

Datometry® Hyper-Q™
Log File Uploader for Azure Synapse Analytics

Version 3.4
May 17, 2021

Copyright © 2021 Datometry Inc.

Copyright © 2021 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

About Datometry Hyper-Q Log File Uploader for Azure Synapse Analytics..	4
Scope of Document.....	4
Intended Audience.....	4
What is Datometry Hyper-Q Log File Uploader?.....	4
Prerequisites	4
Install Datometry Hyper-Q for Azure Synapse Analytics.....	4
Create an Azure Blob Storage Account	4
Prepare a Log Upload Configuration File.....	5
Log Upload Configuration File Parameters	5
Example az_blob_append.ini Configuration File.....	7
Configure Hyper-Q for use with Azure Blob Storage.....	8

About Datometry Hyper-Q Log File Uploader for Azure Synapse Analytics

Scope of Document

Datometry Hyper-Q Log File Uploader for Azure Synapse Analytics describes how to use Azure Blob storage to store Datometry® Hyper-Q™ log files. This document describes:

- System requirements to enable Azure Blob storage for use with Hyper-Q.
- Authentication methods you can use to access Azure Blob storage from the Hyper-Q VM.
- How to configure Hyper-Q to work with Azure Blob storage.

Intended Audience

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

What is Datometry Hyper-Q Log File Uploader?

Logs are essential for identifying and troubleshooting short-term problems but are less effective at identifying long-term trends. Older entries may get overwritten, deleted, or lost. Archiving log files allows you to identify patterns over a longer period of time than rolling log files.

Hyper-Q Log File Uploader lets you archive Hyper-Q log files to Azure Blob storage. Logs stored in Azure Blob storage remain indefinitely or until you delete them. You can also define an access policy within Azure storage that limits access to the logs to certain users.

Prerequisites

Before you begin, ensure that you have the following information available, and that your Microsoft Azure environment meets the following requirements.

Install Datometry Hyper-Q for Azure Synapse Analytics

Verify that you have installed Hyper-Q in the Microsoft Azure environment. To learn how to install Hyper-Q, see *Datometry Hyper-Q Installation for Azure Synapse Analytics*.

Create an Azure Blob Storage Account

You must create an Azure Storage account and have available the following information:

- Azure account URL
- Name of the Blob container you intend to use for storage.
- Allow public access to the container.
- Enable access using one of the following authentication methods:
 - Shared access signature (SAS) tokens.

- Account key.
- Managed Identity (System and User).

Prepare a Log Upload Configuration File

You specify which Hyper-Q log files to upload to Azure Blob storage using a configuration file that you create. You can either create the file using a text editor on your local computer and upload it, or you can create it in place on the Hyper-Q VM.

1. In a text editor, create a file and name it `az_blob_append.ini`.
2. Create a configuration with which to upload the Hyper-Q log files to Azure Blob storage.

See “Example `az_blob_append.ini` Configuration File” on page 5 for a list of configuration parameters and an example Log Upload Configuration File Parameters `az_blob_append.ini` configuration file.

3. Save in UTF-8 format and close the file.

Log Upload Configuration File Parameters

When you prepare your configuration file, you set parameters and values to provide data for the uploading of Hyper-Q log files to Azure Blob storage.

Parameter	Description
<code>prefix</code>	(Optional) A string that is prefixed to the blob name. You can use this to prepend a hostname or other identifier to duplicate files. Additionally, you can add a trailing slash (/) to replicate a file system directory path. By default, <code>prefix</code> is an empty string.
<code>account_url</code>	HTTPs URL to the destination storage account. Anything after the trailing slash (/) is ignored. For example: <code>https://dtm-log.blob.core.windows.net/</code>
<code>container_name</code>	Name of the Blob storage container you intend to use for storage.
<code>account_key</code>	(Optional) This parameter is required if you are authenticating using a Storage Account Key.

	<p>To learn more, see the Microsoft Azure article Choose how to authorize access to blob data in the Azure portal.</p>
storage_access_signature	<p>(Optional) This parameter is required if you are authenticating using a SAS token.</p> <p>Important: Do not enclose the parameter value in quotes.</p> <p>To learn more, see the Microsoft Azure article Create SAS tokens for Document Translation processing.</p>
client_id	<p>(Optional) The Client ID to use for User Managed Identities. This parameter is required if you are authenticating using User Managed Identities.</p> <p>To learn more, see the Microsoft Azure article What are managed identities for Azure resources?</p>

Example az_blob_append.ini Configuration File

This is an example az_blob_append.ini configuration file.

```
[/opt/datometry/logs/error_log*.csv]
prefix=myhost/error_logs/
account_url=https://dtm-log.blob.core.windows.net/
container_name=logs
; account_key=optional_storage_account_key
storage_access_signature=storage_access_signature
; client_id=optional_managed_identity_client_id_for_user_assigned_identities

[/opt/datometry/logs/default_tracer*.csv]
prefix=myhost/default_tracer/
account_url=https://dtm-log.blob.core.windows.net/
container_name=logs
account_key=optional_storage_account_key

[/opt/datometry/logs/another_file_using_managed_identities]
prefix=myhost/another_file/
account_url=https://dtm-log.blob.core.windows.net/
container_name=logs
```

Configure Hyper-Q for use with Azure Blob Storage

You configure Hyper-Q to upload log files to Azure storage using `systemd` timers and services. The `systemd` service lets you schedule tasks using dedicated units called timers.

To schedule log file upload tasks with the `systemd` timers and services:

1. Log into each Hyper-Q VM in the Azure workspace using SSH.
2. Create the file `/opt/datometry/config/az_blob_append.ini`.

If you have not done so, create a `az_blob_append.ini` configuration file as described in “Log Upload Configuration File Parameters” on page 5.

3. Enable and start the `systemd` timer and service units.

```
$ systemctl enable az_blob_append.timer
$ systemctl start az_blob_append.timer
$ systemctl enable az_blob_append
$ systemctl start az_blob_append
```

4. Ensure the timer has a trigger time using the `systemctl status` command.

```
systemctl status az_blob_append.timer
```

5. Verify that the service is being triggered correctly.

The `systemctl status` command should not return a value of more the 60s ago. In the example below the time value is 15s ago.

```
systemctl status az_blob_append
Active: inactive (dead) since Fri 2021-02-26 22:16:51 UTC; 15s ago
```

6. To ensure that there are no configuration issues and that the log files are loading into Azure Blob storage, run the `journalctl` command to view the `systemd` logs.

```
$ journalctl -u az_blob_append
```