[Skip to end of metadata](https://teraworks.teradata.com/display/NGAP/%28Experimental%29+Knox+with+Kerberos+Enabled#page-metadata-end)

* Created by [Rubio, Paul](https://teraworks.teradata.com/display/~pr185024), last modified on [Dec 09, 2014](https://teraworks.teradata.com/pages/diffpagesbyversion.action?pageId=129239675&selectedPageVersions=6&selectedPageVersions=7)

[Go to start of metadata](https://teraworks.teradata.com/display/NGAP/%28Experimental%29+Knox+with+Kerberos+Enabled#page-metadata-start)

 2. Configure Knox with a Secured Hadoop Cluster

Once you have a Hadoop cluster that is using Kerberos for authentication, you have to do the following to configure Knox to work with that cluster.

**2.1. Configure Knox Gateway on the Hadoop Cluster**

To allow the Knox Gateway to interact with a Keberos protected Hadoop cluster, add a knox user and Knox Gateway properties to the cluster.

1. On *every* Hadoop Master perform the following commands:
   1. Create Unix account for Knox:

useradd -g hadoop knox

* 1. Add the following lines to the core-site.xml on each master node near the end of the file:
  2. <property>
  3. <name>hadoop.proxyuser.knox.groups</name>
  4. <value>users</value>
  5. </property>
  6. <property>
  7. <name>hadoop.proxyuser.knox.hosts</name>
  8. <value>$knox-host</value>

</property>

where *$knox-host* is the fully qualified domain name of the host running the gateway.

|  |  |
| --- | --- |
|  | **Note** |
| You can usually find this by running hostname -f. You can define the knox host as \* for local developer testing if Knox host does not have static IP. |

* 1. Add the following lines to the webhcat-site.xml on each master node towards the end of the file:
  2. <property>
  3. <name>hadoop.proxyuser.knox.groups</name>
  4. <value>users</value>
  5. </property>
  6. <property>
  7. <name>hadoop.proxyuser.knox.hosts</name>
  8. <value>$knox-host</value>

</property>

where *$knox-host* is the fully qualified domain name of the host running the gateway.

|  |  |
| --- | --- |
|  | **Note** |
| You can usually find this by running hostname -f. You can define the knox host as \* for local developer testing if Knox host does not have static IP. |

1. On the Oozie host, add the following lines to the oozie-site.xml near the end of the file:
2. <property>
3. <name>oozie.service.ProxyUserService.proxyuser.knox.groups</name>
4. <value>users</value>
5. </property>
6. <property>
7. <name>oozie.service.ProxyUserService.proxyuser.knox.hosts</name>
8. <value>$knox-host</value>

</property>

where *$knox-host* is the fully qualified domain name of the host running the gateway.

|  |  |
| --- | --- |
|  | **Note** |
| You can usually find this by running hostname -f on that host. You could use \* for local developer testing if Knox host does not have static IP. |

**NOTE: The instructions in red is not possible to do with the current version of Ambari.  Specifically, Ambari doesn't allow properties to be added to the custom HUE-HA group.**

**It may be sufficient to add these properties to the default Hive group - this is something that hasn't been attempted.**

1. On the nodes runnin HiveServer2, add the following properties to the hive-site.xml:

<property>  
  <name>hive.server2.enable.doAs</name>  
  <value>true</value>  
</property>

<property>  
  <name>hive.server2.allow.user.substitution</name>  
  <value>true</value>  
</property>

<property>  
  <name>hive.server2.transport.mode</name>  
  <value>http</value>  
  <description>Server transport mode. "binary" or "http".</description>  
</property>

<property>  
  <name>hive.server2.thrift.http.port</name>  
  <value>10001</value>  
  <description>Port number when in HTTP mode.</description>  
</property>

<property>  
  <name>hive.server2.thrift.http.path</name>  
  <value>cliservice</value>  
  <description>Path component of URL endpoint when in HTTP mode.</description>  
</property>

|  |  |
| --- | --- |
|  | **Note** |
| Some of the properties may already be in the hive-site.xml. Ensure that the values match the ones above. |

**2.2. Add Knox Principal to KDC**

On the KDC, create a Kerberos principal keytab for Knox as follows:

1. SSH to the KDC host.
2. Execute kadmin.local to open an interactive session:

kadmin.local

1. Add a key for knox with the following commands:
2. add\_principal knox/$knox-host@EXAMPLE.COM

ktadd -k /etc/security/keytabs/knox.service.keytab knox/$knox-host@EXAMPLE.COM

where: *$knox-host* is the fully qualified domain name of the Knox Gateway and *EXAMPLE.COM* is the name of your KDC Realm.

1. Close the interactive session:

exit

**2.3. Configure Knox Gateway for Keberos**

After preparing the cluster and creating a keytab for Knox, perform the following procedure to complete the configuration.

1. Copy the Knox keytab to Knox host.
2. Add unix account for the knox user on Knox host as follows:

useradd -g hadoop knox

Copy knox.service.keytab created on KDC host on to the Knox host /etc/knox/conf/knox.service.keytab.

1. Change the owner of the file to the knox user and set the premissions as follows:
2. chown knox knox.service.keytab

chmod 400 knox.service.keytab

NOTE: It's unknown what actions need to be performed in steps 4 and 5.

1. Update krb5.conf at /etc/knox/conf/krb5.conf on Knox host.

|  |  |
| --- | --- |
|  | **Tip** |
| You can also copy the *$gateway\_home*/templates/krb5.conf file provided in the Knox binary download and customize it to suit your cluster. |

1. Update the /etc/knox/conf/krb5JAASLogin.conf on Knox host.

|  |  |
| --- | --- |
|  | **Tip** |
| You can also copy the *$gateway\_home*/templates/krb5JAASLogin.conf file provided in the Knox binary download and customize it to suit your cluster. Replace the *$knox-host* with the Knox Gateway FQDN and *EXAMPLE.COM* with the your KDC Realm. |

1. Update *$gateway\_home*/conf/gateway-site.xml on Knox host by changing the following value:
2. <property>
3. <name>gateway.hadoop.kerberos.secured</name>
4. <value>true</value>
5. <description>Boolean flag indicating whether the Hadoop cluster protected by gateway is secured with Kerberos</description>

</property>

1. Restart Knox as follows:
2. su -l knox -c '$gateway\_home/bin/gateway.sh stop'

su -l knox -c '$gateway\_home/bin/gateway.sh start'

1. Redeploy the Cluster Topology as follows:
   1. Redeploy all Clusters using the following command:

$gateway\_home/bin/knoxcli.sh redeploy

* 1. Verify that a new Cluster Topology WAR was created with the following command:

ls -lh /var/lib/knox/data/deployments

A new file for each with the same timestamp is created.

|  |  |
| --- | --- |
|  | **Note** |
| After you do the above configurations and restart Knox, Knox uses SPNego to authenticate with Hadoop services and Oozie. There is no change in the way you make calls to gateway whether you use cURL or Knox DSL. |