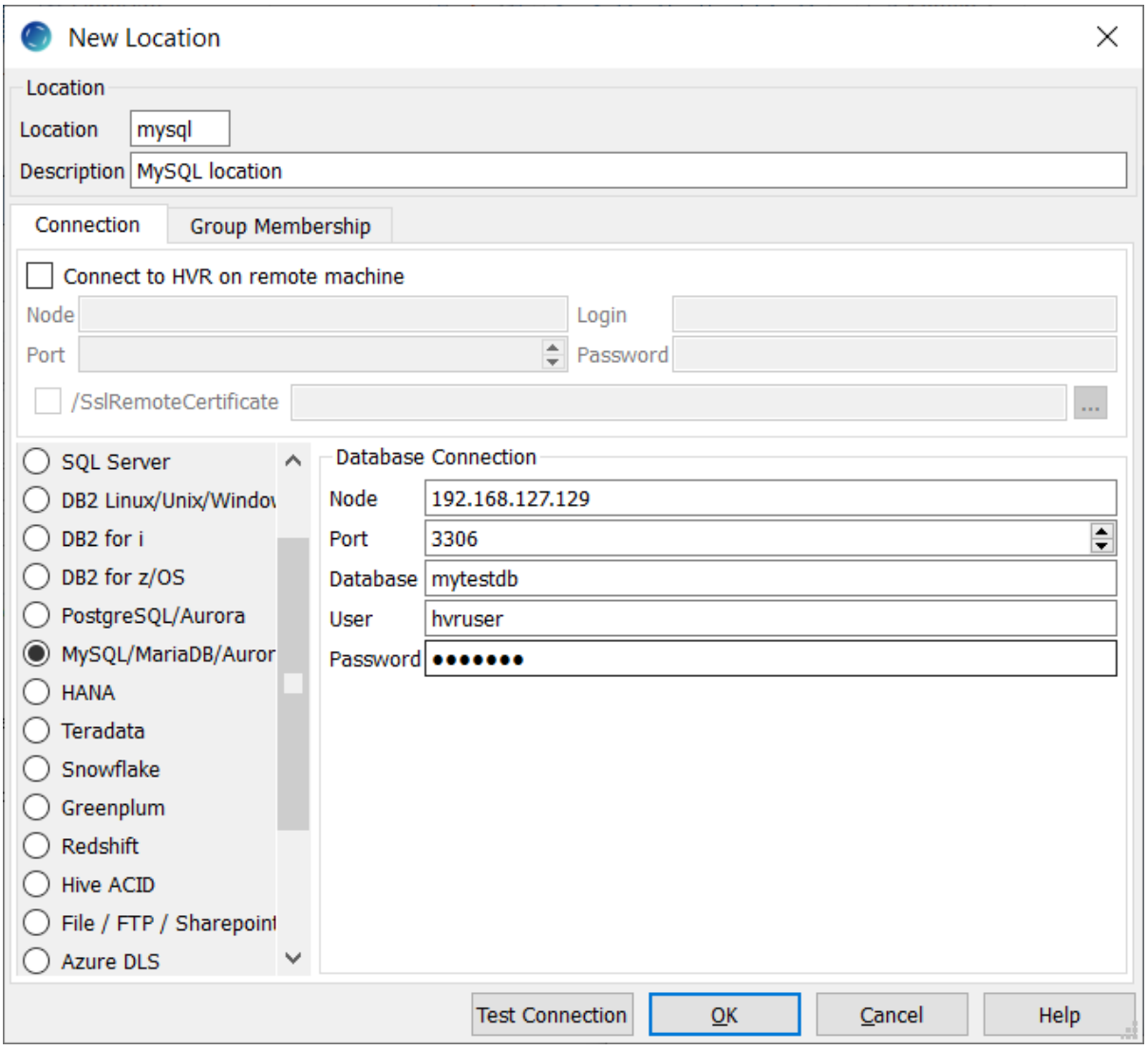
This section describes the requirements, access privileges, and other features of HVR when using MySQL/MariaDB/Aurora MySQL for replication. For information about compatibility and supported versions of MySQL/MariaDB with HVR platforms, see Platform Compatibility Matrix.

For the Capabilities supported by HVR on MySQL, MariaDB, and Aurora MySQL, see Capabilities for MySQL, Capabilities for MariaDB, and Capabilities for Aurora MySQL respectively.

For information about the supported data types and mapping of data types in source DBMS to the corresponding data types in target DBMS or file format, see Data Type Mapping.

Location Connection

This section lists and describes the connection details required for creating MySQL/MariaDB/Aurora MySQL location in HVR. HVR uses MariaDB's native Connector/C interface to connect, read, and write data to MySQL/MariaDB/Aurora MySQL. HVR connects to the MySQL/MariaDB/Aurora MySQL server using the TCP protocol.



Field	Description
Node	The hostname or IP-address of the machine on which the MySQL/MariaDB server is running. **Example:** 192.168.127.129
Port	The TCP port on which the MySQL/MariaDB/Aurora MySQL server is expecting connections. **Example:** 3306
Database	The name of the MySQL/MariaDB/Aurora MySQL database. **Example:** mytestdb
User	The username to connect HVR to MySQL/MariaDB/Aurora MySQL **Database**. **Example:** hvruser
Password	The password of the **User** to connect HVR to MySQL/MariaDB/Aurora MySQL **Database**.

HUB

HVR allows you to create hub database in MySQL/MariaDB/Aurora MySQL. The hub database is a small database which HVR uses to control its replication activities. This database stores HVR catalog tables that hold all specifications of replication such as the names of the replicated databases, the list of replicated tables, and the replication direction.

Grants for Hub

To capture changes from source database or to integrate changes into target database the HVR hub database **User** (e.g., *hvruser*) requires the following grants:

- Permission to create and drop HVR catalog tables.

Capture

Since v5.3.1/13

HVR supports **capturing** changes from MySQL/MariaDB (includes regular MySQL, MariaDB, Amazon RDS for MySQL, and Aurora MySQL) location. HVR uses MariaDB's native Connector/C interface to capture data from MySQL/MariaDB. For the list of supported MySQL, MariaDB or Aurora MySQL versions from which HVR can capture changes, see Capture changes from location in Capabilities.

There are two log read methods supported for capturing changes from MySQL/MariaDB: **SQL** and **DIRECT**. In terms of capture speed and database resource consumption, there is not much difference between using **SQL** or **DIRECT** method for capturing from a MySQL/MariaDB location.

By default, HVR captures changes from MySQL/MariaDB using the **SQL** log read method (**Capture /LogReadMethod**=**SQL**).

The **DIRECT** method requires:

- an HVR agent to be installed on the MySQL/MariaDB source database server
- the Operating System (**OS**) **user** the HVR is running under must have grants to read from binary log files.

From HVR 5.3.1/13 to HVR 5.3.1/20, capturing changes from MySQL using the **DIRECT** connection method is not available. Because of this behavior, the option **Capture /LogReadMethod** is not available for these versions of MySQL.

Grants for Capture

To capture changes from MySQL the **User** requires the following grants:

```
grant replication client on *.* to 'hvruser'@'%' ;

grant replication slave on *.* to 'hvruser'@'%' ;

grant select on *.* to 'hvruser'@'%' ;
```

Binary Logging

In MySQL, the transaction updates are recorded in the binary logs. For HVR to capture changes, the binary logging should be configured in MySQL database. MySQL allows you to define system variables (parameters) at server level (global) and at session level. The configuration for binary logging should be strictly defined as mentioned in this section. Defining parameters not mentioned in this section can lead to HVR not capturing changes.

For more information about binary logging, search for "binary logging" in MySQL Documentation.

If binary logging is not enabled in MySQL, a similar error is displayed in HVR: "hvrinit: F_JD0AC8: The 'SHOW MASTER STATUS' command returned no results. Please check that the binary logging is enabled for the source database. F_JR0015: The previous error was detected during generation of objects for channel hvr_demo. It is possible that objects have been left incomplete."

Binary Logging for Regular MySQL

The following parameters should be defined in MySQL configuration file **my.cnf** (Unix) or **my.ini** (Windows):

- **log_bin=ON** - to enable binary logging in MySQL.
- **binlog_format=ROW** - to set the binary logging format.
- **binlog_row_image=full** or **binlog_row_image=noblob** - to determine how row images are written to the binary log.

binlog_row_image=minimal is not supported in HVR.

Binary Logging for Amazon RDS for MySQL and Aurora MySQL

This section provides information required for configuring binary logging in Amazon RDS for MySQL and Aurora MySQL database.

1. To enable binary logging, perform the steps mentioned in Amazon documentation - How do I enable binary logging for Amazon Aurora for MySQL?.
   While performing the steps to enable binary logging, the following parameters should be defined:
   - **binlog_format=ROW** - to set the binary logging format.
   - **binlog_checksum=CRC32** to enable writing a checksum for each event in the binary log.
2. For Aurora MySQL, the cluster should be restarted after enabling the binary logging. The replication will begin only after restarting the cluster.

3. **Backup retention period in Amazon RDS for MySQL.** Enable automatic backups on the source MySQL DB instance by setting the backup retention period to a value greater than 0. The backup retention period setting defines the number of days for which automated backups are retained. The primary reason for this is that Amazon RDS normally purges a binary log as soon as possible, but the binary log must still be available on the instance to be accessed.

   In Amazon RDS for MySQL, disabling automatic backups may implicitly disable binary logging which will lead to replication issues in HVR.

   To specify the number of hours for RDS to retain binary logs, use the **mysql.rds_set_configuration** stored procedure and specify a period with enough time for you to access the logs.

   The **mysql.rds_set_configuration** stored procedure is only available for MySQL version 5.6 or later.

   The following example sets the retention period to 1 day:

   ```
   call mysql.rds_set_configuration('binlog retention hours', 24);
   ```

   To display the current setting, use the **mysql.rds_show_configuration** stored procedure:

   ```
   call mysql.rds_show_configuration;
   ```

Integrate and Refresh Target

HVR supports integrating changes into MySQL/MariaDB (includes regular MySQL, MariaDB, Amazon RDS for MySQL, and Aurora MySQL) location. This section describes the configuration requirements for integrating changes (using **Integrate** and **refresh**) into MySQL/MariaDB location. For the list of supported MySQL or Aurora MySQL versions, into which HVR can integrate changes, see Integrate changes into location in Capabilities.

HVR uses MariaDB's native Connector/C interface to write data into MySQL/MariaDB during continuous **Integrate** and row-wise **Refresh**. For the methods used during **Integrate** with **/Burst** and Bulk **Refresh,** see section Burst Integrate and Bulk Refresh below.

Grants for Integrate and Refresh Target

To integrate changes into MySQL/MariaDB location, **User** requires the following grants:

- Permission to read and change replicated tables.

```
grant select, insert, update, delete on tbl to hvruser
```

- Permission to create and drop HVR state tables.

Prerequisites for Bulk Load

The two options available to use bulk load during **Refresh** or **Integrate** in MySQL/MariaDB are:

1. Direct loading by the MySQL/MariaDB server. The following conditions should be satisfied to use this option:
   - The **User** should have **FILE** permission.
   - The system variable (of MySQL/MariaDB server) **secure_file_priv** should be set to "" (blank).
2. Initial loading by the MySQL/MariaDB client followed by MySQL/MariaDB server. The following condition should be satisfied to use this option:
   - The system variable (of MySQL/MariaDB client and server) **local_infile** should be enabled.

Burst Integrate and Bulk Refresh

While **HVR Integrate** is running with parameter **/Burst** and Bulk **Refresh**, HVR can stream data into a target database straight over the network into a bulk loading interface specific for each DBMS (e.g. direct-path-load in Oracle), or else HVR puts data into a temporary directory (‘staging file') before loading data into a target database.

For best performance, HVR performs **Integrate** with **/Burst** and Bulk **Refresh** into a MySQL/MariaDB location using staging files. HVR implements **Integrate** with **/Burst** and Bulk **Refresh** (with file staging) into MySQL/MariaDB as follows:

Server File Staging - Direct Loading

1. HVR first stages data to a **server** local staging file (file write)
2. HVR then uses MySQL command '**load data**' to load the data into MySQL/MariaDB target tables

Client File Staging - Initial Loading

1. HVR first stages data to a **client** local staging file (file write)
2. HVR then uses MySQL command '**load data local**' to ingest the data into MySQL/MariaDB target tables

To perform **Integrate** with parameter **/Burst** and Bulk **Refresh**, define action **LocationProperties** on MySQL/MariaDB location with the following parameters:

- **/StagingDirectoryHvr**: a directory local to the MySQL/MariaDB server which can be written to by the HVR user from the machine that HVR uses to connect to the DBMS.
- **/StagingDirectoryDb**: the location from where MySQL/MariaDB will access the temporary staging files.

For MySQL on-premise, you can either define both parameters (**/StagingDirectoryHvr** and **/StagingDirectoryDb**) or define only one parameter (**/StagingDirectoryHvr**).

For MySQL on cloud, you should define only one parameter (**/StagingDirectoryHvr**).

In MySQL bi-directional channel, Bulk **Refresh** may result in looping truncates on either side of the bi-directional channel. For a workaround, see section Bi-directional Replication using MySQL in Managing Recapturing Using Session Names.

Compare and Refresh Source

HVR supports **compare** and **refresh** from (read from) MySQL/MariaDB location. This section describes the configuration requirements for performing **compare** and **refresh** from MySQL/MariaDB (source) location.

Grants for Compare and Refresh Source

To perform **HVR Compare** or **HVR Refresh** (read from MySQL/MariaDB), the **User** requires the following grant to read the replicated tables:

```
grant select on tbl to hvruser
```