

Requirements for Redshift

Contents
<ul style="list-style-type: none">• ODBC Connection• Location Connection• Integrate and Refresh Target<ul style="list-style-type: none">• Burst Integrate and Bulk Refresh

Redshift		
Capture	Hub	Integrate
		

This section describes the requirements, access privileges, and other features of HVR when using Amazon Redshift for replication. For information about compatibility and supported versions of Redshift with HVR platforms, see [Platform Compatibility Matrix](#).

For the [Capabilities](#) supported by HVR on Redshift, see [Capabilities for Redshift](#).

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](#).

For instructions to quickly setup replication using Redshift, see [Quick Start for HVR - Redshift](#).

ODBC Connection

HVR uses ODBC connection to the Amazon Redshift clusters. The Amazon Redshift ODBC driver must be installed on the machine from which HVR connects to the Amazon Redshift clusters. For more information about downloading and installing Amazon Redshift ODBC driver, refer to [AWS documentation](#).

On Linux, HVR additionally requires unixODBC.

For information about the supported ODBC driver version, refer to the HVR release notes (**hvr.rel**) available in **hvr_home** directory or the download page.

Location Connection

This section lists and describes the connection details required for creating Redshift location in HVR.

New Location
✕

Location

Description

Connection

Group Membership

Connect to HVR on remote machine

Node Login

Port Password

/SslRemoteCertificate

- DB2 for z/OS
- PostgreSQL/Aurora
- MySQL/MariaDB/Aurora
- HANA
- Teradata
- Snowflake
- Greenplum
- Redshift
- Hive ACID
- File / FTP / Sharepoint
- Azure DLS
- Azure Blob FS
- HDFS
- S3

Database Connection

Node

Port

Database

User

Password

Linux / Unix

Driver Manager Library

ODBCSYSINI

ODBC Driver

Field	Description
Database Connection	
Node	The hostname or IP-address of the machine on which the Redshift server is running. Example: hvrcluster.ar.78ah9i45.eu-west-1.redshift.amazonaws.com
Port	The port on which the Redshift server is expecting connections. Example: 5439
Database	The name of the Redshift database. Example: mytestdb
User	The username to connect HVR to Redshift Database . Example: hvruser
Password	The password of the User to connect HVR to Redshift Database .
Linux / Unix	
Driver Manager Library	The optional directory path where the ODBC Driver Manager Library is installed. This field is applicable only for Linux/Unix operating system. For a default installation, the ODBC Driver Manager Library is available at /usr/lib64 and does not need to be specified. However, when UnixODBC is installed in for example /opt/unixodbc the value for this field would be /opt/unixodbc/lib .

ODBCSYSINI	<p>The optional directory path where odbc.ini and odbcinst.ini files are located. This field is applicable only for Linux/Unix operating system.</p> <p>For a default installation, these files are available at /etc and do not need to be specified. However, when UnixODBC is installed in for example /opt/unixodbc the value for this field would be /opt/unixodbc/etc.</p> <p>The odbcinst.ini file should contain information about the Amazon Redshift ODBC Driver under the heading [Amazon Redshift (x64)].</p>
ODBC Driver	The user defined (installed) ODBC driver to connect HVR to the Amazon Redshift clusters.

Integrate and Refresh Target

HVR supports integrating changes into Redshift location. This section describes the configuration requirements for integrating changes (using **Integrate** and **refresh**) into Redshift location. For the list of supported Redshift versions, into which HVR can integrate changes, see [Integrate changes into location](#) in [Capabilities](#).

HVR uses the Amazon Redshift ODBC driver to write data to Redshift during continuous **Integrate** and row-wise **Refresh**. However, the preferred methods for writing data to Redshift are **Integrate** with **/Burst** and Bulk **Refresh** using staging as they provide better performance.

Burst Integrate and Bulk Refresh

While **HVR Integrate** is running with 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 Redshift using staging files. HVR implements **Integrate** with **/Burst** and Bulk **Refresh** (with file staging) into Redshift as follows:

1. HVR first connects to S3 using the curl library and writes data into a temporary Amazon S3 staging file(s). This S3 temporary file is written in a CSV format.
2. HVR then uses Redshift SQL '**copy from s3://**' command to load data from S3 temp files and ingest them into Redshift tables.

HVR requires the following to perform **Integrate** with parameter **/Burst** and Bulk **Refresh** on Redshift:

1. An AWS S3 location (bucket) - to store temporary data to be loaded into Snowflake. For more information about creating and configuring an S3 bucket, refer to [AWS Documentation](#).
2. An AWS user with **AmazonS3FullAccess** permission policy - to access the S3 bucket. Alternatively, an AWS user with minimal set of permission can also be used.
 - **s3:GetBucketLocation**
 - **s3:ListBucket**
 - **s3:ListBucketMultipartUploads**
 - **s3:AbortMultipartUpload**
 - **s3:GetObject**
 - **s3:PutObject**
 - **s3>DeleteObject**

```

{
  "Statement": [
    {
      "Sid": <identifier>,
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<account_id>:<user>
/<username>",
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:AbortMultipartUpload"
      ],
      "Resource": "arn:aws:s3:::<bucket_name>/*"
    },
    {
      "Sid": <identifier>,
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<account_id>:<user>
/<username>"
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": "arn:aws:s3:::<bucket_name>"
    }
  ]
}

```

For more information on the Amazon S3 permissions policy, refer to the [AWS S3 documentation](#).

For more information, refer to the following AWS documentation:

- [Amazon S3 and Tools for Windows PowerShell](#)
 - [Managing Access Keys for IAM Users](#)
 - [Creating a Role to Delegate Permissions to an AWS Service](#)
3. Define action **LocationProperties** on the Redshift location with the following parameters:
 - **/StagingDirectoryHvr**: the location where HVR will create the temporary staging files (e. g. **s3://my_bucket_name/**).
 - **/StagingDirectoryDb**: the location from where Redshift will access the temporary staging files. This should be the S3 location that is used for **/StagingDirectoryHvr**.
 - **/StagingDirectoryCredentials**: the AWS security credentials. The supported formats are '**aws_access_key_id="key";aws_secret_access_key="secret_key"**' or '**role="AWS_role"**'. How to get your AWS credential or Instance Profile Role can be found on the AWS documentation web page.
 4. If the S3 bucket used for the staging directory do not reside in the same region as the Redshift server, the region of the S3 bucket must be explicitly specified (this is required for using the Redshift **"copy from"** feature). For more information, search for "COPY from Amazon S3 - Amazon Redshift" in [AWS documentation](#). To specify the **S3 bucket region**, define the following action on the Redshift location:

Group	Table	Action
Redshift	*	Environment /Name=HVR_S3_REGION /Value=s3_bucket_region

