

Cloudera Runtime 7.3.1

Apache Kafka KRaft

Date published: 2023-06-27

Date modified: 2024-12-10

CLOUDERA

<https://docs.cloudera.com/>

Legal Notice

© Cloudera Inc. 2024. 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

KRaft setup.....	4
Extracting KRaft metadata.....	4
Securing KRaft.....	5
Configuring TLS/SSL for KRaft Controllers.....	5
KRaft Ranger authorization.....	6

KRaft setup

Learn how you can set up Kafka KRaft in CDP Public Cloud



Note: Kafka KRaft is available in this version of CDP but is not ready for production deployment. Cloudera encourages you to explore this technical preview feature in non-production environments and provide feedback on your experiences through the [Cloudera Community Forums](#). For more information regarding KRaft limitations and unsupported features, see [Known Issues in Apache Kafka](#).

Kafka KRaft in CDP is implemented in the form of a Kafka service role. The role is called KRaft Controller. In CDP Public Cloud, KRaft Controller roles can be deployed with the Streams Messaging Light Duty, Heavy Duty, and High Availability cluster definitions available in the Data Hub service.

Each of the definitions include an optional KRaft Nodes host group. The KRaft Controller roles are deployed on the nodes of this host group. If you want to deploy a Streams Messaging cluster that uses KRaft for metadata management, you must provision your cluster with at least a single KRaft node (three is recommended). KRaft nodes are also scalable after the cluster is provisioned.

For more information regarding the Streams Messaging cluster definitions, scaling, cluster deployment with Data Hub, as well as KRaft, see the *Related Information*.

Related Information

[Kafka KRaft Overview](#)

[Setting up your Streams Messaging cluster](#)

[Scaling KRaft](#)

[Streams Messaging cluster layout](#)

Extracting KRaft metadata

Learn how to extract Kafka metadata from the `__cluster_metadata` topic. Metadata extracted from this topic can be used for debugging and troubleshooting issues with a Kafka deployment running in KRaft mode.

About this task



Note: Kafka KRaft is available in this version of CDP but is not ready for production deployment. Cloudera encourages you to explore this technical preview feature in non-production environments and provide feedback on your experiences through the [Cloudera Community Forums](#). For more information regarding KRaft limitations and unsupported features, see [Known Issues in Apache Kafka](#).

When Kafka is running in KRaft mode, metadata describing the state of the Kafka cluster is stored in the `__cluster_metadata` topic. This topic can be found in the `/var/local/kraft/data` directory on each KRaft Controller service role host.

In case you encounter any issues when running your deployment in KRaft mode, generally the first step is to print the contents of the `__cluster_metadata` topic. Reviewing the contents of the topic can help in identifying the issues with the cluster.

The contents of the `__cluster_metadata` topic can be printed using the `kafka-dump-log` command with the `--cluster-metadata-decoder` option.

Procedure

1. Log in to one of your cluster hosts that has a KRaft service role (KRaft Controller) deployed on it.

- Find and configure the following properties based on your cluster and requirements.

Table 1: KRaft TLS/SSL configuration properties

Cloudera Manager Property	Description
Enable TLS/SSL for Kraft Controller ssl_enabled	Encrypt communication between clients and KRaft Controller using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)).
KRaft Controller TLS/SSL Server JKS Keystore File Location ssl_server_keystore_location	The path to the TLS/SSL keystore file containing the server certificate and private key used for TLS/SSL. Used when KRaft Controller is acting as a TLS/SSL server. The keystore must be in the format specified in Administration Settings Java Keystore Type .
KRaft Controller TLS/SSL Server JKS Keystore File Password ssl_server_keystore_password	The password for the KRaft Controller keystore file.
KRaft Controller TLS/SSL Server JKS Keystore Key Password ssl_server_keystore_keypassword	The password that protects the private key contained in the keystore used when KRaft Controller is acting as a TLS/SSL server.
KRaft Controller TLS/SSL Trust Store File ssl_client_truststore_location	The location on disk of the trust store, in .jks format, used to confirm the authenticity of TLS/SSL servers that KRaft Controller might connect to. This trust store must contain the certificate(s) used to sign the service(s) connected to. If this parameter is not provided, the default list of well-known certificate authorities is used instead.
KRaft Controller TLS/SSL Trust Store Password ssl_client_truststore_password	The password for the KRaft Controller TLS/SSL Trust Store File. This password is not required to access the trust store; this field can be left blank. This password provides optional integrity checking of the file. The contents of trust stores are certificates, and certificates are public information.
SSL Client Authentication ssl.client.auth	Client authentication mode for SSL connections. This configuration has three valid values, required, requested, and none. If set to required, client authentication is required. If set to requested, client authentication is requested and clients without certificates can still connect. If set to none, which is the default value, no client authentication is required.

- Click Save Changes.
- Restart the Kafka service.

What to do next

TLS/SSL encryption is configured for the KRaft Controller role.

KRaft Ranger authorization

Learn how KRaft integrates with Ranger as well as the default policies and permissions set up for KRaft.



Note: If Ranger authorization is enabled, Kafka still connects to ZooKeeper for auditing. As a result, Kafka's JAAS configuration includes a client entry for ZooKeeper. Additionally, the `-Dzookeeper.sasl.client.username=[**ZOOKEEPER PRINCIPAL SHORTNAME**]` system property is set for the process. This is the result of Ranger's dependency on ZooKeeper. Even though Ranger makes this connection, Kafka does not require or use ZooKeeper for metadata management if it is running in KRaft mode.

KRaft in CDP uses the `KafkaRangerAuthorizer` to authorize requests coming from other entities. In KRaft mode, Kafka brokers forward requests to the controllers and the controllers authorize these requests.

Kraft Controllers run as the `kraft` user. By default, the Kafka resource-based service in Ranger includes a `kraft` internal - topic policy. This policy grants all permission on the `__cluster_metadata` topic for the `kraft` user as well as Describe, Describe Configs, and Consume permissions for the `kafka` user (default user for brokers). By default, other users do not have access to the `__cluster_metadata` topic.

Service Manager

cm_kafka Policies

Policy Name

Policy La

topic



Descript

In addition, the kraft user is added to all default Kafka policies that grant all permissions on Kafka resources.



Ranger

Access Management

Service Manager

cm_kafka Policies

List of Policies : cm_kafka

Search for your policy...

Policy ID ▲	Policy Name
26	all - consumerg
27	all - topic