Open for Innovation
KNIME

Secured Cluster Connection Guide for KNIME Server

KNIME AG, Zurich, Switzerland
Version 4.1 (last updated on 2019-09-23)

Table of Contents

Overview

KNIME Server executes workflows, that may try to access Kerberos-secured services such as Apache Hive™, Apache Impala™ and Apache Hadoop® HDFS™.

This guide describes how to configure KNIME Server so that it can **authenticate** itself against Kerberos and then **impersonate** its own users towards Kerberos-secured cluster services.

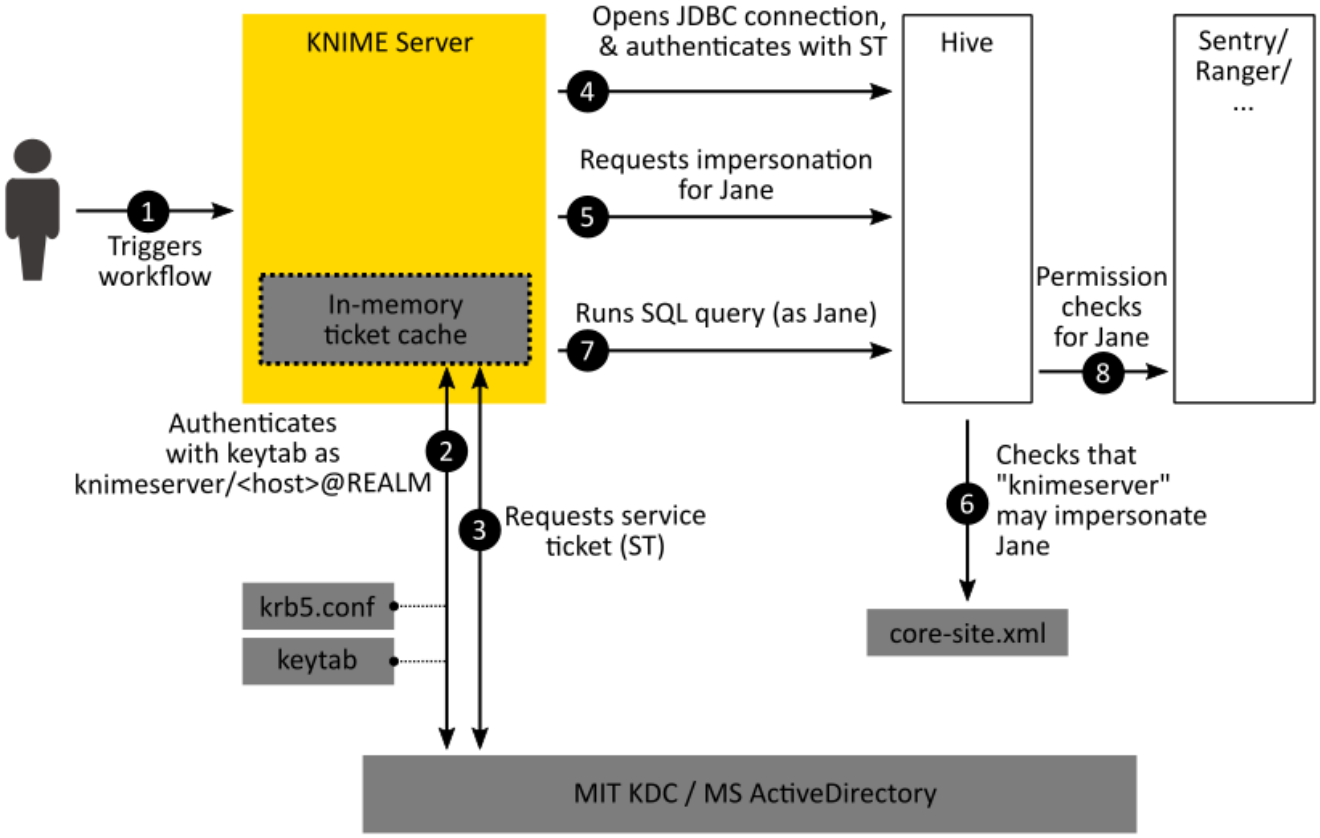
What is user impersonation?

With user impersonation, it does not matter whether a user runs a workflow in KNIME Analytics Platform or on KNIME Server. In both cases, all operations on the cluster will be performed **as that particular user** and the **same permissions and authorization rules** apply. This has the following advantages:

- Workflows that access a secured cluster run without modifications on KNIME Server.
- Authorization to access cluster resources (Hive tables, HDFS files, …) is administered with the usual mechanisms, e.g. Apache Sentry™ or Apache Ranger™.

How does user impersonation work?

Let us assume that a user Jane runs a workflow on KNIME Server. The workflow is supposed to run a Hive query.



The following sequence of events now takes place:

1. She starts a workflow that connects to Hive. This workflow is now executed on KNIME Server, not Jane's machine.
2. When the *Hive Connector* node in the workflow is executed, KNIME Server first checks for a TGT (ticket granting ticket) in its own ticket cache. If there is no TGT, it reads the `krb5.conf` configuration file, connects to the KDC and authenticates itself. Instead of Jane's credentials, it uses the credentials configured on KNIME Server, i.e. a service principal such as `knimeserver/<host>@REALM` and a keytab file. The TGT will be stored in an in-memory ticket cache.
3. To make a JDBC connection to Hive, the Hive JDBC driver on KNIME Server still requires an ST (service ticket), which it now requests from the KDC. The ST is only valid for connections between KNIME Server and the Hive instance.
4. Now, the Hive JDBC driver opens a connection to Hive and authenticates itself with the ST as `knimeserver/<host>@REALM`.
5. Since the workflow was started by Jane, the JDBC driver tells Hive, that all operations shall be performed **as user Jane**.
6. Hive consults the Hadoop `core-site.xml` to verify that KNIME Server is indeed allowed to impersonate Jane. If not, it will return an error.
7. Now, the workflow submits an SQL query via the JDBC connection. The query is

executed on the cluster **as user Jane**.

8. Hive checks whether user Jane has the necessary permissions to run the query. It employs its usual permission checking mechanism, e.g. Apache Sentry™ or Apache Ranger™. The query will succeed or fail, depending on whether **Jane** has the necessary permissions.

Prerequisites

Setting up KNIME Server for Kerberos authentication and user impersonation has the following prerequisites.

- For Kerberos:
  - An existing Kerberos KDC such as MIT Kerberos or Microsoft ActiveDirectory
  - A service principal for KNIME Server. The recommended format is `knimeserver/<host>@<REALM>`, where
    - `<host>` is the fully-qualified domain name of the machine where KNIME Server runs,
    - `<REALM>` is the Kerberos realm.
  - A keytab file for the KNIME Server service principal.
  - A Kerberos client configuration file (`krb5.conf`). The recommended way to obtain this file, is to copy the `/etc/krb5.conf` from a node in the cluster. Alternatively, you can create the file yourself (see Creating your own `krb5.conf`).
- For the cluster:
  - A Kerberos-secured cluster.
  - An account with administrative privileges in the cluster management software, that can configure and restart cluster services. On Cloudera CDH this means a Cloudera Manager account, on Hortonworks HDP this means an Apache Ambari™ account.
- For KNIME Server:
  - An existing KNIME Server installation.
  - An account with administrative privileges on the machine where KNIME Server is installed. This accounts needs to be able to edit the KNIME Server configuration files and restart KNIME Server.

Supported cluster services

KNIME Server supports Kerberos authentication and user impersonation for connections to the following services:

- Apache Hive
- Apache Impala
- Apache Hadoop HDFS (including HttpFS)
- Apache Livy

Setting up Kerberos authentication

This section describes how to set up KNIME Server to authenticate itself against Kerberos.

Kerberos client configuration (krb5.conf)

The KNIME Server Executor needs to read the `krb5.conf` file during Kerberos authentication. Please append the following line to the `knime.ini` file of the KNIME Server Executor:

```
-Djava.security.krb5.conf=<PATH>①
```

① Replace `<PATH>` with the full path to your `krb5.conf` file, for example `C:\krb5.conf`. The file must be readable by the KNIME Server Executor process. It is recommended to store the file outside of the KNIME Server Executor installation folder, to avoid accidentally deleting it during upgrades.

For reference, Possible locations for `krb5.conf` describes the process by which KNIME Server Executor locates the `krb5.conf` file.

Kerberos principal and keytab file

KNIME Server Executor needs credentials during Kerberos authentication. Please specify the service principal and the keytab file by adding the following lines to the KNIME Server `preferences.epf`:

```
/instance/org.knime.bigdata.commons/org.knime.bigdata.config.kerberos.user=<PRINCIPAL>①
/instance/org.knime.bigdata.commons/org.knime.bigdata.config.kerberos.keytab.file=<PATH>
②
```

① Replace `<PRINCIPAL>` with the service principal, for example `knimeserver/host.company.com@REALM`.

② Replace `<PATH>` with the full path to the keytab file, for example `C:\knimeserver.keytab`. The file must be readable by the KNIME Server Executor process. It is recommended to store the keytab file outside of the KNIME Server Executor installation folder, to avoid accidentally deleting it during upgrades.

Setting up proprietary JDBC drivers (optional)

Each KNIME Server Executor is a headless instance of KNIME Analytics Platform. If the *KNIME Big Data Connectors* extension is installed, KNIME Server Executor includes a fully-functional embedded JDBC driver for Hive and Impala. If you choose to use this driver **(recommended)**, then you can skip this section.

The currently embedded JDBC driver is the open-source Apache Hive™ JDBC driver version `1.1.0-cdh5.13.0` (Release Notes). The driver has been verified to be compatible with:

- HDP 2.2.4 and later
- CDH 5.3 and later

The following subsections only apply if you choose to set up a proprietary JDBC driver for Hive/Impala. These drivers are vendor-specific. Please consult the subsection that applies to your Hadoop vendor:

- Cloudera JDBC drivers (Hive and Impala)
- Hortonworks Hive JDBC drivers

Cloudera JDBC drivers (Hive and Impala)

1. Download the newest JDBC drivers from the Cloudera website (login required):
   - Hive JDBC Connector for Cloudera Enterprise
   - Impala JDBC Connector for Cloudera Enterprise
2. Once downloaded, extract the contained `Cloudera_HiveJDBC41_<version>.zip` (`Cloudera_ImpalaJDBC41_<version>.zip`) file into an empty folder. Note that you need one folder for the Hive driver and another one for the Impala driver.
3. Verify that the resulting two folders contain JAR files.

4. It is recommended to place the folders outside of the KNIME Server Executor installation folder, to avoid accidentally deleting them during upgrades.

Now, configure KNIME Server Executor to load the JDBC drivers from these folders by adding the following lines to the KNIME Server `preferences.epf`:

```
/instance/org.knime.workbench.core/database_drivers=<FOLDERLIST>①
/instance/org.knime.workbench.core/database_timeout=86400②
```

① Replace `<FOLDERLIST>` with a semicolon-separated list of the folders, in which the JAR files reside. Note that the file` format requires characters such as the colon (':') and the backslash ('\') to be escaped. For example `C:\Drivers\HiveJDBC41` needs to be written as `C\:\\Drivers\\HiveJDBC41`.

② This increases the database connect and query timeout from its default 15s to 1 day, which prevents long-running queries from timing out.

Hortonworks Hive JDBC drivers

1. Download the newest Hive JDBC driver from the Hortonworks website.
    - Locate the **Hortonworks JDBC Driver for Apache Hive**.
    - Download the JDBC 4.1 driver.
2. Once downloaded, extract the ZIP file into an empty folder.
3. Verify that the folder now contains JAR files.
4. It is recommended to place the folder outside of the KNIME Server Executor installation folder, to avoid accidentally deleting them during upgrades.

Now, configure KNIME Server Executor to load the JDBC drivers from these folders by adding the following lines to the KNIME Server `preferences.epf`:

```
/instance/org.knime.workbench.core/database_drivers=<FOLDERLIST>①
/instance/org.knime.workbench.core/database_timeout=86400②
```

① Replace `<FOLDERLIST>` with a semicolon-separated list of the folders, in which the JAR files reside. Note that the file` format requires characters such as the colon (':') and the backslash ('\') to be escaped. For example `C:\Drivers\HiveJDBC41` needs to be written as `C\:\\Drivers\\HiveJDBC41`.

② This increases the database connect and query timeout from its default 15s to 1 day, which prevents long-running queries from timing out.

Setting up user impersonation

This section describes how to set up both ends of user impersonation, which requires configuration on two sides: KNIME Server **and** the cluster.

User impersonation on KNIME Server

By default, KNIME Server tries to impersonate its users on Kerberos-secured connections towards the following cluster services:

- HDFS (including httpFS)
- Apache Livy
- Apache Hive

Impersonation for HDFS and Apache Livy is done automatically and does not require any further setup. Connections to Apache Hive require further setup steps depending on the used JDBC driver as described below.

To activate user impersonation for the **embedded** Apache Hive JDBC driver, add the following lines to the KNIME Server `preferences.epf`:

```
/instance/org.knime.bigdata.commons/org.knime.bigdata.config.kerberos.jdbc.impersonation
.flag=true
/instance/org.knime.bigdata.commons/org.knime.bigdata.config.kerberos.jdbc.impersonation
.param=hive.server2.proxy.user\={1}
```

This will append the `hive.server2.proxy.user` JDBC parameter to every JDBC connection made via the embedded JDBC driver. The placeholder `{1}` is automatically replaced by the login name of the KNIME Server user.

The **embedded** Apache Hive JDBC Driver (for Impala) does not support impersonation. For impersonation when connecting to Impala please setup the **proprietary** driver as described below.

To activate user impersonation for **proprietary** Simba based JDBC drivers, such as the ones provided by Cloudera and Hortonworks, add the following lines to the KNIME Server `preferences.epf`:

```
/instance/org.knime.bigdata.commons/org.knime.bigdata.config.kerberos.jdbc.impersonation
.flag=true
/instance/org.knime.bigdata.commons/org.knime.bigdata.config.kerberos.jdbc.impersonation
.param=DelegationUID\={1}
```

This will append the `DelegationUID` JDBC parameter to every JDBC connection made via the proprietary JDBC drivers. The placeholder `{1}` is automatically replaced by the login name of the KNIME Server user.

Please check the driver documentation for the appropriate impersonation parameter if you are using any third party JDBC driver.

To deactivate user impersonation for all cluster services see Deactivating user impersonation on KNIME Server.

User impersonation on Apache Hadoop™ and Apache Hive™

Apache Hadoop™ and Apache Hive™ consult the `core-site.xml` file to determine whether KNIME Server is allowed to impersonate users.

Changing the `core-site.xml` file must be done via Ambari (on HDP) or Cloudera Manager (on CDH). A restart of the affected Hadoop services is required.

Please add the following settings to the Hadoop `core-site.xml` on your cluster:

```
<property>
    <name>hadoop.proxyuser.knimeserver.hosts</name>①
    <value>*</value>
</property>
<property>
    <name>hadoop.proxyuser.knimeserver.groups</name>①
    <value>*</value>
</property>
```

① If you have created a service principal for KNIME Server other than `knimeserver/<host>@<REALM>`, then adjust the property name accordingly.

User impersonation on Apache Impala™

Apache Impala™ requires a configuration setting to determine whether KNIME Server is allowed to impersonate users.

It is recommended to also enable Apache Sentry™ authorization in Apache Impala™. Otherwise Impala continues to perform all read and write operations with the privileges of the `impala` user.

The required steps are similar to Configuring Impala Delegation for Hue. In Cloudera Manager, navigate to **Impala** > **Configuration** > **Impala Daemon Command Line Argument Advanced Configuration Snippet (Safety Valve)** and add the following line:

```
-authorized_proxy_user_config='hue=*;knimeserver=*'
```

Then click **Save** and restart all Impala daemons. Please note:

- This will make `hue` and `knimeserver` the only services that can impersonate users in Impala. If other services should be allowed to do the same, they need to be included here as well.
- If you have created a service principal for KNIME Server other than `knimeserver/<host>@<REALM>`, then adjust the above setting accordingly.

Advanced configuration

This section offers supplementary material to address more advanced setups.

Creating your own `krb5.conf`

A minimal `krb5.conf` can look like this:

```
[libdefaults]
default_realm = MYCOMPANY.COM①

[realms]
MYCOMPANY.COM = {
  kdc = kdc.mycompany.com②
}
```

① The name of your Kerberos realm.

② The fully-qualified domain name or IP of the Kerberos KDC.

Adjust these values as appropriate for your setup. Depending on your setup, more configuration settings may be necessary. The `krb5.conf` format is fully described as part of the MIT Kerberos Documentation.

Possible locations for `krb5.conf`

Like any other Java program, KNIME Server Executor tries to read the `krb5.conf` file from a set of default locations. KNIME Executor tries the following locations in the given order:

1. First it checks whether the `java.security.krb5.conf` system property is set. If so, it will try to read the file from the location specified in this system property. The system property can be set as described in Kerberos client configuration (krb5.conf).
2. Otherwise, it will try to read the `krb5.conf` from the Java Runtime Environment of KNIME Server Executor:
    - `<knime-executor-installation>/plugins/org.knime.binary.jre.<version>/jre/lib/security/krb5.conf`
3. If this fails too, it will try the following operating system dependent locations:
    - Windows: `C:\Windows\krb5.ini`
    - Linux: `/etc/krb5.conf`

You can place `krb5.conf` in any of the above locations. It is however recommended to set the `java.security.krb5.conf` system property in `knime.ini`.

Deactivating user impersonation on KNIME Server

By default, KNIME Server tries to impersonate its users on Kerberos-secured connections. To completely deactivate user impersonation (not recommended) on these connections, add the following line to the KNIME Server `preferences.epf`:

```
/instance/org.knime.bigdata.commons/org.knime.bigdata.config.kerberos.impersonation.enabled=false
```

All operations performed on these connections are now subject to the authorization rules that apply to the KNIME Server service principal.

Troubleshooting

Activating Kerberos debug logging

If Kerberos authentication fails, activating Kerberos debug logging may provide insight into why this is happening. To activate Kerberos debug logging, add the following line to the KNIME Server `preferences.epf`:

```
/instance/org.knime.bigdata.commons/org.knime.bigdata.config.kerberos.logging.enabled=true
```

Then, restart the KNIME Server Executor and run a workflow that accesses a Kerberos-secured service. The `knime.log` will then contain Kerberos debug messages. You can find the `knime.log` on the KNIME Server machine under:

```
<server-repository>/runtime/runtime_knime-rmi_<suffix>/.metadata/knime/knime.log
```

Depending on the configuration, `<suffix>` is either a number, a username, or a combination of both.

KNIME AG
Technoparkstrasse 1
8005 Zurich, Switzerland
www.knime.com
info@knime.com