Cloudera Runtime 7.2.11

Managing Apache HBase Security

Date published: 2020-02-29
Date modified: 2022-12-21



https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2026. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

HBase authentication....4
Configuring HBase servers to authenticate with a secure HDFS cluster....4
Configuring secure HBase replication....4
Configure the HBase client TGT renewal period....5
Disabling Kerberos authentication for HBase clients....5

HBase authorization....8

Configuring TLS/SSL for HBase....8
Prerequisites to configure TLS/SSL for HBase....8
Configuring TLS/SSL for HBase Web UIs....8
Configuring TLS/SSL for HBase REST Server....9
Configuring TLS/SSL for HBase Thrift Server....9
Configuring HSTS for HBase Web UIs....10
Configuring the TLS versions for HBase....10

HBase authentication

You must establish a mechanism for HBase servers and clients to securely identify themselves with HDFS, ZooKeeper, and each other. This ensures that hosts are who they claim to be.

To enable HBase to work with Kerberos security, you must enable Kerberos Authentication for Cloudera Runtime and use Kerberos authentication for ZooKeeper. This means that HBase Master, RegionServer, and client hosts must each have a Kerberos principal for authenticating to the ZooKeeper ensemble.

Before you start configuring Kerberos authentication for HBase ensure that:

- Kerberos is enabled for the cluster.
- Kerberos principals for Cloudera Manager Server, HBase and ZooKeeper hosts exist and are available for use.

Cloudera Manager automatically configures authentication between HBase to ZooKeeper and sets up the HBase Thrift gateway to support impersonation (doAs). However, you must manually configure the HBase REST service for Kerberos, as it uses Simple authentication by default, instead of Kerberos.



Note: Impersonation (doAs) cannot be used with Thrift framed transport (TFramedTransport) because SASL does not work with Thrift framed transport.

Although an HBase Thrift server can connect to a secured Hadoop cluster, access is not secured from clients to the HBase Thrift server. To encrypt communication between clients and the HBase Thrift Server you must configure TLS/SSL for HBase Thrift Server.

Configuring HBase servers to authenticate with a secure HDFS cluster

You can configure HBase servers to authenticate with a secure HDFS cluster using Cloudera Manager.

Procedure

1. In Cloudera Manager, select the HBase service.
2. Click the Configuration tab.
3. Use the HBase (Service-Wide) and Security filters.
4. Find the Kerberos Principal property.
5. Ensure the Kerberos principal for the HBase service was generated.
6. Find the HBase Secure Authentication property.
7. Select kerberos as authentication type.
8. Click Save Changes.
9. Restart the role.
10. Restart the HBase service.

Configuring secure HBase replication

You must configure cross realm support for Kerberos, ZooKeeper, and Hadoop to configure secure HBase replication.

About this task

There must be at least one common encryption mode between the two realms.



Note: HBase peer-to-peer replication from a non-Kerberized cluster to a Kerberized cluster is not supported.

Procedure

1. Create krbtgt principals for the two realms.

   For example, if you have two realms called EXAMPLE.com and COMPANY.TEST, you need to add the following principelas: krbtgt/EXAMPLE.COM@COMPANY.TEST and krbtgt/COMPANY.TEST@EXAMPLE.COM
2. Add the two principals at both realms.

```
kadmin: addprinc -e "<enc_type_list>" krbtgt/EXAMPLE.COM@COMPANY.TEST
kadmin: addprinc -e "<enc_type_list>" krbtgt/COMPANY.TEST@EXAMPLE.COM
```

Add rules creating short names in ZooKeeper:

3. Add a system level property in java.env, defined in the conf directory.

   The following example rule illustrates how to add support for the realm called EXAMPLE.COM and have two members in the principal (such as service/instance@EXAMPLE.COM):

```
-Dzookeeper.security.auth_to_local=RULE:[2:\$1@\$0](.*@\\QEXAMPLE.COM\\E
$)s/@\\QEXAMPLE.COM\\E$//DEFAULT
```

   This example adds support for the EXAMPLE.COM realm in a different realm. So, in the case of replication, you must add a rule for the primary cluster realm in the replica cluster realm. DEFAULT is for defining the default rule

Add rules for creating short names in the Hadoop processes:

4. Add the hadoop.security.auth_to_local property in the core-site.xml file in the replica cluster.

   For example to add support for the EXAMPLE.COM realm:

```
<property>
  <name>hadoop.security.auth_to_local</name>
  <value>
    RULE:[2:$1@$0](.*@\QEXAMPLE.COM\E$)s/@\QEXAMPLE.COM\E$//
  DEFAULT
  </value>
</property>
```

Configure the HBase client TGT renewal period

You must configure the HBase Client TGT Renewal Period to a value that allows the user enough time to finish HBase client processes.

An HBase client user must have a Kerberos principal which typically has a password that only the user knows. Configure the maxrenewlife setting for the client's principal to a value that allows the user enough time to finish HBase client processes before the ticket granting ticket (TGT) expires.

For example, if the HBase client processes require up to four days to complete, you should create the user's principal and configure the maxrenewlife setting by using this command:

```
kadmin: addprinc -maxrenewlife 4days
```

Disabling Kerberos authentication for HBase clients

Client applications that run on legacy products and do not have Kerberos enabled, fail to connect to Cloudera Operational Database instances that have Kerberos enabled. You can disable Kerberos authentication in your Cloudera Operational Database instances so that HBase or Phoenix clients can connect seamlessly.

About this task

You can also disable the Kerberos authentication using Cloudera Operational Database CLI while creating an operational database.

You can use the --disable-kerberos option while running the `create-database` command to disable the Kerberos authentication.

`cdp opdb create-database --environment-name` *ENVIRONMENT_NAME* `--database-name` *DATABASE_NAME* `--disable-kerberos`

For example,

```
cdp opdb create-database --environment-name cdp_7215 --database-name cod1 --
disable-kerberos
```

Important: Cloudera recommends that you do not disable Kerberos authentication. You might want to disable it for testing purposes until the legacy applications are restructured to use Kerberos authentication. When you disable Kerberos authentication, Cloudera Operational Database becomes vulnerable to malicious clients.

Procedure

1. In Cloudera Manager, go to the HBase service.
2. In the Configuration tab, set hbase.security.authentication=simple under HBase Secure Authentication and hbase.thrift.security.qop=none under HBase Thrift Authentication.

Figure 1: HBase security authentication

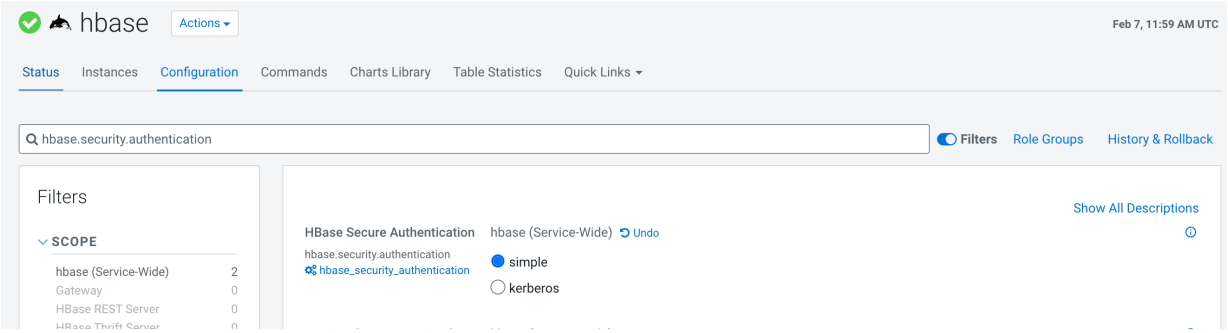
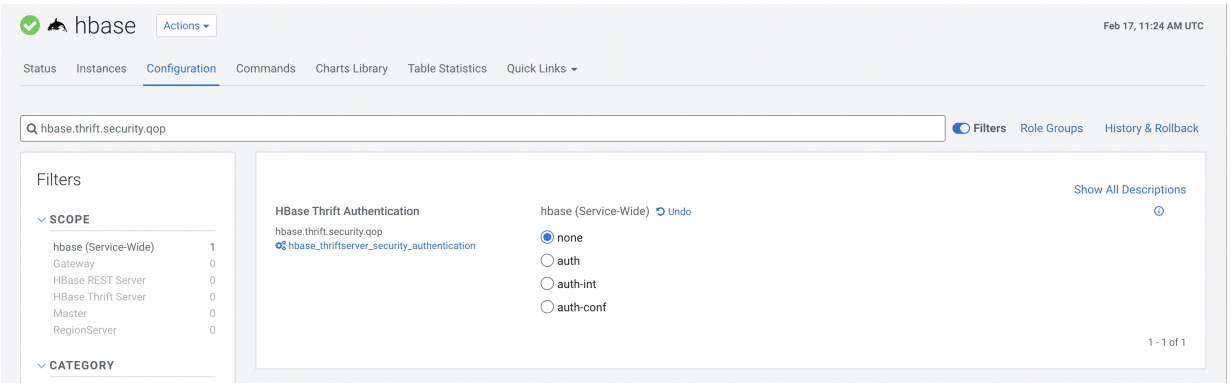


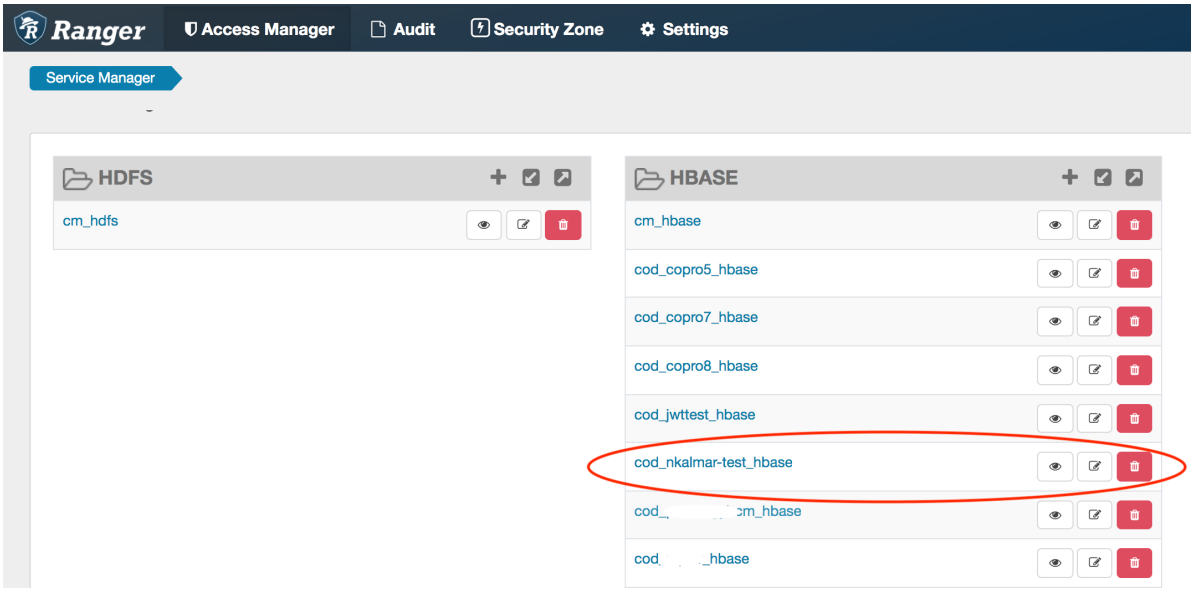
Figure 2: HBase thrift authentication



3. In the Ranger web UI, add the phoenix user. This resolves the impersonation issue.
   a) On the Data lake page, select the Ranger service.

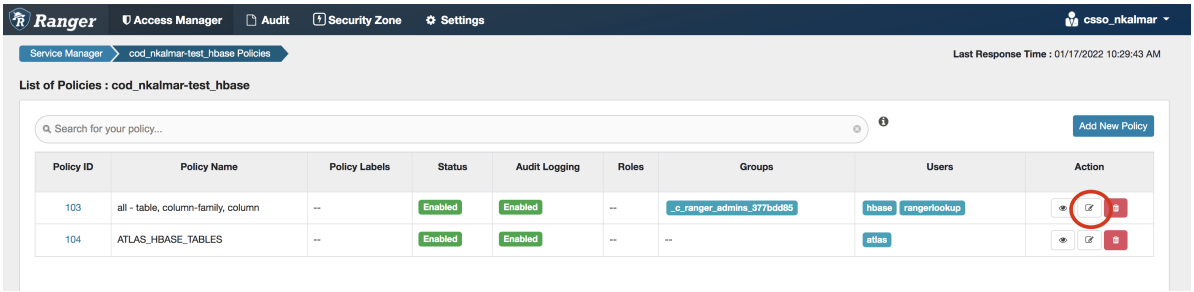
b) On the Ranger Service Manager web UI, find and select the HBase policy for your Cloudera Operational Database instance.

Figure 3: Ranger service manager



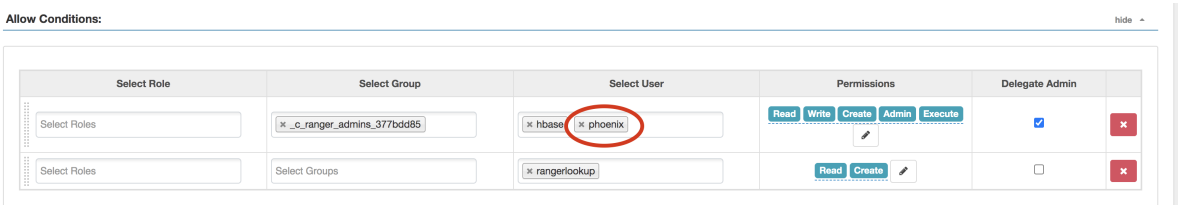
c) Click on the edit button in the Action column to edit the *ALL - TABLE, COLUMN-FAMILY, COLUMN* policy.

Figure 4: Ranger service manager policy



d) In the Ranger service manager edit policy page, add the phoenix user and save the modified policy.

Figure 5: Add phoenix user



Results

Kerberos authentication is disabled in your Cloudera Operational Database instance and legacy HBase or Phoenix clients without having Kerberos enabled, can connect to your Cloudera Operational Database instance.

Related Information

Cloudera Operational Database CLI command reference GitHub repository

Cloudera CLI BETA command reference GitHub repository

HBase authorization

After configuring HBase authentication, you must define rules on resources that are allowed to access. Apache Ranger manages access control through a user interface that ensures consistent policy administration across Cloudera Data Lake components.

Once a user has been authenticated, their access rights must be determined. Authorization defines user access rights to resources. Authorization is concerned with who or what has access or control over a given resource or service. For example, a user may be allowed to create a policy and view reports, but not allowed to edit users and groups. HBase rules can be defined for individual tables, columns, and cells within a table.

For more information about how to set up HBase authorization using Ranger, see *Configure a Resource-based Service: HBase*, and then *Configure a Resource-based Policy: HBase*.

Related Information

Configure a Resource-based Service: HBase

Configure a Resource-based Policy: HBase

Configuring TLS/SSL for HBase

Once all the prerequisites are fulfilled, you can configure TLS/SSL for HBase Web UIs, HBase REST Server and HBase Thrift Server.

Prerequisites to configure TLS/SSL for HBase

Before configuring TLS/SSL for HBase, ensure that all prerequisites are fulfilled.

- Before enabling TLS/SSL, ensure that keystores containing certificates bound to the appropriate domain names will need to be accessible on all hosts on which at least one HBase daemon role is running.
- Keystores for HBase must be owned by the hbase group, and have permissions 0440 (that is, readable by owner and group).
- You must specify absolute paths to the keystore and truststore files. These settings apply to all hosts on which daemon roles of the HBase service run. Therefore, the paths you choose must be valid on all hosts.
- Cloudera Manager supports the TLS/SSL configuration for HBase at the service level. Ensure you specify absolute paths to the keystore and truststore files. These settings apply to all hosts on which daemon roles of the service in question run. Therefore, the paths you choose must be valid on all hosts.

  An implication of this is that the keystore file names for a given service must be the same on all hosts. If, for example, you have obtained separate certificates for HBase daemons on hosts node1.example.com and node2.example.com, you might have chosen to store these certificates in files called hbase-node1.keystore and hbase-node2.keystore (respectively). When deploying these keystores, you must give them both the same name on the target host — for example, hbase.keystore.

Configuring TLS/SSL for HBase Web UIs

You can configure TLS/SSL for HBase Web UIs using Cloudera Manager.

Procedure

1. In Cloudera Manager, select the HBase service.
2. Click the Configuration tab.
3. Use the Scope / HBase (Service-Wide) filter.

4. Search for tls/ssl.
5. Check Web UI TLS/SSL Encryption Enabled.
6. Edit the HBase TLS/SSL properties according to your configuration.

Table 1: HBase TLS/SSL Properties

Property	Description
HBase TLS/SSL Server JKS Keystore File Location	Path to the keystore file containing the server certificate and private key used for encrypted web UIs.
HBase TLS/SSL Server JKS Keystore File Password	Password for the server keystore file used for encrypted web UIs.
HBase TLS/SSL Server JKS Keystore Key Password	Password that protects the private key contained in the server keystore used for encrypted web UIs.

7. Click Save Changes.
8. Restart the HBase service.

Configuring TLS/SSL for HBase REST Server

You can configure TLS/SSL for HBase REST Server using Cloudera Manager.

Procedure

1. In Cloudera Manager, select the HBase service.
2. Click the Configuration tab.
3. Search for tls/ssl rest.
4. Check Enable TLS/SSL for HBase REST Server.
5. Edit the HBase REST Serer TLS/SSL properties according to your configuration.

Table 2: HBase TLS/SSL Properties

Property	Description
HBase REST Server TLS/SSL Server JKS Keystore File Location	The path to the TLS/SSL keystore file containing the server certificate and private key used for TLS/SSL. Used when HBase REST Server is acting as a TLS/SSL server. The keystore must be in JKS format.file.
HBase REST Server TLS/SSL Server JKS Keystore File Password	The password for the HBase REST Server JKS keystore file.
HBase REST Server TLS/SSL Server JKS Keystore Key Password	The password that protects the private key contained in the JKS keystore used when HBase REST Server is acting as a TLS/SSL server.

6. Click Save Changes.
7. Restart the HBase service.

Configuring TLS/SSL for HBase Thrift Server

You can configure TLS/SSL for HBase Thrift Server using Cloudera Manager.

Procedure

1. In Cloudera Manager, select the HBase service.
2. Click the Configuration tab.

3. Search for tls/ssl thrift.
4. Check Enable TLS/SSL for HBase Thrift Server over HTTP.
5. Edit the HBase REST Server TLS/SSL properties according to your configuration.

Table 3: HBase TLS/SSL Properties

Property	Description
HBase Thrift Server over HTTP TLS/SSL Server JKS Keystore File Location	Path to the TLS/SSL keystore file (in JKS format) with the TLS/SSL server certificate and private key. Used when HBase Thrift Server over HTTP acts as a TLS/SSL server.
HBase Thrift Server over HTTP TLS/SSL Server JKS Keystore File Password	Password for the HBase Thrift Server JKS keystore file.
HBase Thrift Server over HTTP TLS/SSL Server JKS Keystore Key Password	Password that protects the private key contained in the JKS keystore used when HBase Thrift Server over HTTP acts as a TLS/SSL server.

6. Click Save Changes.
7. Restart the HBase service.

Configuring HSTS for HBase Web UIs

You can configure HBase to include HTTP headers to enforce the HTTP Strict Transport Security (HSTS) ensuring that a web browser does not load the service information using HTTP.

About this task

Additionally, all attempts to load the information using HTTP will automatically be converted to HTTPS.

Procedure

1. Go to the HBase service.
2. Click the Configuration tab.
3. Select Advanced under Category.
4. Set the following HSTS credentials in HBase Service Advanced Configuration Snippet (Safety Valve) for hbase-site.xml.

```
<property>
<name>hbase.http.filter.hsts.value</name>
<value>max-age=63072000;includeSubDomains;preload</value>
</property>
<property>
<name>hbase.http.filter.csp.value</name>
<value>default-src https: data: 'unsafe-inline' 'unsafe-eval'</value>
</property>
```

5. Restart the HBase service.

Configuring the TLS versions for HBase

Know how to configure the TLS versions for HBase using Cloudera Manager. It is essential to configure these versions to enforce modern security standards, access advanced encryption, and streamline management.

About this task

Standardized TLS version configuration in Cloudera Manager unifies encryption parameters across managed services by using a dual-parameter model and a central feature flag. Cloudera Manager manages Transport Layer Security (TLS) versions through the TLS_ADVANCED_CONTROL feature flag. This feature flag controls whether services utilize legacy configurations or adopt the standardized architecture introduced in Cloudera Manager 7.13.2 SP1. The dual-parameter model separates internal tracking from properties emitted into service configuration files, which allows two distinct configurations to resolve to the same configuration file property.

The feature flag determines parameter behavior and visibility:

- When the feature flag is disabled (OFF), services exhibit legacy behavior, maintaining previous parameters and default values to prevent disruption to existing deployments.
- When the feature flag is enabled (ON), services automatically transition to the unified specifications, access TLS v1.3 options, and ignore legacy customer values in favor of centralized defaults.

The framework automatically handles parameter visibility and emission into configuration files based on the active state of the feature flag. The following table compares the legacy and unified parameter specifications for TLS version control:

The unified model defaults to the Copy from Global setting. This option enables services to directly inherit version requirements from the global Cloudera Manager configuration. By default, the central manager resolves this setting to TLS v1.2 unless you manually update the global properties. Older protocol versions below TLS v1.2 are completely removed from the unified specification to enforce modern security standards.

Procedure

1. Log in to the Cloudera Manager as an administrator.
2. Select the HBase service.
3. Click the Configuration tab.
4. Search for *tls_version*.
5. Select any of the following Supported SSL/TLS versions for HBASE-1 (Service-Wide) or HBase Web UIs, HBase REST Server, HBase Thrift Server, HBase Master, or HBase RegionServer.
   - TLSv1.2: Enable TLS 1.2 exclusively.
   - TLSv1.3: Enable TLS 1.3 exclusively.
   - TLSv1.2, and TLSv1.3: Enable both the versions simultaneously.
   - Copy From Global Settings: Inherit cipher settings directly from Cloudera Manager's global TLS configuration.

Related Information

Transport Layer Security Support and Custom Protocol Configuration
Enabling TLS for database connection