Apache Kafka KRaft

Date published: 2023-06-27
Date modified: 2024-04-03

CLOUDERA

Legal Notice

Contents

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.

2. Run the kafka-dump-log command with the --cluster-metadata-decoder option. For example:

```
kafka-dump-log --cluster-metadata-decoder --files /var/local/kraft/data/
__cluster_metadata-0/00000000000000000000.log
```

Securing KRaft

Learn about KRaft security and security configuration in CDP.

> **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 you deploy Kafka in KRaft mode a set of specialized broker roles, KRaft Controller roles, are deployed on your cluster. KRaft Controllers communicate with brokers to serve their requests and to manage Kafka's metadata. The connection between controllers and brokers can be secured using TLS/SSL encryption, TLS/SSL authentication, and/ or Kerberos authentication.

By default KRaft Controllers inherit the security configuration of the parent Kafka service. For example, if TLS/SSL is enabled for Kafka, then Cloudera Manager automatically enables TLS/SSL for the KRaft Controllers in the cluster. As a result, if you configure security for the Kafka service, no additional configuration is required to secure KRaft Controllers.

However, if required, some security properties related to encryption and authentication can be configured separately for KRaft Controllers.

• TLS/SSL encryption and authentication

TLS/SSL configuration can be configured separately as the KRaft Controller role has its own set of TLS/ SSL properties. You can enable or disable TLS/SSL as well as configure the key and truststore that the KRaft Controller roles use. For more information see, Configuring TLS/SSL for KRaft Controllers on page 5.

• Kerberos authentication

Kerberos cannot be enabled or disabled separately for KRaft Controllers. The default Kerberos principal for KRaft controllers, the kraft user, can be changed using the Role-Specific Kerberos Principal Kafka service property.

> **Important:** Cloudera Manager configures CDP services to use the default Kerberos principal names. Cloudera recommends that you do not change the default Kerberos principal names. If it is unavoidable to do so, contact Cloudera Professional Services because it requires extensive additional custom configuration.

Ranger authorization

In addition to encryption and authentication, the default principal that KRaft Controllers run as is integrated with Ranger. For more information on the default policies set up for the user, see KRaft Ranger authorization on page 6.

Configuring TLS/SSL for KRaft Controllers

Learn how to configure TLS/SSL for KRaft Controllers.

Procedure

1. In Cloudera Manager, select the Kafka service.
2. Go to Configuration.

3. 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.

4. Click Save Changes.

5. Restart the Kafka service.

What to do next
TLS/SSL encryption is configured for the KRaft Controller role.
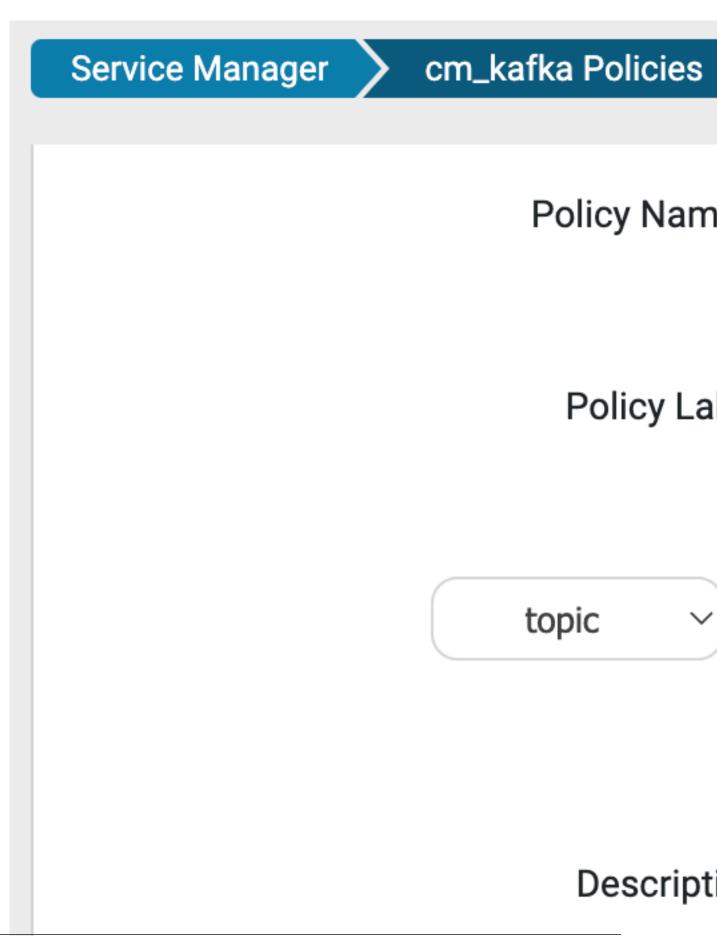
KRaft Ranger authorization

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.user name=*[**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 in ternal - 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 Nam

Policy Lal

topic ⌄

Descripti

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

List of Policies : cm_kafka

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