Requirements for SingleStore

Since v5.7.5/4

	SingleStore	
Capture	**Hub**	**Integrate**
❌	❌	✅

Contents

This section describes the requirements, access privileges, and other features of HVR when using SingleStore (formerly MemSQL) for replication.

For information about compatibility and supported versions of SingleStore with HVR platforms, see Platform Compatibility Matrix.

For the Capabilities supported by HVR on SingleStore, see Capabilities for SingleStore.

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 SingleStore location in HVR. HVR uses MariaDB's native Connector/C interface to connect, read, and write data to SingleStore. HVR connects to the SingleStore server using the TCP protocol.

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

SingleStore Server Time Zone

To use the **timestamp** data type in SingleStore database, the SingleStore server's time zone must be set to **UTC** or **+00:00** using the **default_time_zone** configuration parameter. For more information about setting the time zone, refer to article Setting the Time Zone in SingleStore documentation.

For example, to set the time zone to **UTC** on a host, run the command:

```
sdb-admin update-config --key default_time_zone --value "+00:00" --all
```

While updating a time zone on a host, it is required to update it identically on all hosts in the cluster and then the cluster must be restarted.

Integrate and Refresh Target

HVR supports integrating changes into SingleStore location. This section describes the configuration requirements for integrating changes (using **Integrate** and **refresh**) into SingleStore location. For the list of supported SingleStore 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 SingleStore 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 SingleStore location, the database **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 SingleStore are:

1. Direct loading by the SingleStore server. The following conditions should be satisfied to use this option:
   - The **User** should have **FILE** permission.
   - The system variable (of SingleStore server) **secure_file_priv** should be set to "" (blank).
2. Initial loading by the SingleStore client followed by SingleStore server. The following condition should be satisfied to use this option:
   - The system variable (of SingleStore 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 SingleStore location using staging files. HVR implements **Integrate** with **/Burst** and Bulk **Refresh** (with file staging) into SingleStore 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 SingleStore 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 SingleStore target tables

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

- **/StagingDirectoryHvr**: a directory local to the SingleStore 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 SingleStore will access the temporary staging files.

You can either define both parameters (**/StagingDirectoryHvr** and **/StagingDirectoryDb**) or define only one parameter (**/StagingDirectoryHvr**).

Compare and Refresh Source

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

Grants for Compare and Refresh Source

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

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