

Datometry® Hyper-Q™
Installation for Azure Synapse Analytics

Version 3.x
May 17, 2021

Copyright © 2021 Datometry Inc.

Copyright © 2020-1 Datometry Inc., All rights reserved.

Datometry believes the information in this publication to be accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” Datometry MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any Datometry software described in this publication requires an applicable software license.

Datometry and Datometry Hyper-Q are trademarks of Datometry Inc. All other trademarks used herein are the property of their respective owners.

Table of Contents

1	About Datometry Hyper-Q Installation for Azure Synapse Analytics	4
	Scope of Document.....	4
	Intended Audience.....	4
2	What is Datometry Hyper-Q?	4
	Hyper-Q Solution Architecture	5
	Terminology	5
3	Prerequisites	6
	Datometry Hyper-Q Software and License	6
	Access to Microsoft ODBC Suite	6
	Azure Synapse Instance	6
	Azure Subscription Privileges and Credentials.....	7
4	Set up Datometry Metadata Store	8
	Create the Schema for Metadata Store	8
	Create a Dedicated User to Access Metadata Store	9
5	Installation and Setup	9
	Create a Hyper-Q Virtual Machine.....	9
	Install or Register Microsoft ODBC Drivers	10
	Configure Hyper-Q for Use with Azure Synapse	12
6	After You Install Hyper-Q	13
	Start the Hyper-Q Service	13
	Stop the Hyper-Q Service.....	13
	Check the Status of the Hyper-Q Service	13
	Validate the Hyper-Q Installation	14
	Configure Hyper-Q to Start Automatically at Boot Time	14
7	Troubleshooting the Hyper-Q Installation	14
	Collecting Log Files for Troubleshooting a Hyper-Q Installation.....	14
	Hyper-Q Fails to Start.....	15
	Client Cannot Connect to Hyper-Q	15
	Client Connects to Hyper-Q but Cannot Connect to Azure Synapse.....	15
	The Metadata Store is Not Accessible	15
	Missing Metadata Store Permissions.....	15
	Data encryption enabled	16
8	Example dtm.ini Configuration File	17

1 About Datometry Hyper-Q Installation for Azure Synapse Analytics

Scope of Document

Datometry Hyper-Q Installation for Azure Synapse Analytics describes how to install Datometry® Hyper-Q™ for use with Microsoft Azure Synapse Analytics. This document describes:

- How to create a virtual machine on Azure.
- Install all required software packages for Hyper-Q.
- Configure Hyper-Q to work with an Azure Synapse instance.

Intended Audience

This document is intended for experienced Linux system administrators who are familiar with Microsoft Azure, virtual machine technology, and data center operations.

2 What is Datometry Hyper-Q?

Datometry Hyper-Q is a database virtualization platform that makes Teradata applications instantly interoperable with Azure Synapse Analytics. Applications originally written or configured for Teradata can run directly on Azure Synapse without changing SQL code or APIs. To do so, Hyper-Q is situated between your database applications and Azure Synapse. Sitting in the data path, Hyper-Q accepts all communication from the various applications and translates Teradata-specific SQL statements into T-SQL for Azure Synapse.

Hyper-Q provides full functional and accurate emulation of all commonly used Teradata features. Even complex features such as recursion or Global Temporary Tables, which do not have a direct corresponding implementation on Azure Synapse, are emulated by Hyper-Q. Additionally, Hyper-Q supports loaders and utilities commonly used in Extract, Transform, and Load (ETL) or Extract, Load, and Transform (ELT) processing.

Hyper-Q Solution Architecture

Hyper-Q, together with Azure Synapse, provides a complete replacement of legacy Teradata appliances. Figure 1 depicts the architecture of the Hyper-Q for Azure Synapse Analytics solution.

Hyper-Q connects client applications with Azure Synapse. In a production environment, two redundant Hyper-Q instances in conjunction with a load balancer provide high-availability and fault-tolerance.

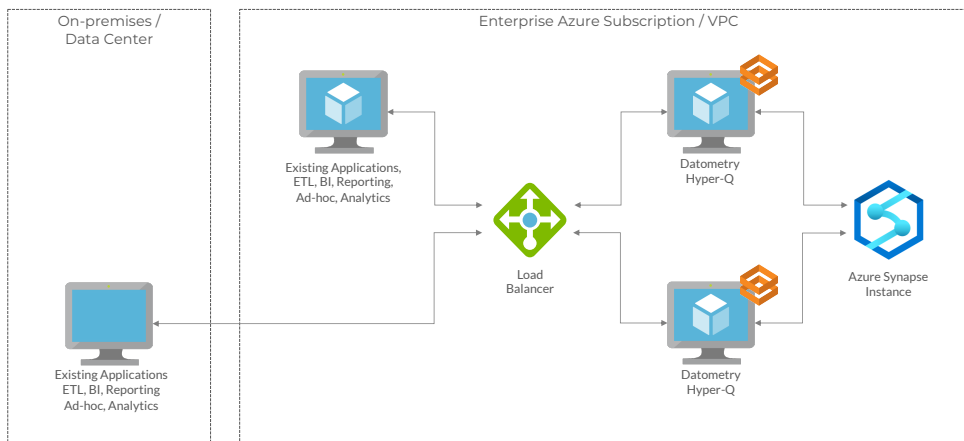


Figure 1. Hyper-Q for Azure Synapse architecture with multiple Hyper-Q instances

Terminology

We use the following terminology in this guide:

- *Service* refers to the Azure Synapse service in your subscription.
- *Azure Synapse instance* or *instance* refers to the actual data warehouse. An instance is identified by its hostname, port, and instance name.
- *Master database* or *master* refers to a special database within the same service. The master database is identified by its hostname, port, and the term **master** as instance name.

To learn more about Azure Synapse, see [Azure Synapse Analytics](#) in the Microsoft Azure documentation.

3 Prerequisites

Verify that the Microsoft Azure environment into which you want to deploy Hyper-Q meets the following requirements.

Datometry Hyper-Q Software and License

Datometry makes available the following items needed for the installation. These files need to be copied to the Hyper-Q VM for installation.

Filename	Description
<code>hyperq-version-sqlldw.tar.gz</code>	RPM packages with which to install the Hyper-Q software.
<code>dtm.lic</code>	License file specific to the customer's subscription plan.

Access to Microsoft ODBC Suite

Access to the Microsoft ODBC Suite URL from within your Azure subscription is required to register the Microsoft ODBC driver and `mssql-tools` package. Verify that you have access to the following URL: <https://packages.microsoft.com/config/rhel/7>

Note: If the Microsoft ODBC Suite URL is not accessible from within your Azure subscription due to restrictive policies, you can install these packages manually. You must first download the ODBC drivers outside of your Azure subscription, and then upload them to the VIRTUAL MACHINE you create to run Hyper-Q.

Azure Synapse Instance

You must have available the name of the Azure Synapse instance, hostname, and network port to which clients can connect. To learn more, see [Azure Synapse Analytics](#) in the Microsoft Azure documentation.

Azure Subscription Privileges and Credentials

You must have the following privileges within the Azure subscription and credentials on the Azure Synapse instance to which you are connecting Hyper-Q. You need these credentials to install, not to operate, the system. The installation automatically creates an account with which to operate Hyper-Q.

Note: Avoid using highly privileged user IDs such as an overall Azure administrative account. Datometry defaults to the customer's information access management (User Management) policies for individuals that need access to the system. Password requirements are as recommended by the customer.

Permissions in Azure Subscription

The user performing the installation must have an account in the Azure subscription, with permissions to provision virtual machines within that subscription. To learn more, see [Azure Synapse Analytics](#) in the Microsoft Azure documentation.

Permissions in Azure Synapse

You must create several database objects in the Azure Synapse instance. Therefore, you must have sufficient permissions to perform the following operations in the Azure Synapse Analytics instance that you are connecting to Hyper-Q:

- Create a schema.
- Create a user.
- Create a role.
- Add a user to the newly created role.
- Permission to create a login in the master database

4 Set up Datometry Metadata Store

Hyper-Q maintains Metadata Store in Azure Synapse. Metadata Store contains annotations for advanced emulations of complex SQL constructs.

Create the Schema for Metadata Store

Using a query editor such as Synapse Studio, create the following objects in the Azure Synapse instance.

```
CREATE SCHEMA [__DTM_MDSTORE];
CREATE TABLE [__DTM_MDSTORE].[MDSTORE_TBL] (
    [schemaname] VARCHAR (256),
    [objectname] VARCHAR (256),
    [colname] VARCHAR (256),
    [propname] VARCHAR (256),
    [propvalue] VARCHAR (MAX),
    [seqno] INT
) WITH (DISTRIBUTION = ROUND_ROBIN, HEAP);
CREATE TABLE [__DTM_MDSTORE].[MDSTORE_WRKTBL] (
    [schemaname] VARCHAR (256),
    [objectname] VARCHAR (256),
    [colname] VARCHAR (256),
    [propname] VARCHAR (256),
    [propvalue] VARCHAR (MAX),
    [seqno] INT
) WITH (DISTRIBUTION = ROUND_ROBIN, HEAP);
GRANT SELECT ON SCHEMA::[__DTM_MDSTORE] TO PUBLIC;
```

Note: Granting SELECT access to PUBLIC on the schema of Metadata Store is critical for the emulation of various Teradata functions, including those using system tables so that Hyper-Q can properly process queries and return result sets.

Create a Dedicated User to Access Metadata Store

Metadata Store is maintained and accessed using a specific user ID. The user ID to be used for this purpose is set in the `dtm.ini` configuration file. To learn more, see “Example `dtm.ini` Configuration File” on page 17.

Note: Metadata Store does not require a separate login/user to be created as shown below. However, it is best practice to create such a dedicated user whose authorization is limited only to Metadata Store. If you decide not to create a dedicated user, skip the remainder of this section.

In order to set up a dedicated user ID for accessing Metadata Store first create a login with a valid password.

1. Connect to the master database of the Azure Synapse instance.

```
CREATE LOGIN [DTM_MDS_LOGIN] WITH PASSWORD = 'password';
```

2. Connect to the default database of the Azure Synapse instance and execute the following commands to associate the login accordingly.

```
CREATE USER [DTM_MDS_USER] FOR LOGIN [DTM_MDS_LOGIN];
CREATE ROLE [DTM_MDS_ROLE];
ALTER AUTHORIZATION ON SCHEMA::[__DTM_MDSTORE] TO [DTM_MDS_ROLE];
GRANT SELECT, INSERT, UPDATE, DELETE, ALTER ON
    SCHEMA::[__DTM_MDSTORE] TO [DTM_MDS_ROLE];
GRANT CREATE TABLE TO [DTM_MDS_ROLE];
GRANT CREATE VIEW TO [DTM_MDS_ROLE];
GRANT CREATE PROCEDURE TO [DTM_MDS_ROLE];
EXEC sp_addrolemember 'DTM_MDS_ROLE', 'DTM_MDS_USER';
```

5 Installation and Setup

This section describes how to set up an individual Hyper-Q instance. For a setup that includes multiple active instances of Hyper-Q the following steps need to be repeated for each instance.

Create a Hyper-Q Virtual Machine

Create a virtual machine for Hyper-Q following the procedures for provisioning virtual machines in Azure. To learn more, see [Azure Virtual Machine](#) in the Microsoft Azure documentation.

The virtual machine must meet the following system requirements.

Resource	Minimum	Recommended
VM type	D8v3	D16v3
Disk	Standard SSD, 500GB	Premium SSD, 1TB
OS	RedHat Enterprise Linux (RHEL) 7.x or CentOS 7.x	

Provision the Virtual Machine

Once you create and instantiate a virtual machine that meets the above requirements, it is now the Hyper-Q VM.

Required Ports for Hyper-Q Virtual Machine

Hyper-Q uses designated ports for communication. Configure the Hyper-Q VM so that it permits inbound connections using the following ports.

Port	Purpose
22	SSH connection for maintenance during operations.
1025	Default port for all existing Teradata clients.
5432	Default port for Datometry Client Tools and Utilities.

Note: Certain RedHat virtual machine images come preconfigured with the `iptables` software firewall active, allowing only incoming SSH connections (Port 22). To learn how to view and modify firewall configuration, see <https://www.redhat.com/sysadmin/secure-linux-network-firewall-cmd>

Important: Configure all ports that are not required for your deployment to drop connection requests and not be open for any application or system traffic. As an additional security measure, restrictions should be placed on the available ports to only allow traffic from IP addresses known to be acceptable associated with your corporate network.

Install or Register Microsoft ODBC Drivers

You must either register or install Microsoft ODBC drivers.

Register the Microsoft ODBC Driver Suite

Registering the Microsoft repositories does not install the ODBC driver suite directly but registers the repository. The installation uses this repository to automatically download and install the ODBC software.

1. Verify that you can access the ODBC repository URL from within your Azure subscription.

```
https://packages.microsoft.com/config/rhel/7/prod.repo
```

2. Log in as superuser and register the Microsoft ODBC repositories.

```
$ sudo su
# curl https://packages.microsoft.com/config/rhel/7/prod.repo >
  /etc/yum.repos.d/mssql-release.repo
# exit
```

(Optional) Install the Microsoft ODBC Driver Suite from RPMs

As an alternative to registering the ODBC drivers, you can manually install the drivers from RPMs.

1. Obtain the most recent version of the `msodbcsql17` and `mssql-tools` packages from:
<https://packages.microsoft.com/rhel/7/prod/>
2. Use the `yum` command to install both packages.

```
$ sudo yum install msodbcsql17-VSN.x86_64.rpm  
mssql-tools-VSN.x86_64.rpm
```

Install the Hyper-Q Software

To install the software, perform the following steps:

1. Copy the `hyperq-version-sqldw.tar.gz` to the Hyper-Q VM.
2. Un-compress the file.

```
$ tar xvf hyperq-version-sqldw.tar.gz
```

This creates the following four RPM files:

```
hyperq-version.x86_64.rpm  
hyperq-monitors-version.x86_64.rpm  
hyperq-sqldw-version.x86_64.rpm  
hyperq-system-version.x86_64.rpm
```

3. Install the software packages as superuser, replacing the `version` placeholder with the actual version number.

```
$ sudo yum install hyperq-version.x86_64.rpm  
hyperq-monitors-version.x86_64.rpm  
hyperq-sqldw-version.x86_64.rpm  
hyperq-system-version.x86_64.rpm
```

Installation Results

The installation process performs the following steps:

- Creates the group `dtm` and a user ID `dtm` at the OS level.
- Installs all binaries and support files under `/opt/datometry`.
- Creates a service `dtm` within the system.
- Adjusts system controls by installing the file `/etc/sysctl.d/dtm.conf` and applies its configuration settings to the Hyper-Q VM. Changes take effect immediately without requiring a reboot.

Note: The installation creates the service, however, it does not enable automatically starting the service after a reboot. See “Configure Hyper-Q to Start Automatically at Boot Time” on page 14.

Configure Hyper-Q for Use with Azure Synapse

The configuration parameters for Hyper-Q are in the file `/opt/datometry/config/dtm.ini`. Modify the following lines to configure Hyper-Q for use with the Azure Synapse environment.

Note: All string values on the left-hand side of an assignment in `dtm.ini` must be enclosed in double quotes. For example: `"DTM_MDS_USER"`.

For an example of a `dtm.ini` configuration file, see “Example dtm.ini Configuration File” on page 17

In the `[endpoints]` section, locate the line below and replace `asa_database` with the name of your Azure Synapse instance.

```
"endpoint".database = asa_database
```

4. In `[gateways]` section, locate the line below and replace `asa_host` with the hostname of your Azure Synapse instance.

```
"gateway".host = asa_host
```

5. In the `[metadata_stores]` section, locate the lines below and replace `asa_database`, `asa_username`, and `asa_user_password` with the name of your Azure Synapse instance, and the username and password to be used to access Metadata Store.

```
"mdstore".database = asa_database  
"mdstore".user = asa_username  
"mdstore".password = asa_user_password
```

Install the Hyper-Q License File

Datometry provides you with a unique license file, `dtm.lic`, which is subject to your license agreement. Ensure that:

- The license is valid. The expiration date in the file corresponds to the terms of your license agreement.
- The license file is in `/opt/datometry/config`, and is readable by the user `dtm:dtm`.
- There is only one file ending in the `.lic` extension in the directory `/opt/datometry/config`.

Hyper-Q services will not start if any of the above conditions are not met.

Install Hyper-Q User Defined Functions

Hyper-Q utilizes user defined functions (UDFs) to implement standard functionality. These are referred to as *Standard UDFs*. The UDFs must be installed in the Azure Synapse instance and updated with each new Hyper-Q release.

1. Locate the UDF source files in the folder `/opt/datometry/dtm/udf/standard`.
2. (Optional) If using an MD store setup deviating from the default outlined in this guide, you must adjust the schema in the UDF source files to match it by replacing `__DTM_MDSTORE` with the appropriate schema name.
3. Execute the SQL contained in all UDF source files directly on the Azure Synapse instance using a database client to connect directly to the instance. Suitable clients include `sqlcmd`, SSMS, and SQL Workbench/J.

Note: Some database clients require explicit batch handling to split multiple requests in the same file. To learn how to process batch files with `sqlcmd`, see KB article: <https://support.datometry.com/hc/en-us/articles/360046308391>

6 After You Install Hyper-Q

After you install Hyper-Q, you start and manage the deployment using the `systemctl` utility.

Start the Hyper-Q Service

Use `systemctl` to start the Hyper-Q service.

```
$ sudo systemctl start dtm
```

The service is now operational, and applications can connect to the Hyper-Q service and query Azure Synapse.

Stop the Hyper-Q Service

Use `systemctl` to stop the Hyper-Q service.

```
$ sudo systemctl stop dtm
```

Check the Status of the Hyper-Q Service

Use `systemctl` to perform a status check for the Hyper-Q service.

```
$ systemctl status dtm
```

Validate the Hyper-Q Installation

You can validate that Hyper-Q is correctly installed and fully operational.

1. Log into Hyper-Q using a Teradata client and valid credentials for the Azure Synapse Analytics instance Hyper-Q is associated with
2. Execute the command `dtm show version`.
3. This returns the version of the installed Hyper-Q software. Compare the returned version numbers with the version of the RPM packages you installed. For more information on the RPM packages, see “Install the Hyper-Q Software” on page 11.

```
dtm show version
```

4. Using the Teradata SQL `SHOW VIEW view_name` statement to display a view in your Azure Synapse instance.

Confirm the definition of the view displays correctly.

```
SHOW VIEW view_name
```

Configure Hyper-Q to Start Automatically at Boot Time

The installation registers Hyper-Q as a service with the system but does not enable it to start at boot time. You can configure Hyper-Q to automatically start when you boot the Hyper-Q VM.

After rebooting the Hyper-Q VM, run the command:

```
$ sudo systemctl enable dtm
```

7 Troubleshooting the Hyper-Q Installation

The Hyper-Q installation troubleshooting topics provide solutions to problems that you might encounter during the Hyper-Q deployment process.

For support FAQs and troubleshooting information, visit the Datometry Help Center at: <https://support.datometry.com/>

Collecting Log Files for Troubleshooting a Hyper-Q Installation

Hyper-Q logs critical startup messages as well as errors at runtime to the file `error_log-DATE_TIME.csv` located in the directory `/opt/datometry/logs`.

At startup and at midnight a new log file is created with the current date and time as part of the filename.

Hyper-Q Fails to Start

Hyper-Q will fail to start for the following reasons.

Missing License File

- Copy a valid license file to `/opt/datometry/config/dtm.lic`.

Duplicate License Files

- Ensure only one file ending in `.lic` is located in `/opt/datometry/config`.

Expired License

1. Request a new license file from Datometry Support.
2. Ensure the new license has not expired by inspecting the file, then install license file.

To learn more, see “Install the Hyper-Q License File” on page 1211.

Client Cannot Connect to Hyper-Q

If your client cannot connect to Hyper-Q verify that the endpoint is correctly configured.

- Ensure the database name and port configuration in the "endpoints" section of the `dtm.ini` file is correct.

Client Connects to Hyper-Q but Cannot Connect to Azure Synapse

The connection to the Azure Synapse instance is not configured correctly.

- Ensure database name and port configuration in "gateways" section of the `dtm.ini` file is correct.

The Metadata Store is Not Accessible

If the Metadata Store is not accessible, the client receives the error “DTM5306: Invalid MDStore table”.

Ensure database, username, and password configuration in the "metadata_stores" section of the `dtm.ini` file is correct, and that the credentials being used can access Metadata Store in Azure Synapse.

Missing Metadata Store Permissions

If permissions have not been assigned to Metadata Store, queries involving the catalog (Information Schema, DBC) will fail.

- Grant `SELECT` permissions on the Metadata Store schema to `PUBLIC`. See “Create the Schema for Metadata Store” on page 8.

Data encryption enabled

Either the client cannot log in the database or executing queries on the session fails with “Cannot call a method on closed connection” or a similar error message.

- Ensure the client is not requesting data encryption.

8 Example dtm.ini Configuration File

The following is an example of the `dtm.ini` file used to configure Hyper-Q for Azure Synapse .

Placeholder Name	Replacement Value
<code>asa_database</code>	Name of the Azure Synapse instance
<code>asa_hostname</code>	Hostname of Azure Synapse instance
<code>asa_username</code>	User ID for Metadata Store access
<code>asa_user_password</code>	Password for the Metadata Store user

```
version = "3.3.34"
```

```
[endpoints]
```

```
"endpoint".port = 1025
"endpoint".database = "asa_database"
"endpoint".app_protocol = odbc
"endpoint".type = teradata
"endpoint".version = "140000"
"endpoint".is_case_sensitive = false
```

```
[gateways]
```

```
"gateway".name = "gateway_msadw"
"gateway".is_odbc_gateway = true
"gateway".connection_string = "Driver=ODBC Driver 17 for SQL
Server;Server=__HOST__;Port=__PORT__;Database=__DATABASE__);Uid=__USER__);Pwd=__PASSWORD__"
"gateway".host = "asa_hostname"
"gateway".port = 1433
"gateway".tls = prefer
"gateway".ipv = 4
"gateway".type = msadw
"gateway".version = "100000"
"gateway".bulk_load_utility = bcp
"gateway".bulk_load_chunk_size = 100MB
"gateway".bulk_load_max_errors = 2000000000
"gateway".bulk_load_detect_encoding = false
"gateway".locale = "en_US.iso88591"
"gateway".locale = "en_US.CP1252"
```

[metadata_stores]

```
"mdstore".name = "MDSTORE"  
"mdstore".gateway = "gateway_msadw"  
"mdstore".user = "asa_username"  
"mdstore".password = "asa_user_password"  
"mdstore".database = "asa_database"  
"mdstore".mdstore_schema = "__DTM_MDSTORE"  
"mdstore".mdstore_table = "MDSTORE_TBL"  
"mdstore".mdstore_worktable = "MDSTORE_WRKTBL"
```

[pools]

```
"defpool".gateway = ["gateway_msadw"]  
"defpool".name = "pool_msadw"  
"defpool".backlog = 10  
"defpool".capacity = 100  
"defpool".active = 30  
"defpool".tx_sync = shared
```

[policies]

```
"defpolicy".protocol = all  
"defpolicy".database = all  
"defpolicy".user = all  
"defpolicy".ip = "0.0.0.0"  
"defpolicy".ip_prefix = 0  
"defpolicy".pool = "pool_msadw"  
"defpolicy".auth_method = passthru
```