Cloudera Runtime 7.2.10

# Managing Apache HBase Security

**Date published: 2020-02-29**
**Date modified: 2022-01-19**

**https://docs.cloudera.com/**

# Legal Notice

# Contents

# 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.

## Configure 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.

## Configure 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
```

# 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 Platform (CDP) 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.

## Configure 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.

## Configure 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.

## Configure 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.

# Configure 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.