How-To Library

Informatica®

Configuring Big Data Management® to Access an SSL Enabled Hadoop Cluster

Abstract

SSL certificates create a foundation of trust by establishing a secure connection between the Hadoop cluster and the Informatica® domain. When you configure the Informatica domain to communicate with an SSL-enabled cluster, the Developer tool client can import metadata from sources on the cluster, and the Data Integration Service can run mapping jobs on the cluster.

Supported Versions

- Informatica Big Data Management 10.1.1 Hotfix 1, 10.2

Table of Contents

Overview

When you configure the Informatica domain to communicate with an SSL-enabled cluster, the Developer tool client can import metadata from sources on the cluster, and the Data Integration Service can run mapping jobs on the cluster.

To access an SSL-enabled cluster, edit the connection string to enable SSL and import security certificates to clients.

For version 10.1.1x, you must then configure the Data Integration Service properties with the location of SSL truststore files on the cluster. The cluster might use different locations for truststore files, depending on the resource location. For example, cluster might store truststore files in one location for data in HDFS and truststore files in another location for Hive data, or truststore files in a single location for both HDFS and Hive.

Steps for Version 10.1.1 Hotfix 1

Perform the steps in this section if you are configuring access to SSL-enabled Cloudera or Hortonworks clusters from Big Data Management version 10.1.1x.

Creating and Configuring Security Certificates and Truststore Files

When you use custom, special, or self-signed security certificates to secure the Hadoop cluster, Informatica clients that connect to the cluster require these certificates to be present in the client machine truststore.

Import Security Certificates

To connect to the Hadoop cluster to develop a mapping, the Developer tool requires security certificate aliases on the machine that hosts the Developer tool. To run a mapping, the machine that hosts the Data Integration Service requires these same certificate alias files.

Perform the following steps from the Developer tool host machine and from the Data Integration Service host machine:

1. Run the following command to export the certificates from the cluster:

   ```
   keytool -export -alias <alias name> -keystore <custom.truststore file location> -file
   <exported certificate file location> -storepass <password>
   ```

   For example:

   ```
   <java home>/jre/bin/keytool -export -alias <alias name> -keystore ~/custom.truststore -
   file ~/exported.cer
   ```

   The command produces a certificate file.
2. Choose to import security certificates to an SSL-enabled domain or a domain that is not SSL-enabled.
   - If the domain is SSL-enabled, import the certificate file to the following location: `<Informatica installation directory>/services/shared/security/infa_truststore.jks`
   - If the domain is not SSL-enabled, import the certificate file to the following location: `<Informatica installation directory>/java/jre/lib/security/cacerts`

Configure the Data Integration Service to Use Truststore File Paths

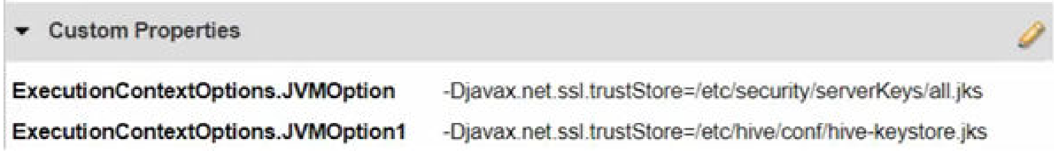
To enable the Data Integration Service to access truststore files on the Hadoop cluster, perform the following steps to configure Data Integration Service properties with truststore file paths:

1. Get the location of truststore files from the cluster manager administration web page.
   If you do not have access to the cluster manager, ask the cluster administrator for the path to the truststore files for the resources you want to access.
2. To enable access to Hive sources in Native mode, copy the truststore files to the corresponding location in the Hadoop distribution directory of the Informatica domain machine.
   For example, if the truststore file is located at `/etc/security/serverKeys/all.jks` on the cluster, copy the file to the same location in the Hadoop distribution directory on the domain machine: `/etc/security/serverKeys/all.jks`. Create the directory if it does not exist.
3. In the Administrator tool, select the Data Integration Service in the **Domain Navigator**. Click the **Properties** tab to display Data Integration Service properties.
4. Click the **Edit** icon for Custom Properties.
   The **Edit Custom Properties** dialog box appears.
5. Enter the following name for the custom property: JVMOption.
   If a JVMOption custom property exists, then increment the name with an integer like JVMOption1.
6. In the **Value** pane for the property, type the following value:

   ```
   -Djavax.net.ssl.trustStore=<path to the truststore file on the cluster>
   ```

   For example:

   ```
   -Djavax.net.ssl.trustStore=/etc/security/serverKeys/all.jks
   ```
7. Click **OK**.
8. To access additional resources that use truststore files in different locations, repeat steps 4-7. Increment the custom property name with an integer.
   The following image shows custom properties that allow the Data Integration Service to access truststore files in two different locations:



Custom Properties

ExecutionContextOptions.JVMOption	-Djavax.net.ssl.trustStore=/etc/security/serverKeys/all.jks
ExecutionContextOptions.JVMOption1	-Djavax.net.ssl.trustStore=/etc/hive/conf/hive-keystore.jks

Note: The Data Integration service converts the name of custom properties to add the "ExecutionContextOptions." prefix when you recycle the service.

9. Recycle the Data Integration Service.
10. If you have ever run a mapping on the Blaze engine, you must stop the Grid Manager application on the Data Integration Service host machine.
    a. Run the following command to list existing YARN applications:
    ```
    yarn application -list
    ```
    b. In the list of YARN applications, identify the application ID for the Blaze Grid Manager.
    c. Run the following command to stop the Grid Manager application:
    ```
    yarn application -kill <application ID>
    ```

Now you can run the mapping from the Developer tool. The Data Integration System imports the truststore certificates at run time.

Steps for Version 10.2

Perform the steps in this section if you are configuring access to SSL-enabled Hadoop clusters from Big Data Management version 10.2.

Step 1. Configure the Hive Connection for SSL-Enabled Clusters

If you created the Hive connection when you created cluster configurations, the cluster configuration creation wizard enables access to a cluster that uses SSL. If you manually created a Hive connection, you must configure the connection string properties to enable access to a cluster that uses SSL.

If you manually created a Hive connection, add the following property-value pair to the metadata connection string and data access connection string properties:

```
ssl=true
```

For example:

```
jdbc:hive2://<hostname>:<port>/<db>;ssl=true
```

Note: Insert the `ssl=true` flag before the Kerberos principal element when you create the Hive connection manually.

Step 2. Import Security Certificates from an SSL-Enabled Cluster

When you use custom, special, or self-signed security certificates to secure the Hadoop cluster, Informatica services that connect to the cluster require these certificates to be present on the machines that run the application services. Use the keytool utility to import certificates from the cluster.

For more information about the keytool utility, refer to the Oracle documentation.

Note: If a MapR cluster is SSL-enabled, you do not have to import the security certificates. Make sure that the MapR client on the Data Integration Service and Metadata Access Service machines is configured to access an SSL-enabled cluster. For more information about installing and configuring the MapR client, see the *Informatica Big Data Management Hadoop Integration Guide*.

If a Cloudera CDH or Hortonworks HDP cluster uses SSL, import security certificates from the cluster to the Data Integration Service and Metadata Access Service machines.

1. Run the following keytool -exportcert command on the cluster to export the certificates:

   ```
   keytool -exportcert
   -alias <alias name>
   -keystore <custom.truststore file location>
   -file <exported certificate file location>
   -storepass <password>
   ```

   Where:

   - -alias specifies the alias name associated with the truststore file.
   - -keystore specifies the location of the truststore file on the cluster.
   - -file specifies the file name and location for the exported certificate file.
   - -storepass specifies the password for the keystore on the cluster.

   The keytool -exportcert command produces a certificate file associated with the alias.

2. Run the following keytool -importcert command on one Data Integration Service machine to import the security certificates:

   ```
   keytool -importcert -trustcacerts
   -alias <alias name>
   -file <exported certificate file location>
   -keystore <java cacerts location>
   -storepass <password>
   ```

   Where:

   - -alias specifies the alias name associated with the certificate file.
   - -file specifies the file name and location of the exported certificate file.
   - -keystore specifies the location of the truststore file on the domain.
   - -storepass specifies the password for the keystore on the domain.

   Important: Import the certificate files one time and then copy them to all machines that host the Data Integration Service and Metadata Access Service. If the Data Integration Service runs on a grid, mappings that you push to the Hadoop environment can fail with initialization errors due to inconsistent binary hex values.

   Depending on whether the Informatica domain uses SSL, you specify the keystore location as follows:

   - If the domain is SSL-enabled, import the certificate file to the following location:
     `<Informatica installation directory>/services/shared/security/infa_truststore.jks`
   - If the domain is not SSL-enabled, import the certificate file to the following location:
     `<Informatica installation directory>/java/jre/lib/security/cacerts`

   The keytool -importcert command imports the security certificates to the keystore location you specify.

Example: Import Security Certificates

The big data environment includes a Cloudera CDH cluster that uses SSL and an Informatica domain that does not use SSL. You export the security certificate for the user bigdata_user1 from the custom.keystore on the Cloudera CDH cluster to the file exported.cer. Then, you import the export.cer certificate file to the Informatica domain location.

1. Run the following export command:

   ```
   keytool -exportcert -alias bigdata_user1 -keystore ~/custom.truststore -file ~/
   exported.cer
   ```

2. Run the following import command on the Data Integration Service machine:

   ```
   keytool -importcert -alias bigdata_user1 -file ~/exported.cer -keystore <Informatica
   installation directory>/java/jre/lib/security/cacerts
   ```

3. Copy the certificate file to all other machines that host the Data Integration Service and the Metadata Access Service.

Author

Mark Pritchard