

Securing Cloudera Search

Date published: 2020-01-27

Date modified:



Legal Notice

© Cloudera Inc. 2025. 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

Cloudera Search Security Overview.....	4
Configure TLS/SSL encryption for Solr.....	4
Cloudera Search Authentication.....	5
Configure Kerberos Authentication for Solr.....	6
Enable Kerberos Authentication in Solr.....	6
Set Proxy Server Authentication for Clusters Using Kerberos.....	6
Overview of Proxy Usage and Load Balancing for Search.....	8
Enable LDAP Authentication in Solr.....	8
Enabling Solr Clients to Authenticate with a Secure Solr.....	9
Enable Ranger Authorization in Solr.....	11

Cloudera Search Security Overview

Cloudera Search Security covers the following security aspects:

- Securing network communication

Cloudera Search supports TLS for encrypting communications over a network.

For information on securing communications over a network, see [Encrypting Data in Transit](#).

- Authentication

Cloudera Search supports Kerberos and LDAP for authentication.

For information on enabling Kerberos for Cloudera Search, see [Configuring Authentication in Cloudera Manager](#).

- Authorization

Cloudera Search supports Apache Ranger for authorization.

For information on enabling Ranger for authorization, see [Using Ranger to Provide Authorization in CDP](#).

Related Information

[Enable Kerberos Authentication in Solr](#)

[Enable Ranger Authorization in Solr](#)

Configure TLS/SSL encryption for Solr

Before you begin

Minimum required role: Configurator (Also provided by Cluster Administrator, Full Administrator)

- The Solr service must be running.
- Keystores for Solr must be readable by the solr user. This could be a copy of the Hadoop services' keystore with permissions 0440 and owned by the solr group.
- Truststores must have permissions 0444 (that is, readable by all).
- Specify absolute paths to the keystore and truststore files. These settings apply to all hosts on which daemon roles of the Solr service run. Therefore, the paths you choose must be valid on all hosts.
- In case there is a DataNode and a Solr server running on the same host, they can use the same certificate.

For more information on obtaining signed certificates and creating keystores, see [Encrypting Data in Transit](#). You can also view the upstream documentation located [here](#).

Procedure

1. Open the Cloudera Manager Admin Console and go to the Solr service.
2. Click the Configuration tab.
3. Select Scope All .
4. In the Search field, type TLS/SSL to show the Solr TLS/SSL properties.

5. Edit the following properties according to your cluster configuration.



Note: These values must be the same for all hosts running the Solr role.

Table 1: Solr TLS/SSL Properties

Property	Description
Enable TLS/SSL for Solr	Check this field to enable TLS for Solr.
Solr 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 Solr is acting as a TLS/SSL server. The keystore must be in JKS format.
Solr TLS/SSL Server JKS Keystore File Password	Password for the Solr JKS keystore.
Solr TLS/SSL Client Trust Store File	Required in case of self-signed or internal CA signed certificates. The location on disk of the truststore, in .jks format, used to confirm the authenticity of TLS/SSL servers that Solr might connect to. This is used when Solr is the client in a TLS/SSL connection. This truststore must contain the certificate(s) used to sign the service(s) being connected to. If this parameter is not provided, the default list of well-known certificate authorities is used instead.
Solr TLS/SSL Client Trust Store Password	The password for the Solr TLS/SSL Certificate Trust Store File. This password is not required to access the truststore: this field can be left blank. This password provides optional integrity checking of the file. The contents of truststores are certificates, and certificates are public information.

6. Enter a Reason for Change, and then click Save Changes to commit your changes.
7. Launch the Stale Configuration wizard to restart the Solr service and any dependent services.

What to do next

If Ranger authorization has been enabled for the Solr service, you need to update the Solr Collection URL (for a resource-based policy) or Solr URL (for a resource-based service) from `http://host_ip:8983/solr` to `https://host_ip:8985/solr` on the Ranger Admin Web UI.

Related Information

[Configure a resource-based policy: Solr](#)

[Configure a resource-based service: Solr](#)

[Encrypting Data in Transit](#)

[Enabling SSL](#)

Cloudera Search Authentication

Cloudera Search continues to use simple authentication with the anonymous user as the default configuration, but Search also supports changing the authentication scheme to Kerberos. All required packages are installed during the installation or upgrade process. Additional configuration is required before Kerberos is available in your environment.

When authentication is enabled, only specified hosts and users can connect to Solr. Authentication also verifies that clients connect to legitimate servers. This feature prevents spoofing such as impersonation and man-in-the-middle attacks. Search supports Kerberos and LDAP authentication.

Cloudera Search supports a variety of combinations of authentication protocols:

Table 2: Authentication Protocol Combinations

Solr Authentication	Use Case
No authentication	Insecure cluster
Kerberos only	The Hadoop cluster has Kerberos turned on and every user (or client) connecting to Solr has a Kerberos principal.

Solr Authentication	Use Case
Kerberos and LDAP	The Hadoop cluster has Kerberos turned on. External Solr users (or clients) do not have Kerberos principals but do have identities in the LDAP server. Client authentication using LDAP requires that Kerberos is enabled for the cluster. Using LDAP alone is not supported.

Once you are finished setting up authentication, configure Ranger authorization. Authorization involves specifying which resources can be accessed by particular users when they connect through Search. For more information, see *Using Ranger to Provide Authorization in CDP*.

Related Information

[Using Ranger to Provide Authorization in CDP](#)

Configure Kerberos Authentication for Solr

Solr supports Kerberos authentication. All necessary packages are installed when you install Search. To enable Kerberos, see [Configuring Authentication in Cloudera Manager](#).

Enable Kerberos Authentication in Solr

Secure access to your Solr service by enabling Kerberos authentication.

About this task

Besides securing access to the Solr service, enabling Kerberos authentication is a prerequisite of both configuring LDAP authentication and Ranger authorization.

Before you begin

Solr supports Kerberos authentication. All necessary packages are installed when you install Search.

Kerberos authentication must be configured in Cloudera Manager for the cluster where Solr is deployed. For more information, see [Configuring Authentication in Cloudera Manager](#).

Procedure

1. In Cloudera Manager select the Solr service.
2. Select Configuration and find the Solr Secure Authentication property.
3. Select the Kerberos option.
4. Click Save Changes.
5. Restart the Solr service.

Results

Kerberos authentication for Solr is enabled.

Related Information

[Configuring Authentication in Cloudera Manager](#)

Set Proxy Server Authentication for Clusters Using Kerberos

In a cluster using Kerberos, applications check host credentials to verify that the host they are connecting to is the same one that is actually processing the request, to prevent man-in-the-middle attacks. To clarify that the load-balancing proxy server is legitimate, you need to perform these extra Kerberos setup steps.

About this task

This procedure assumes you are starting with a Kerberos-enabled cluster.

Procedure

1. Choose the host you will use for the proxy server. Based on the Kerberos setup procedure, it should already have an entry `solr/PROXY_HOST@REALM` in its keytab.
2. Navigate to Solr service Configuration Category Main .
3. Set the value of Solr Load Balancer to `<HOSTNAME>:<PORT>`, specifying the hostname and port of the proxy host.
4. Click Save Changes.
5. Launch the Stale Configuration wizard to restart the Solr service and any dependent services.

Cloudera Manager transparently handles the keytab and dependent service updates by setting `SOLR_AUTHENTICATION_KERBEROS_PRINCIPAL=*` under `/etc/default/solr` and by generating a merged keytab that includes the HTTP principal of the load balancer in addition to the own HTTP principal of the Solr server.

6. You can verify that the merged keytabs have been created and they contain the HTTP principal for both the load balancer and the particular Solr server by checking the process directory of Solr in `/var/run/cloudera-scm-agent/process`:
For example:

```
# klist -kte 291-solr-SOLR_SERVER/solr.keytab
Keytab name: FILE:291-solr-SOLR_SERVER/solr.keytab
KVNO Timestamp          Principal
-----
-----
    2 01/21/20 06:08:05 HTTP/loadbalancer.example.com@EXAMPLE.COM (des3-cbc-sha1)
    2 01/21/20 06:08:05 HTTP/loadbalancer.example.com@EXAMPLE.COM (arcfour-hmac)
    2 01/21/20 06:08:05 HTTP/loadbalancer.example.com@EXAMPLE.COM (des-hmac-sha1)
    2 01/21/20 06:08:05 HTTP/loadbalancer.example.com@EXAMPLE.COM (des-cbc-md5)
    2 01/21/20 06:08:05 HTTP/solrserver1.example.com@EXAMPLE.COM (des3-cbc-sha1)
    2 01/21/20 06:08:05 HTTP/solrserver1.example.com@EXAMPLE.COM (arcfour-hmac)
    2 01/21/20 06:08:05 HTTP/solrserver1.example.com@EXAMPLE.COM (des-hmac-sha1)
    2 01/21/20 06:08:05 HTTP/solrserver1.example.com@EXAMPLE.COM (des-cbc-md5)
    2 01/21/20 06:08:05 solr/solrserver1.example.com@EXAMPLE.COM (des3-cbc-sha1)
    2 01/21/20 06:08:05 solr/solrserver1.example.com@EXAMPLE.COM (arcfour-hmac)
    2 01/21/20 06:08:05 solr/solrserver1.example.com@EXAMPLE.COM (des-hmac-sha1)
    2 01/21/20 06:08:05 solr/solrserver1.example.com@EXAMPLE.COM (des-cbc-md5)
```

Related Information

[Enable Kerberos Authentication in Solr](#)
[Stale Configurations](#)

Overview of Proxy Usage and Load Balancing for Search

See the advantages of configuring a proxy server for the Solr service.

- Applications connect to a single well-known host and port, rather than keeping track of the hosts where the Solr service is running. This is especially useful for non-Java Solr clients such as web browsers or command-line tools such as curl.



Note: The Solr Java client (solrj) can inspect Zookeeper metadata to automatically locate the individual Solr servers, so load-balancing proxy support is not necessary.

- If any host running the Solr service becomes unavailable, application connection requests still succeed because you always connect to the proxy server rather than a specific host running the Solr server.
- Users can configure an SSL terminating proxy for Solr to secure the data exchanged with the external clients without requiring SSL configuration for the Solr cluster itself. This is relevant only if the Solr cluster is deployed on a trusted network and needs to communicate with clients that may not be on the same network. Many of the advantages of SSL offloading are described in [SSL Offloading, Encryption, and Certificates with NGINX](#).
- The "coordinator host" for each Search query potentially requires more memory and CPU cycles than the other hosts that process the query. The proxy server can issue queries using round-robin scheduling, so that each connection uses a different coordinator host. This load-balancing technique lets the hosts running the Solr service share this additional work, rather than concentrating it on a single machine.

Related Information

[Set Proxy Server Authentication for Clusters Using Kerberos](#)

Enable LDAP Authentication in Solr

You can configure LDAP-based authentication using Cloudera Manager at the Solr service level.

About this task

Solr supports LDAP authentication for external Solr clients including:

- Command-line tools
- curl
- Web browsers
- Solr Java clients

In some cases, Solr does not support LDAP authentication. Use Kerberos authentication instead in these cases. Solr does not support LDAP authentication with:

- Search indexing components including the MapReduce indexer and Lily HBase indexer.
- Solr internal requests such as those for replication or querying.
- Hadoop delegation token management requests such as GETDELEGATIONTOKEN or RENEWDELEGATIONTOKEN.

Before you begin

- Configuring LDAP authentication requires that Kerberos authentication is already configured and enabled in Solr.
- For secure LDAP connections, it is a prerequisite that TLS/SSL has been configured and enabled in Solr.

Procedure

1. In Cloudera Manager select the Solr service.
2. Click the Configuration tab.
3. Select Scope Solr .
4. Select Category Security .

5. Select Enable LDAP Authentication.
6. Enter the LDAP URL in the LDAP URL property.

To configure a TLS encrypted LDAP connection, select one of the following options:

- `ldaps://<LDAP_SERVER>:<PORT>`

The default port is 636.

OR

- `ldap://<LDAP_SERVER>:<PORT>`

The default port is 389.

Select Enable LDAP TLS. This is not required when using an LDAP URL with prefix `ldaps://`, because that already specifies TLS.

To configure LDAP with unencrypted transmission of usernames and passwords, set `ldap://<ldap_server>:<port>`, without setting Enable LDAP TLS.

7. Configure only one of following mutually exclusive parameters:
 - LDAP BaseDN: Replaces the username with a "distinguished name" (DN) of the form: `uid=userid,ldap_base DN`. Typically used for OpenLDAP server installation.
 - Active Directory Domain: Replaces the username with a string `username@ldap_domain`. Typically used for Active Directory server installation.
8. Launch the Stale Configuration wizard to restart the Solr service and any dependent services.

Related Information

[Stale Configurations Wizard](#)

Enabling Solr Clients to Authenticate with a Secure Solr

The process of enabling Solr clients to authenticate with a secure Solr is specific to the client.

Cloudera Search supports the following options:

- Using Kerberos and curl
- Using `solrctl`
- Using a `jaas.conf` File
- This enables technologies including:
 - Command line solutions
 - Java applications
 - The `MapReduceIndexerTool`

Secure Solr requires that the CDP components it interacts with are also secure. Secure Solr interacts with HDFS, ZooKeeper and optionally HBase, MapReduce, and NiFi.

Using Kerberos and curl

You can use Kerberos authentication with clients such as curl. To use curl, begin by acquiring valid Kerberos credentials and then run the desired command. For example, you might use commands similar to the following:

```
$ kinit -kt username.keytab username
$ curl --negotiate -u foo:bar http://solrserver:8983/solr/
```



Note: Depending on the tool used to connect, additional arguments may be required. For example, with curl, --negotiate and -u are required. The username and password specified with -u is not actually checked because Kerberos is used. As a result, any value such as foo:bar or even just : is acceptable. While any value can be provided for -u, note that the option is required. Omitting -u results in a 401 Unauthorized error, even though the -u value is not actually used.

Using solrctl

If you are using solrctl to manage your deployment in an environment that requires Kerberos authentication, you must have valid Kerberos credentials, which you can get using kinit. For more information on solrctl, see [solrctl Reference](#).

Using a jaas.conf File

Some applications, such as those using the SolrJ library, require a Java Authentication and Authorization Service (JAAS) configuration file. You can use a file name other than jaas.conf, in the following examples jaas-client.conf is used.

Creating a JAAS configuration file:

- If you are authenticating using kinit to obtain credentials, you can configure the client to use your credentials cache by creating a jaas-client.conf file with the following contents:

```
Client {
  com.sun.security.auth.module.Krb5LoginModule required
  useKeyTab=false
  useTicketCache=true
  principal="<USER>@EXAMPLE.COM";
};
```

Replace <USER> with your username, and EXAMPLE.COM with your Kerberos realm.

- If you want the client application to authenticate using a keytab, modify jaas-client.conf as follows:

```
Client {
  com.sun.security.auth.module.Krb5LoginModule required
  useKeyTab=true
  keyTab="/path/to/user.keytab"
  storeKey=true
  useTicketCache=false
  principal="<user>@EXAMPLE.COM";
};
```

Replace /PATH/TO/USER.KEYTAB with the keytab file you want to use and <USER>@EXAMPLE.COM with the principal in the keytab. If you are using a service principal that includes the hostname, make sure that it is included in the jaas.conf file (for example, solr/solr01.example.com@EXAMPLE.COM).

Example usage of a JAAS configuration file:

- Command line

Set the property when invoking the program. For example, if you were using a jar, you might use:

```
java -Djava.security.auth.login.config=/home/user/jaas-client.conf -jar
app.jar
```

- Java applications

Set the Java system property java.security.auth.login.config. For example, if the JAAS configuration file is located on the filesystem as /home/user/jaas-client.conf, the Java system property java.security.auth.login.config must

be set to point to this file. Setting a Java system property can be done programmatically, for example using a call such as:

```
System.setProperty("java.security.auth.login.config", "/home/user/jaas-client.conf");
```

- The MapReduceIndexerTool

The MapReduceIndexerTool uses SolrJ to pass the JAAS configuration file. Using the MapReduceIndexerTool in a secure environment requires the use of the HADOOP_OPTS variable to specify the JAAS configuration file. For example, you might issue a command such as the following:

```
HADOOP_OPTS="-Djava.security.auth.login.config=/home/user/jaas-client.conf" \  
hadoop jar MapReduceIndexerTool
```

- Configuring the hbase-indexer CLI

Certain hbase-indexer CLI commands such as replication-status attempt to read ZooKeeper hosts owned by HBase. To successfully use these commands in Solr in a secure environment, specify a JAAS configuration file with the HBase principal in the HBASE_INDEXER_OPTS environment variable. For example, you might issue a command such as the following:

```
HBASE_INDEXER_OPTS="-Djava.security.auth.login.config=/home/user/hbase-jaas.conf" \  
hbase-indexer replication-status
```

Related Information

[solrctl Reference](#)

Enable Ranger Authorization in Solr

Add a Ranger service to enable access control in Solr.

Before you begin

- Ranger authorization requires that Kerberos authentication is enabled in Solr.

About this task

Ranger restrictions are consistently applied regardless of the way users attempt to complete actions. For example, restricting access to data in a collection consistently restricts that access, whether queries come from the command line, from a browser, or through the admin console.

Procedure

1. In Cloudera Manager select the Solr service.
2. Select Configuration and find the RANGER Service property.
3. Check the checkbox next to the name of the Ranger service that you want this Solr service to depend on.
4. Click Save Changes.
5. Restart the Solr service.

Results

Ranger authorization for Solr is enabled. The Solr service depends on the selected Ranger service for authorization.

Related Information

[Configure a resource-based service: Solr](#)

[Configure a resource-based policy: Solr](#)