HDFS Authentication and Kerberos

Contents

HVR supports connecting to both insecure and Kerberos-secured HDFS clusters. Information on setting up Hadoop for HVR can be found in Requirements for HDFS.

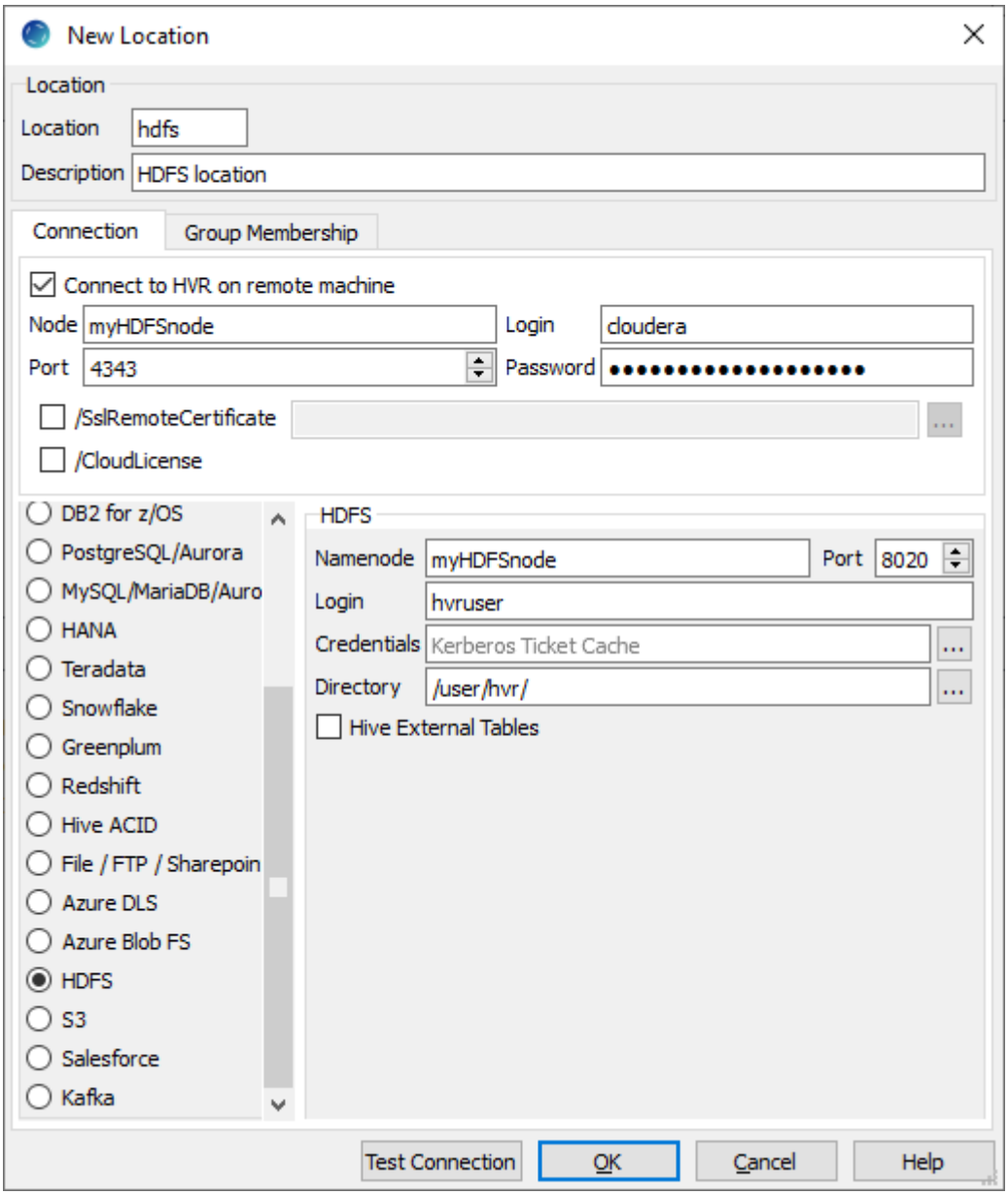
Insecure HDFS clusters are protected at network level by restricting which hosts can establish a connection. Any software that is able to establish a connection can claim to act as any user on HDFS system.

Secure HDFS clusters are protected by Kerberos authentication. Both HDFS servers (Hadoop **NameNode**, Hadoop **DataNode**) and HDFS clients (HVR) authenticate themselves against a central Kerberos server which grants them a ticket. Client and server exchange their tickets, and both verify each other's identity. HVR must have access to cluster configuration files (in **$HVR_HADOOP_HOME**) in order to verify **NameNode**'s and **DataNode**'s Kerberos identities.

Insecure Clusters

Insecure clusters require only a HDFS username. Enter this user in **Login** field of the location dialog. This username will be used in HDFS file permissions, such as any files created by HVR in HDFS will be owned by this user.

If the **Login** field is empty, HVR's operating system username will be used (If HVR is running on a remote location, remote operating system username will be used).



Kerberized Clusters

Accessing the kerberized clusters require authentication against a Kerberos server which can be achieved in the following two ways :

- Accessing Kerberized Clusters with Ticket Cache File
- Accessing Kerberized Clusters with Keytab File

To access the kerberized clusters, HVR requires the following:

- If HVR is installed on a separate server (outside the Hadoop cluster),

    a. Kerberos should be installed and configured on the server where HVR is installed. The configuration should be same as that of the hadoop cluster's configuration.
    b. Configure the Hadoop client. For more information, see section Hadoop Client in Requirements for HDFS.
    c. Using the cluster manager's web interface, Hadoop client configuration files should be downloaded from the Hadoop cluster to the server where HVR is installed.

Client Configuration Files

Client configuration files are required if Kerberos authentication is used in the Hadoop cluster or else they can be useful for debugging. Client configuration files contain settings for different services like HDFS, and others. If the HVR integrate server is not part of the cluster, it is recommended to download the configuration files for the cluster so that the Hadoop client knows how to connect to HDFS.

The client configuration files for Cloudera Manager or Ambari for Hortonworks can be downloaded from the respective cluster manager's web interface. For more information about downloading client configuration files, search for "Client Configuration Files" in the respective documentation for Cloudera and Hortonworks.

- If HVR is installed on a Hadoop edge node, the steps mentioned above are not required for HVR to connect to the kerberized cluster.
- If the Kerberos server issues tickets with strong encryption keys then install Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files onto the JRE installation that HVR uses. JCE can be downloaded from Oracle website.

To verify the connectivity between HVR and HDFS, execute the following command on the server where HVR is installed:

```
$HADOOP_HOME/bin/hadoop fs -ls hdfs://cluster/
```

Accessing Kerberized Clusters with Ticket Cache File

A short-term ticket is issued by authenticating against Kerberos server with **kinit** command. This ticket usually expires in a timeframe of a few hours or a few days depending on the Kerberos configuration. An interactive operating system shell connection will renew Kerberos ticket as required, but this would not automatically happen on a non-interactive HVR setup.

- Command **kinit** can be used to obtain or renew a Kerberos ticket-granting ticket. For more information about this command, refer to MIT Kerberos Documentation.
- Command **klist** lists the contents of the default Ticket Cache file, also showing the default filename. For more information about this command, refer to MIT Kerberos Documentation.

By default, HVR is configured for the path of the Kerberos Ticket Cache file, and assumes tickets will be renewed by the user as needed. HVR will pick up any changes made to the Ticket Cache file automatically. It is user's responsibility to set up periodic renewal jobs to update the file before Kerberos tickets expire.

The Ticket Cache file must be located on the HVR remote location if HVR is running on a remote machine. The file must have correct file system permissions for HVR process to read.

To use a Ticket Cache file with HVR, enter the Kerberos principal's user part to **Login** field and full path of the Ticket Cache file to **Credentials** field in the location dialog.

- **Alternative configuration 1**
  Leave **Credentials** field empty, and configure your channel with:
  Environment /Name=HVR_HDFS_KRB_TICKETCACHE /Value=*ticketcache.file*
- **Alternative configuration 2**
  Leave **Login** and **Credentials** field empty, and configure your channel with:
  Environment /Name=HVR_HDFS_KRB_PRINCIPAL /Value=*full_principal_name (e.g. username@REALM)*
  Environment /Name=HVR_HDFS_KRB_TICKETCACHE /Value=*ticketcache.file*

Accessing Kerberized Clusters with Keytab File

Non-interactive daemons can also authenticate to the Kerberos server using a **Keytab** file. A **keytab** file holds a list of entries, consisting of Kerberos principal name, key version, encryption type, and an encryption key. Encryption key is generated with the password provided at creation time. It is crucial that an entry is added to this file for each possible encryption type your Kerberos server might ask for. It is not possible for a Kerberos client to respond to an encryption type not found in the file.

The **keytab** files can be copied across computers, they are not bound to the host they were created on. The **keytab** file is only used to acquire a real ticket from the Kerberos server when needed. If the file is compromised, it can be revoked from the Kerberos server by changing password or key version.

Keytab files are created using the **ktutil** command. Depending on your system Kerberos package, usage will vary. For more information about this command, refer to MIT Kerberos Documentation.

The **keytab** file must be located on the HVR remote location if HVR is running on a remote machine. The file must have correct file system permissions for HVR process to read.

To use a **keytab** file with HVR, leave **Login** and **Credentials** fields of the location dialog blank and configure your channel with:

> **Environment /Name=HVR_HDFS_KRB_PRINCIPAL /Value=***full_principal_name (e.g. username@REALM)*
>
> **Environment /Name=HVR_HDFS_KRB_KEYTAB /Value=***keytab.file*

Accessing Kerberized Clusters with HDFS Impersonation

In most cases, your HDFS cluster will be configured to map the username part of your Kerberos principal as the HDFS username (*"username"* in principal *"username@KERBEROS.REALM"*). If you need to use a different HDFS username than your Kerberos principal, and your cluster is configured to allow this setup, then you can configure HVR in the following way.

- To use Ticket Cache with HDFS impersonation, set your HDFS impersonate username in the **Login** entry of the Location dialog, leave **Credentials** entry blank, and define **$HVR_HDFS_KRB_PRINCIPAL** and **$HVR_HDFS_KRB_TICKETCACHE** environment actions as described in the **Accessing Kerberized Clusters with Ticket Cache File** section above.
- To use **keytab** with HDFS impersonation, set your HDFS impersonate username in the **Login** entry of the Location dialog, leave **Credentials** entry blank, and define **$HVR_HDFS_KRB_PRINCIPAL** and **$HVR_HDFS_KRB_KEYTAB** environment actions as described in the **Accessing Kerberized Clusters with Keytab File** section above.