Requirements for FTP, SFTP, and SharePoint WebDAV

Contents

General

HVR supports different kinds of file locations; regular ones (a directory on a local file system), FTP file locations, SFTP file locations, and WebDAV file locations (this is the protocol used by HVR to connect to Microsoft SharePoint).

Generally, the behavior of HVR replication is the same for all of these kinds of file locations; capture is defined with action **Capture** and integration is defined with **Integrate**. All other file location parameters are supported and behave normally.

If HVR will be using a file protocol to connect to a file location (e.g. FTP, SFTP or WebDAV), it can either connect with this protocol directly from the hub machine, or it first connects to a remote machine with HVR's own remote protocol and then connect to the file location from that machine (using FTP, SFTP or WebDAV).

A small difference is the timing and latency of capture jobs. Normal file capture jobs check once a second for new files, whereas if a job is capture from a non local file location, then it only checks every 10 seconds. Also if **Capture** is defined without **/DeleteAfterCapture**, then the capture job may have to wait for up to a minute before capturing new files; this is because these jobs rely on comparing timestamps, but the file timestamps in the FTP protocol have a low granularity (minutes not seconds).

A proxy server to connect to FTP, SFTP or WebDAV can be configured with action **LocationProperties /Proxy**.

HVR uses the cURL library to communicate with the file systems.

Integrate Limitations

By default, for file-based target locations, HVR does not replicate the **delete** operation performed at the source location.

FTPS, WebDAV: Server Authentication

FTP and WebDAV support certificates for authentication. These are held in the certificate directory **$HVR_HOME/lib/cert** (see section Files in Hvr). The following files are included or can be easily created:

- **ca-bundle.crt**: Used by HVR to authenticate SSL servers (FTPS, secure WebDAV, etc). It can be overridden by creating new file *host* **.pub_cert** in this same certificate directory. No authentication is done if neither file is found. Delete or move both files to disable FTPS authentication. This file can be copied from e.g. **/usr/share/ssl/certs/ca–bundle.crt** on Unix/Linux.
- *host* **.pub_cert**: Used to override **ca-bundle.crt** for server verification for *host* .
  FTP connections can be unencrypted or they can have three types of encryption; this is called FTPS, and should not be confused with SFTP. These FTPS encryption types are SSL/TLS implicit encryption (standard port: 990), SSL explicit encryption and TLS explicit encryption (standard port: 21).

Note that if the FTP/SFTP connection is made via a remote HVR machine, then the certificate directory on the remote HVR machine is used, not the one on the hub machine.

FTPS, WebDAV: Client Authentication

Normally, clients (HVR) are authenticated using a username and password. In addition to this, you can use client-side SSL certificates.
To set this up, you need to have a public/private key pair in PEM format. Place them in **$HVR_HOME/lib/cert** as: *client***.pub_cert** and *client***.priv_key**, and add the environment variables to a file location:

- **Environment /Name=HVR_SSL_CLIENT_CERT /Value=** *client* **.pub_cert**
- **Environment /Name=HVR_SSL_CLIENT_KEY /Value=** *client* **.priv_key**
  If you have a password/phrase on your key, set:
- **Environment /Name=HVR_SSL_CLIENT_KEY_PASSWD /Value=** *passwd*
  You can encrypt this password for HVR by running:

  ```
  $ hvrcrypt client passwd
  ```

- If you have a PKCS#12 or PFX file, you can convert it to PEM format using openssl:

  ```
  $ openssl pkcs12 -in cert.p12 -clcerts -nokeys -out client.pub_cert
  $ openssl pkcs12 -in cert.p12 -nocerts -out client.priv_key
  ```

SFTP: Server Authentication

A file (named **$HVR_CONFIG/files/known_hosts**) is used internally during SFTP connections. It is updated the first time HVR connects to reach the SFTP machine. On Unix or Linux this file can be initialized by copying it from **$HOME/.ssh/known_hosts**.

Note that if the FTP/SFTP connection is made via a remote HVR machine, then the certificate directory on the remote HVR machine is used, not the one on the hub machine.

SFTP: Client Authentication

Normally, clients (HVR) are authenticated using a username and password. Instead of this, you can use client-side keys.
To set this up, you need to have a public/private key pair. Place them in **$HVR_HOME/lib/cert**, and add the environment variables to a file location:

- **Environment /Name=HVR_SSH_PUBLIC_KEY /Value=** *client***.pub**
- **Environment /Name=HVR_SSH_PRIVATE_KEY /Value=** *client*
  If you have a password/phrase on your private key, you can set it in the password field of the location.

You can generate keys using:

```
$ ssh-keygen -f client
```

WebDAV: Versioning

HVR can replicate to and from a WebDAV location which has versioning enabled. By default, HVR's file integrate will delete the SharePoint file history, but the file history can be preserved if action **LocationProperties /StateDirectory** is used to configure a state directory (which is the then on the HVR machine, outside SharePoint). Defining a **/StateDirectory** outside SharePoint does not impact the 'atomicity' of file integrate, because this atomicity is already supplied by the WebDAV protocol.