Cloudera Runtime 7.3.2

Using Streams Messaging Manager

Date published: 2023-10-25
Date modified: 2026-03-31



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

Monitoring Kafka 5
Monitoring Kafka clusters 5
Monitoring Kafka producers 6
Monitoring Kafka topics 9
Monitoring Kafka brokers 14
Monitoring Kafka consumers 16
Monitoring log size information 19
Monitoring lineage information 23

Managing Kafka topics 25
Creating a Kafka topic 25
Modifying a Kafka topic 26
Deleting a Kafka topic 26

Managing Alert Policies and Notifiers 26
Creating a notifier 28
Updating a notifier 29
Deleting a notifier 29
Creating an alert policy 29
Updating an alert policy 31
Enabling an alert policy 31
Disabling an alert policy 31
Deleting an alert policy 32
Component types and metrics for alert policies 32

Monitoring end-to-end latency 36
Enabling interceptors 37
Monitoring end to end latency for Kafka topic 39
End to end latency use case 43

Monitoring Kafka cluster replications (Streams Replication Manager) 48
Viewing Kafka cluster replication details 49
Searching Kafka cluster replications by source 50
Monitoring Kafka cluster replications by quick ranges 50
Monitoring status of the clusters to be replicated 50
Monitoring topics to be replicated 51
Searching by topic name 52
Monitoring throughput for cluster replication 52
Monitoring replication latency for cluster replication 53
Monitoring checkpoint latency for cluster replication 54
Monitoring replication throughput and latency by values 55

Managing and monitoring Kafka Connect using Streams Messaging Manager 56
The Kafka Connect UI 56
Deploying and managing Kafka Connect connectors in Streams Messaging Manager 61
Deploying a Kafka Connect connector in Streams Messaging Manager 61
Pausing, resuming, restarting, and deleting a Kafka Connect connector in Streams Messaging Manager 65
Reconfiguring Kafka Connect connectors in Streams Messaging Manager 66
Connector configuration features in Streams Messaging Manager 67

Managing and monitoring Cruise Control rebalance 72
Cruise Control dashboard in Streams Messaging Manager UI 72
Using the Rebalance Wizard in Cruise Control 78

Monitoring Kafka

Learn how to monitor Kafka clusters, producers, consumers topics brokers, log size and lineage using Streams Messaging Manager.

Monitoring Kafka clusters

The overview page provides you with tools to see a snapshot of the Kafka cluster you are monitoring. After you select the Kafka cluster to monitor, you can see the total number of producers, brokers, topics, and consumer groups in that cluster. You can also monitor producer and consumer metrics.

Configure Apache Kafka for Streams Messaging Manager

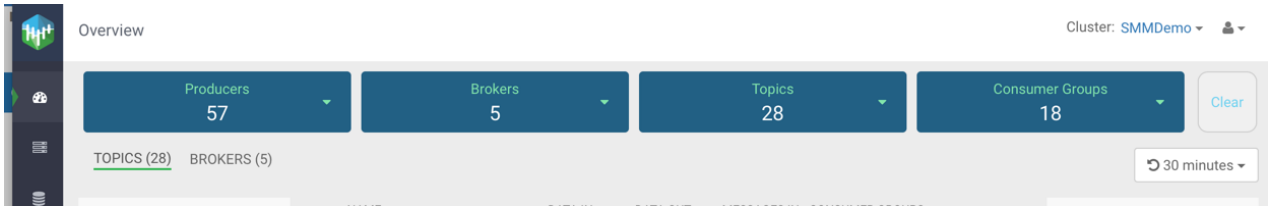
After you have installed and configured Apache Kafka, you must set one configuration parameter to enable Kafka and Streams Messaging Manager to communicate.

1. Select Kafka from your cluster drop-down, and then select the Configuration tab.
2. Ensure that the Enable Producer Metrics check box is selected.

Viewing cluster overview information

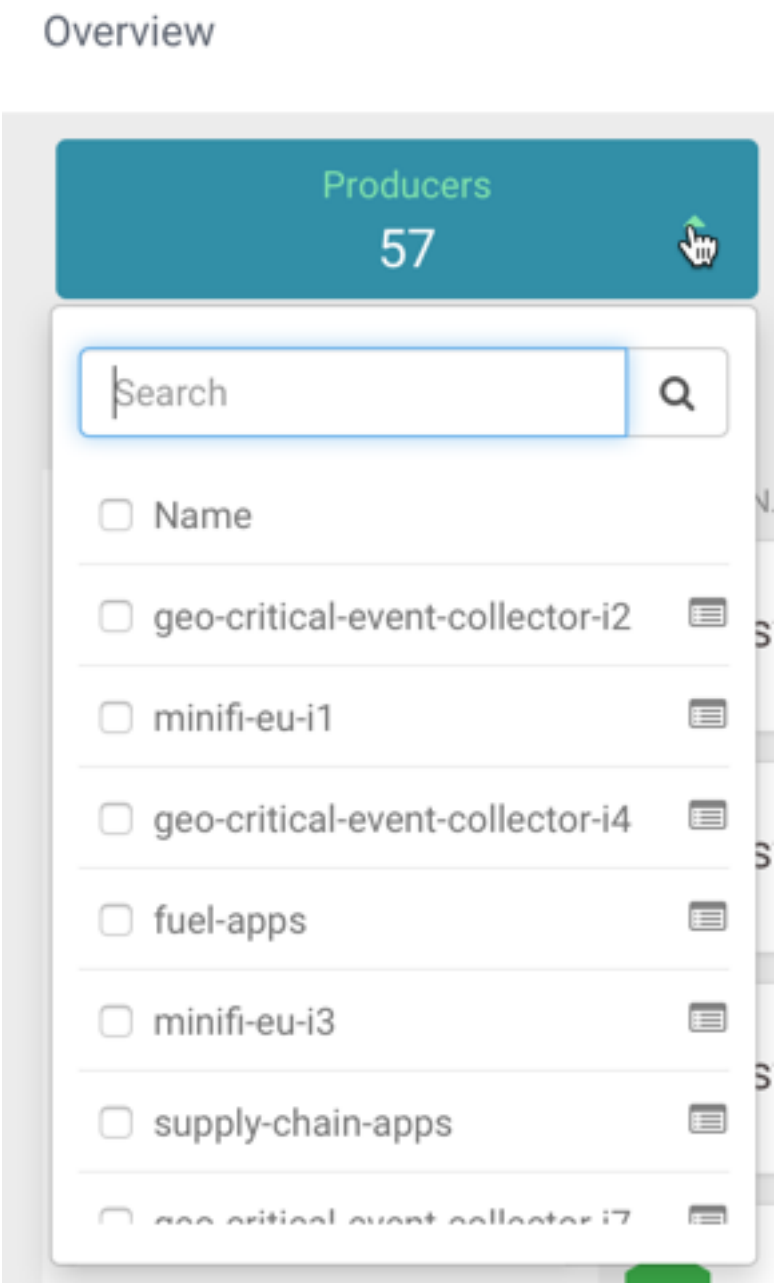
You can use the Overview tab to review information about your Kafka cluster. This page gives you information about total number of producers, brokers, topics, and consumer groups. It also provides more detailed metrics about producers and consumers.

Review the Producers, Brokers, Topics, and Consumer Groups information at the top of your page to understand how many of each are contained in your Kafka cluster.

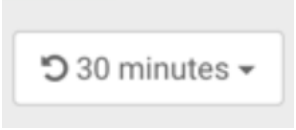


You can click the drop-down arrow in any of the boxes to view a list of Kafka resource. Select one or more Kafka resource to filter your view to just those resource. You can also search for a specific resource. You can click clear at any time to return to the full overview.

Note: Searching supports regular expressions. Invalid regular expressions are interpreted as exact match searches.



You can select the time period you want to view the metrics for, on the top-right corner of the page. If Cloudera Manager is configured as a metrics backend, the metrics (for example, topic > partition > producermetrics) which are used for time periods larger than 6 hours are calculated asynchronously, and take time to refresh.



Monitoring Kafka producers

By monitoring Kafka producers, you can track the active and inactive producers in your cluster. You can also change the period of time after which a producer is considered inactive.

Understanding producer naming conventions

The producers you interact with in Streams Messaging Manager are named based on the client.id property you added when creating Kafka producers.

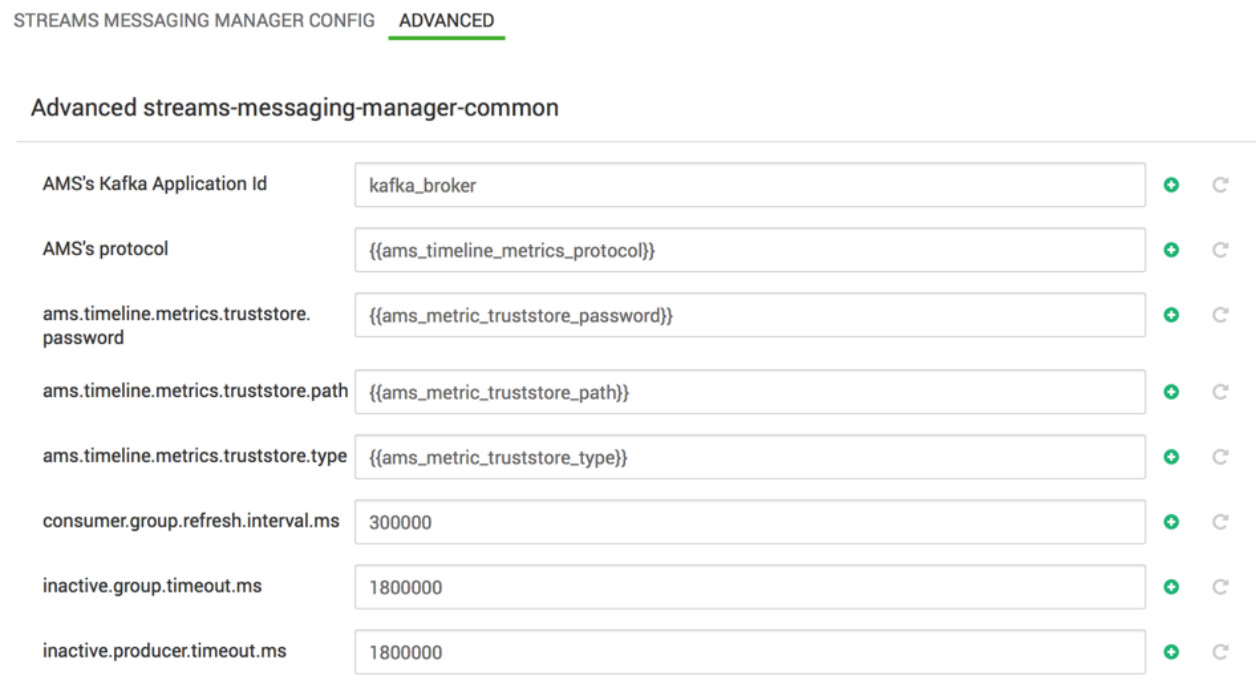
Active vs. passive producers

On the Overview page, producers are referred to as active or passive. Producers are active when they are producing messages over a designated time period.

On the Producers page, passive producers are referred to as inactive.

You can set the period of time after which a producer is considered inactive in the Streams Messaging Manager Configs screen.

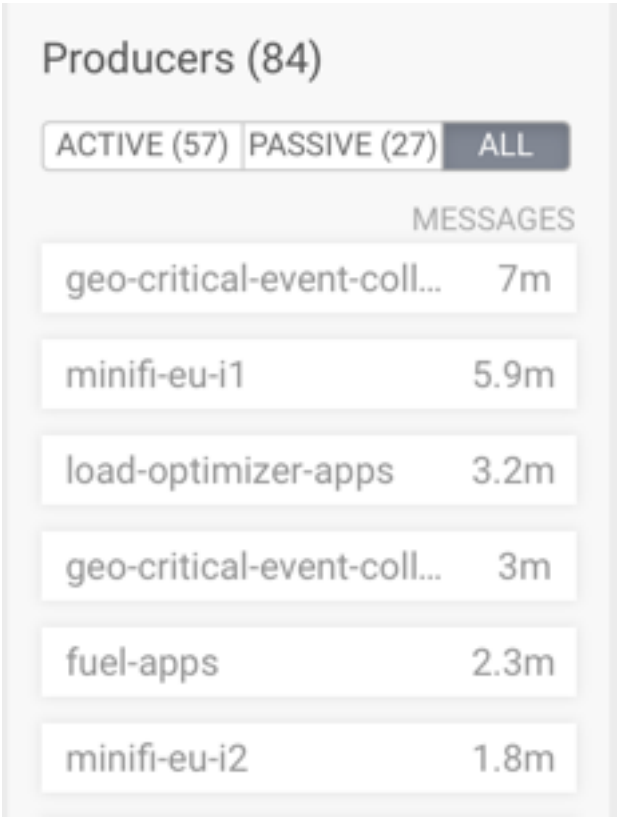
1. Select Streams Messaging Manager from the services pane.
2. Click Configs and select Advanced streams-messaging-manager-common from the Advanced tab.
3. Update inactive.producer.timeout.ms to change the period of time after which a producer is considered inactive. This value is specified in milliseconds.



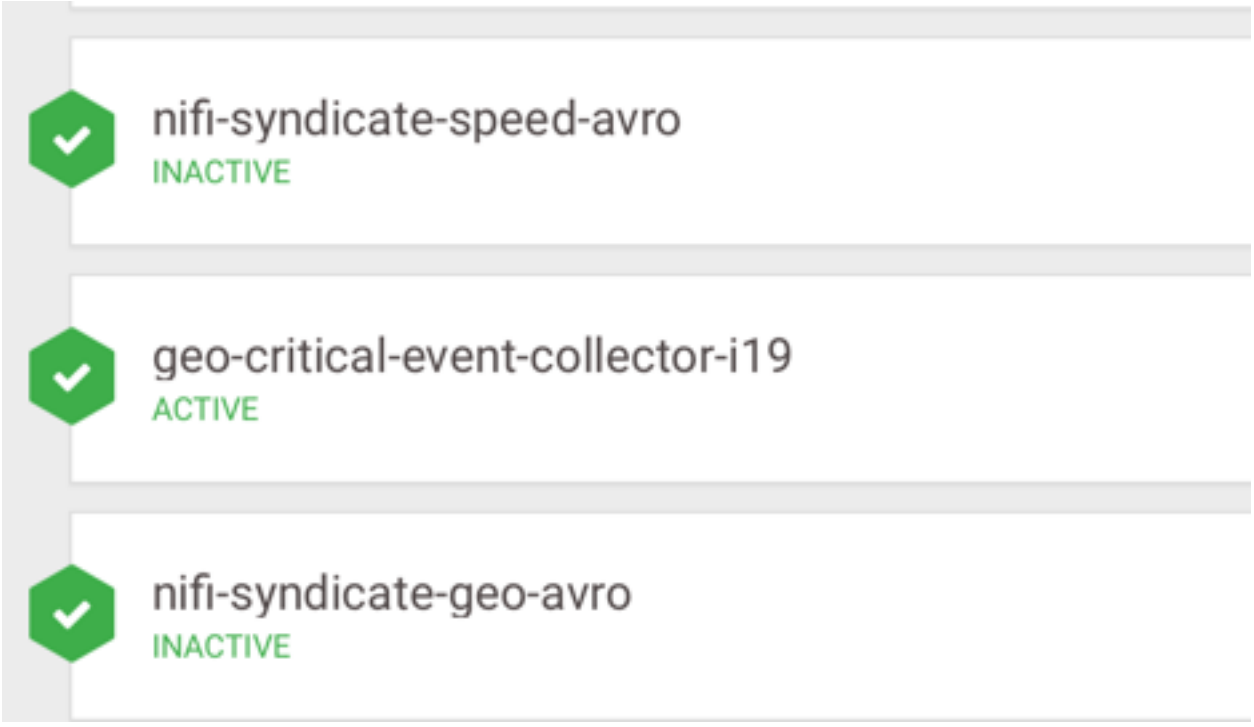
Identifying a producer state

There are two ways to identify whether a producer is active or passive.

From the Producer pane in the Overview page, use the Active, Passive, and All tabs to view only active producers, only passive producers, or all of them. This allows you to see the total number of active and passive producers.



From the Producers page, each producer is listed with the status visible.

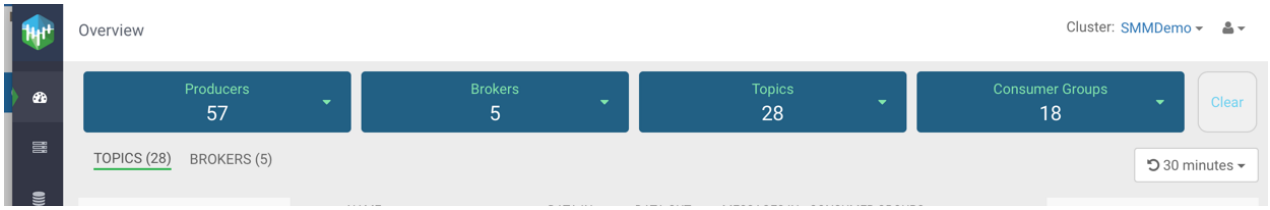


Monitoring Kafka topics

By monitoring Kafka topics, you can track the total number of topics in your cluster and details about the topics. You can also monitor Grafana metrics for the topics in your cluster.

Viewing the total number of topics in your cluster

You can see the total number of topics in your Kafka cluster on the Overview page.



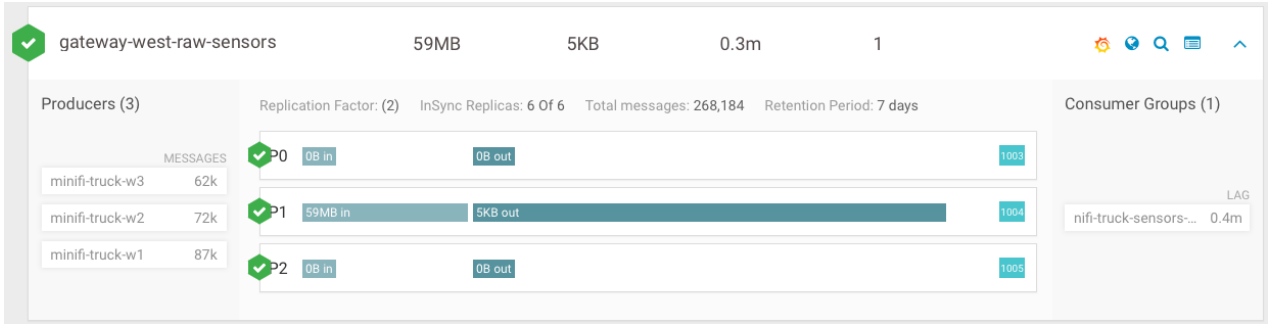
Detailed information about topics

The Topics page contains a number of useful details about your Kafka topics. This page helps you answer the following questions:

- How can I see if the replicas in this topic are in sync?
- How do I see this topic's retention rate?
- How can I see the replication factor for this topic?
- How do I see the producers and consumers that are connected to this topic?
- How do I find the total number of messages going into this topic, over a specified time range?

To access this detailed topic information:

1. From the left navigation pane, click Topics.
2. Identify the topic about which you want information. You can either scroll through the list of topics, or use the Search bar at the top left of the page.
3. Click the green hexagon at the left of the topic to view details.



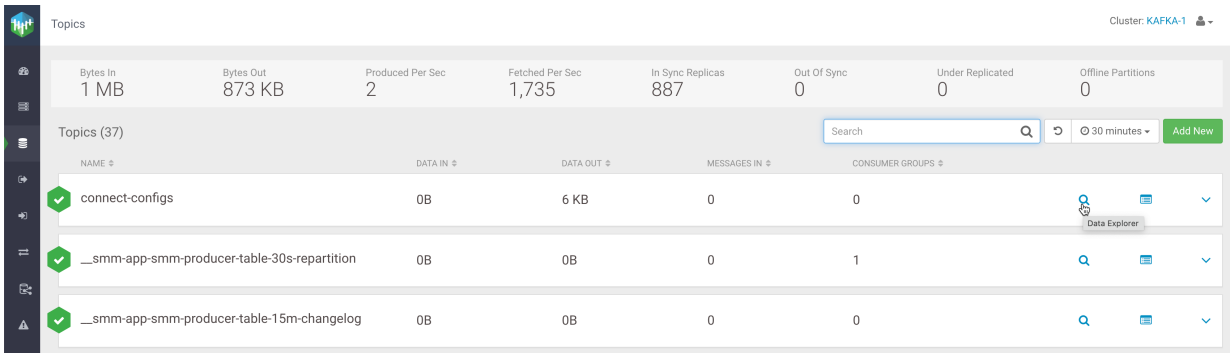
Viewing topic messages using Data Explorer

Data Explorer is a simple Kafka consumer within Streams Messaging Manager. It enables you to view the content of a Kafka topic. You can select any Kafka topic and any partition within that topic, and view messages from the selected partition.

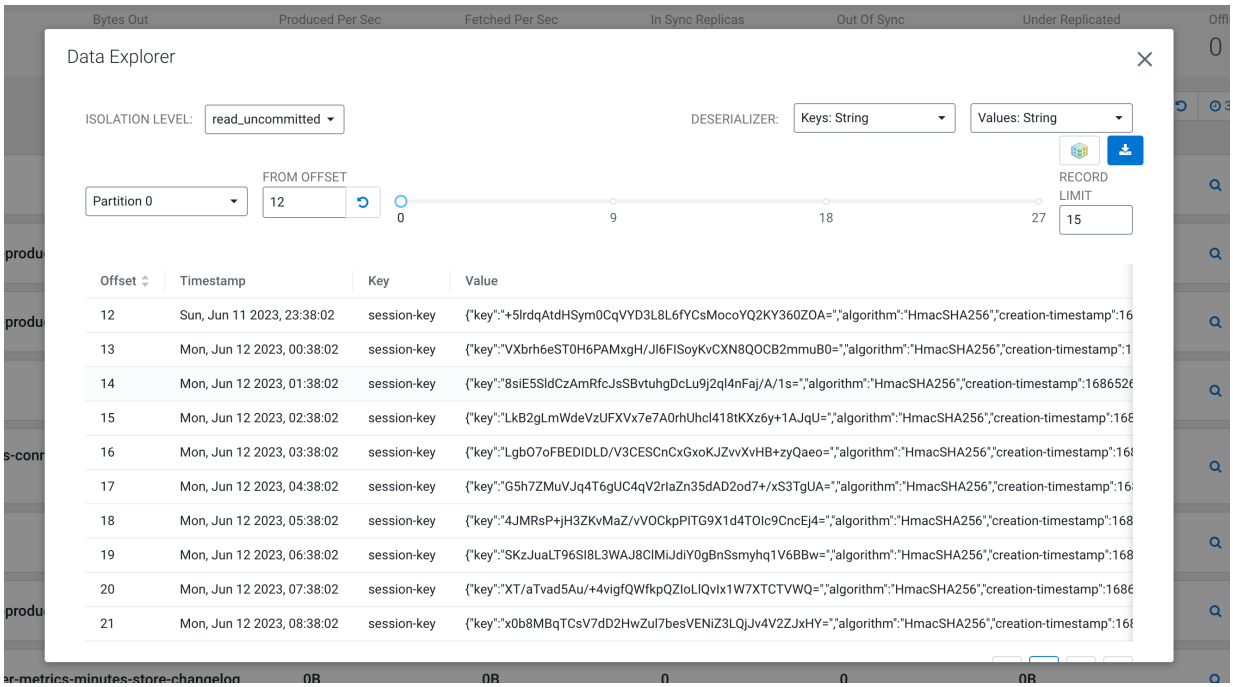
You can reach Data Explorer in two ways. One way is from the **Topics** page, and the other is from the **Overview** page. In both pages, you need to either click the magnifier icon, or navigate to the **Topic Details** page and then select the Data Explorer tab. The following steps describe the process:

1. Log in to the Streams Messaging Manager UI.
2. From the left navigation pane, click Topics.

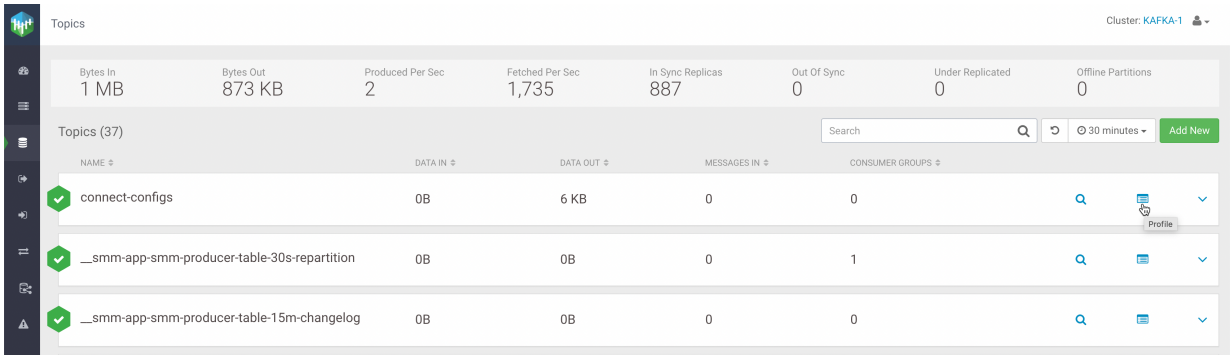
3. Identify the topic for which you want the message information. You can either scroll through the list of topics, or use the Search bar to find a topic.
4. Click the Data Explorer icon for that topic.



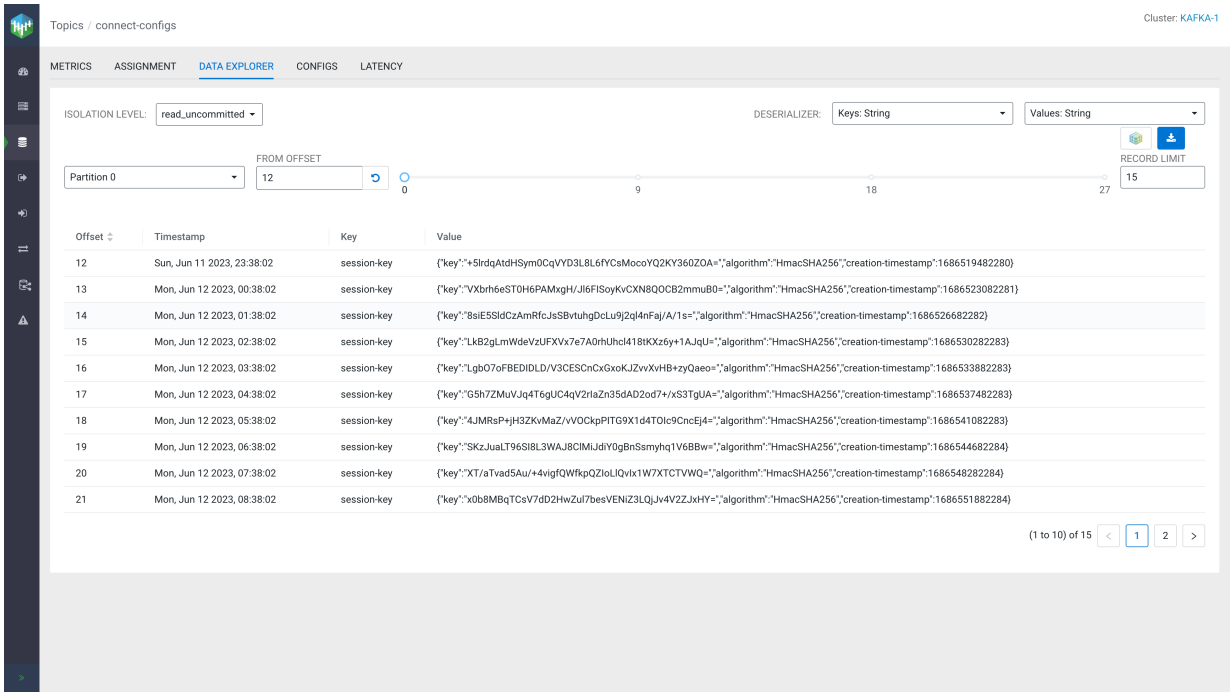
The Data Explorer dialog appears.



Alternatively, you can click the Profile icon for that topic.



Then go to the Data Explorer tab.



5. Select any of the following modes in the Isolation Level option:
   - read_committed
   - read_uncommitted

   The isolation level specifies whether uncommitted transactional messages should be read. By default, it is set to read_uncommitted.
6. Select the deserializer types for the Keys and Values options.

   For example, if you select Avro, Streams Messaging Manager uses the schema that can be found in Schema Registry to deserialize the messages.
7. Select a Partition.

   The Kafka topic must have partitions to select from.
8. Select a value for the From Offset field.

   You can also use the selection bar to select an offset value. The maximum value is the offset of the last message.

   Click ↻ to refresh the partition offset range and to fetch the latest messages.
9. Select a Record Limit.

   The record limit value is the number of messages that are fetched starting from the message offset number selected in the From Offset field.
10. To see long messages, click show more beside a message.

    The message opens in a dialog or a new tab based on the size of the message.
11. Click the Schema Registry icon to go to the related page in the Schema Registry UI.



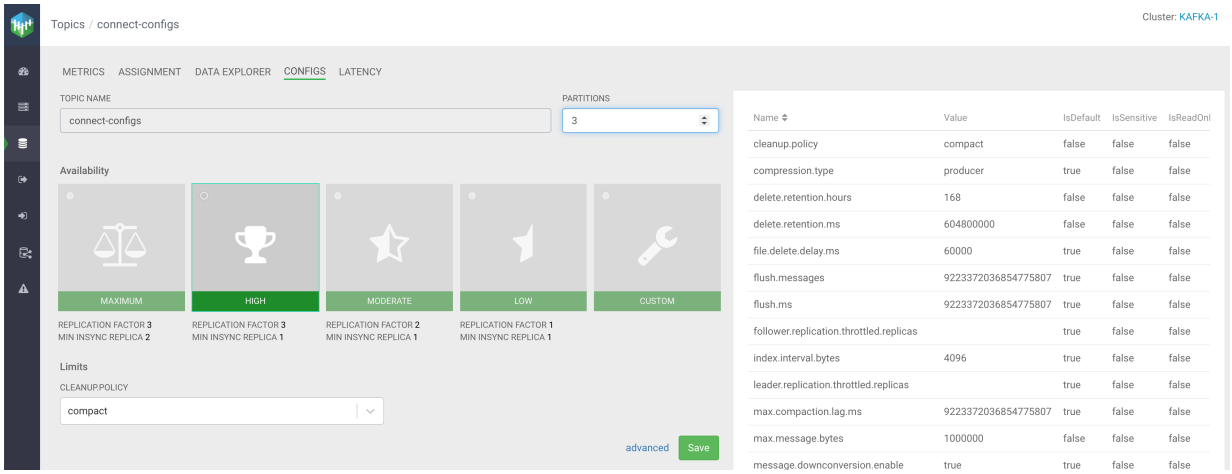
Increasing topic partition

You can increase the number of partitions of a topic

Warning: Increasing the partition numbers can impact the use of keys on messages. The data that is already written is not redistributed using the new partition, but remains on the partition on which it was located.

The following steps describe the process:

1. Log in to the Streams Messaging Manager UI.
2. From the left navigation pane, click Topics.
3. Identify the topic for which you want to increase the topic partition number. You can either scroll through the list of topics, or use the Search bar to find a topic.
4. Click the Profile icon for that topic.
5. Go to the Configs tab.



6. Enter the partition number in Partitions or increase it using the arrows.
7. Click Save.

   A dialog appears to ensure the increment.
8. Click Yes.
9. Refresh the page to see the change.

Getting current state of the topic partitions (Tech Preview)

The experimental Assignment tab, on the topic details page, shows the current state of the topic. It shows some topic-level statistics and the replica assignment of all partitions. If rack awareness is used in the Kafka cluster, the replica assignment is shown in a rack-based view. If the rack IDs follow the format of multi-level rack IDs, the rack IDs are rendered as a hierarchy. For more information on rack awareness, see *Kafka rack awareness*.

To go to the Assignment tab, click the Profile icon for a topic from the Overview or Topics page, and then click the Assignment tab. You can view the following statistics there:

- Number of offline partitions

  Shows the number of offline partitions in the topic. A partition is offline if it does not have a leader. Partitions can become offline if all their in-sync replicas are offline.
- Number of under-min-ISR partitions

  Shows the number of under-min-ISR partitions in the topic. A partition is in an under-min-ISR state if the number of in-sync replicas is lower than the value set in the min.insync.replicas property of the topic. The minimum in-sync replicas configuration defines how many replicas must acknowledge a produced message before the produce request is considered successful.
- Number of under-replicated partitions

Shows the number of under-replicated partitions in the topic. A partition is under-replicated if it has at least one out-of-sync replica.

- Number of unevenly distributed partitions

  This appears if rack awareness is being used.

  Important: This statistics is based on standard rack awareness, and does not respect the multi-level rack IDs. This means that a partition might be shown as evenly distributed while in terms of multi-level rack awareness it should be considered unevenly distributed.

  Shows the number of unevenly distributed partitions in the topic. A partition is unevenly distributed if the difference between the maximum and minimum number of replicas in a rack is greater than one. This typically suggests that the partition does not meet expected durability guarantees or that it causes uneven load on the cluster. If a partition is unevenly distributed, try reassigning them. In most cases, unevenly distributed partitions become evenly distributed across the racks after reassignment.
- Number of unused racks

  This appears if rack awareness is being used.

  Shows the number of racks which are currently not used by this topic. A rack is unused if the topic has no replicas residing in that rack. This typically suggests that the partition does not meet expected durability guarantees because it is not using all available racks (physical locations) to store replicas of the data. This is expected and does not cause issues for non-critical topics that have an intentionally low replication factor.

In the replica assignment table, replicas are shown with different colors:

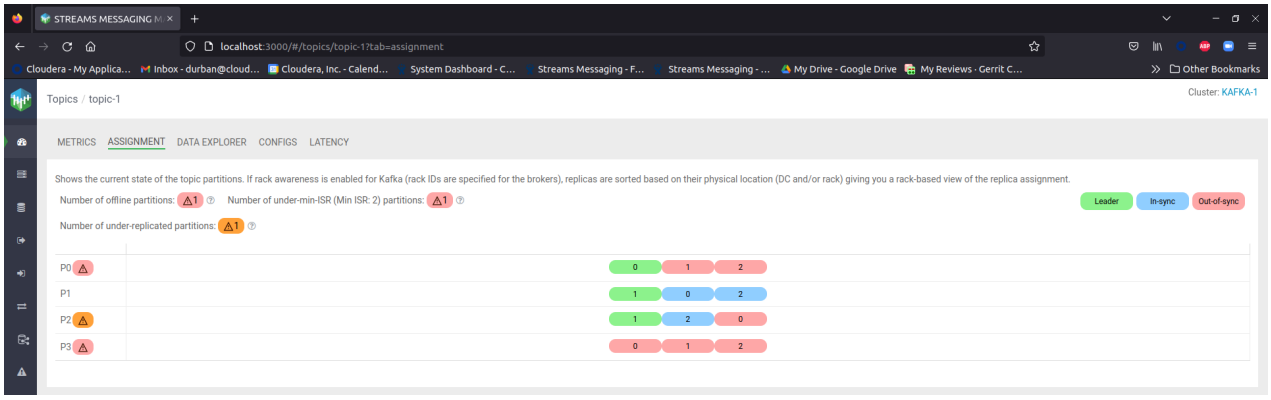
- Leader replicas are green
- In-sync replicas are blue
- Out-of-sync replicas are red

In addition, offline partitions and out-of-sync partitions are red. Under-replicated partitions, unevenly distributed racks, and unused racks are orange. When a specific partition or rack has one of these issues, a warning sign appears next to it.

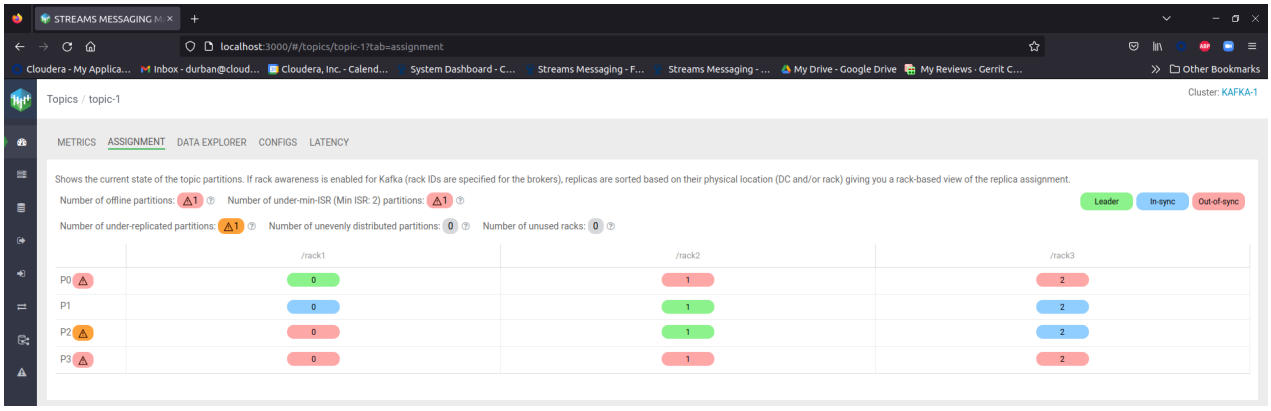
Leader and in-sync replicas also act as links to the corresponding broker details page.

In the table header and in the first column, warning icons are shown if the specific column or row is affected by one of the issues listed in the topic statistics.

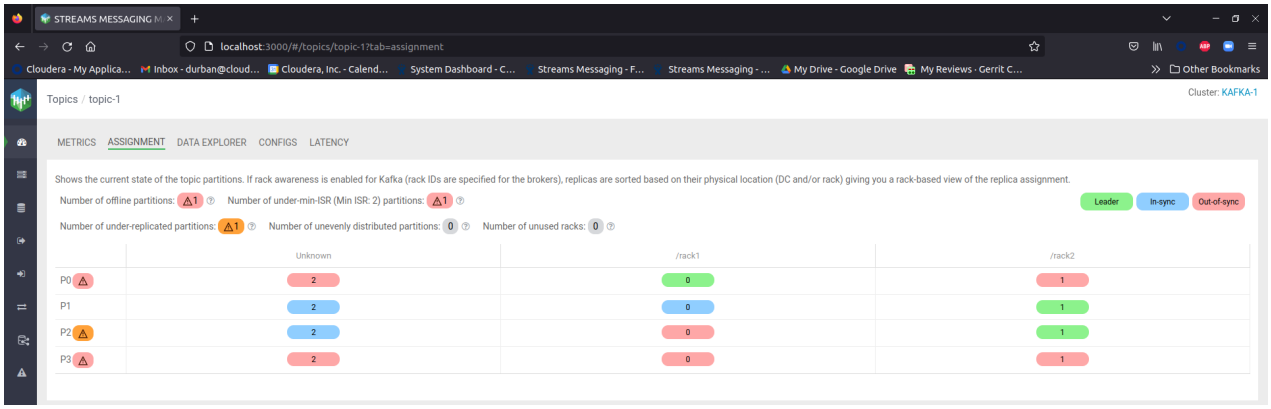
The following image shows the Assignment tab without racks:



The following image shows the Assignment tab with racks:



The following image shows the Assignment tab with racks, but with one broker metadata unknown, which adds an extra Unknown column to the table:



Related Information

Kafka rack awareness

Monitoring Kafka brokers

By monitoring Kafka brokers, you can track various details about brokers including the host where the broker is located, disk space used by the broker, throughput, messages coming in, partitions, and replicas.

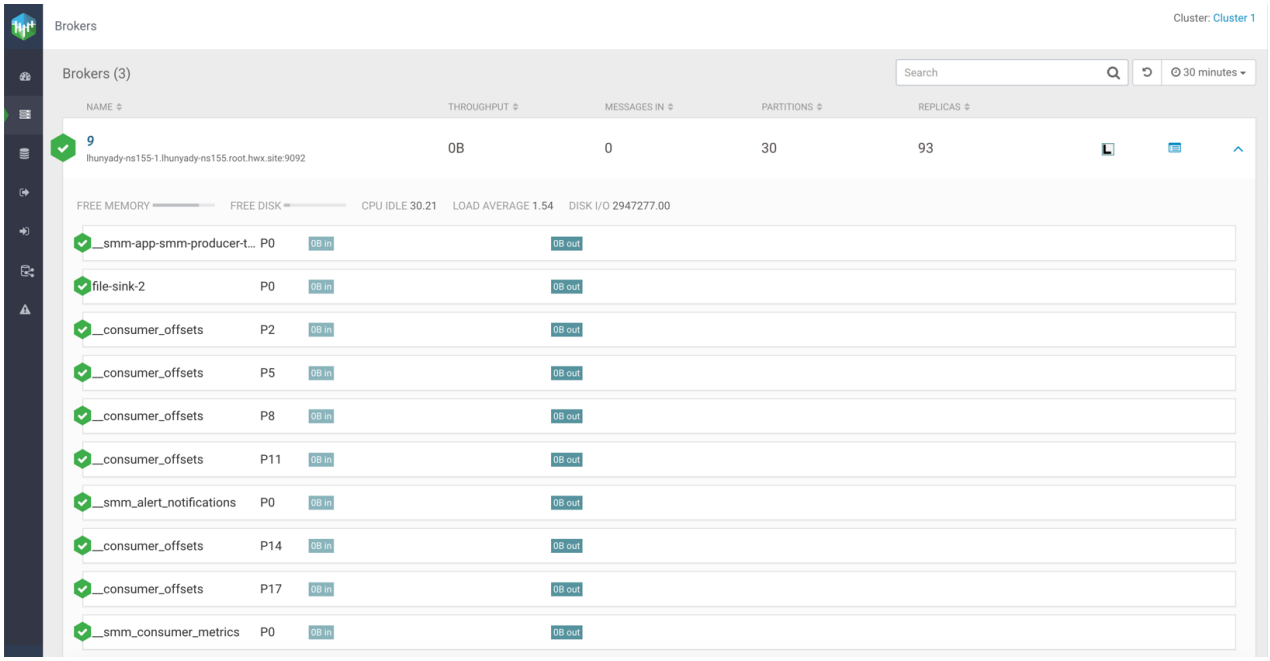
Detailed broker information

The Brokers page contains a number of useful details about your Kafka brokers. This page helps you answer the following questions:

- On what host is my broker located?
- Is my broker running out of disk space?

To access detailed broker information:

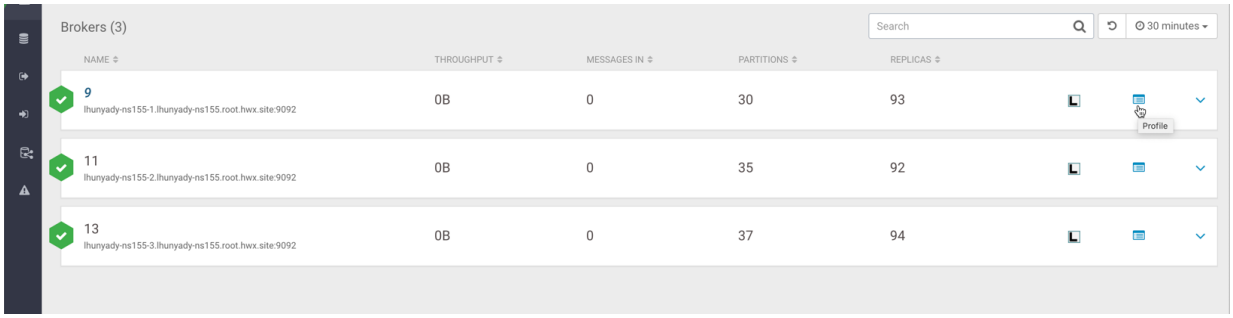
1. From the left navigation pane, click Brokers.
2. Identify the broker about which you want information. You can either scroll through the list of brokers, or use the Search bar at the top left of the page.
3. Click the green hexagon at the left of the broker to view details.



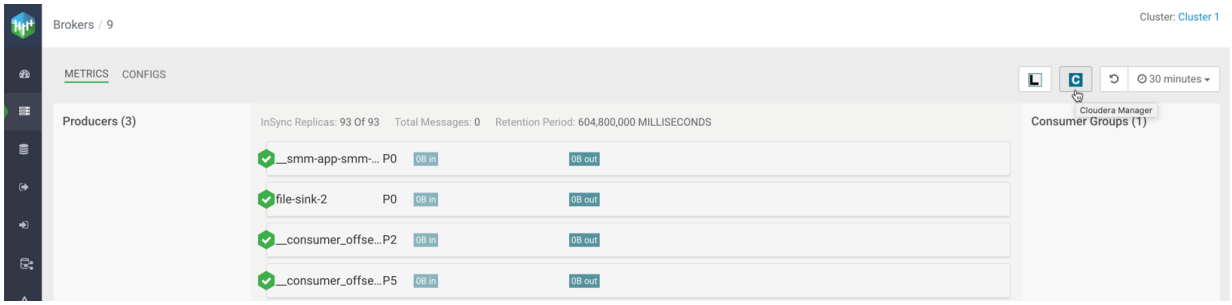
Viewing additional details about the broker host

You can view additional details about the broker host from Cloudera Manager. To access this information perform the following steps:

1. From the left navigation pane, click Brokers.
2. Identify the broker about which you want information. You can either scroll through the list of brokers, or use the Search bar at the top left of the page.
3. Click the Profile icon on the right side of the broker view.



4. Click the Cloudera Manager icon on the right side of the header.



You can track additional details about the broker host in the Metrics and Configs tabs.

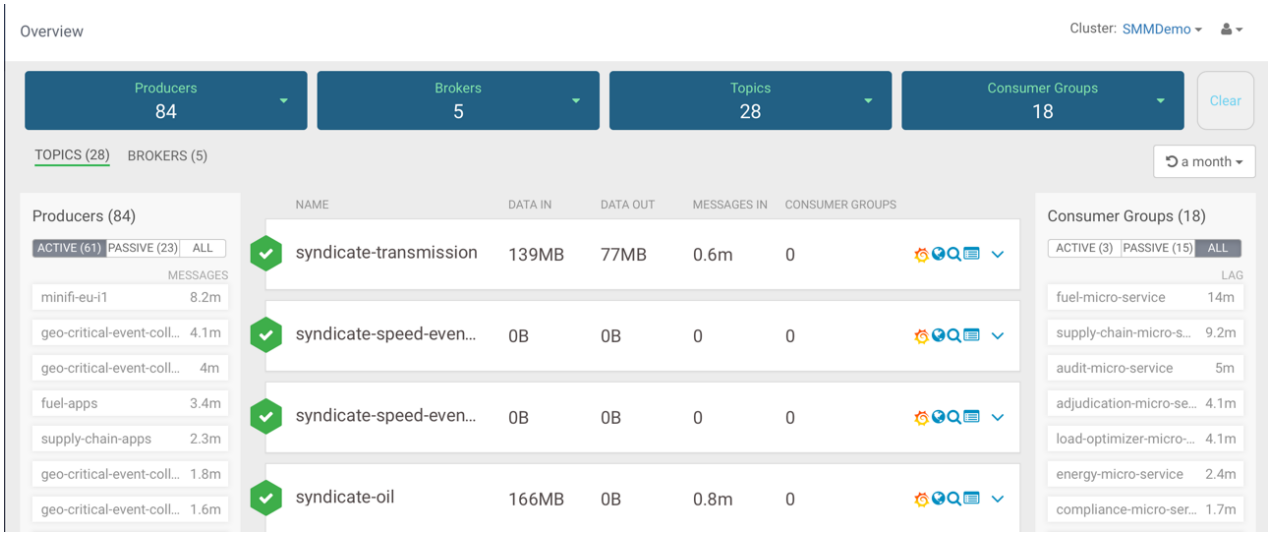
Monitoring Kafka consumers

By monitoring Kafka consumer groups, you can track active and passive consumer groups, or all consumer groups, which use the default internal __consumer_offsets topic to store the consumed offset information. You can track additional details about consumer groups. You can also track details including number of consumers and consumer instances included in a group and consumer group lag in the consumer group profile.

Streams Messaging Manager displays consumer groups that have offsets stored in Kafka's internal topic __consumer_offsets, which is also the default store if the auto.commit.enable property is set to true for consumers. Streams Messaging Manager does not display consumer groups that have offsets stored anywhere else other than this default store.

Viewing summary information about consumer groups

The Overview page gives you summary information about consumer groups on the right side of the page. You can use the Active, Passive, and All tabs to view consumer groups only in the Active or Passives, or all of the consumer groups, which use the default internal __consumer_offsets topic to store the consumed offset information. Use the Lag tab to sort consumer groups based on ascending or descending amounts of lag.



Viewing details about a consumer group

To access detailed consumer group information:

1. From the left navigation pane, click Consumer Group.
2. Identify the consumer group about which you want information. You can either scroll through the list of consumer groups, or use the Search bar at the top left of the page.
3. Click the green hexagon at the left of the consumer group to view details.



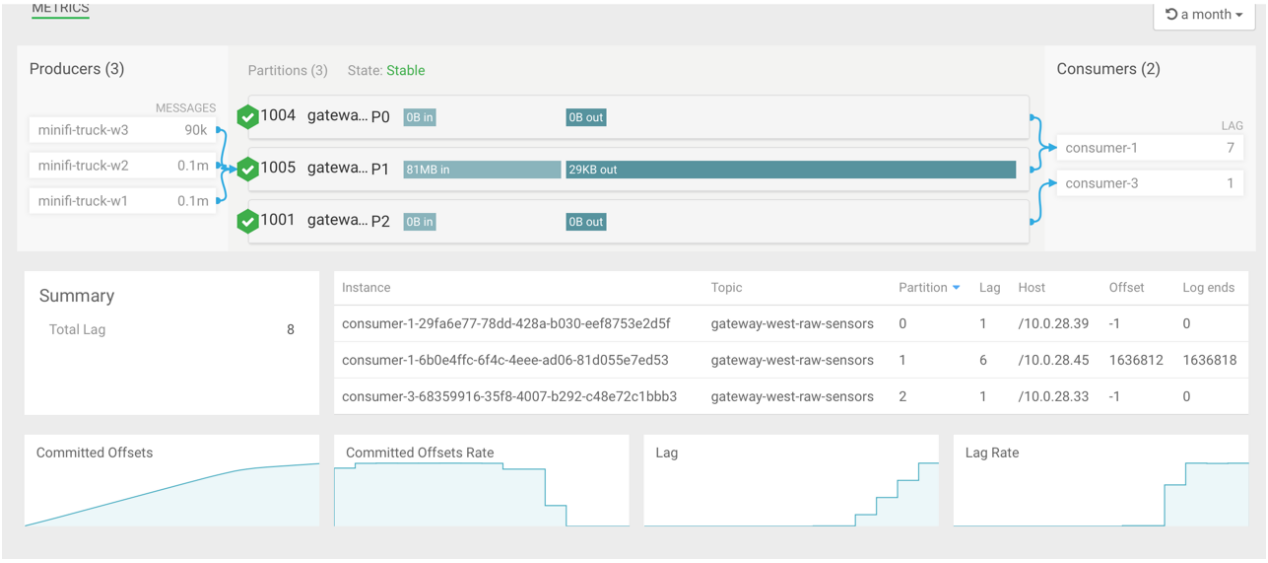
Viewing the consumer group profile

The Consumer Group profile displays detailed information about each consumer group, including:

- The number of consumers included in the group.
- The number of consumer instances in the group.
- Details about consumer group lag.

To access the Consumer Group profile:

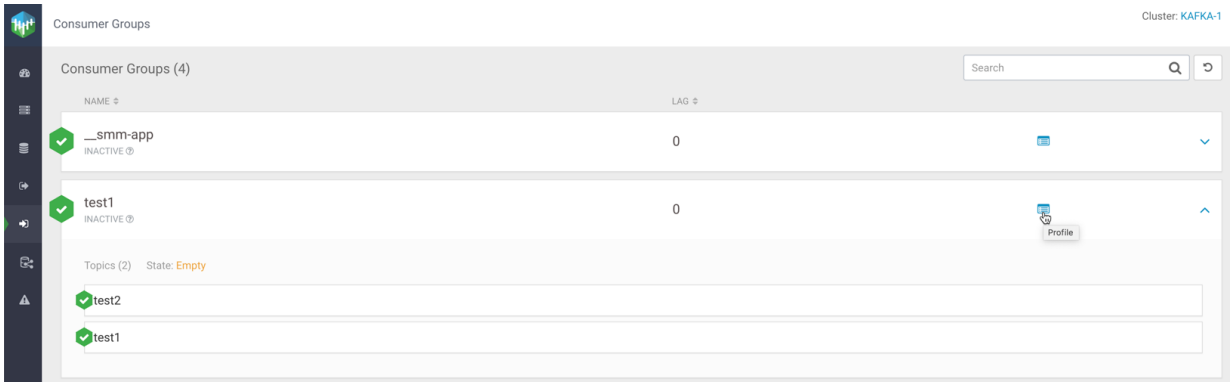
1. From the Consumer Group page, select the consumer group for which you want to view the profile.
2. Click the profile icon in the upper right of the Consumer Group tile.



Resetting consumer offset

To reset the offset of a consumer group, perform the following steps:

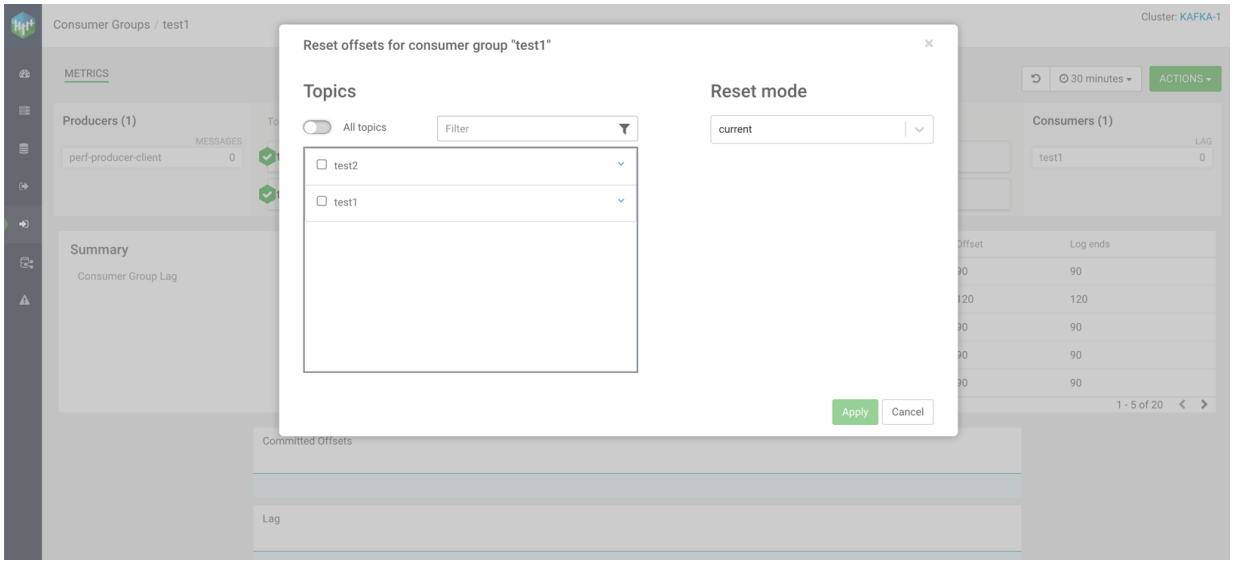
1. From the left navigation pane, click Consumer Group.
2. Choose the consumer group for which you want to reset the offset, and click the Profile icon.



Note: Resetting offsets is only possible for those groups whose state is Empty or Dead. Attempting to reset offset of any other group results in an error.

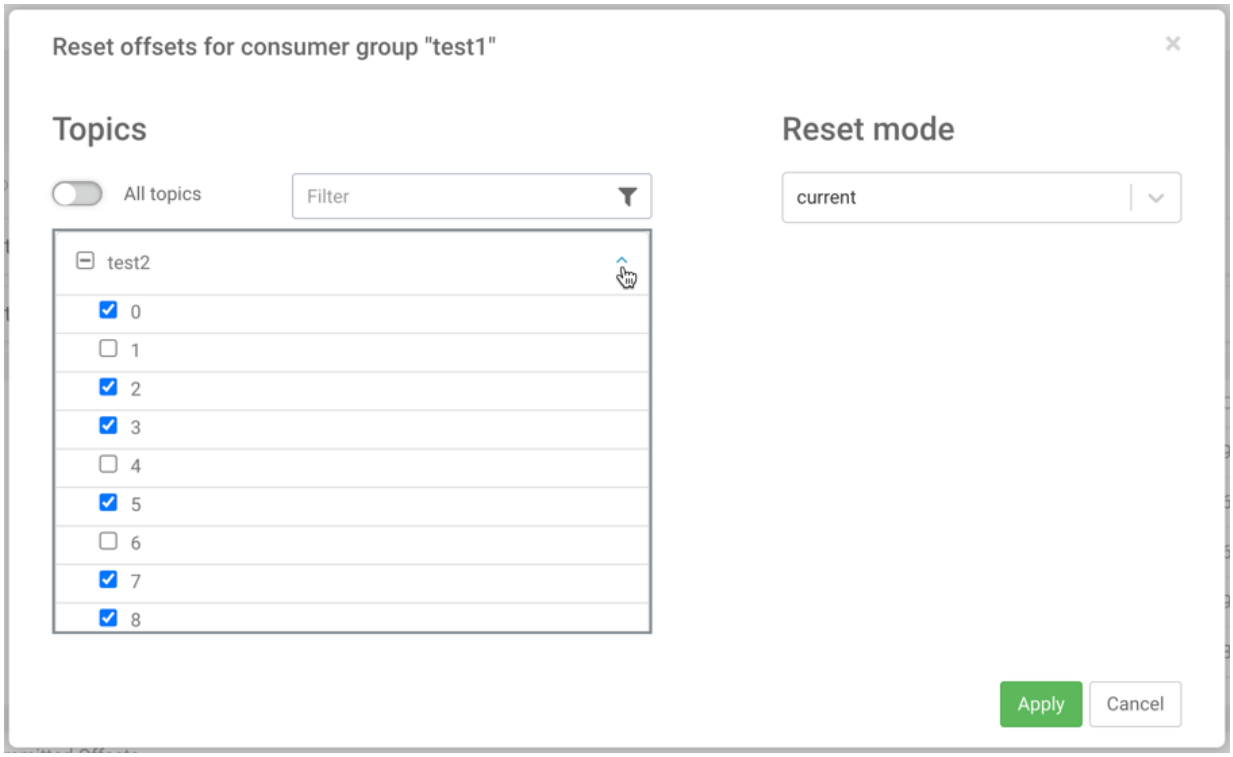
3. In the Metrics page, click Actions Reset offset .

   The Reset offsets for consumer group dialog appears.



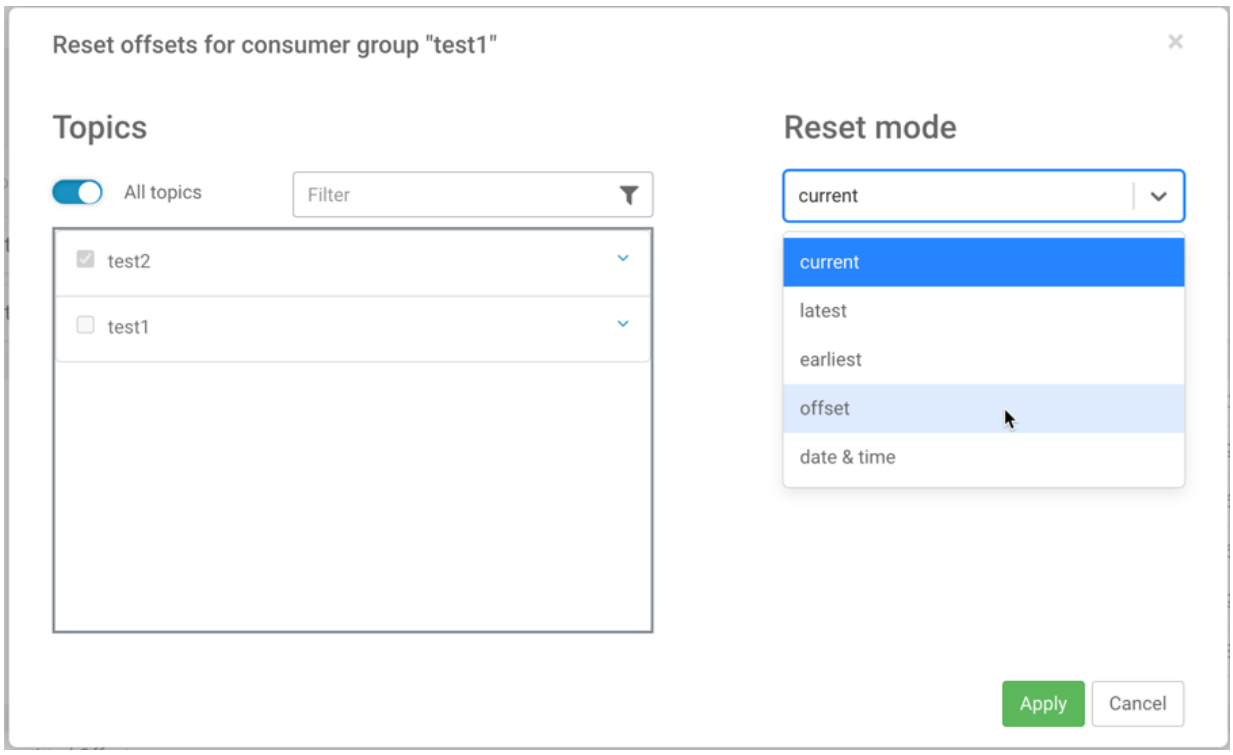
4. Select the topic and partition(s) you want to reset.

   You can use the arrow beside each topic to display the partitions of that topic, and select partition(s) as required. You can also select all topics and all related partitions by selecting the All topics option. You can use the Filter option to find a specific topic.



5. Select the reset option from the Reset mode.

   The available options are as shown in the following image:



6. Click Apply.

Monitoring log size information

The Streams Messaging Manager UI shows log size related information about brokers, topics, and partitions. Furthermore, warning messages appear when log directory related errors happen.

The feature uses a cache mechanism. The following cache attributes are set by default. You can modify these attributes in the Streams Messaging Manager configuration page of the Cloudera Manager UI:

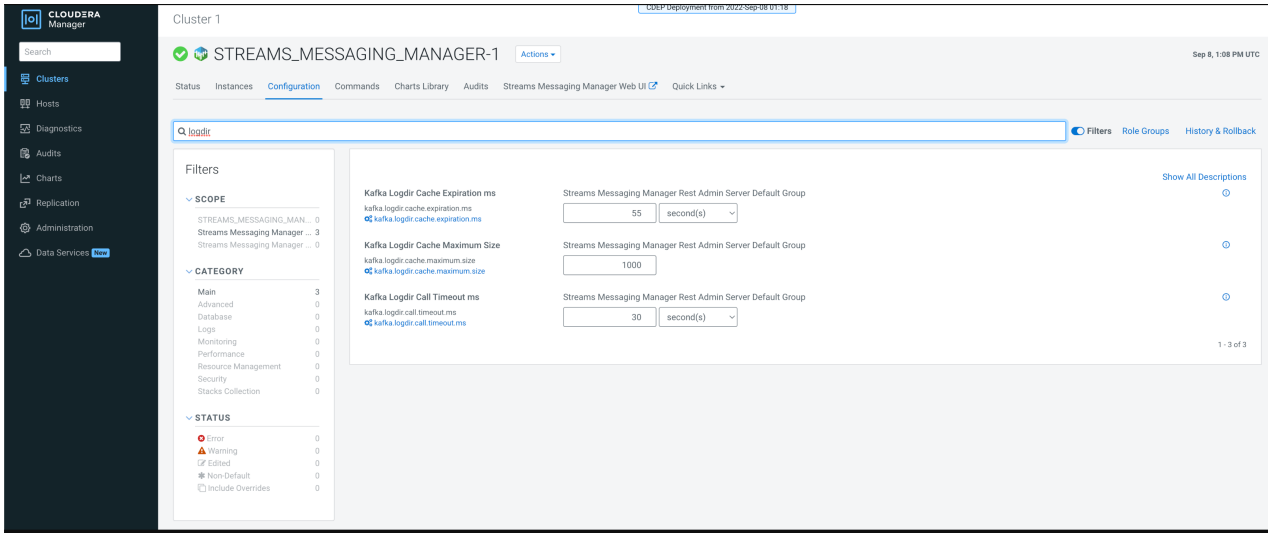
- Kafka Logdir Cache Expiration ms

  Sets cache expiration time.
- Kafka Logdir Cache Maximum Size

  Sets maximum cache size.
- Kafka Logdir Call Timeout ms

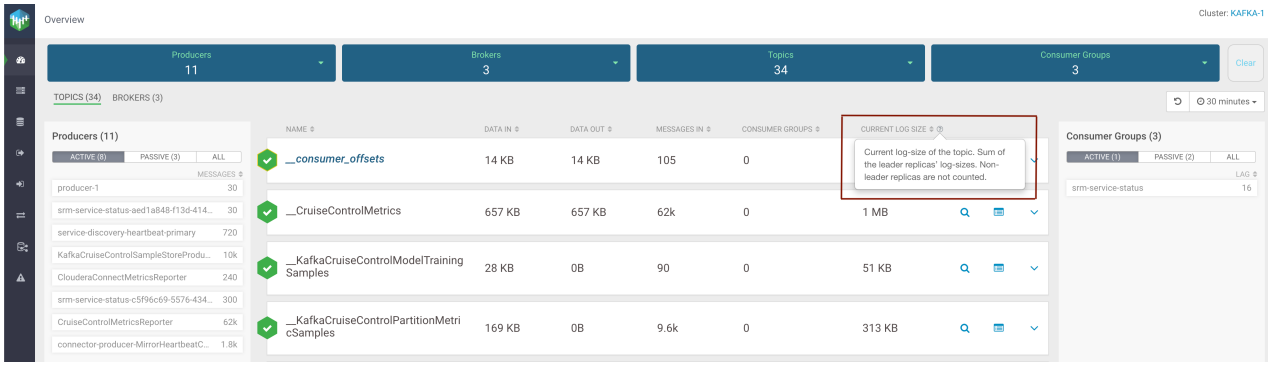
  Sets timeout of the Kafka query.

The following image shows the properties in Cloudera Manager:

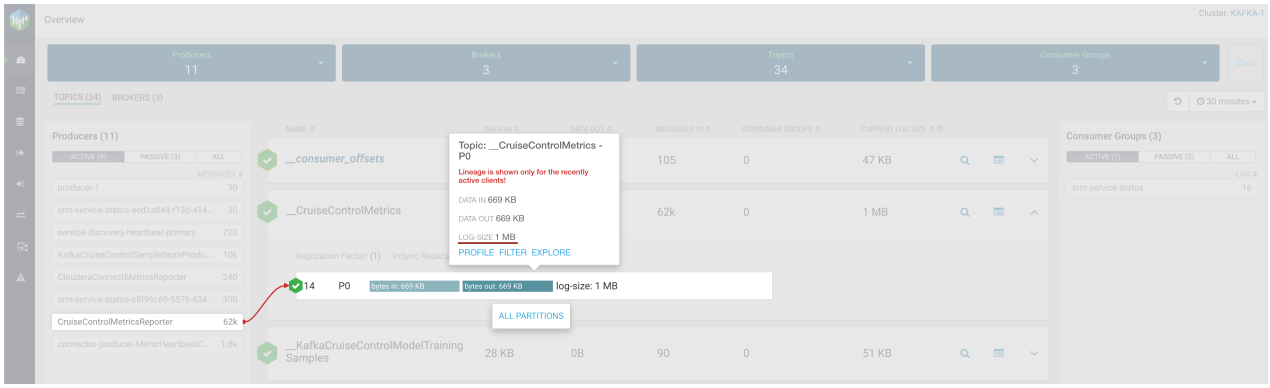


Log size details for topics

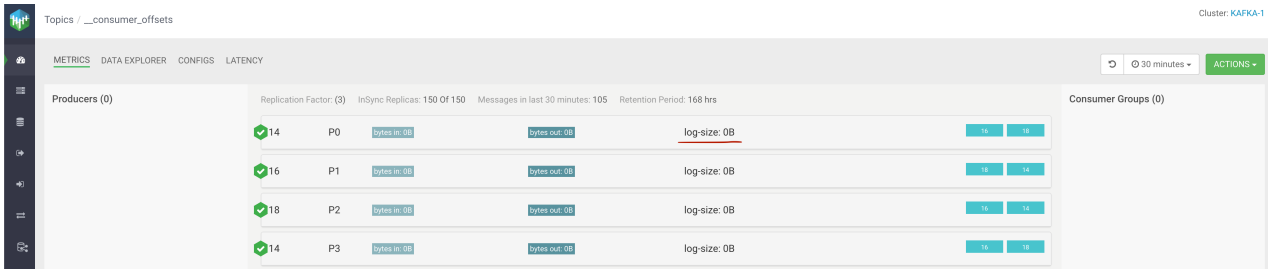
On the Overview page of the Streams Messaging Manager UI, the Topics tab contains a sortable column called Current Log Size which shows the data with the measure unit. This log size is calculated by the leader partitions; so, the follower partitions are not included. The column contains a tooltip which is a question-mark symbol. The topic related log-size improvements or information are also available on the Topics page.



The Topics tab on the Overview page represents topics and you can select one of them. In this case the topic's leader partitions are viewable and here you can find the leader partition log-size. If you click a partition, a pop-up window appears, where log-size information is displayed.

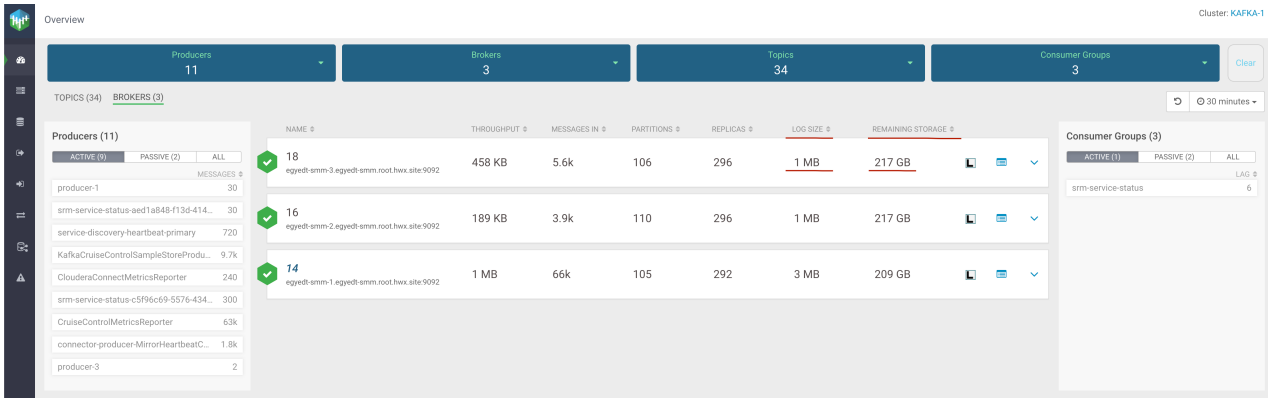


If you click Profile for a topic, log-size of replica-leaders appears, hosted by the broker.

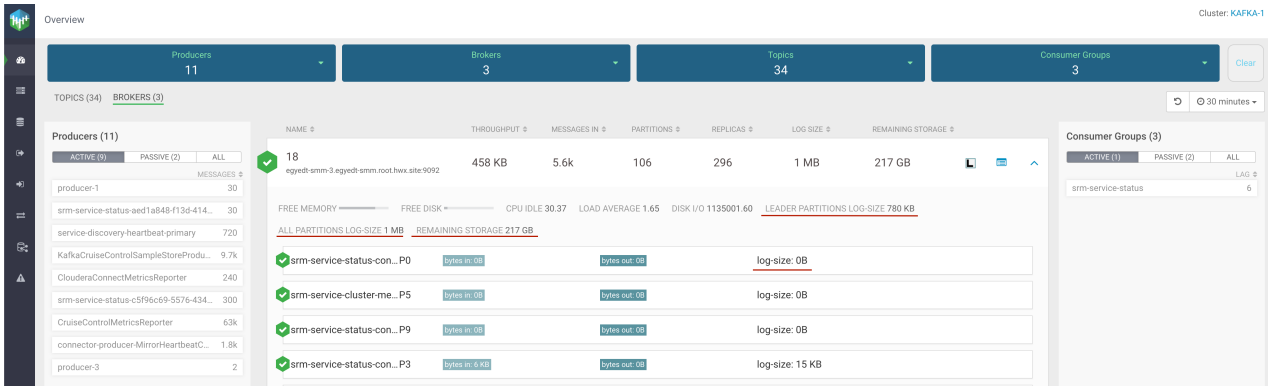


Log size details for brokers

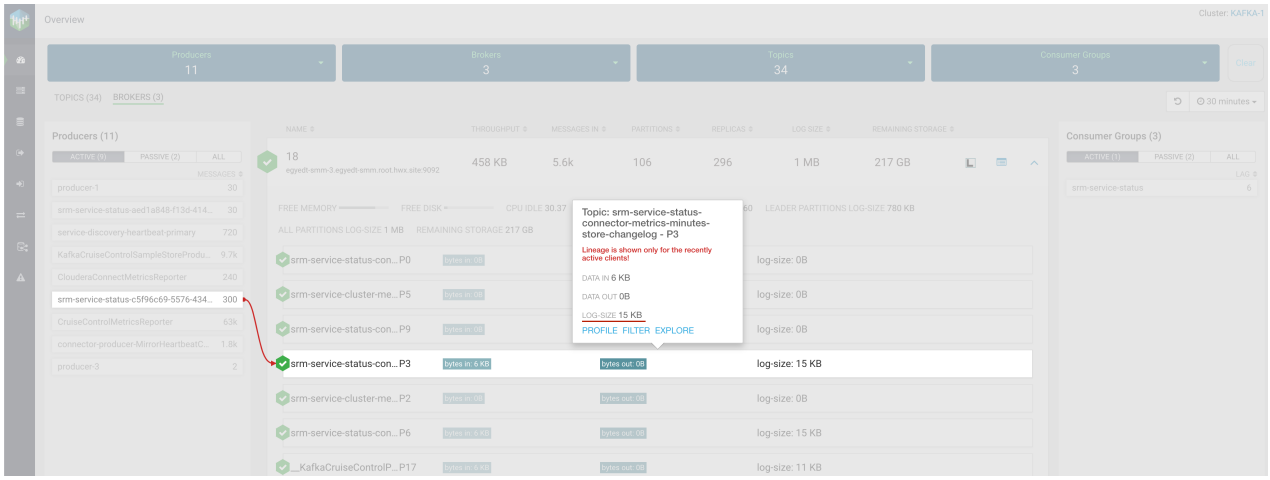
On the Overview page, the Brokers tab contains two sortable columns called Log Size and Remaining Storage. The log-size is calculated for all the partitions including the follower partitions. The remaining storage column contains the available capacity in all of the configured log directories. The broker related log-size improvements or information are also available on the Brokers page.



If you click the arrow icon for a broker, the details of that broker are displayed. The Leader Partition Log-Size shows the sum of leader partitions' log-size. The All Partitions Log-Size field's value is equal to the sortable column's value called Log Size (naming is different to make the sortable column names as short as possible). The Remaining Storage sortable column and field is the same (value and name).

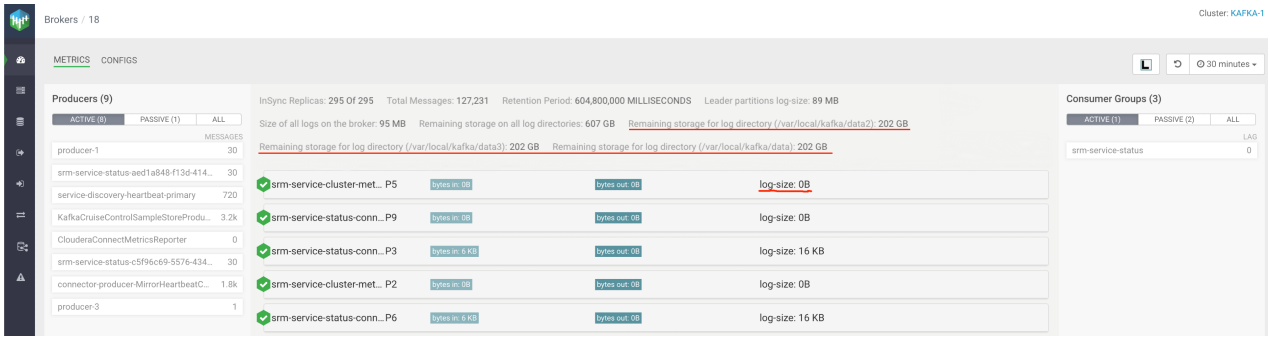


If you click a specific partition, a pop-up window appears where log-size information is displayed.

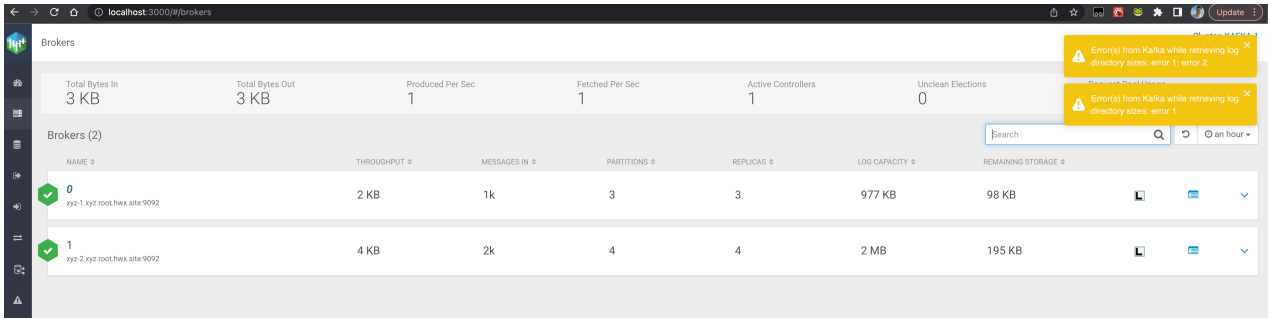


If you click Profile of a broker, then the metrics details for that broker are displayed. The log-size information per partition is available. Furthermore, the sum of leader partitions' log-size, remaining storage size, and all partitions log-size details are also available. The Metrics tab on the Brokers page contains the available storage size per log directory.

Note: There is no advantage of creating more log directories on the same disk, so this should be avoided.



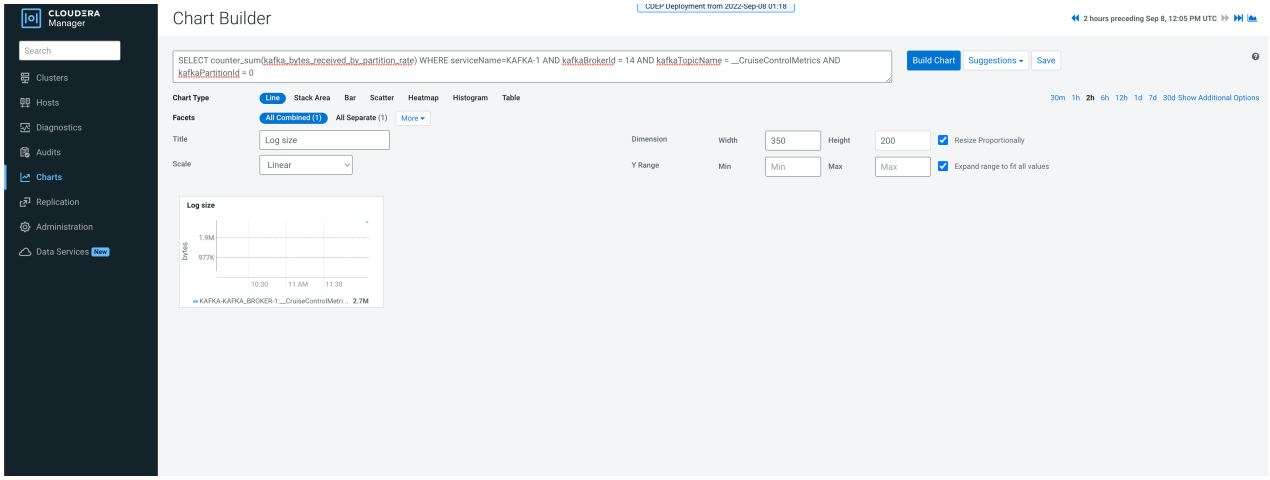
If there is any issue with the query on the log directory information, then a warning message appears. There can be multiple warnings if more than one broker is related. But no new warning message from a specific broker appears until the previous message disappears.



Cloudera Manager chart builder

For historical information, you can use the chart builder provided in Cloudera Manager. The following command and screenshot show Kafka log-size data from the past where the broker ID and topic name with the topic partition are specified.

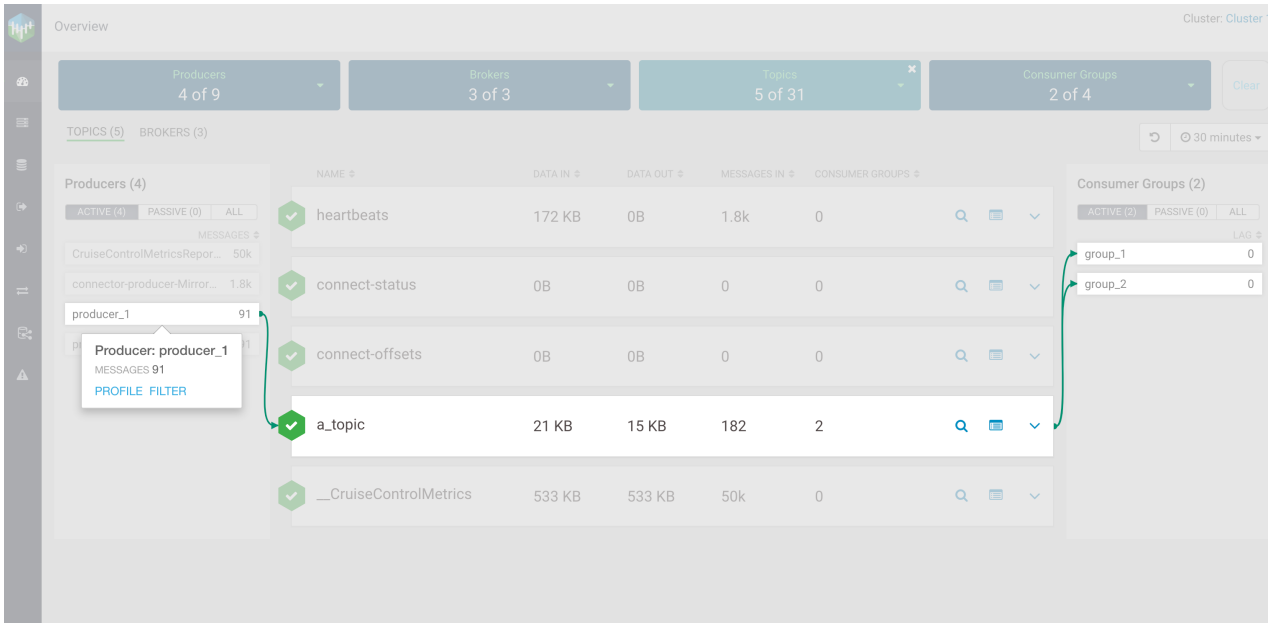
```
SELECT counter_sum(kafka_bytes_received_by_partition_rate) WHERE serviceName
=KafkaServiceName AND kafkaBrokerId = KafkaBrokerId AND kafkaTopicName = Kaf
kaTopicName AND kafkaPartitionId = kafkapartitionId
```



Monitoring lineage information

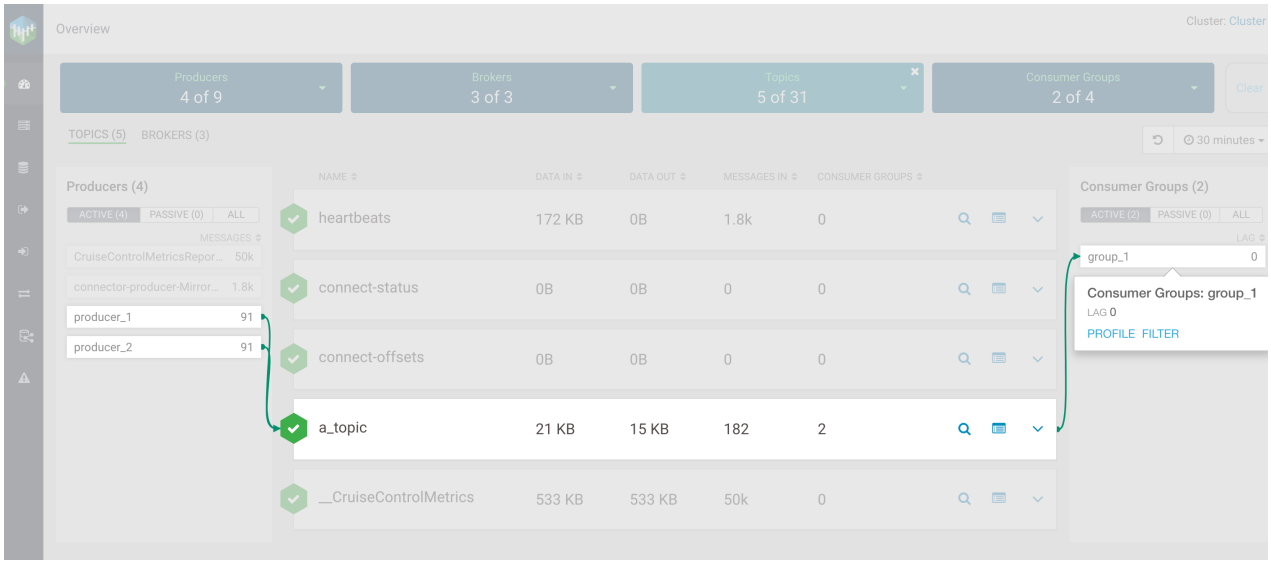
Learn how you can visualize the lineage between producers and consumers.

To check which topics a producer is producing to, and which consumers consume from those topics, go to the Overview page and click on a single producer on the Producer pane. For example, click producer_1, as shown in the following image:



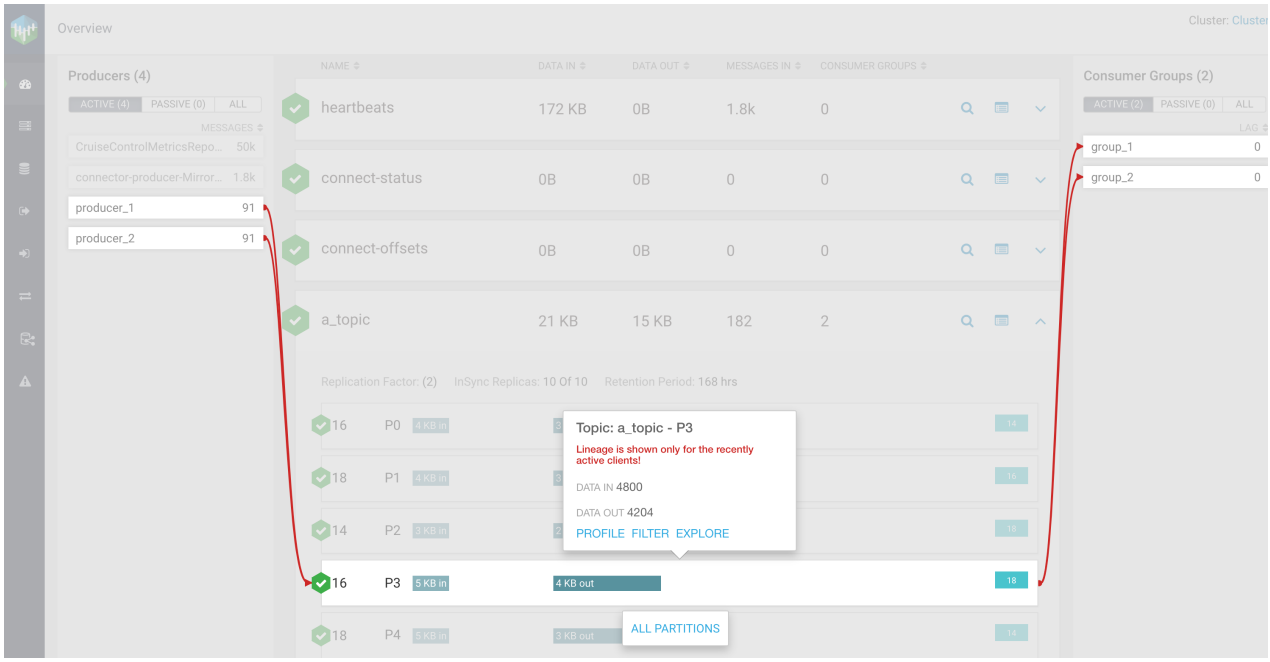
After you click producer_1, you can see that it produces to a topic called a_topic, and that both consumer groups (group_1 and group_2) consume from that topic.

This works the other way round as well. If you click on a single consumer group, you see what topics it consumes from and which producers produce to those topics. For example, click group_1, as shown in the following image:



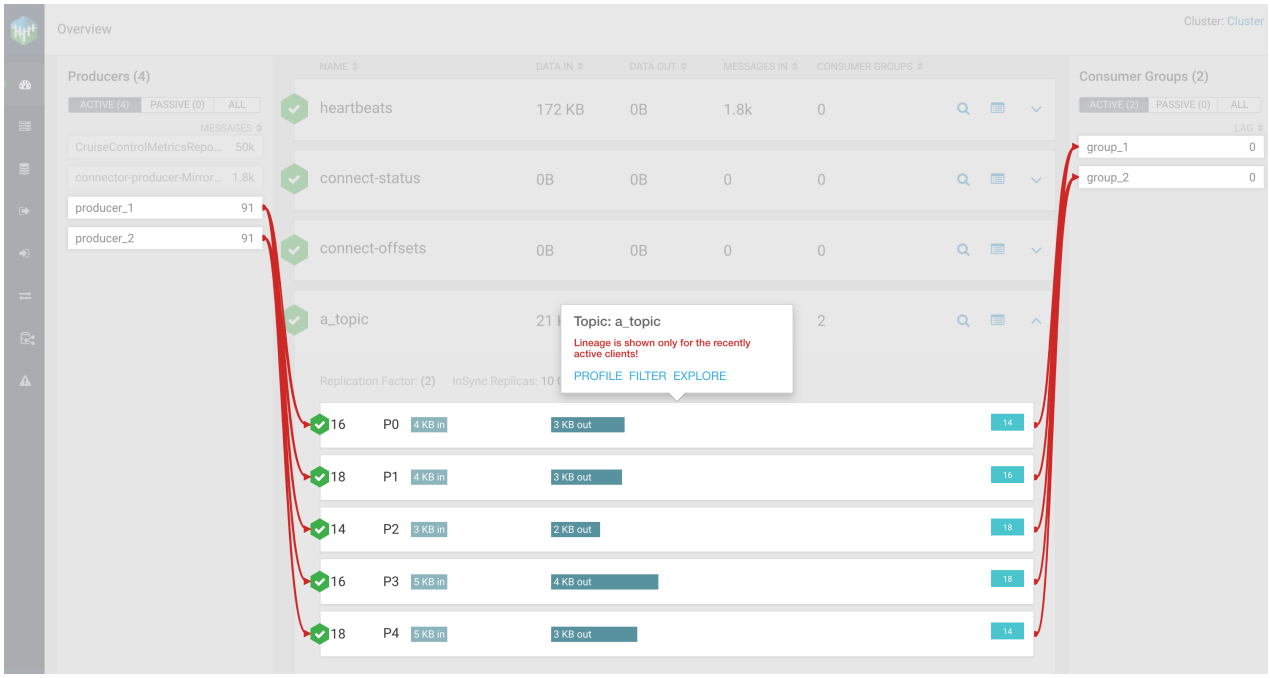
After you click group_1 consumer group, you can see that it consumes from the topic called a_topic, and that two producers produce to that topic (producer_1 and producer_2).

If you are interested in a more detailed view and want to check the lineage information for a single partition, you can do that as well, however, it is important to note that the lineage information is provided exclusively for the last 30 minutes. For example, click P3, as shown in the following image:



After you click P3 partition in the topic called a_topic, you can see that producer_1 and producer_2 produce to that partition, and group_1 and group_2 consume from it.

If you click the All Partitions button, you are shown the lineage information for every partition in a single topic.



You can also access the lineage information from the experimental endpoints. You can find the endpoints at the *Streams Messaging Manager REST API Reference*.

Related Information

Streams Messaging Manager REST API Reference

Managing Kafka topics

Learn how to create, modify, and delete Kafka topics using Streams Messaging Manager.

Creating a Kafka topic

Learn how to create Kafka topics by using Streams Messaging Manager UI.

About this task

You can create a new Kafka topic by navigating to the Topics page and providing information prompted in the Add New pop-up.

Procedure

1. Navigate to the Topics page.
2. Click Add New.
3. Provide the following information:
   - Topic name
   - Number of partitions
   - Level of availability
   - Cleanup policy
4. Streams Messaging Manager automatically sets Kafka topic properties. You can manually adjust the properties by clicking advanced.

For more information on configuration properties, see Apache Kafka documentation.

5. Click Save.

Related Information

Apache Kafka documentation: Topic-Level Configs

Modifying a Kafka topic

Learn how to modify Kafka topics by using Streams Messaging Manager UI.

About this task

You can modify Kafka topic properties by navigating to the Topics page, and editing content in the Configs tab of the topic Profile.

Procedure

1. Navigate to the Topics page and select the topic you want to modify.
2. Select the Profile icon.
3. Select the Configs tab.
4. Update the configurations.

   Note: You cannot change the topic name or the replication factor.

   In the simple version, you can update the number of partitions at partitions and the cleanup.policy.

   You can click advanced to access the additional properties. For more information on configuration properties, see Apache Kafka documentation.
5. Click Save.

Related Information

Apache Kafka documentation: Topic-Level Configs

Deleting a Kafka topic

Learn how to delete Kafka topics by using Streams Messaging Manager UI.

About this task

You can delete Kafka topics by navigating to the Topics page, and using the Delete Topic option from the topic Profile.

Procedure

1. Navigate to the Topics page and select the topic you want to delete.
2. Click the Profile icon.
3. Click the Actions button and select Delete Topic.
4. Confirm that you want to delete the topic.

Managing Alert Policies and Notifiers

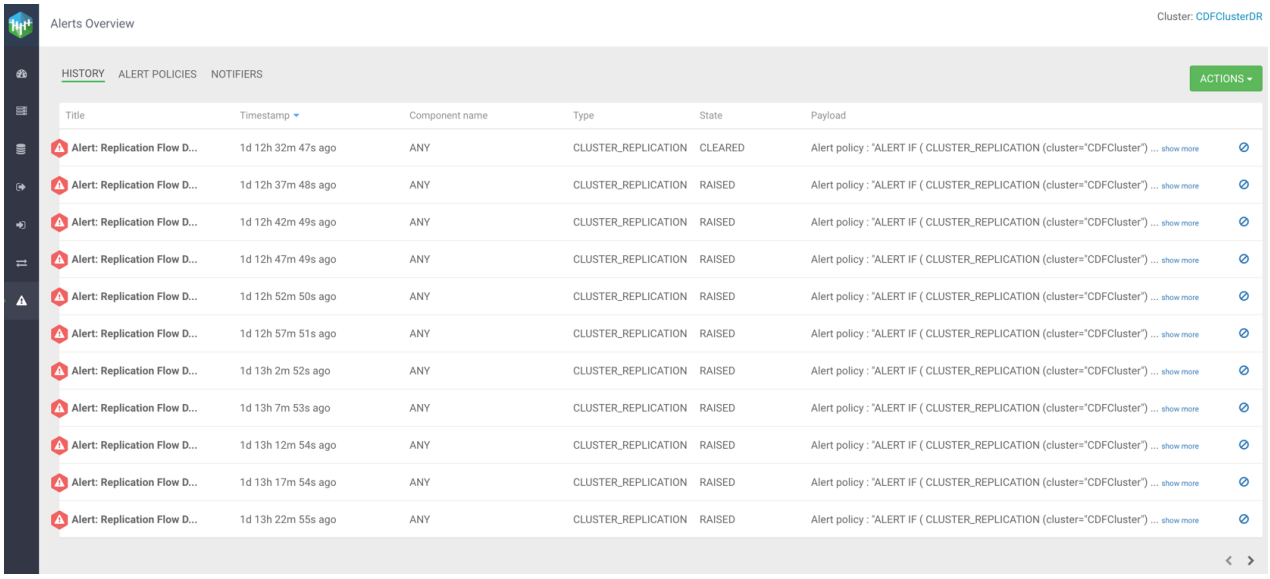
Learn what Streams Messaging Manager Alert Policies and Notifiers are and how you can manage them.

Alert Policies in Streams Messaging Manager are user configured alerts that automatically trigger when configured conditions are met. An alert contains the details of the policy including the alert message and the conditions that triggered the alert. Once an alert is triggered, a notification is sent out by Streams Messaging Manager using the Notifier associated with the Alert Policy. You can use these alerts and notifications to monitor the health of Kafka, as well as to identify and troubleshoot problems.

For example, assume that you want to monitor your Kafka topics to ensure that data production and consumption is happening without interruptions. To do this you could set up an Alert Policy that triggers if the bytes consumed from or produced to your topics fall below a specified threshold. When the alert is triggered, Streams Messaging Manager sends out a notification using the Notifier associated with the Alert Policy immediately notifying you of a potential issue.

UI overview

Alerts in the navigation sidebar takes you to the **Alerts Overview** page, which you use to view and manage Alert Policies and Notifiers.



The page consists of three tabs. These are as follows:

For HISTORY

The **HISTORY** tab lists all triggered alerts. You can check the title, timestamp details, component name, type, state, and payload information of an alert. You can click show more to check complete payload details for an alert.

Click ACTIONS Mark All As Read to mark all the alerts as read. You can also click ⊘ Dismiss to mark an individual alert as read.

For ALERT POLICIES

The **ALERT POLICIES** tab lists available Alert Policies. You can create new Alert Policies with ADD NEW. Additionally, you can manage existing policies.

For NOTIFIERS

The **NOTIFIERS** tab lists available Notifiers. You can create new Notifiers with ADD NEW. Additionally, you can manage existing Notifiers.

Creating a notifier

Learn how to use Streams Messaging Manager to create a notifier in your environment.

About this task

Perform the following steps to create a notifier:

Procedure

1. From the left navigation pane, select Alerts.
   The Alerts Overview page appears.
2. Click NOTIFIERS.
3. Click ADD NEW to create a new notifier.
   The Notifier window appears.
4. Configure the following properties:

Configuration	Property	Description
Common Notifier Configuration	NAME	Enter a unique name for the notifier.
	DESCRIPTION	Enter an optional description for the notifier.
	PROVIDER	Choose one of the following providers: • Email • HTTP
	NOTIFIER RATE LIMIT COUNT	Select the number of allowed notifications.
	NOTIFIER RATE LIMIT DURATION	Select the number of allowed notifications with respect to given duration in SECONDS, MINUTES, or HOURS.
Email Notifier Configuration	FROM ADDRESS	Enter the email address to use for SMTP mail command. Default is admin@localhost.
	TO ADDRESS	Enter one or multiple email addresses that you want to send the notification to.
	USERNAME	Enter the username for SMTP.
	PASSWORD	Enter the password for SMTP.  **Note:** The password must be reentered when editing the configuration of existing notifiers.
	SMTP HOSTNAME	Enter the SMTP server that you want to connect to. Default is localhost.
	SMTP PORT	Enter the SMTP server port that you want to connect to. Default is 25.
	ENABLE AUTH	Select to enable authentication.
	ENABLE SSL/STARTTLS	Select to enable SSL. This is applicable when you enable authentication. You can either select SSL or STARTTLS.
	PROTOCOL	Enter the protocol to use to send emails. Default is SMTP.
	ENABLE DEBUG	Select to enable debug mode to trace any issue in the email notifier. Disabled by default.
HTTP Notifier Configuration	URL	Enter the target service URL.

Configuration	Property	Description
	CONNECTION TIMEOUT (MSECS)	Select the connection timeout in milliseconds for creating the initial connection. Default is 30 seconds.
	READ TIMEOUT (MSECS)	Select the read timeout in milliseconds for waiting to read data. Default is 30 seconds.

5. Click Save.

Updating a notifier

Learn how to use Streams Messaging Manager to update a notifier that you create in your environment.

About this task

Perform the following steps to update a notifier:

Procedure

1. From the left navigation pane, select Alerts.
   The Alerts Overview page appears.
2. Click NOTIFIERS.
3. Find the notifier you want to update from the list of available notifiers, and click the pencil icon beside the notifier.
   The Notifier window appears.
4. Edit the properties.

   Note: The password of email notifiers must be reentered when you update the configuration of the notifier.

5. Click Save.

Deleting a notifier

You can use Streams Messaging Manager to delete a notifier that you create in your environment. You can delete a notifier only if the notifier is not mapped to an alert policy.

About this task

Perform the following steps to delete a notifier:

Procedure

1. From the left navigation pane, select Alerts.
   The Alerts Overview page appears.
2. Click NOTIFIERS.
3. Find the notifier you want to delete from the list of available notifiers, and click the delete icon beside the notifier.
4. Click Yes.

Creating an alert policy

You can use Streams Messaging Manager to create an alert policy in your environment.

About this task

Perform the following steps to create an alert policy:

Procedure

1. From the left navigation pane, select Alerts.
   The Alerts Overview page appears.
2. Click ALERT POLICIES.
3. Click ADD NEW to create a new alert policy.
   The Alert Policy window appears.
4. Configure the following properties:

Property	Description
NAME	Enter a unique name for the alert policy.
DESCRIPTION	Enter a description for the alert policy.
EXECUTION INTERVAL IN SECONDS	Enter the execution interval in seconds to run the alert policy periodically after the given time interval.
EXECUTION DELAY IN SECONDS	Enter the execution delay in seconds to delay the execution of the alert policy. This is applicable only when the last execution of the alert policy triggered an alert. Ideally, this value should not be less than the value you enter for the EXECUTION INTERVAL IN SECONDS option.
ENABLE	Choose to enable or disable the alert policy.
COMPONENT TYPE	Select one of the following component types for the IF policy: • Broker • Consumer • Producer • Topic • Latency • Cluster • Cluster replication
TARGET NAME	Select the target name for the IF policy. You can add multiple WITH conditions by clicking the plus icon beside TARGET NAME.
TARGET CLUSTER NAME	Select the name of the target cluster for the IF policy. The property is visible when you select Cluster replication component type.
SOURCE CLUSTER NAME	Select the name of the source cluster for the IF policy. The property is visible when you select Cluster replication component type.
TOPIC NAME	Select the topic name for the IF policy. The property is visible when you select Latency or Cluster replication component type. For Cluster replication, the topic name represents the name of the topic on the target cluster.
CONSUMER NAME	Select the consumer name for the IF policy. The property is visible when you select Latency component type.
ATTRIBUTE	Select the attribute for the policy.
CONDITION	Select the condition for the policy.

Property	Description
VALUE	Select the value for the policy. You can add multiple attributes, conditions, and values by clicking the plus icon beside VALUE.
NOTIFICATION	Select a notifier.
PREVIEW	Displays the alert that you configure. For example, IF [COMPONENT_TYPE]: [TARGET_NAME] has [METRIC] [CONDITION] [VALUE] THEN notify by [NOTIFICATION]

5. Click Save.

Updating an alert policy

You can use Streams Messaging Manager to update an alert policy in your environment.

About this task

Perform the following steps to update an alert policy:

Procedure

1. From the left navigation pane, select Alerts.
   The Alerts Overview page appears.
2. Click ALERT POLICIES.
3. Find the alert policy that you want to update, and click the pencil icon beside the alert policy.
   The Alert Policy window appears.
4. Edit the properties.
5. Click Save.

Enabling an alert policy

You can use Streams Messaging Manager to enable an alert policy in your environment.

About this task

Perform the following steps to enable an alert policy:

Procedure

1. From the left navigation pane, select Alerts.
   The Alerts Overview page appears.
2. Click ALERT POLICIES.
3. Find the alert policy that you want to enable, and click the enable icon beside the alert policy.
   The alert policy is enabled.

Disabling an alert policy

You can use Streams Messaging Manager to disable an alert policy in your environment.

About this task

Perform the following steps to disable an alert policy:

Procedure

1. From the left navigation pane, select Alerts.
   The Alerts Overview page appears.
2. Click ALERT POLICIES.
3. Find the alert policy that you want to disable, and click the enable icon beside the alert policy.
   The alert policy is disabled.

Deleting an alert policy

You can use Streams Messaging Manager to delete an alert policy in your environment.

About this task

Perform the following steps to delete an alert policy:

Procedure

1. From the left navigation pane, select Alerts.
   The Alerts Overview page appears.
2. Click ALERT POLICIES.
3. Find the alert policy that you want to delete, and click the delete icon beside the alert policy.
4. Click Yes.

Component types and metrics for alert policies

You create an alert policy for a component type. The component type drives the list of metrics to select for creating a threshold. Learn the different component types and supported metrics for each component type.

The following table lists the component types and metrics for an alert policy:

Table 1: Component Types and Metrics

Metric	Description	Suggested Alert
Topic		
UNDER REPLICATED PARTITIONS COUNT	Total number of partitions that are under replicated for a topic.	Value > 0.
BYTES IN PER SEC	Bytes per second coming in to a topic.	Two kinds of alert can be configured. • Alert-1: Value = 0, raises an alert when the topic becomes idle. • Alert-2: Value > max_bytes_in_expected, raises an alert when the topic input load is higher than usual.
BYTES OUT PER SEC	Bytes per second going out from a topic. It does not count the internal replication traffic.	Two kinds of alert can be configured. • Alert-1: Value = 0, raises an alert when the topic becomes idle. • Alert-2: Value > max_bytes_out_expected, raises an alert when the topic output load is higher than usual.
OUT OF SYNC REPLICA COUNT	Total number of replicas that are not in sync with the leader for a topic.	Value > 0, raises an alert if there are out of sync replicas for the topic.

Metric	Description	Suggested Alert
TOPIC PARTITION CONSUMPTION PERCENTAGE	Percentage of bytes consumed per topic partition compared according to the configured parameter retention.bytes. If retention.bytes is not configured, any condition involving this metric would be false.	Value > max_expected_value, raises an alert if the topic partition reaches a certain consumption percentage.
TOPIC PARTITION BYTES IN PER SEC	Bytes per second coming in to a topic partition.	Two kinds of alert can be configured. • Alert-1: Value = 0, raises an alert when the topic partition becomes idle. • Alert-2: Value > max_bytes_in_expected, raises an alert when the topic partition input load is higher than usual.
TOPIC PARTITION BYTES OUT PER SEC	Bytes per second coming out of a topic partition.	Two kinds of alert can be configured. • Alert-1: Value = 0, raises an alert when the topic partition becomes idle. • Alert-2: Value > max_bytes_out_expected, raises an alert when the topic partition output load is higher than usual.
Producer		
IS PRODUCER ACTIVE	Checks whether a producer is active.	Value is False.
MILLISECONDS LAPSED SINCE PRODUCER WAS ACTIVE	Milliseconds passed since the producer was last active.	Value > max_producer_idle_time, raises an alert if the producer did not produce for max_producer_idle_time ms.
Cluster		
ACTIVE CONTROLLER COUNT	Number of brokers in the cluster reporting as the active controller in the last interval.	Value != 1.
ONLINE BROKER COUNT	Number of brokers that are currently online.	Depends on the application. For example, you can raise an alert if the number of brokers falls below the min.insync.replicas configured for the producer. • Alert-1: Value < min.insync.replicas, raises an alert when producer could not send any messages. • Alert-2: Value = min.insync.replicas, raises an alert that denotes if any one of the remaining brokers goes down, then producer would not be able to send messages.
UNCLEAN LEADER ELECTION COUNT	Number of unclean partition leader elections in the cluster reported in the last interval.	Value > 0.
UNDER REPLICATED PARTITIONS COUNT	Total number of topic partitions in the cluster that are under replicated.	Value > 0.
LEADER ELECTION PER SEC	Rate of partition leader elections.  **Important:** Alert policies based on LEADER ELECTION PER SEC do not trigger if the monitored Kafka cluster is running in KRaft mode.	Depends on the number of partitions in the application.
OFFLINE PARTITIONS COUNT	Total number of topic partitions, in the cluster, that are offline.	Value > 0.

Metric	Description	Suggested Alert
NETWORK PROCESSOR AVG IDLE PERCENT	Average fraction of time the network processor threads are idle across the cluster.	Two kinds of alert can be configured. • Alert-1: Value = 0, raises an alert when the network processor threads are busy. • Alert-2: Value > network_processor_idle_percentage, raises an alert when the network_processor_idle_percentage is higher than usual. Note: Value range is 0 to 1.
REQUEST HANDLER POOL AVG IDLE PERCENT	Average fraction of time the request handler threads are idle across the cluster.	Two kinds of alert can be configured. • Alert-1: Value = 0, raises an alert when the request handler threads are busy. • Alert-2: Value > request_handler_idle_percentage, raises an alert when the request_handler_idle_percentage is higher than usual. Note: Value range is 0 to 1.
BROKER BYTES IN DEVIATION PERCENTAGE	Percentage by which a broker bytes in per second has deviated from the average bytes in per second of all the alive brokers.	Value > max_byte_in_deviation_percentage, raises an alert if a broker is seeing more than max_byte_in_deviation_percentage incoming traffic compared to average incoming traffic seen by all the brokers.
BROKER BYTES OUT DEVIATION PERCENTAGE	Percentage by which a broker bytes out per second has deviated from the average bytes out per second of all the alive brokers.	Value > max_byte_out_deviation_percentage, raises an alert if a broker is seeing more than max_byte_out_deviation_percentage outgoing traffic compared to average outgoing traffic seen by all the brokers.
ZOOKEEPER SESSION EXPIRATION PER SEC	Average rate at which brokers are experiencing zookeeper session expiration per second.	If this value is high, it can lead to controller fail over and leader changes. Raises an alert if value > 0.
Consumer		
CONSUMER GROUP LAG	How far consumer groups are behind the producers.	Depends on the application.
IS CONSUMER ACTIVE	Checks whether a consumer is active.	Value is False.
MILLISECONDS LAPSED SINCE CONSUMER WAS ACTIVE	Milliseconds passed since the consumer was last active.	Value > max_consumer_idle_time, raises an alert if the consumer did not consume for max_consumer_idle_time ms.
Broker		
BYTES IN PER SEC	Number of bytes per second produced to a broker.	Two kinds of alert can be configured. • Alert-1: Value = 0, raises an alert when the broker becomes idle. • Alert-2: Value > max_bytes_in_expected_per_broker, raises an alert when the broker input load is higher than usual.
ZOOKEEPER SESSION EXPIRATION PER SEC	Rate at which brokers are experiencing Zookeeper session expirations per second.	If this value is high, it can lead to controller fail over and leader changes. Raises an alert if value > 0.

Metric	Description	Suggested Alert
TOTAL PRODUCE REQUESTS PER SEC	Total number of produce requests to a broker per second.	Depends on the application. Two kinds of alert can be configured. • Alert-1: Value =0, raises an alert when there are no produce requests for the last 15 minutes. • Alert-2: Value > usual_num_producer_requests_expected to detect the spike in the number of requests.
PARTITION IMBALANCE PERCENTAGE	The partition imbalance for a broker. It is calculated as: (abs(average_no_of_partitions_per_broker - actual_no_of_partitions_per_broker) / average_no_of_partitions_per_broker) * 100	Value > 10 %
BYTES OUT PER SEC	Number of bytes per second fetched from a broker. It does not count the internal replication traffic.	Two kinds of alert can be configured. • Alert-1: Value = 0, raises an alert when the broker becomes idle. • Alert-2: Value > max_bytes_out_expected_per_broker, raises an alert when the broker output load is higher than usual.
IS BROKER DOWN	Checks whether a broker is down.	Value is True.
TOTAL PRODUCE REQUEST LATENCY	Latency of produce requests to this broker at the 99th percentile (in ms).	Value > max_expected_latency_ms.
ISR SHRINKS PER SEC	Rate at which brokers are experiencing InSync Replica Shrinks (number of shrinks per second).	Value > 0.
TOTAL FETCH CONSUMER REQUEST LATENCY	Latency of fetch consumer requests to this broker at 99th percentile (in ms).	Value > max_expected_latency_ms.
REQUEST HANDLER POOL AVG IDLE PERCENT	Average fraction of time the request handler threads are idle.	Two kinds of alert can be configured. • Alert-1: Value = 0, raises an alert when the request handler threads are busy. • Alert-2: Value > request_handler_idle_percentage, raises an alert when the request_handler_idle_percentage is higher than usual. Note: Value range is 0 to 1.
NETWORK PROCESSOR AVG IDLE PERCENT	Average fraction of time the network processor threads are idle.	Two kinds of alert can be configured. • Alert-1: Value = 0, raises an alert when the network processor threads are busy. • Alert-2: Value > network_processor_idle_percentage, raises an alert when the network_processor_idle_percentage is higher than usual. Note: Value range is 0 to 1.
Cluster Replication		
REPLICATION LATENCY	15 minutes average replication latency in milliseconds.	Value > max_expected_replication_latency, raises an alert if the replication latency is greater than max_expected_replication_latency.
REPLICATION THROUGHPUT	15 minutes average replication throughput in bytes per second.	Value < min_expected_throughput, raises an alert if throughput during replication is low. This could happen because of network issues.

Metric	Description	Suggested Alert
CHECKPOINT LATENCY	15 minutes average checkpoint latency in milliseconds.	Value > max_expected_checkpoint_latency, raises an alert if the checkpoint latency is greater than max_expected_replication_latency.
REPLICATION STATUS	Replication status of a replication pipeline.	Value != ACTIVE, raises an alert if the replication is not active.
Latency		
END TO END LATENCY	15 minutes average of end to end latency in ms.	Value > max_expected_latency, raises an alert if the end to end latency is greater than max_expected_latency.

Monitoring end-to-end latency

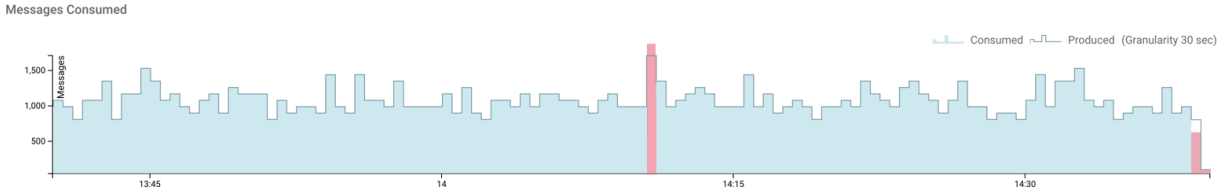
You can use Streams Messaging Manager UI to monitor end-to-end latency in Kafka topics. By monitoring latency, you get a complete understanding of the time taken by a consumer to consume a message that is produced in a topic. You can also monitor the number of consumed messages across all the consumer groups for a topic within the selected time range.

Use the latency feature to achieve the following goals:

- Verify whether end-to-end processing time SLAs are met.
- Identify slow or lagging consumers.
- Verify whether messages are overconsumed or underconsumed.

You can find details about the number of messages produced in a topic, the number of messages consumed from a topic, and latency details during the consumption of the messages in the following two graphs in the Streams Messaging Manager UI:

- Messages Consumed. The graph provides you the overall produced message count and consumed message count across all the consumer groups for a topic within the selected time range. Any discrepancy in the produced and consumed message count is highlighted in red.



In the preceeding image, the linear formation represents the number of messages produced in last one hour, and the filled area represents the number of messages consumed in last one hour with a granularity of 30 seconds. The blue area signifies that all the produced messages are consumed. The red area represents a discrepancy in the produced and consumed message count and can either mean that messages are overconsumed or underconsumed.

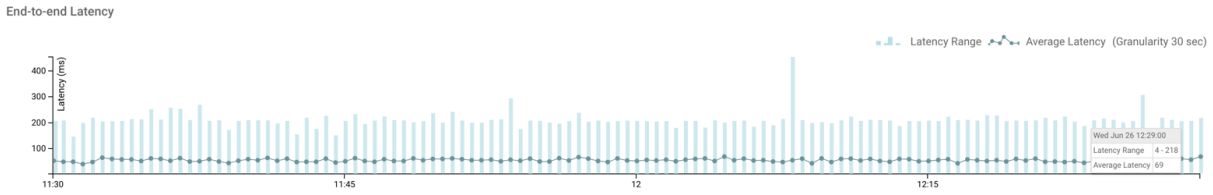
In the image, there are two red areas. The first red area, from the left, indicates that the number of consumed messages is more than the number of produced messages. This represents an overconsumption of messages which can occur when a consumer group offset is reset to an older offset to reprocess messages, or when producers or consumers are shut down in an unclean manner.

The last red area indicates that the number of consumed messages is less than the number of produced messages. This represents an underconsumption of messages which can occur when the consumer group offset is set to a newer offset causing the consumer group to skip processing some messages.

The far right section of the graph shows the current processing window where consumers are still consuming the produced messages. Therefore this area is expected to be marked red and indicates an undercsonumption of messages.

All other areas in the image are blue, which indicates that all the produced messages are consumed.

- End-to-end Latency. The End-to-end Latency graph provides you the details of the latency range and average latency in consuming the messages, which are produced in a particular topic, in milliseconds within the selected time range.



In the above image, the vertical lines represent the latency range, and the dotted line represents the average latency in consuming the produced messages in last one hour with a granularity of 30 seconds. You can see that at 12:29:00 hours on Wed Jun 26, the latency range was between 4 - 218 milliseconds and the average latency was 69 milliseconds.

Note: You can also create alerts to receive notifications based on the conditions that you configure in the alert policy for monitoring latency in your system. For more information about creating alerts to monitor latency, see the *Managing Alert Policies* guide.

Metric granularity

Streams Messaging Manager uses the REST API to display metrics. All the metrics are available for querying at two different granularities: 30 seconds and 15 minutes. The metrics are pre-aggregated for the defined buckets. Depending on how long the data is queried, granularity and varying dimensions of the topic, partition, consumer group ID and client ID, the data is calculated and rendered as JSON. Go through the following details before you start monitoring latency using Streams Messaging Manager:

- When you select a period newer than current time by 24 hours, the data is retrieved from REST server with metrics granularity of 30 seconds.
- When you select a period older than current time by 24 hours, the data is retrieved from REST server with metrics granularity of 15 minutes.
- Streams Messaging Manager UI polls the API for updates periodically (every 30 seconds if the selected period is newer than current time by 24 hours and every 15 minutes otherwise).
- By default, the 30 seconds granularity metrics are stored for 24 hours and the 15 minutes granularity metrics are stored for 2 weeks.

For information on REST APIs used in Streams Messaging Manager, see the *REST API Reference* guide.

Related Information

Managing Alert Policies and Notifiers

Streams Messaging Manager REST API Reference

Enabling interceptors

You need to enable interceptors for consumers, producers, and KafkaStreams applications to enable Streams Messaging Manager to fetch the metrics. If you do not enable the interceptors, you can not see any metrics in Streams Messaging Manager.

Interceptors publish the metrics to Kafka periodically. Metrics include counts on the producer side, and count average latency, and minimum and maximum latencies on the consumer side.

Steps to enable interceptors in your application

Add the following jar to the classpath of the application or as a dependency in the application:

```
<dependency>
```

```
 <groupId>com.hortonworks.smm</groupId>
 <artifactId>monitoring-interceptors</artifactId>
</dependency>
```

Steps to enable consumer interceptor

Perform the following steps to enable consumer interceptor:

1. Add the interceptor.classes property to consumer configuration that gets passed to the KafkaConsumer constructor.
2. Configure the client.id property as follows:

```
KafkaConsumer<Integer, String> createKafkaConsumer(String bootstrapServe
rs, String groupId, String clientIdentifier) {
    Properties properties = new Properties();
    properties.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapSe
rvers);
    properties.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, "org.ap
ache.kafka.common.serialization.IntegerDeserializer");
    properties.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, "org.a
pache.kafka.common.serialization.StringDeserializer");
    properties.put(ConsumerConfig.GROUP_ID_CONFIG, groupId);
    properties.put(ConsumerConfig.CLIENT_ID_CONFIG, clientIdentifier);
    //Add ConsumerInterceptor like this
    properties.put(ConsumerConfig.INTERCEPTOR_CLASSES_CONFIG,
       "com.hortonworks.smm.kafka.monitoring.interceptors.MonitoringCon
sumerInterceptor");
    return new KafkaConsumer<Integer, String>(properties);
}
```

Note: We recommend you to configure the client.id property. It helps in identifying the consumer instance. If you do not configure it, the latency metrics fetch the default consumer ID.

Steps to enable producer interceptor

Add the interceptor.classes property to producer configuration that gets passed to the KafkaProducer constructor, as follows:

```
KafkaProducer<Integer, String> createKafkaProducer(String bootstrapServers)
{
    Properties properties = new Properties();
    properties.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers
);
    properties.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, "org.apach
e.kafka.common.serialization.IntegerSerializer");
    properties.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, "org.apac
he.kafka.common.serialization.StringSerializer");
    //Add ProducerInterceptor like this
    properties.put(ProducerConfig.INTERCEPTOR_CLASSES_CONFIG,
       "com.hortonworks.smm.kafka.monitoring.interceptors.MonitoringProduce
rInterceptor");
    return new KafkaProducer<Integer, String>(properties);
}
```

Steps to enable interceptors in KafkaStreams applications

Add the producer.interceptor.classes and consumer.interceptor.classes properties to Kafka Streams configurations, as follows:

```
void startKafkaStreams(StreamsBuilder builder) {
```

```
    KafkaStreams kstreams = new KafkaStreams(builder.build(), getKafkaStrea
msConfiguration());
    kstreams.start();
}
Properties getKafkaStreamsConfiguration() {
    Properties config = new Properties();
    config.put(StreamsConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
    config.put(StreamsConfig.APPLICATION_ID_CONFIG, appId);
    config.put(StreamsConfig.CLIENT_ID_CONFIG, clientId);

    //Add producer interceptor like this
    config.put(
      StreamsConfig.PRODUCER_PREFIX + ProducerConfig.INTERCEPTOR_CLASSES_CO
NFIG,
      "com.hortonworks.smm.kafka.monitoring.interceptors.MonitoringProduce
rInterceptor");
    //Add consumer interceptor like this
    config.put(
      StreamsConfig.CONSUMER_PREFIX + ConsumerConfig.INTERCEPTOR_CLASSES_C
ONFIG,
      "com.hortonworks.smm.kafka.monitoring.interceptors.MonitoringConsum
erInterceptor");

    return config;
}
```

Note: MonitoringProducerInterceptor publishes the producer metrics to the "__smm_producer_metrics" topic and MonitoringConsumerInterceptor publishes the consumer metrics to the "__smm_consumer_metrics" topic from their respective client applications. So if Authorization is enabled on kafka cluster, ensure that your client applications have access to the aforementioned topics.

Monitoring end to end latency for Kafka topic

To monitor end-to-end latency of Kafka topics, you can monitor the count of consumed messages across all Kafka consumer groups and the time taken by all Kafka consumer groups to consume messages that are produced in a Kafka topic, in a graphical way. You can also monitor the same for each Kafka consumer group, each client in a Kafka consumer group, and each partition in a Kafka topic.

About this task

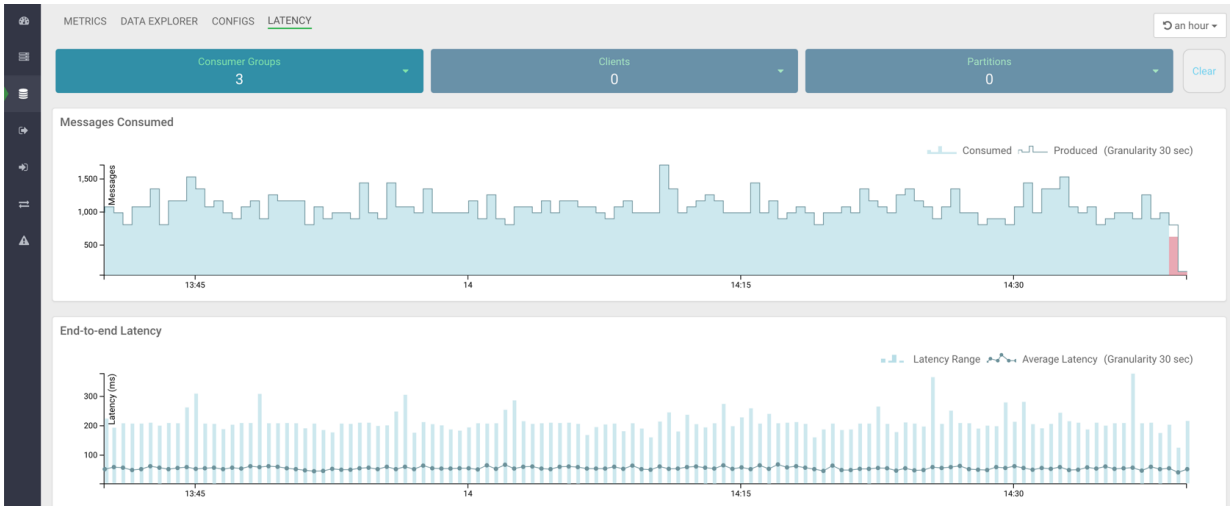
Perform the following steps to monitor end-to-end latency in the Streams Messaging Manager UI:

Procedure

1. Go to Topics in the Streams Messaging Manager UI.
2. Select the topic you want to verify the details about.
3. Click the Profile icon  beside the topic you select.

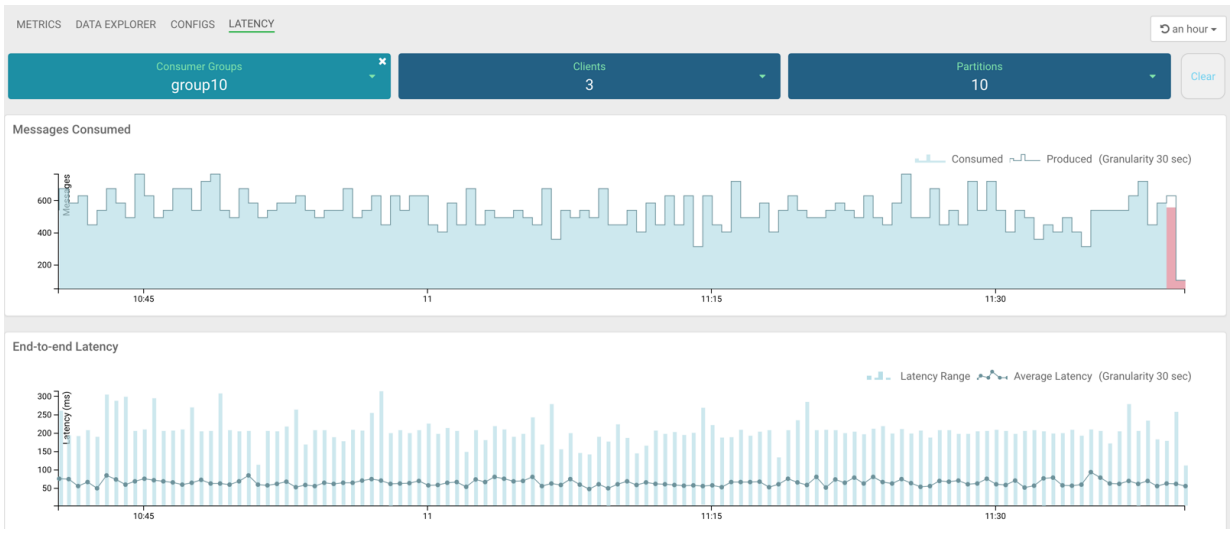
   This takes you to the Metrics page where you can find the Messages Consumed and End-to-end Latency graphs along with other topic details. On the Metrics page, these two graphs provide you an aggregated result of latency and count of consumed messages across all consumer groups.
4. To verify the details by individual consumer groups, clients, and partitions, go to the Latency tab.

   The **Latency** page appears as shown in the following image:



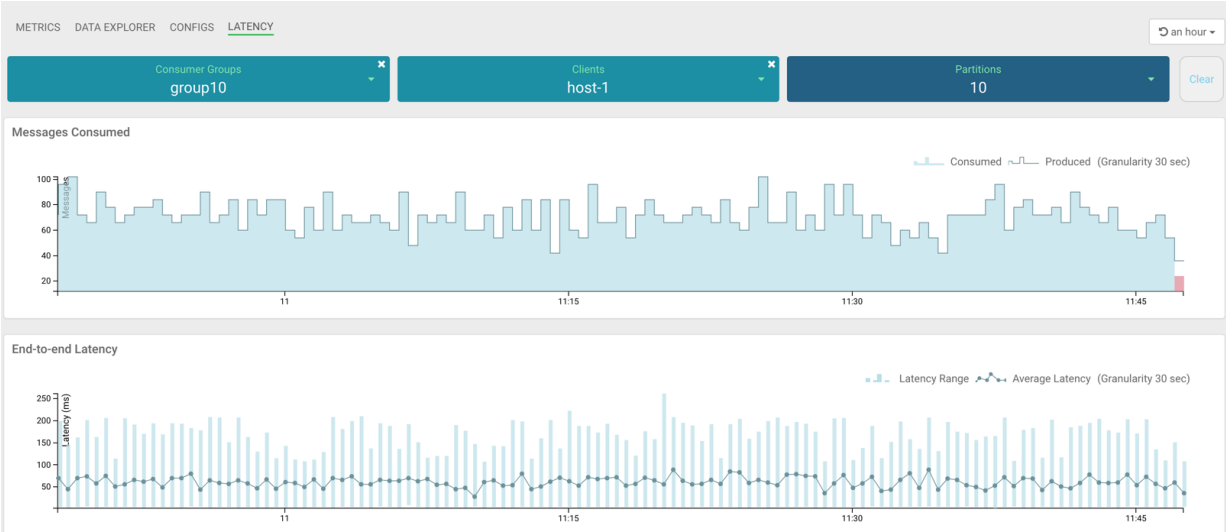
The Latency view gives you a powerful snapshot of the end-to-end latency scenario: number of consumer groups for a topic, number of clients inside a particular consumer group, and number of partitions in a topic along with the Messages Consumed and End-to-end Latency graphs.

5. Select any consumer group from the Consumer Groups drop-down, as shown in the following image:



In the image, group10 consumer group is selected. The Latency tab displays that there are 3 clients in the group10 consumer group, and 10 partitions are available in the topic.

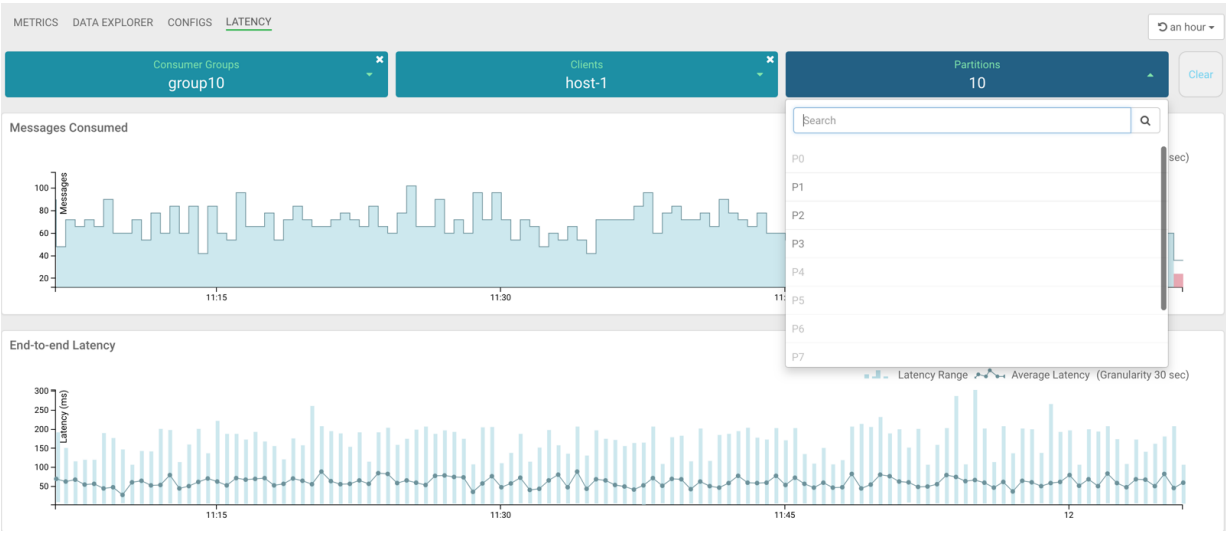
6. Select any client from the Clients drop-down, as shown in the following image:



In the image, host-1 client is selected. At this instance, the Messages Consumed and End-to-end Latency graphs display data for only the host-1 client. Here you can monitor the number of messages produced, number of messages consumed, latency range, and average latency for host-1 only. Hover your mouse over the graphs and get data at any point of time in the selected time range. You can see in the Messages Consumed graph that host-1 consumed all the messages that are produced and actively consuming data at the latest time. You can see in the End-to-end Latency graph that the latency range and average latency are under 250 milliseconds.

7. To get details about the partitions from where host-1 is consuming data, click Partitions.

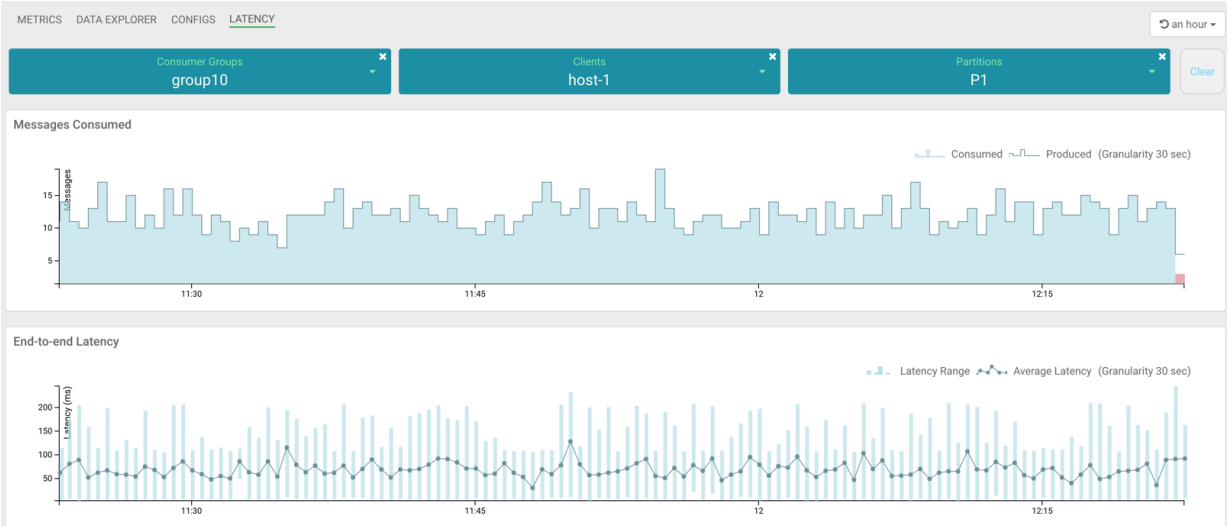
   The list of partitions in the topic appears, as shown in the following image:



In the image, you can see that host-1 is consuming data from 3 partitions: P1, P2, and P3. Other partitions are inactive for host-1.

8. Select any active partition from the list.

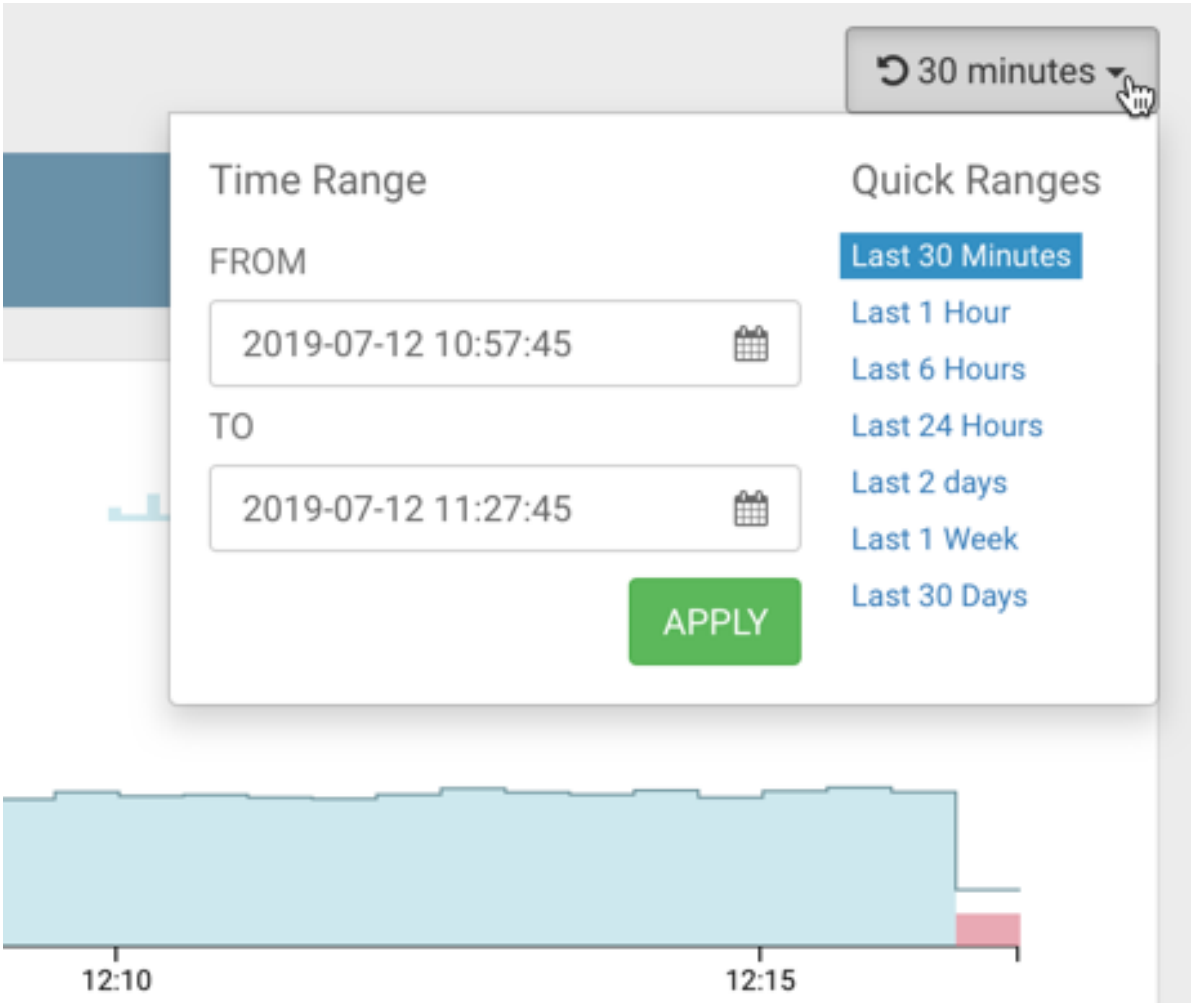
   The Latency tab displays the transaction details between host-1 and the selected partition (for example, P1), as shown in the following image:



Now you have got details for host-1 client. Likewise, you can fetch details for other clients.

9. Follow steps 6 through 8 to fetch data for all other clients.
10. Follow steps 5 through 8 to fetch data for all other consumer groups.

- To clear all the selections at once, click the Clear button at the top-right corner of the page.
- To clear selection for Consumer Groups, Clients, or Partitions, click the delete icon located on each drop-down.
- To select a different time range, use the Time Range and Quick Ranges options at the top-right corner of the page, as shown in the following image:



End to end latency use case

You can use end-to-end latency feature available in Streams Messaging Manager to handle real-time scenarios including measurement of end-to-end processing time, identification of slow or lagging consumers, and verification of over-consumption or under-consumption of messages.

Verify whether end-to-end processing time SLAs are met

A service-level agreement (SLA) is a commitment between a service provider and a service user. Particular aspects of the service are agreed between the service provider and the service user. The most common component of SLA is that the services should be provided to the user as agreed upon in the contract. For example, you agreed upon an average latency value and a maximum latency value for message consumption with Cloudera. Therefore, after a producer produces messages, if the messages take the agreed amount of time to be consumed by consumers, the SLA would be met.

1. Go to Topics in the Streams Messaging Manager UI.
2. Select the topic you want to verify the details about.
3. Click the Profile icon beside the topic you select.
4. Check the latency graph and see if the average and maximum latencies are as expected.
5. If the latencies are not as expected, go to the Latency tab.
6. Check whether the number of clients is as expected. If not, then you might want to check the missing client instance.

7. If the number of clients are as expected, then check if there is any spike in the message count. Select a period of 1 week in the Time Range pane and see if there is a surge in incoming messages that could explain the time SLA violation.
8. If the time SLA appears to be breached even after all the checks turned positive, go to Use Case 2.

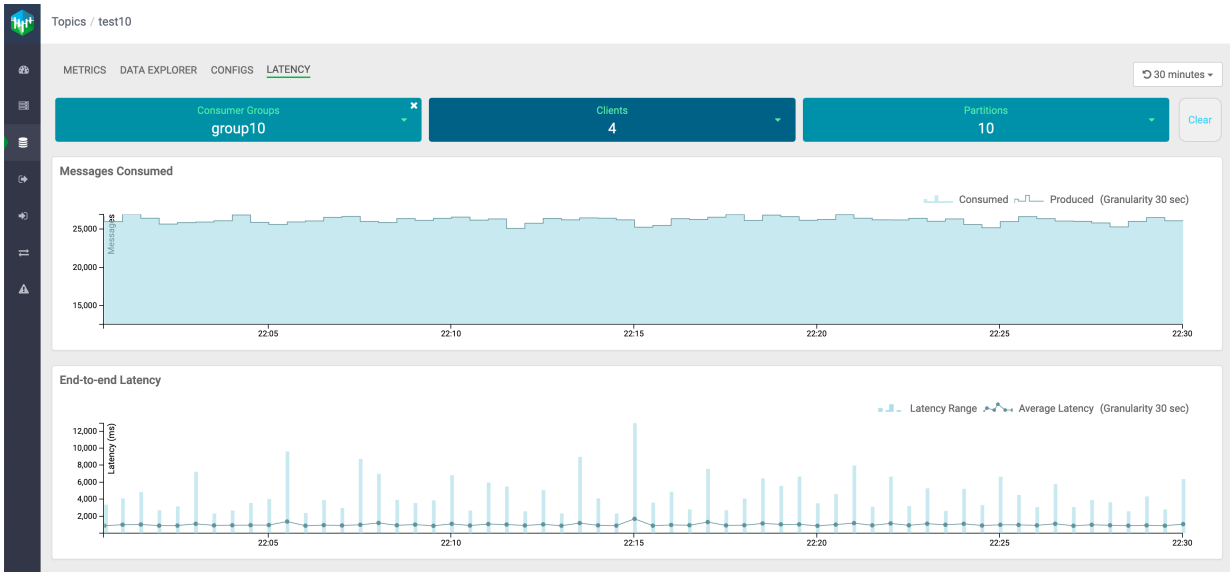
Identify slow or lagging consumers

A stream based application flow involves application polling for messages, fetching and processing the messages, performing an optional blocking operation like interacting with database or local file system and then application polling for the messages again. However, the delay due to message processing, system bottleneck, or external bottleneck might become very long in some scenarios and you might want to understand which of the process instances is facing issues.

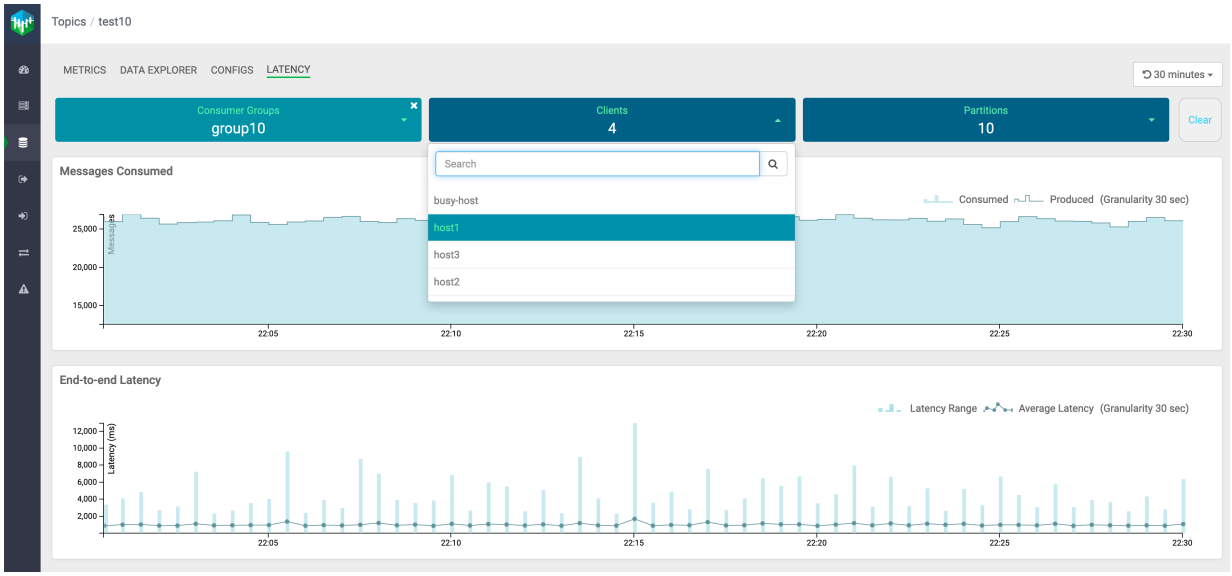
1. Go to Topics in the Streams Messaging Manager UI.
2. Select the topic you want to verify the details about.
3. Click the Profile icon beside the topic you select.
4. Go to the Latency tab.
5. After you select a group, inspect the latency and message counts for each client.

   This could lead you to the slow consumer.

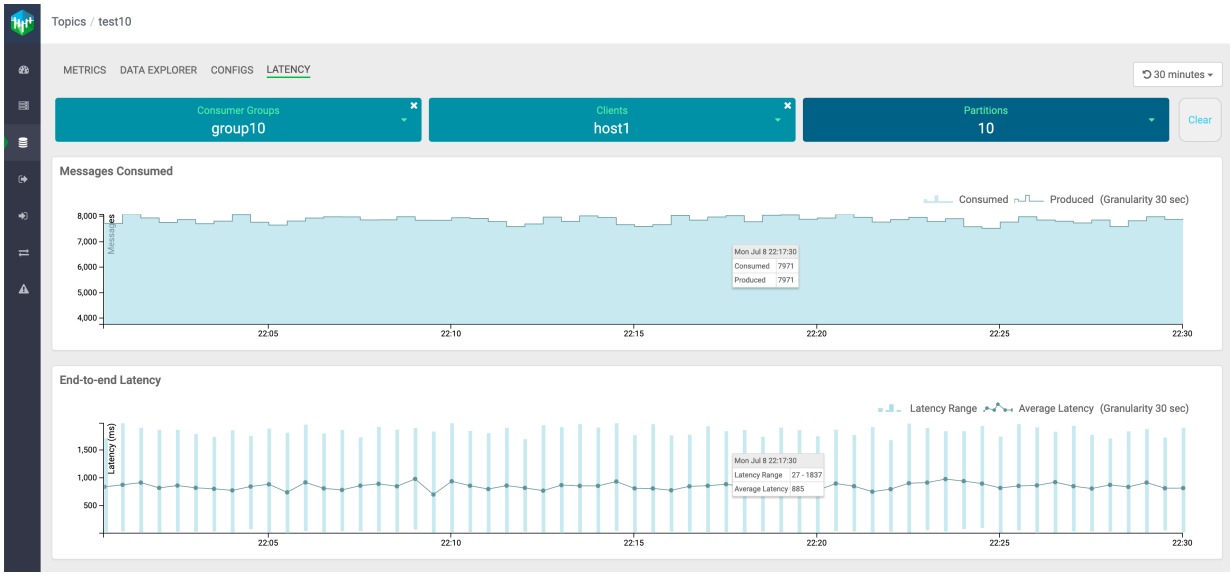
   Let us walk through an example.



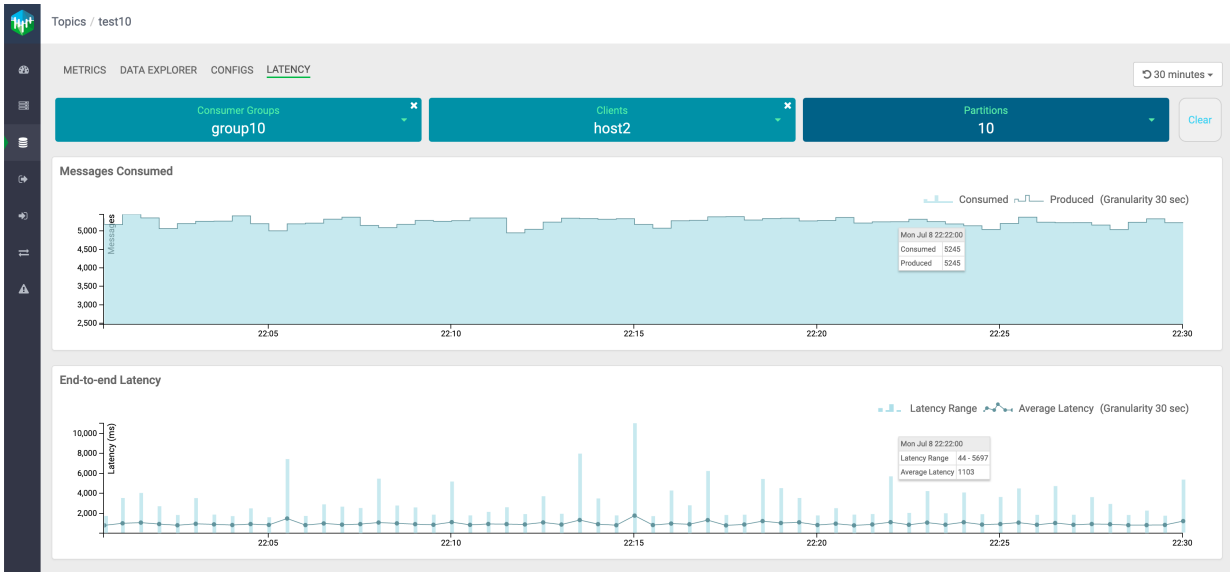
   In the above image, you select group10 consumer group to inspect the latency and message counts for each client.



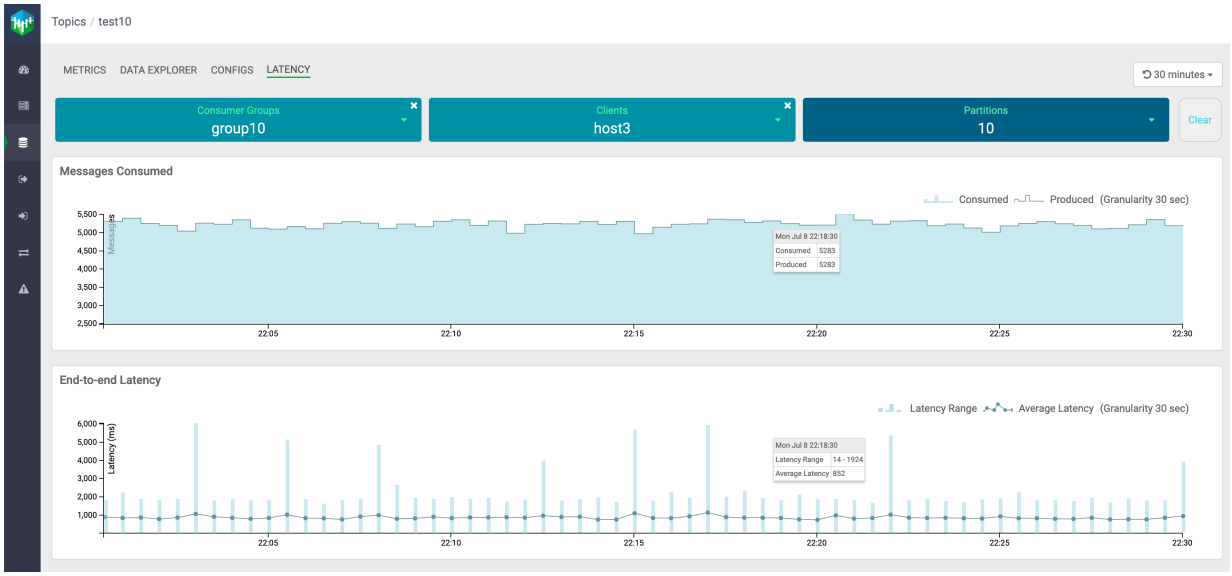
In the above image, you see the list of active clients for group10: host1, host2, host3, and busy-host. Now you need to select each client and inspect the latency and message counts.



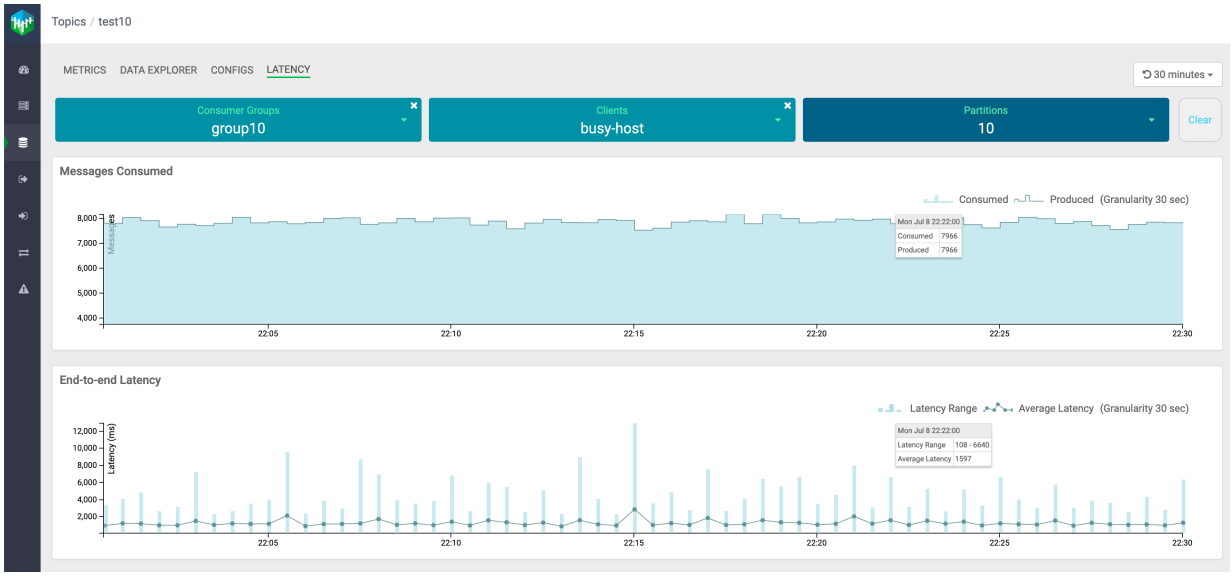
In the above image, you can see that host1 consumed all the messages produced, and the average latency and latency range are under a good range.



In the above image, you can see that host2 consumed all the messages produced. Also, there are few spikes in latency range, but the average latency is under a good range.



In the above image, you can see that host3 consumed all the messages produced. Also, there are occasional spikes in latency range, but the average latency is under a good range.



In the above image, you can see that busy-host consumed all the messages produced. There are frequent spikes in latency range, and both latency range and average latency are high. Hence, busy-host is the slow consuming client.

It is possible that all the clients are experiencing longer latencies. This could imply that there is a common bottleneck. For example, clients are interacting with external store over network and having delays in consuming the messages due to network issues.

If there is a single client that is experiencing slowness, you must check the message counts for other clients, and system parameters like CPU and memory.

This addresses your need to identify the slow consuming application.

Verify whether messages are overconsumed or underconsumed

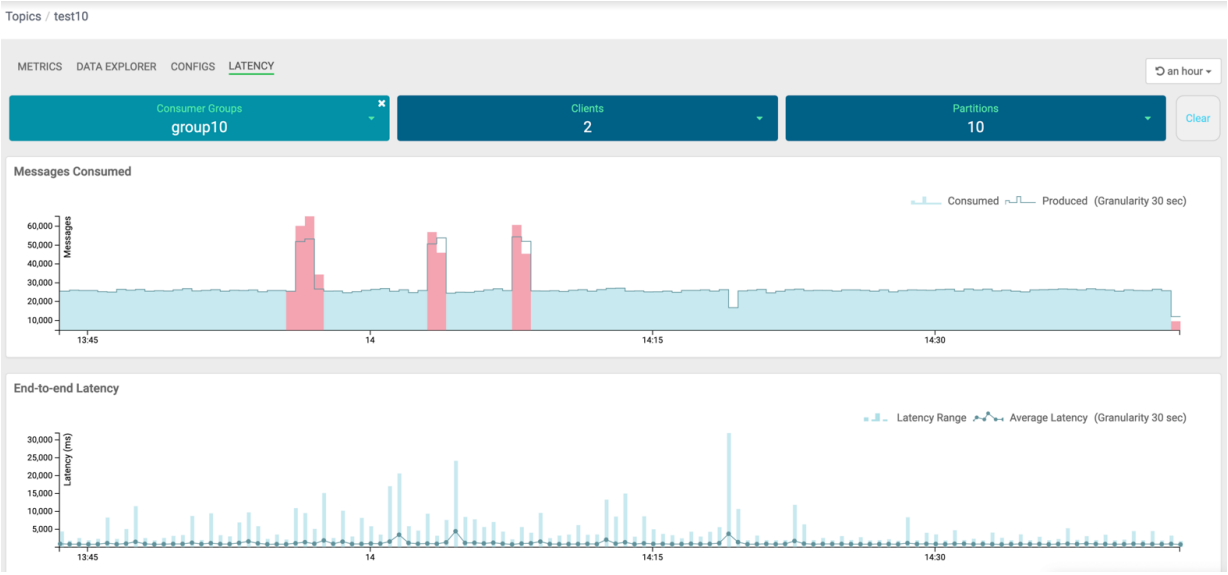
There could be an overconsumption of messages. This might occur due to the following reasons:

- If the producers and consumers are shut down in an unclean manner or the producers and consumers went down in an unexpected manner. For example, Kafka producer produced some messages but it shut down before the producer received any acknowledgement from the broker. Similarly, Kafka consumer consumed a few messages but got shut down before it could commit the offset at this latest point.
- If the consumer is reset to an older offset (reprocessing scenarios).

If the consumer is reset to a new offset (real-time application requirement), there could be an underconsumption of messages.There could be over or under consumption of messages if the cluster is in an unhealthy state.

1. Go to Topics in the Streams Messaging Manager UI.
2. Select the topic you want to verify the details about.
3. Click the Profile icon beside the topic you select.
4. Go to the Latency tab.
5. After you select a group, inspect the produced message and consumed message counts for each client in Messages Consumed graph.

   This helps you to verify whether the consumers are consuming all the messages that are produced in a topic. You might also identify occurrences of any overconsumption or underconsumption of messages.



In the image, you can see that for group10 consumer group, there are three red spikes in the Messages Consumed graph.

The first spike, from the left, indicates that the number of consumed messages is more than the number of produced messages. So, this is an overconsumption of messages.

The second and the third spike indicate overconsumption of messages followed by an underconsumption of messages.

Monitoring Kafka cluster replications (Streams Replication Manager)

Get started with monitoring Streams Replication Manager using Streams Messaging Manager.

Streams Replication Manager is an enterprise-grade replication solution that enables fault tolerant, scalable and robust cross-cluster Kafka topic replication. Streams Replication Manager provides the ability to dynamically change configurations and keeps the topic properties in sync across clusters at high performance. Streams Replication Manager also delivers custom extensions that facilitate installation, management and monitoring making Streams Replication Manager a complete replication solution that is built for mission critical workloads.

Streams Messaging Manager is capable of integrating with Streams Replication Manager enabling you to monitor your Streams Replication Manager replications on the Streams Messaging Manager UI. You can monitor status of the Kafka cluster replications, number of topics associated with the replication, throughput, replication latency, and checkpoint latency. Additionally, the alerting features of Streams Messaging Manager are also supported for replications. This allows you to create receive alerts in connection to your replications.

The following tasks and concepts walk you through the **Cluster Replications** section of the Streams Messaging Manager UI, which is your main hub in Cloudera where you view and monitor details regarding your Streams Replication Manager replications.

Note: The **Cluster Replications** page is only available on the Streams Messaging Manager UI if integration between Streams Messaging Manager and Streams Replication Manager is enabled. Integration is enabled when you add the Streams Replication Manager service to your cluster. However, integration can be enabled separately, after service installation as well.

Related Information

Managing Alert Policies and Notifiers

Integrating Streams Replication Manager with Streams Messaging Manager

Viewing Kafka cluster replication details

You can view and monitor your Streams Replication Manager replications in the Cluster Replications section of the Streams Messaging Manager UI. Learn more about what information you can view on the UI regarding replications.

About this task

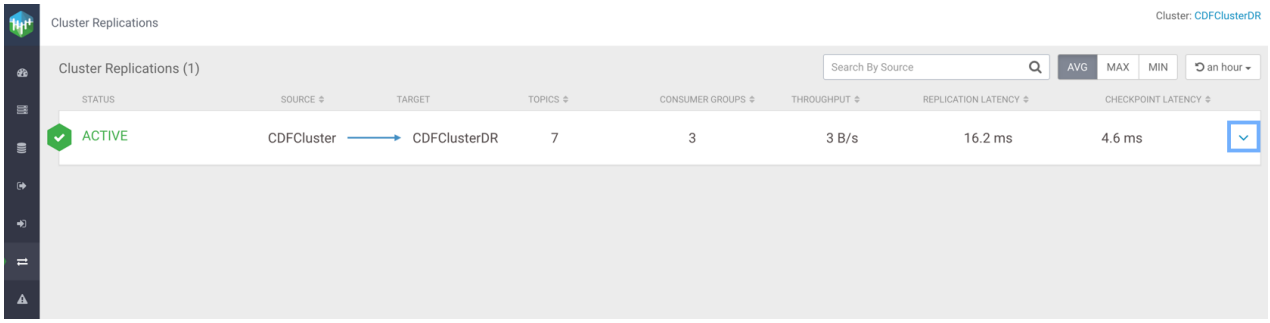
Note: Which exact replications are available for monitoring on the UI is determined by the configuration of the Streams Replication Manager Service roles that Streams Messaging Manager integrates with. You will only be able to view the replications that target the cluster(s) that the Streams Replication Manager Service roles target. Replications targeting other clusters in your deployment will not be visible. If you want to view all replications in your deployment using a single Streams Messaging Manager service, you must either enable the Remote Querying feature in Streams Replication Manager (recommended), or reconfigure the Streams Replication Manager Service roles that integrate with Streams Messaging Manager to target all clusters in your deployment.

You can view the status of the replications, source cluster names, target cluster names, number of topics to be replicated, number of consumer groups, throughput of the replication, replication latency, and checkpoint latency. Streams Messaging Manager also displays two graphs, one represents the throughput of the replication and the other displays the replication latency of the replication along with the details of each topic to be replicated.

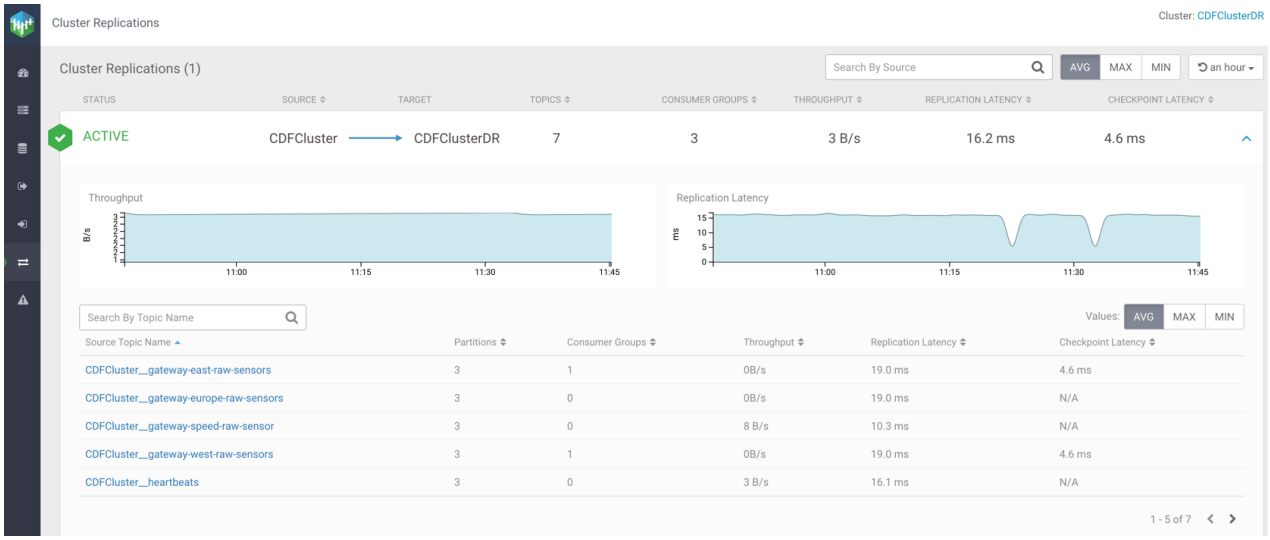
Perform the following steps to view the details of a cluster replication:

Procedure

In the Cluster Replications page, click a cluster replication or the drop-down icon beside a replication, as shown in the following image:



The replication details appear as shown in the following image:



Related Information

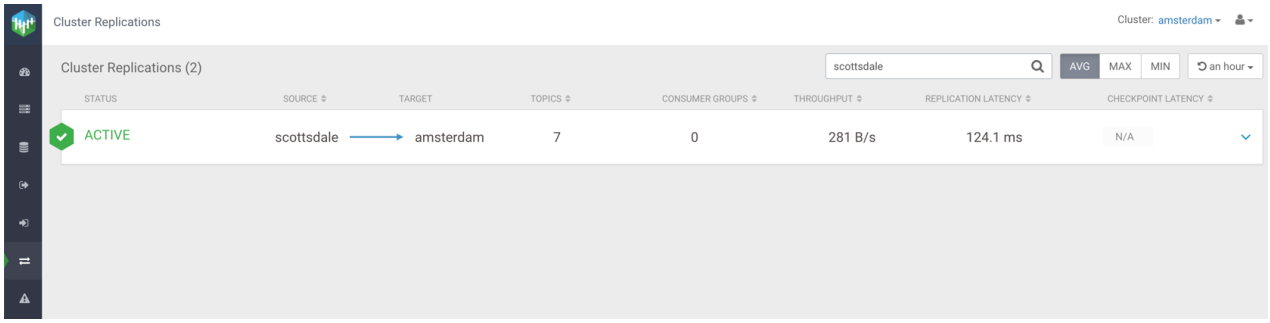
Streams Replication Manager Service
Remote Querying

Searching Kafka cluster replications by source

Learn how to filter your Kafka cluster replications by source cluster after you select the Kafka cluster which acts as the target cluster.

About this task

You can use the Search By Source bar at the top right of the page to search cluster replications by the source cluster name. For example, if the destination cluster is amsterdam, and you enter scottsdale in the Search By Source bar, Streams Messaging Manager displays the Kafka cluster replication between scottsdale and amsterdam, as shown in the following image:



Monitoring Kafka cluster replications by quick ranges

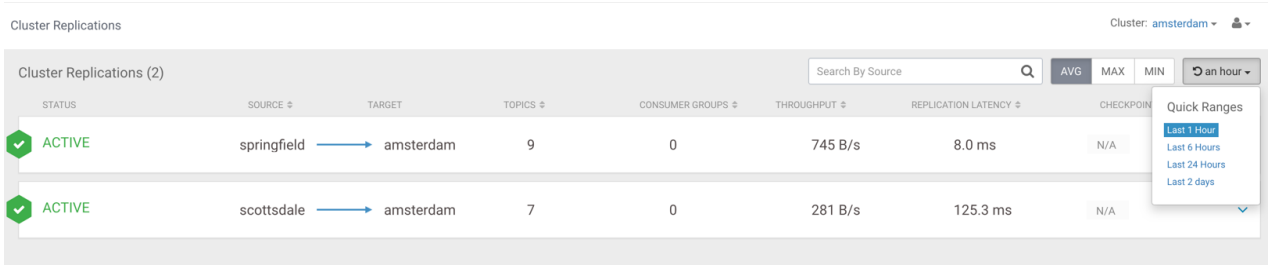
Learn how to filter your cluster replications by specific time periods after you select the Kafka cluster which acts as the target cluster.

About this task

You can filter the cluster replications by time range. You can choose any of the following filter values from the drop-down to display the cluster replication details in the selected time range:

- Last 1 Hour
- Last 6 Hours
- Last 24 Hours
- Last 2 Days

The following image shows the Quick Ranges drop-down:



Monitoring status of the clusters to be replicated

You can monitor the status of Kafka cluster replications in the Status column in the Cluster Replications page. You can see whether a cluster replication is active, inactive, or showing a warning.

The status has three variations:

- Active

  Indicates that the cluster replication is running.
- Inactive

  Indicates that the cluster replication is not running.
- Warning

  Indicates that the cluster replication is facing issues.

If the status of a Kafka cluster replication shows inactive or warning, check the logs, and troubleshoot the replication.

Monitoring topics to be replicated

You can monitor the number of topics associated with a Kafka cluster replication in the Topics column in the Clus ter Replications page. You can also monitor details on target topic name, partitions, consumer groups, throughput, replication latency, and checkpoint latency.

Click on the cluster replication to fetch topic details. Streams Messaging Manager displays the following details about the topics:

- Target Topic Name

  Name of the topic at the source.
- Partitions

  Number of partitions of the topic at the source.
- Consumer Groups

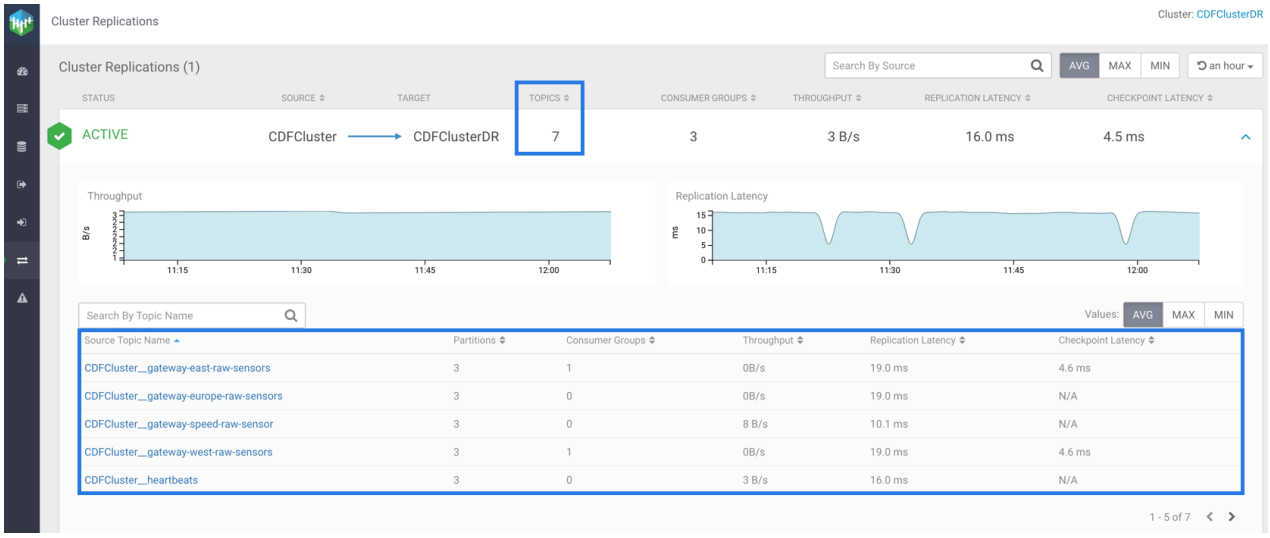
  Number of consumer groups consuming data from the topic.
- Throughput

  Data replicated between the source cluster and the target cluster per second from a topic. Throughput is measured in bytes per second. By default, Streams Messaging Manager displays the average throughput. You can also fetch the maximum or minimum throughputs for a topic by clicking the MAX or MIN button located above the topic details.
- Replication Latency

  Amount of time taken for a message of a topic to get replicated from the source cluster to the target cluster. Replication latency is measured in milliseconds. By default, Streams Messaging Manager displays the average replication latency. You can also fetch the maximum or minimum replication latency for a topic by clicking the MAX or MIN button located above the topic details.
- Checkpoint Latency

  Amount of time taken for a message of a topic to checkpoint on the target cluster after the message is committed on the source cluster. Checkpoint latency is measured in milliseconds. By default, Streams Messaging Manager displays the average checkpoint latency. You can also fetch the maximum or minimum checkpoint latency for a topic by clicking the MAX or MIN button located above the topic details.

The following diagram shows details of topics in a cluster replication:

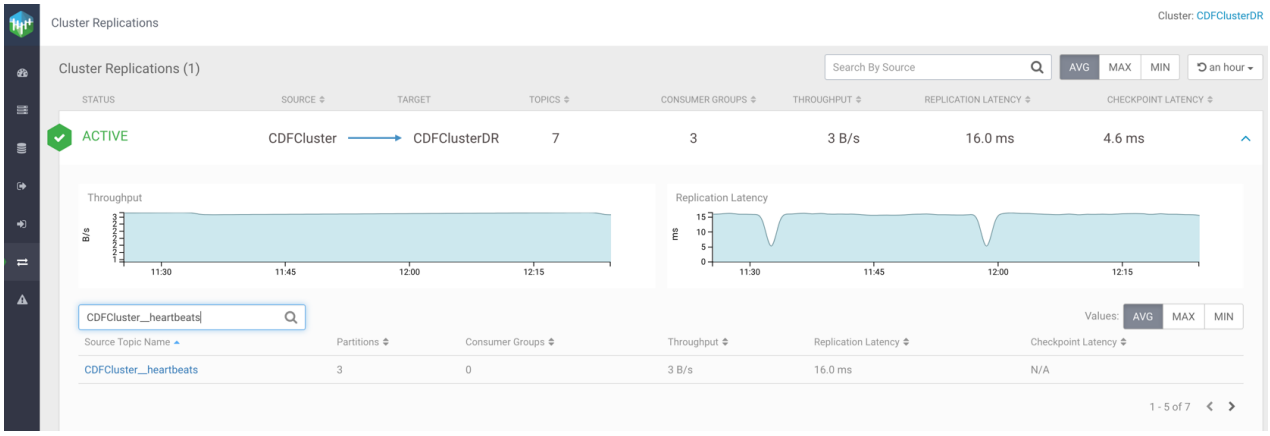


In the image, you can see that the number of topics to be replicated from CDFCluster to CDFClusterDR is 7, and topic details including topic names, number of partitions for the topics at the source cluster, number of consumer groups consuming messages from each topic, throughput, replication latency, and checkpoint latency of each topic.

Searching by topic name

Learn how to search a Kafka topic by name in a Kafka cluster replication and fetch details for that topic. After you find a topic, you can see the topic name, partitions, consumer groups, throughput, replication latency, and checkpoint latency for that topic.

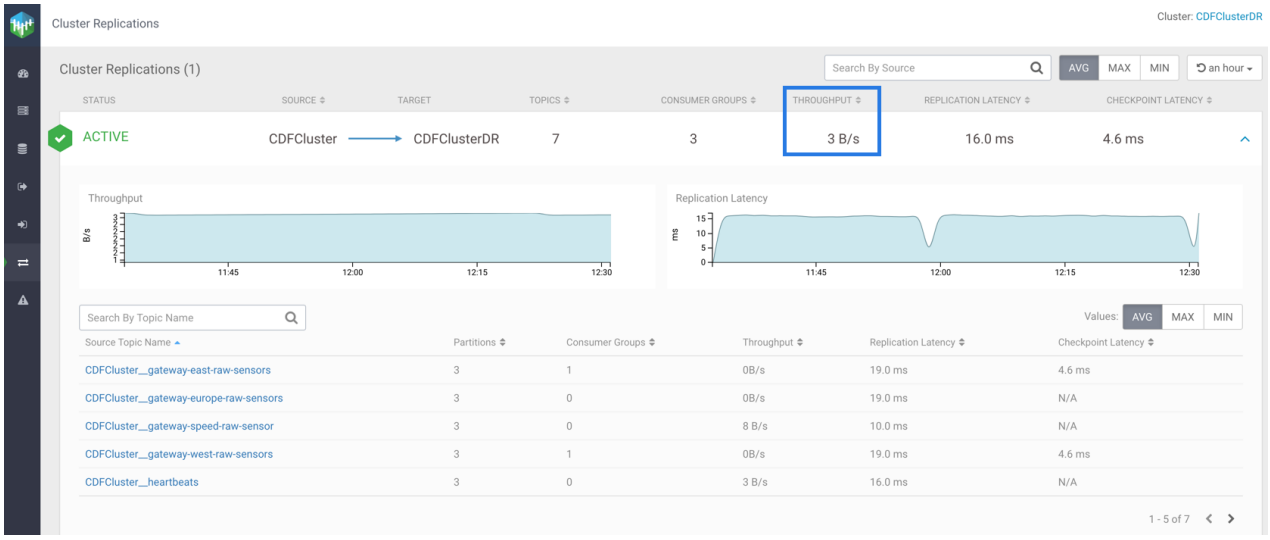
You can use the Search By Topic Name bar to search a Kafka topic by name and get details of that topic. The following image shows details of the CDFCluster__heartbeats topic:



Monitoring throughput for cluster replication

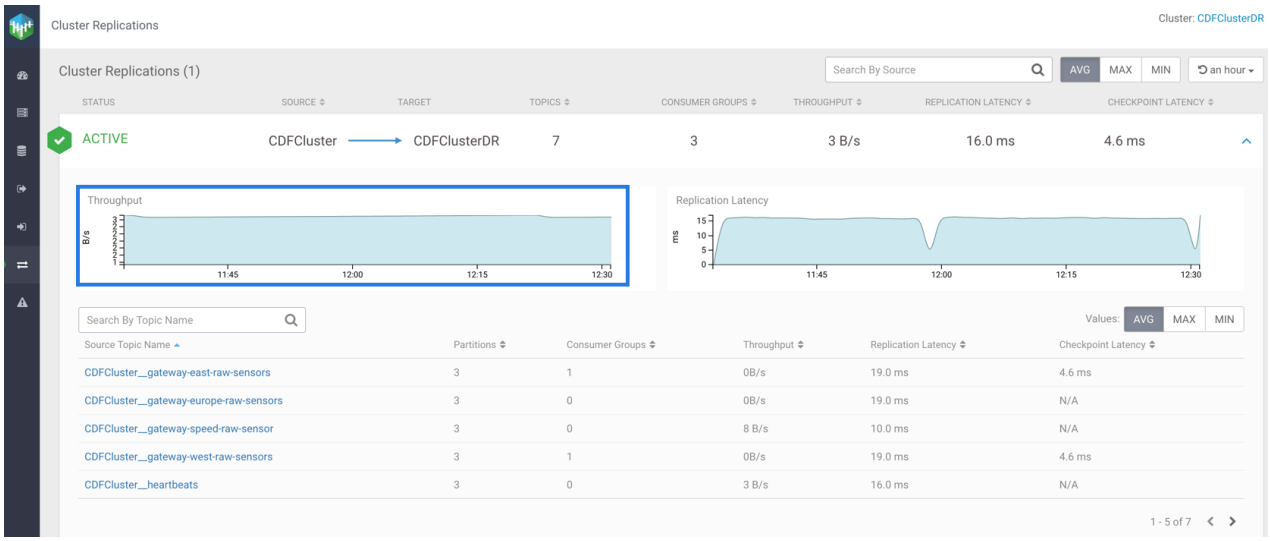
Learn how to monitor the throughput of a Kafka cluster replication in Streams Messaging Manager. You can monitor average, maximum, and minimum throughput of a cluster replication. You can also monitor the throughput of a cluster replication graphically.

Throughput is defined as the data replicated between the source cluster and the destination cluster per second. Throughput is measured in bytes per second.



In the image, you can see the average throughput for the CDFCluster to CDFClusterDR replication is 3 bytes per second. You can fetch the maximum or minimum throughput for the replication by clicking the MAX or MIN button located above the cluster replication.

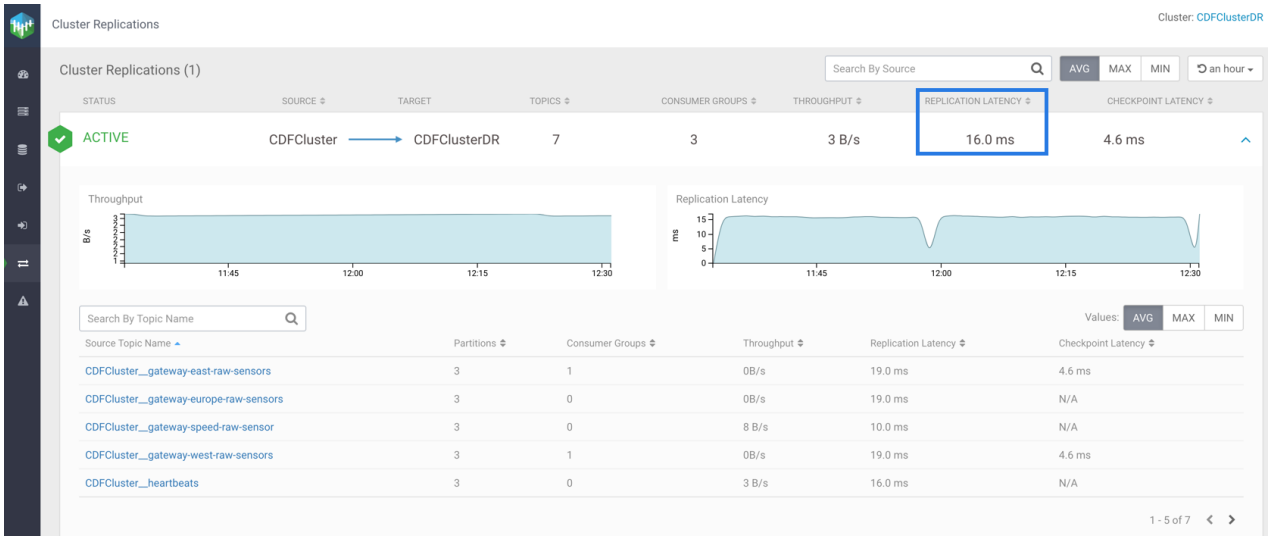
You can also monitor the throughput of a cluster replication graphically. Streams Messaging Manager displays the Throughput graph for each cluster in the cluster details. The following image shows the graph for throughput for CDFCluster to CDFClusterDR replication:



Monitoring replication latency for cluster replication

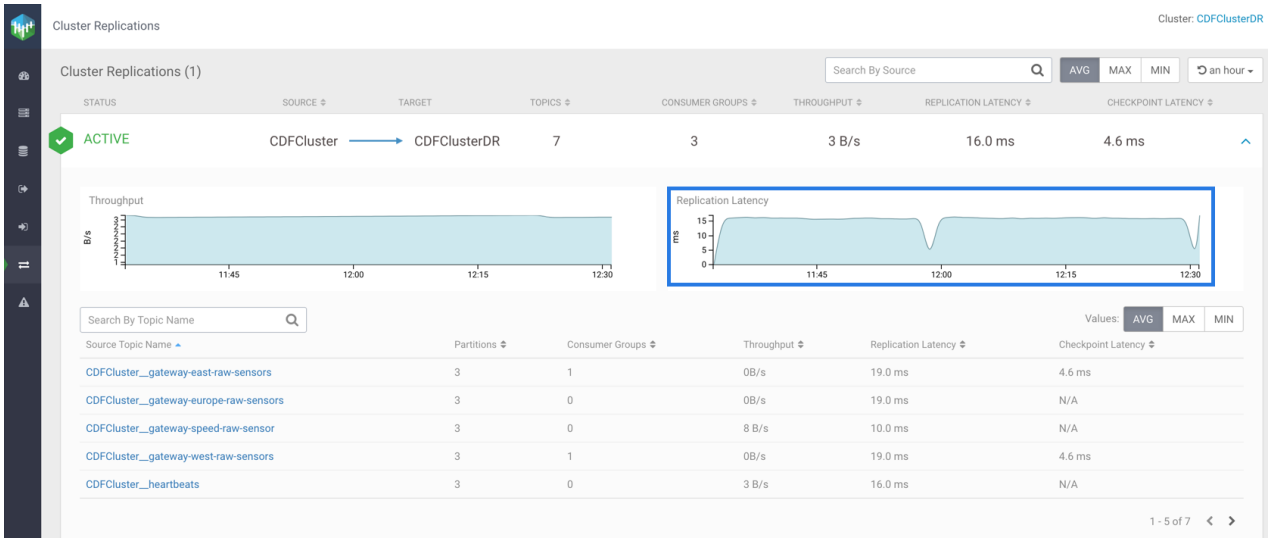
Learn how to monitor the replication latency of a Kafka cluster replication in Streams Messaging Manager. You can monitor average, maximum, and minimum replication latency of a cluster replication. You can also monitor the replication latency of a cluster replication graphically.

Replication latency is defined as the amount of time taken for a message to get replicated from the source cluster to the destination cluster. Replication latency is measured in milliseconds.



In the image, you can see the average replication latency for the CDFCluster to CDFClusterDR replication is 16.0 milliseconds. You can fetch the maximum or minimum replication latency for the replication by clicking the MAX or MIN button located above the cluster replication.

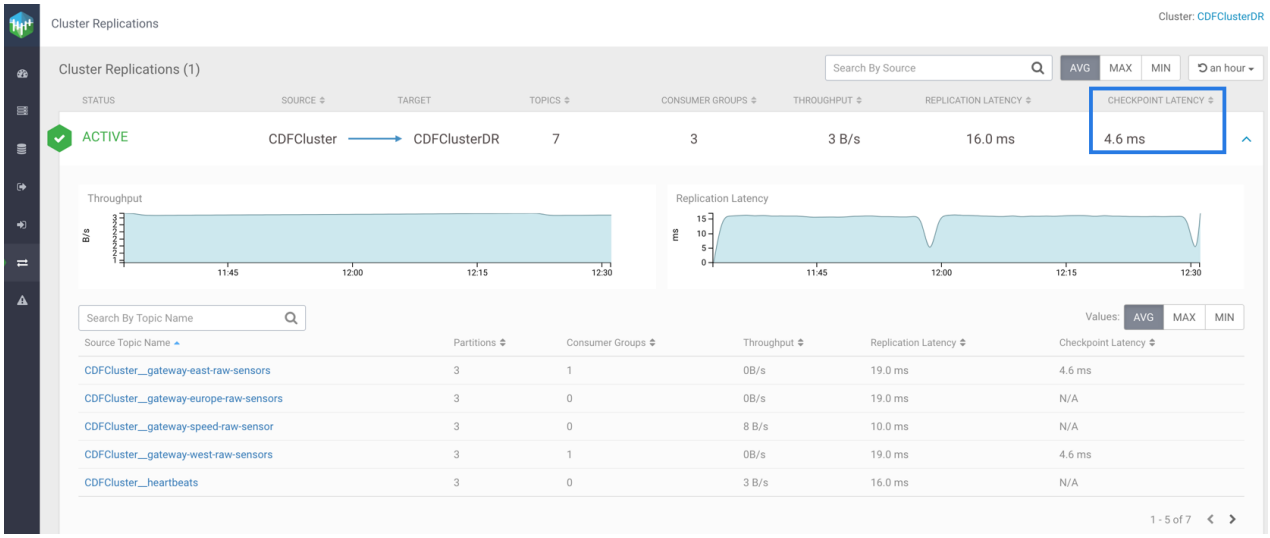
You can monitor the replication latency of a cluster replication graphically. Streams Messaging Manager displays the Replication Latency graphs for each cluster in the cluster details. The following image shows the graph for replication latency for CDFCluster to CDFClusterDR replication:



Monitoring checkpoint latency for cluster replication

Learn how to monitor the checkpoint latency of a Kafka cluster replication in Streams Messaging Manager. You can monitor average, maximum, and minimum checkpoint latency of a cluster replication. You can also monitor the checkpoint latency of a cluster replication graphically.

Checkpoint latency is defined as the amount of time taken for a message of a topic to checkpoint on the target cluster after the message is committed on the source cluster. Checkpoint latency is measured in milliseconds.



In the image, you can see the average checkpoint latency for the CDFCluster to CDFClusterDR replication is 4.6 milliseconds. You can fetch the maximum or minimum replication latency for the replication by clicking the MAX or MIN button located above the cluster replication.

Note: If the checkpoint latency for a cluster replication or a topic appears as Not Available, then that means there are no consumer groups defined.

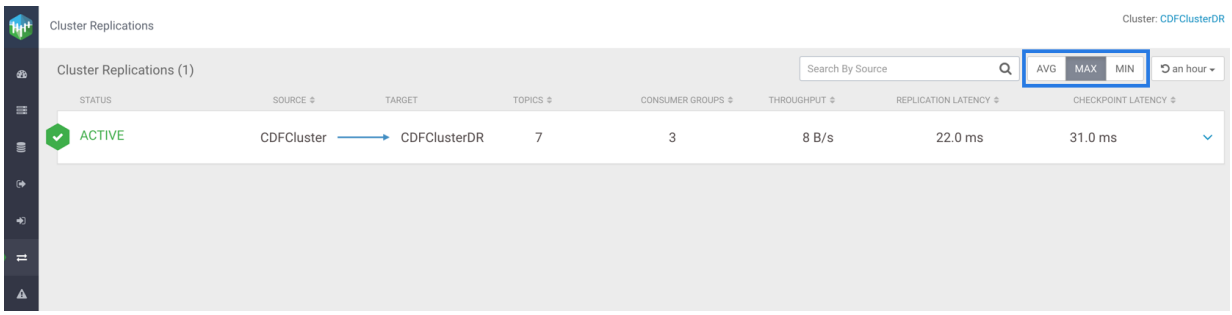
Monitoring replication throughput and latency by values

Learn how to fetch the average, maximum, and minimum values for throughput, replication latency, and checkpoint latency of a Kafka cluster replication in Streams Messaging Manager.

You can do this at the following levels:

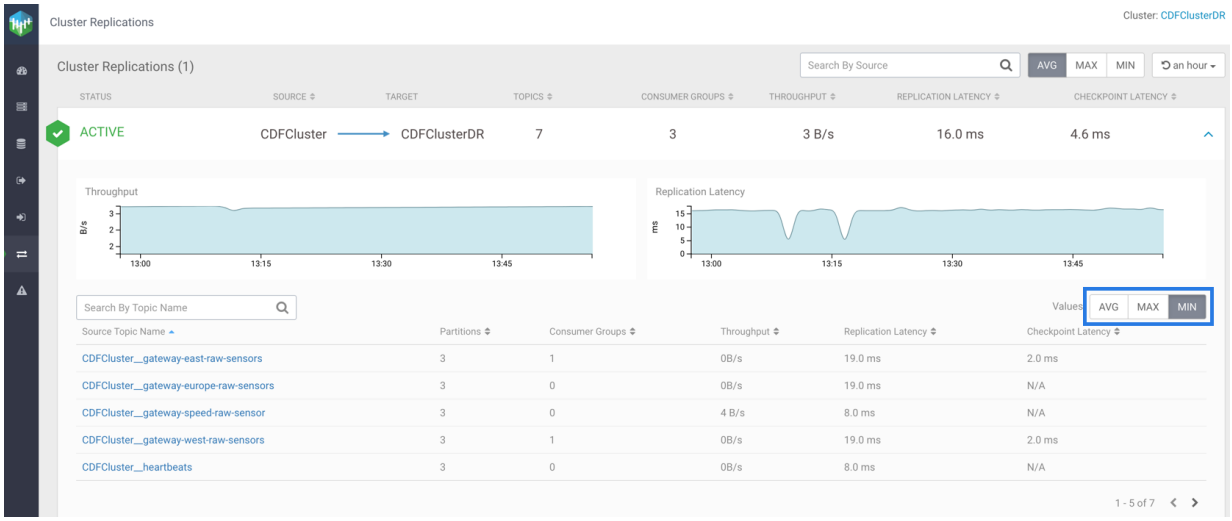
- Cluster Replication Level

  Click the AVG, MAX, or MIN buttons, as shown in the following image, to fetch average, maximum, or minimum values of throughput, replication latency, and checkpoint latency for cluster replications.



- Topic Level

  Click the AVG, MAX, or MIN buttons, as shown in the following image, to fetch average, maximum, or minimum values of throughput, replication latency, and checkpoint latency for topics.



Managing and monitoring Kafka Connect using Streams Messaging Manager

Get started with Kafka Connect in Streams Messaging Manager.

Kafka Connect is a tool for streaming data between Apache Kafka and other systems in a reliable and scalable fashion. Kafka Connect makes it simple to quickly define connectors that move large collections of data into and out of Kafka. Source connectors can ingest entire databases or collect metrics from all your application servers into Kafka topics, making the data available for stream processing with low latency. Sink connectors can deliver data from Kafka topics into secondary storage and query systems or into batch systems for offline analysis.

Kafka Connect in Cloudera is shipped with many differentCloudera-developed as well as publicly available sink and source connectors. Each of which cover a specific use case for streaming data. In addition to the connectors available by default, installing custom developed or third-party connectors is also possible. All connectors can be deployed, managed, and monitored using the Streams Messaging Manager UI (recommended), Streams Messaging Manager REST API, or Kafka Connect REST API.

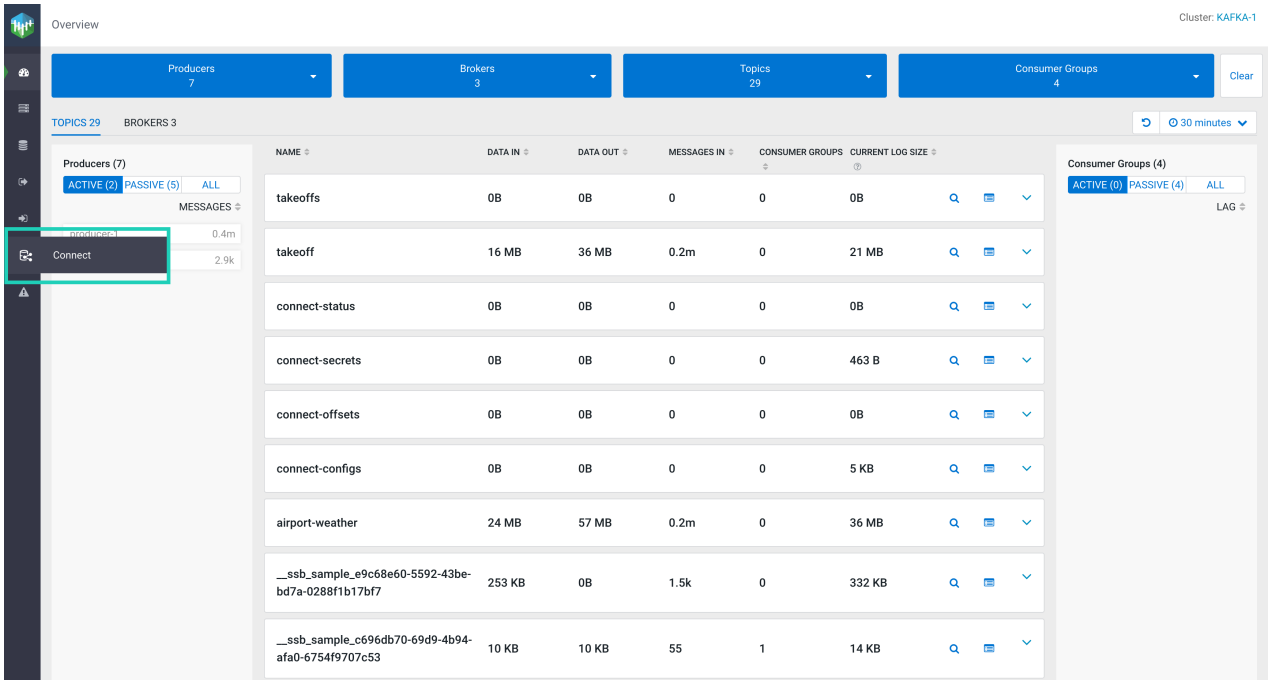
Related Information

Connectors

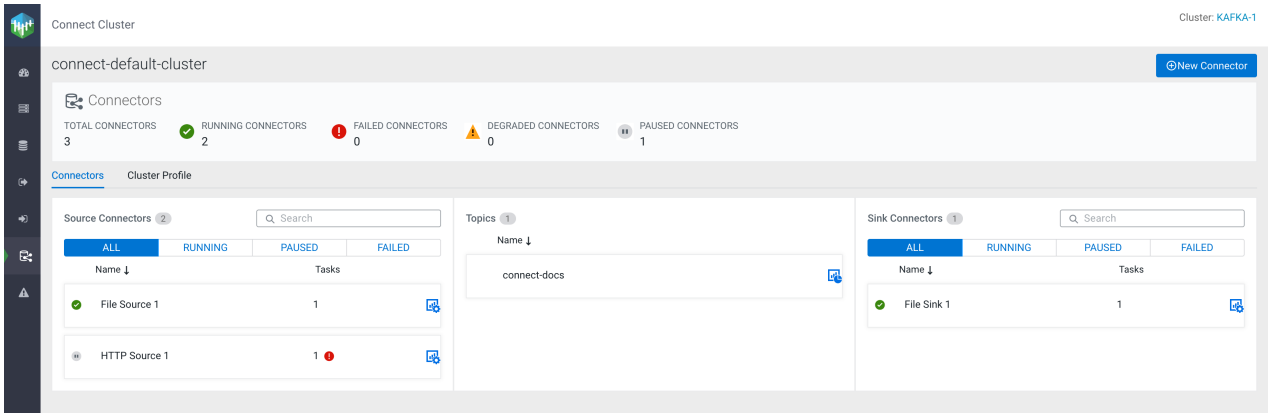
The Kafka Connect UI

Learn about the Kafka Connect section in the Streams Messaging Manager UI, which you can use to deploy, manage, and monitor Kafka Connect connectors.

The **Connect** section in Streams Messaging Manager is your main hub in Cloudera where you deploy, manage, and monitor Kafka Connect connectors. This section of the UI is only available if you deployed Kafka Connect role instances under your Kafka service and Streams Messaging Manager is configured to integrate with Kafka Connect. For detailed instructions on how to set up Kafka Connect and integrate it with Streams Messaging Manager, see *Kafka Connect Setup*.



Clicking Connect, opens the **Connect Cluster** page with the **Connectors** tab open. This is the default page of the Kafka Connect UI. It provides a high level overview of the connectors and the connect cluster.



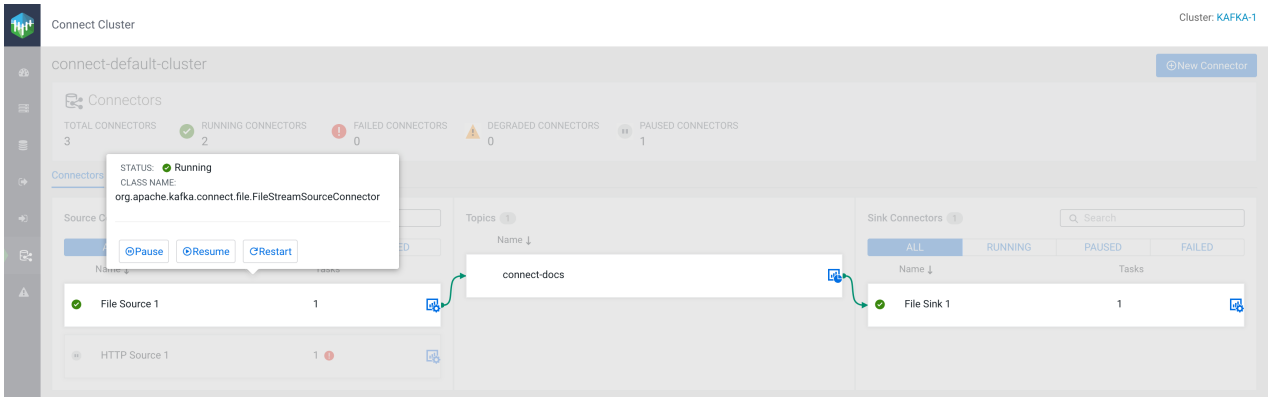
At the top-right corner of the **Connect Cluster** page, you can see the name of the cluster. The New Connector option below the cluster name starts the **Connector Setup** wizard, which you use to deploy connectors in your cluster. For detailed steps on how to deploy a new connector, see *Deploying and Managing connectors* .

The **Connectors** section provides some basic metrics regarding Kafka Connect including total connectors, running connectors, failed connectors, degraded connectors, and paused connectors. Under the Connectors section you can find two tabs, Connectors and Cluster Profile. The two tabs enable you to monitor and manage your connectors and the connect cluster.

Connectors tab

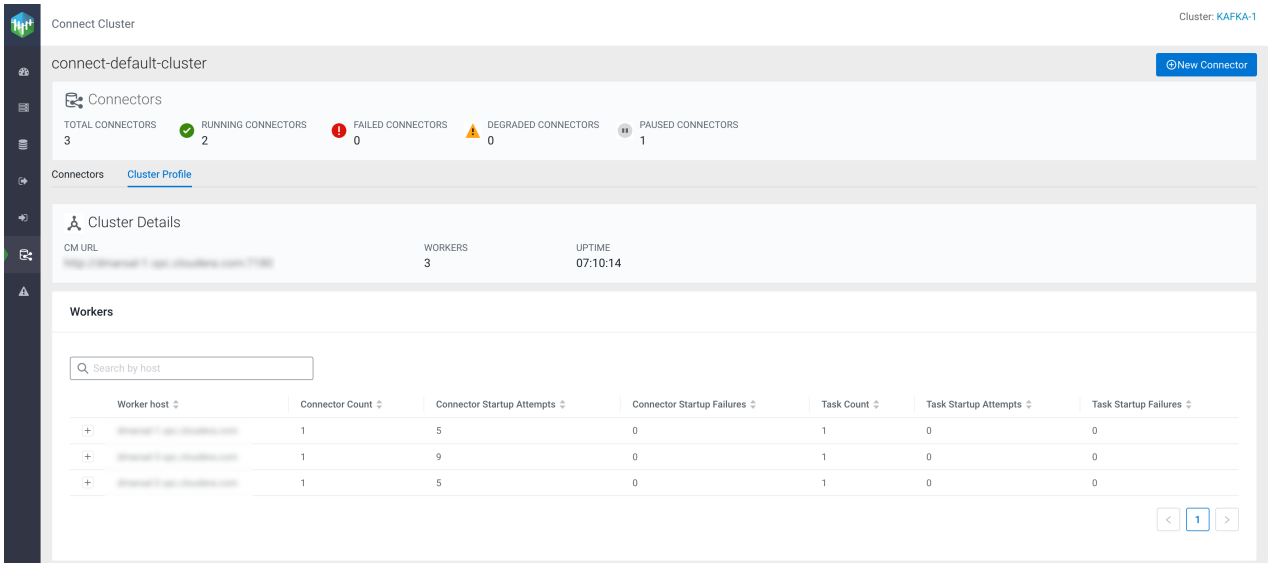
In the Connectors tab, you can view details of the source connectors, topics, and sink connectors in the cluster. The Source Connectors and Sink Connectors sections show all, running, paused, and failed connectors with connector name and associated task details. Both Source Connectors and Sink Connectors sections contain a Search option that enables you to search for particular connector details. The Topics section shows the Kafka topics where data is read from or written to.

Hovering over a specific connector displays a context menu where you can view the class name and status of the connector. The context menu also provides quick options to pause, resume, and restart the connector. Clicking on any of your deployed connectors or topics visualizes how the data is flowing into or out of the selected entities.

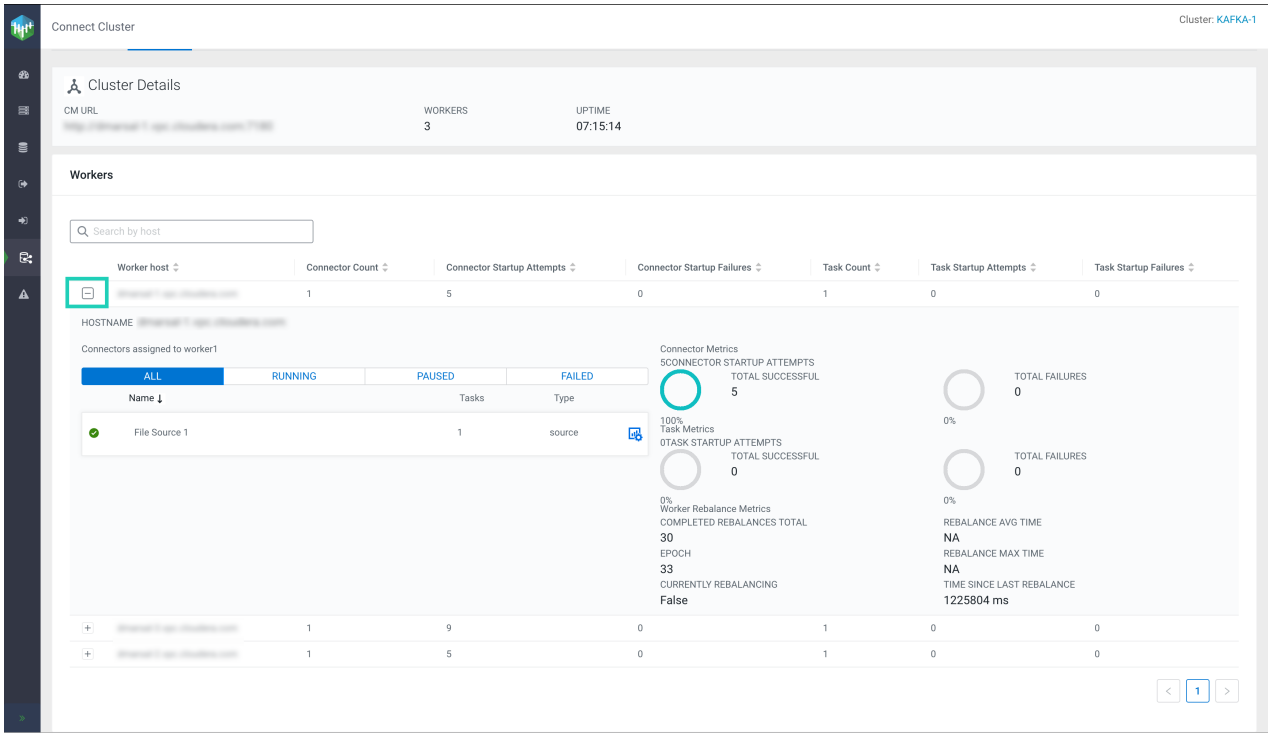


Cluster Profile tab

In the Cluster Profile tab, you can view details of the connect cluster and the workers. The tab includes information regarding your cluster such as the Cloudera Manager URL, the number of workers deployed in the cluster, as well as an uptime counter. Additionally, the **Workers** section lists all worker hosts available in the cluster. This section includes basic worker-level information, such as how many connectors are deployed on a worker, success and failure rates, and more.



If you click the + option found next to each host, additional details and metrics are displayed that are specific to the worker host you selected. The information includes a list of connectors assigned to the worker, as well detailed connector, task, and worker rebalance metrics. Using the Search by host field, you can search for worker details by host. Additionally, you can filter the list of connectors deployed on the worker host based on their status.



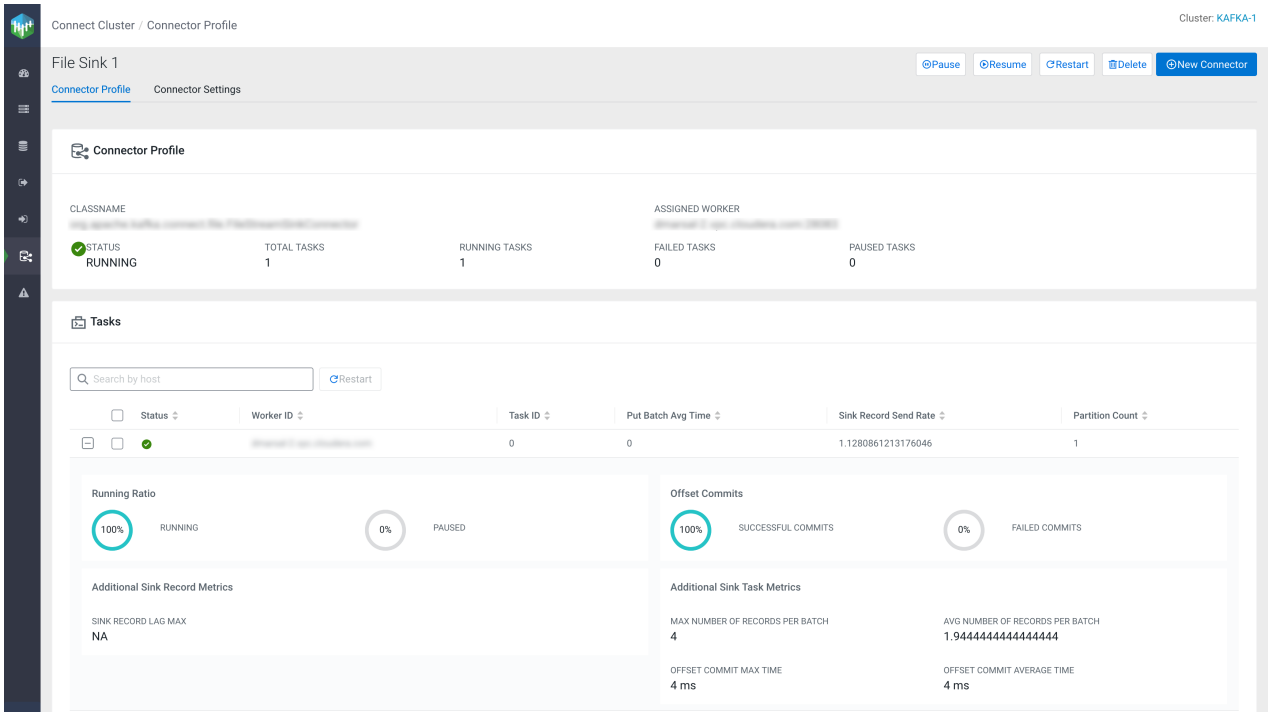
The Connector Profile

After you deploy a connector, you can monitor the details of the connector, modify its configuration, and carry out various management actions.

To monitor or manage a specific connector, click (Connector Profile) next to the connector's name. Clicking the icon redirects you to the **Connector Profile** page of the selected connector. The **Connector Profile** page enables you to monitor the details of the connector, monitor and manage its tasks, and edit its configuration. The page consists of two tabs **Connector Profile** and **Connector Settings**.

Connector Profile tab

The Connector Profile tab enables you to monitor details of the connector and its tasks. Additionally, a few basic management options are also available.



This tab is organized into multiple sections. The various sections enable you to do the following:

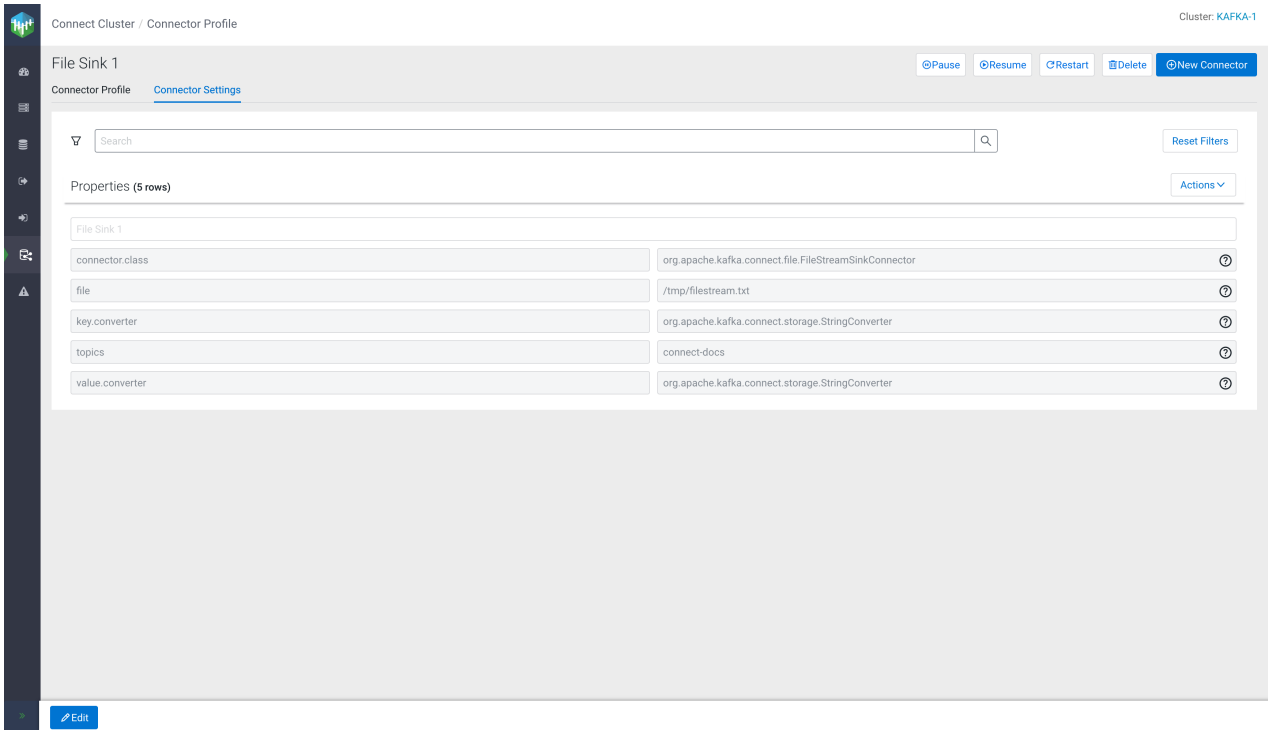
- The **Connector Profile** section provides you with details regarding the Classname, Assigned Worker, Status, Total Tasks, Running Tasks, Failed Tasks, and Paused Tasks.
- In the **Tasks** section, you can view and monitor Status, Worker ID, Task ID, and various other details regarding connector tasks.

  Clicking + next to a task displays detailed information and metrics about the selected task. In addition to viewing status and metrics, the **Tasks** section also allows you to restart a particular task. This can be done by selecting the task you want to restart and clicking the Restart option found within the **Tasks** section. The Search by host option enables you to search for particular task details by host.
- Using the buttons in the top right-hand corner you can pause, resume, restart, and delete the connector, or deploy a new connector.

Connector Settings tab

The Connector Settings tab enables you to review and edit the configuration of the connector. By default editing the connector configuration is disabled. To enabled editing, click Edit in the bottom left-hand corner. In addition to reconfiguring the connector, you can pause, resume, restart, and delete the connector, or deploy a new connector with the buttons in the top right-hand corner of the page.

Tip: The **Connector Settings** tab is identical in functionality to the **Connector Configuration** page in the **Connector Setup** wizard. For detailed information on the various options and features available related to configuration, see *Connector configuration features in Streams Messaging Manager*.



Related Information

Kafka Connect Setup
Integrating Kafka Connect with Streams Messaging Manager
Connectors
Deploying and managing Kafka Connect connectors in Streams Messaging Manager
Connector configuration features in Streams Messaging Manager

Deploying and managing Kafka Connect connectors in Streams Messaging Manager

Learn how to use the Streams Messaging Manager UI to deploy new Kafka Connect connectors using the Connector Setup wizard, as well as how to reconfigure, pause, resume, restart, and delete existing connectors. Additionally, learn about the various features and options Streams Messaging Manager provides for editing connector configurations.

Deploying a Kafka Connect connector in Streams Messaging Manager

Kafka Connect connectors are deployed in Streams Messaging Manager using the Connector Setup wizard. Learn how to deploy a new connector using the wizard.

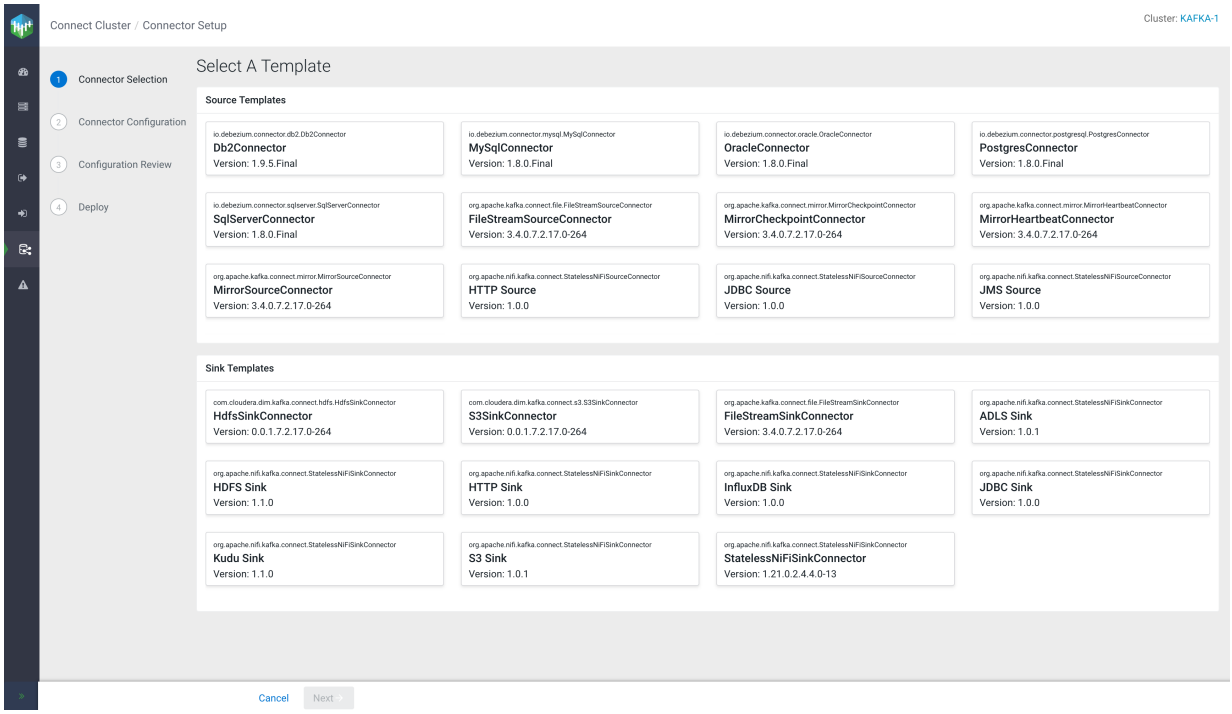
Before you begin

- By default, you can use the Connector Setup wizard to deploy any connector that is shipped with Cloudera Runtime. The only exceptions are the FileStream example connectors (FileStreamSourceConnector and FileStreamSinkConnector). Although these connectors are part of the Kafka distribution and are shipped with Cloudera Runtime, they are not available by default in the Connector Setup wizard. This is because neither connector is considered production ready. If you want to deploy an instance of these connectors, you must install them first. For more information, see Installing Kafka Connect connectors.
- Third-party connectors can be deployed using the wizard, but must be installed first. For more information, see Installing Kafka Connect connectors.
- Before deploying any connector, Cloudera recommends that you review the documentation of the connector. For more information regarding each connector shipped in Cloudera, see Connectors.

Procedure

1. Click **Connect** in the navigation sidebar.
2. Click the New Connector option.

   This option is available on all Kafka Connect related pages of the Streams Messaging Manager UI. Clicking this option starts the **Connector Setup** wizard and redirects you to the **Select A Template** page. This page contains all connectors available for deployment in the form of selectable cards.

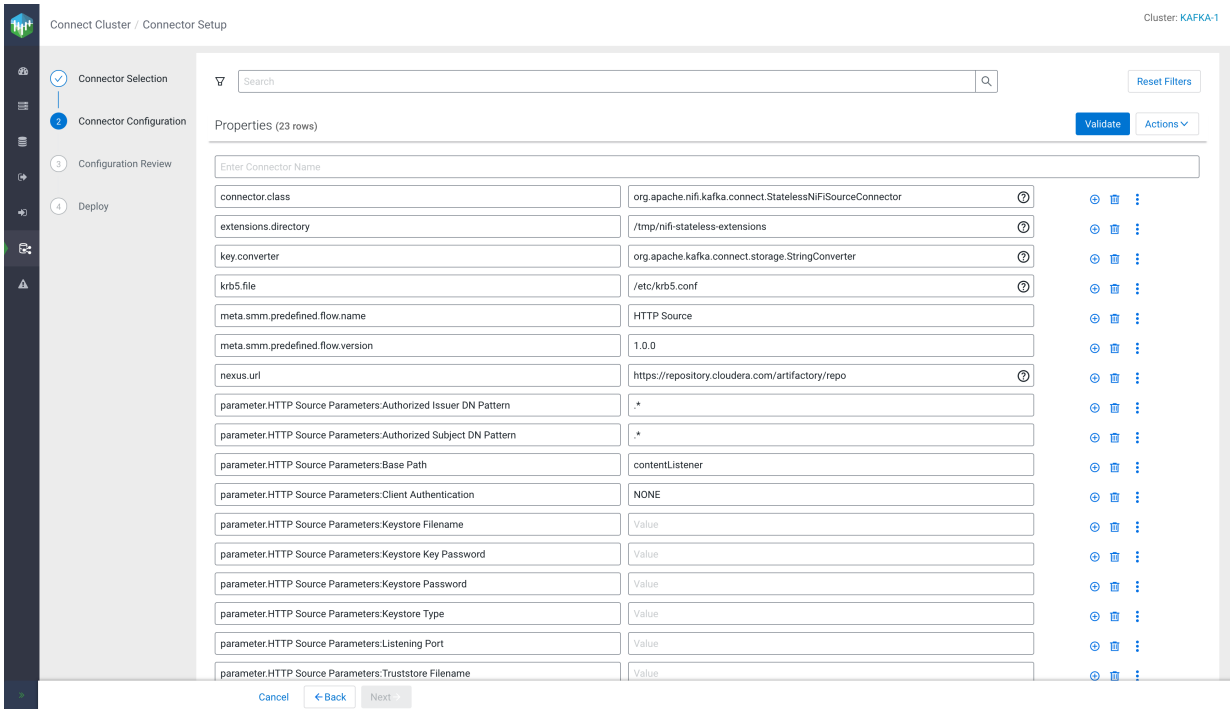


   Each card includes the following information about the connector:

   - The connector's fully qualified class name.
   - The connector's display name. If no display name is available, the card includes the unqualified classname.
   - The version of the connector.

   By default, the page includes all connectors shipped with Cloudera Runtime. Third-party connectors that you install manually are also visible on this page following installation.
3. Select a connector from the **Source Templates** or **Sink Templates** section.

   The Connector Configuration page appears.



Most connectors shipped with Cloudera Runtime come with a default configuration template to ease configuration. If a template is available for a specific connector, the property keys and values are automatically populated when you select the connector. The properties and values included in the templates depend on the selected connector. In general, the templates include all mandatory properties that are required for successful deployment. However, most connectors also have a number of additional properties that might not be part of the template. As a result, Cloudera recommends that you always review the documentation for the specific connector that you want to deploy.

4. Configure the properties of the connector.

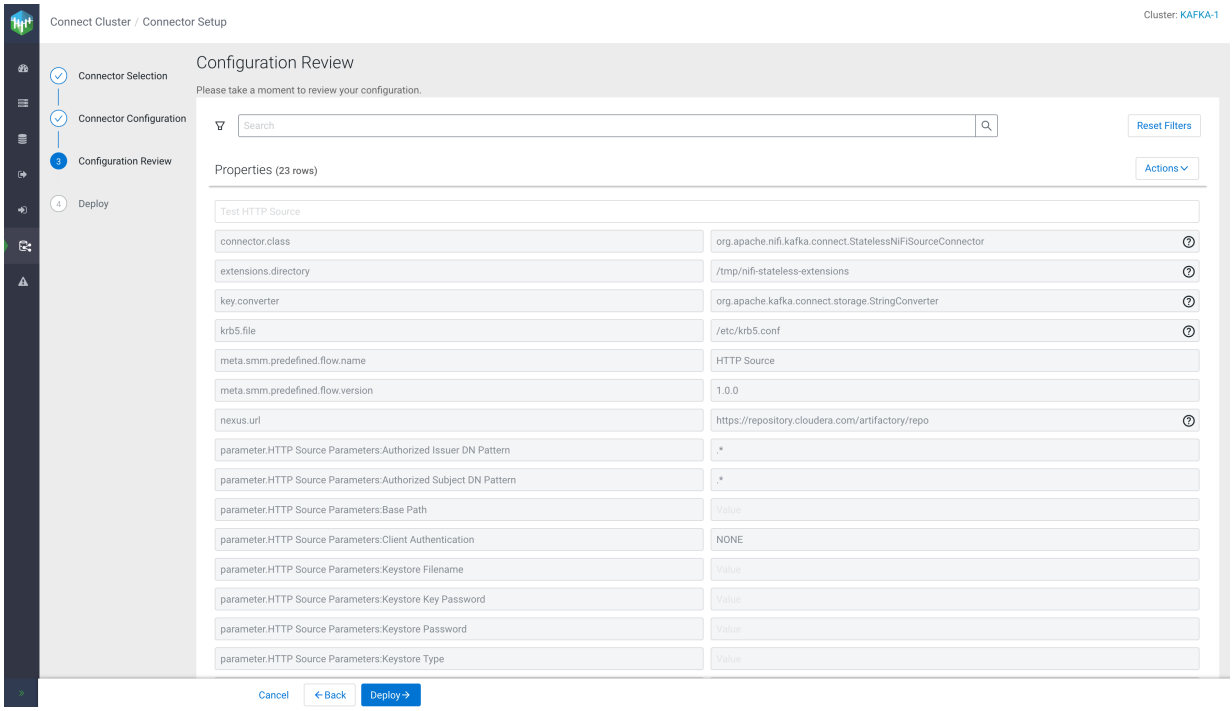
   The Connector Configuration page includes various features and options that you can use to configure your connector. Each row in the **Properties** section represents the key and value of a property. Using the different buttons and other options available, you can add, delete, import, or otherwise modify the configuration. For more details on the configuration features available on this page, see *Connector configuration features in Streams Messaging Manager*.

5. Click Validate after you are done configuring the connector.

   Validating the configuration is mandatory when you deploy or modify a connector. If Streams Messaging Manager finds any errors in the configuration, the properties that contain errors are highlighted in red, and an error message with the details regarding the configuration issue is displayed. Resolve any errors until validation passes.

6. Click Next.

   The Next option is disabled until validation passes. After you click Next, the **Configuration Review** page appears.



7. Review your connector configuration.

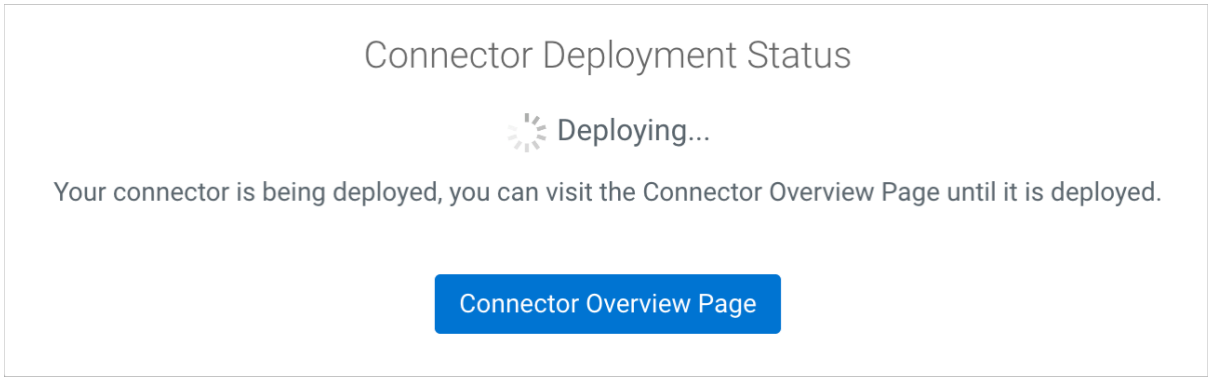
   The **Configuration Review** page allows you to review the connector configuration. Use this page to ensure that all properties are correctly configured for your use case.

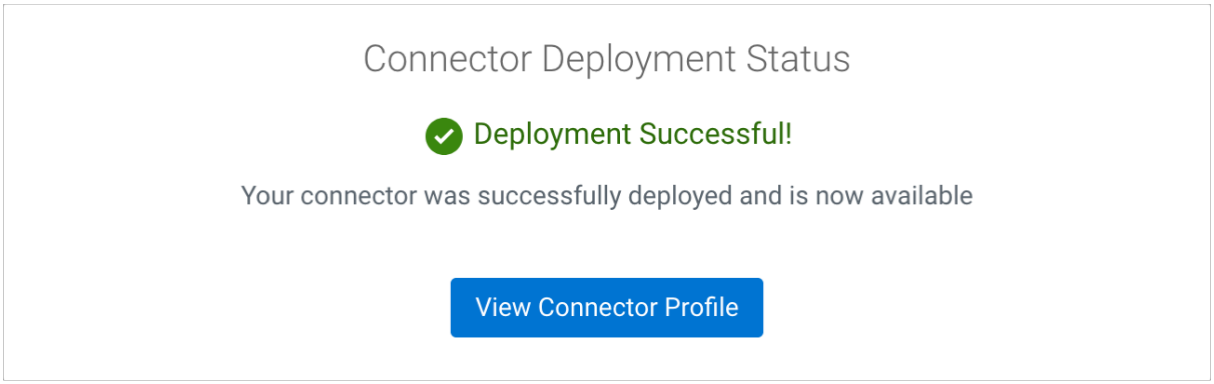
   Most configuration options are disabled on this page. However, you can use the search and filtering options to search for properties. Additionally, you can export the connector configuration for later use with Actions Export . If you find any errors, go back to the **Connector Configuration** page, make any necessary changes, and revalidate the configuration before continuing.

8. Click Deploy to deploy the connector.

   After you click Deploy, the **Connector Deployment Status** modal is displayed that shows the status of the deployment.

   Note: Connector deployment is not instantaneous. You might need to wait a few seconds for deployment to finish.





What to do next

After the connector is deployed, you can monitor the connector on the Connect Cluster page or the connector's profile page. For more information, see *The Kafka Connect UI*.

Related Information

Connectors

Streams Messaging Reference

The Kafka Connect UI

Connector configuration features in Streams Messaging Manager

Pausing, resuming, restarting, and deleting a Kafka Connect connector in Streams Messaging Manager

Learn how to pause, resume, restart, or delete an existing Kafka Connect Connector in Streams Messaging Manager.

Procedure

1. Click **Connect** in the navigation sidebar.
2. Identify the connector you want to manage.
3. Click (Profile) beside the connector.
4. Click Pause, Resume, Restart, or Delete.

   If you select Delete, you must confirm the action in a pop-up window.

   Tip: The Pause, Resume, and Restart options are also available if you hover over the connector on the **Connect Cluster** page.

Related Information

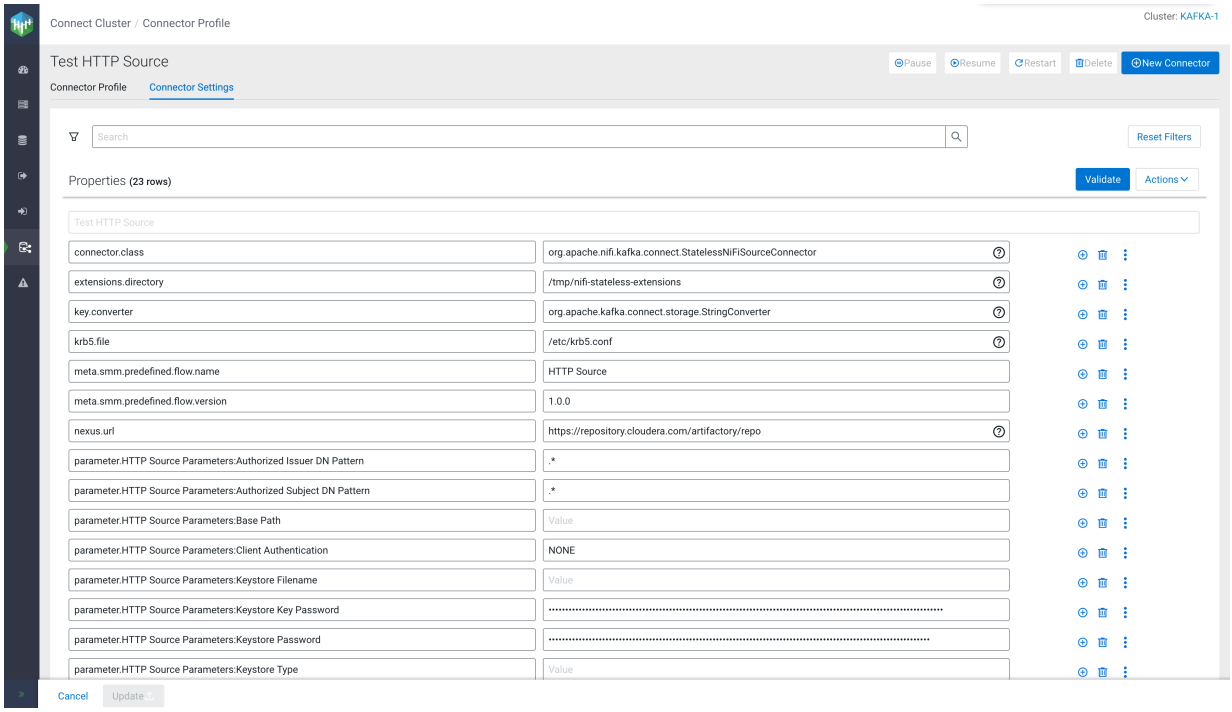
Connectors
Streams Messaging Reference
The Kafka Connect UI

Reconfiguring Kafka Connect connectors in Streams Messaging Manager

Learn how to edit the configuration of a running Kafka Connect connector using the Streams Messaging Manager UI.

Procedure

1. Click **Connect** in the navigation sidebar.
2. Identify the connector you want to modify.
3. Click (Profile) beside the connector.
4. Go to the Connector Settings tab.
5. Click Edit at the bottom-left corner of the page.
   All properties except connector name become editable.



6. Configure the properties of the connector.
   The Connector Settings page includes various features and options that you can use to configure your connector. Each row in the **Properties** section represents the key and value of a property. Using the different buttons and other options available, you can add, delete, import, or otherwise modify the configuration. For more details on the configuration features available on this page, see *Connector configuration features in Streams Messaging Manager*.
7. Click Validate after you are done configuring the connector.
   Validating the configuration is mandatory when you deploy or modify a connector. If Streams Messaging Manager finds any errors in the configuration, the properties that contain errors are highlighted in red, and an error message with the details regarding the configuration issue is displayed. Resolve any errors until validation passes.
8. Click Update .

Related Information

Connectors
Streams Messaging Reference
The Kafka Connect UI
Connector configuration features in Streams Messaging Manager

Connector configuration features in Streams Messaging Manager

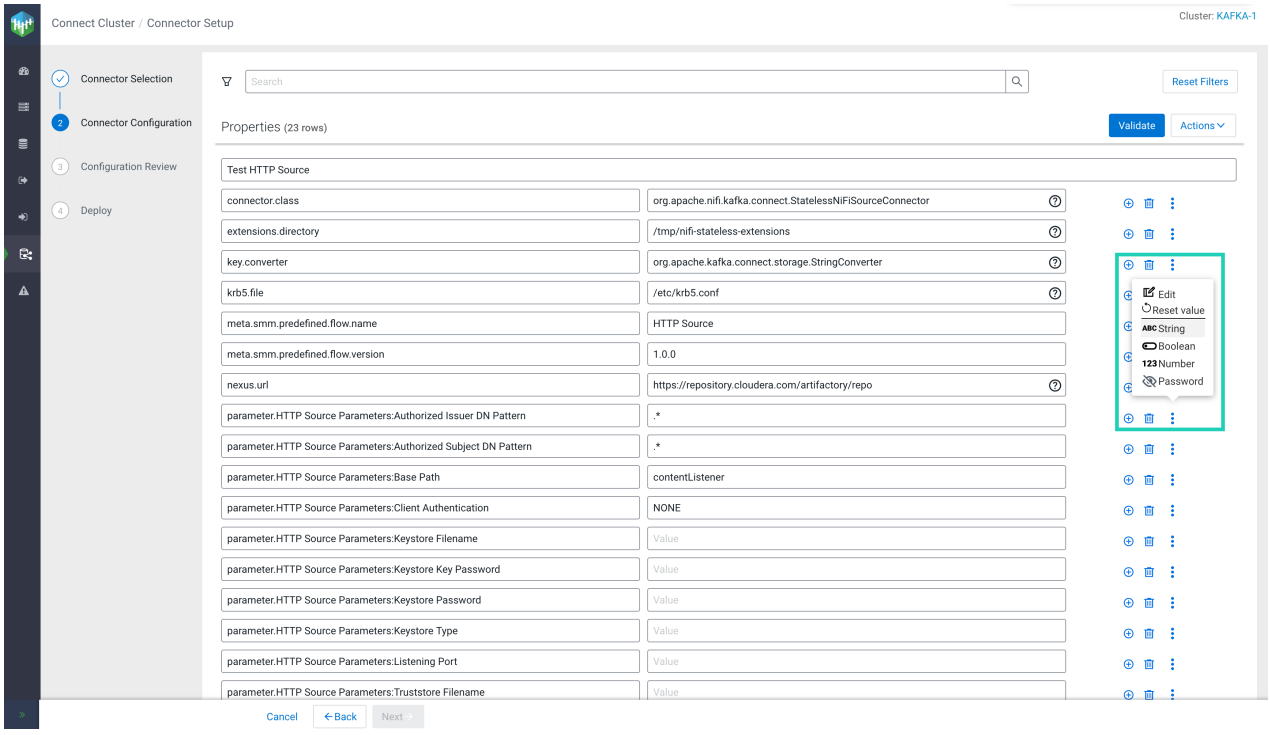
Learn about the various configuration features and options that you can use when configuring Kafka Connect connectors in Streams Messaging Manager.

The Streams Messaging Manager UI includes two pages where you configure Kafka Connect connector properties. The **Connector Configuration** step of the **Connector Setup** wizard and the Connector Profile Connector Settings tab. Both pages include various features and options that are designed to help you with connector configuration. The following sections go over each of the configuration options available on the UI.

Note: The only difference between **Connector Configuration** and **Connector Settings** is that the former is used to configure new connectors, the latter is used to reconfigure already running connectors. Otherwise, the two pages are identical in functionality.

Configuring properties

Each row on the **Connector Configuration** page represents the key (name) and configuration value of a specific property. You can use the icons next to each property to add or remove properties. Clicking ⋮ next to a property opens a context menu that includes additional configuration options. The options available in the ⋮ context menu depend on the property. For example, you can edit and reset the connector.class property, but you cannot configure its type.



Adding properties

Click ⊕ to add a new property. An empty row appears for the new property. Enter property keys in the left field and property values in the right field. When you select a key field, a list of suggested

properties appear that are valid for the connector. The list is automatically filtered when you start typing.

Deleting properties

Click to delete a property. Use Actions Remove to remove all properties from the configuration with the exception of connector.class.

Editing properties

Click Edit to edit the configuration value of a property in a separate window. Use this option if the value you add is long, complex, or consists of multiple lines.

If you are deploying a NiFi Stateless Sink or Source connector, the modal that opens with for the flow.snapshot property is unique and includes the Browse... and Save and Enhance options. Browse... enables you to upload a flow definition JSON from your machine, Save and Enhance adds the parameters specified in your flow definition to the connector configuration. For more information on flow.snapshot configuration, see *Configuring flow.snapshot for Stateless NiFi connectors*.

Resetting properties

You can use Reset value to reset the value of a single property. Alternatively, you can use Actions Reset to reset the full configuration. The reset options behave uniquely depending on the configuration page that you use them on.

When deploying new connectors using the **Connector Configuration** page of the wizard:

- Reset value resets the value to the default value set in the configuration template. This option is not available for properties that are not part of the template.
- Actions Reset resets all configuration properties and reverts the configuration to the default configuration template.

When editing the configuration of existing connectors on the Connector Profile Connector Settings page:

- Reset value resets the value to the last value that the connector was deployed with. This option is not available for new properties that you add to the configuration while editing the configuration.
- Actions Reset resets all configuration properties and reverts to the configuration that the connector was last deployed with.

Type configuration

The ABC String , Boolean, 123 Number, and Password options enable you to set the type of the property. Setting the type of a property changes the property's value field depending on the type you select.

- ABC String sets the type of the property to string and changes the input field to a single line text box.
- 123 Number sets the type to number and changes the input field to a text box that only allows negative and positive whole numbers and decimals.
- Boolean sets the type of the property to boolean and changes the input field to a checkbox.
- Password sets the type of the property to password.

Selecting Password hides (locks) the value on the UI and stores it in a secure manner. Once the connector is deployed, properties marked as passwords are encrypted and stored securely in an internal Kafka topic. The actual values only resolve at runtime. These values cannot be retrieved or otherwise read from the configuration of the connector. In most cases, you use this option to hide passwords or other types of sensitive credentials found in connector configurations, however, you can use this option for any property value that your organization considers sensitive.

Tip: The password option is the UI implementation of the Kafka Connect Secrets storage feature. For more information on the feature, and how Kafka Connect handles values you mark as sensitive data, see *Kafka Connect Secrets Storage*.

Property search and filtering

At the top of the page you can find a search bar that you can use to search and filter for connector properties. Search only works for property keys, values are disregarded. When you search for a string, the **Properties** section of the page is automatically filtered and only the relevant properties are shown. The number of matches is displayed next to **Properties**.

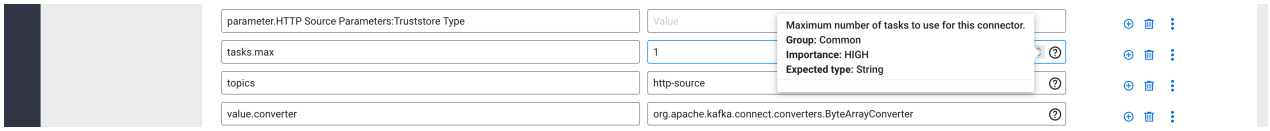
In addition to searching, you can filter properties based on their group and importance. To do so, click the filter icon and select the relevant group and importance that you want to search for. To clear all applied filters, click Reset Filters.



Viewing property help

Hovering over (Help), found next to some property values, displays information about that property. What information is displayed is property dependent, but at minimum you are presented with a description of the property.

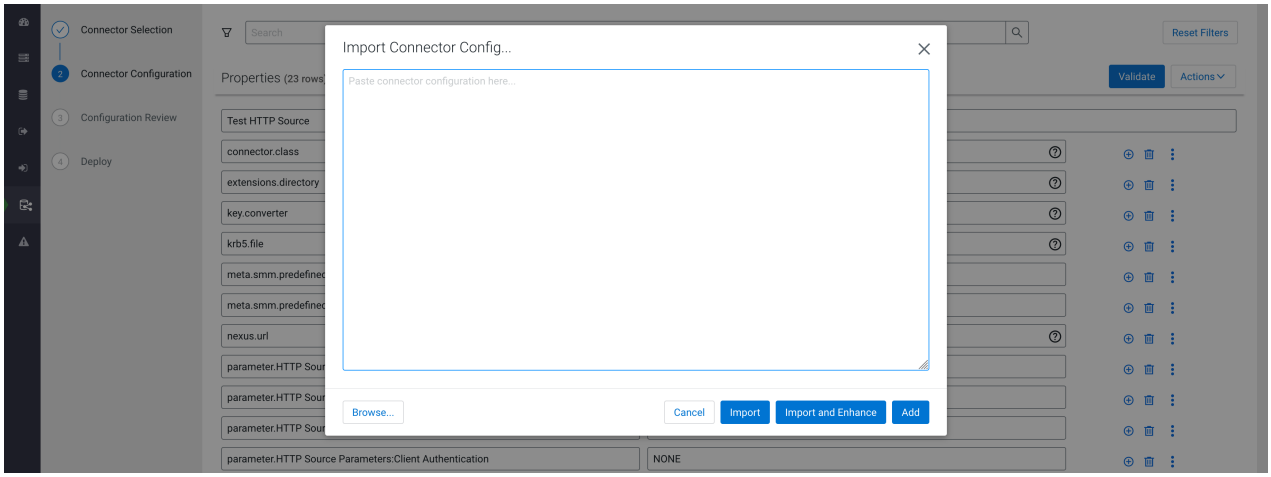
If available, the group, importance, and expected type of a property is also displayed. The icon is only available for properties that have their metadata (description, type, group, and so on) defined.



Tip: The information displayed in the help context menu is the property metadata defined with `ConfigDef`. If you are developing your own connector and want help information to be available in Streams Messaging Manager, ensure that you define property metadata in your connector code using the `ConfigDef` class. Additionally, if you specify a display name for the property in `ConfigDef`, Streams Messaging Manager will print the display name above the key field of the property.

Importing configurations

To import a configuration, click Actions Import . Clicking Import opens the **Connector Config…** modal.



Using the modal you can:

- Type or paste your configuration.
- Use Browse… to upload the contents of a file from your machine.

The configuration that you add can be a full configuration that contains all properties necessary for the connector. Alternatively, it can be a partial configuration that contains a select number of properties that you want to add to the configuration. The configuration you include in the text box must be in JSON format. For example:

```
{
  "tasks.max": 1,
  "key.converter": "org.apache.kafka.connect.storage.StringConverter",
  "value.converter": "com.cloudera.dim.kafka.connect.converts.AvroConvert
er",
  "value.converter.passthrough.enabled": true,
  "value.converter.schema.registry.url": "http://schema-registry:9090/api/v1
",
  "topics": "avro_topic",
  .
  .
  .
}
```

After adding properties, you can use the Import, Import and Enhance, or Add options to import the configuration. Each option imports the properties in a different way.

- Import

  This option populates the **Properties** section with the keys and values you added in the **Import Connector Config...** modal. This option overwrites all existing properties in the configuration.
- Import and Enhance

  This option populates the **Properties** section with the keys and values you added in the **Import Connector Config...** modal. Additionally, properties that are most likely needed for the respective connector are also added.

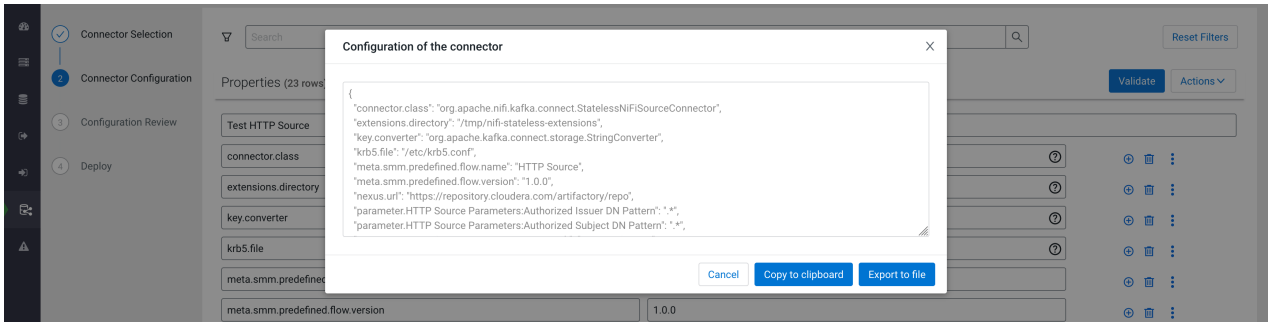
  Import and Enhance is used specifically for the NiFi Stateless Source and NiFi Stateless Sink connectors. When this option is used, Streams Messaging Manager parses the dataflow specified in flow.snapshot, extracts all parameters that are available in the dataflow, and adds them to the connector configuration. This option overwrites all existing properties in the configuration.
- Add

  This option amends the existing configuration with the properties from the **Import Connector Config...** modal. Use this option if you want to batch import additional properties to your configuration. For example, you can use this option to batch add security properties.

Exporting Configurations

To export a configuration, click Actions Export . Clicking Export opens the **Configuration of the connector** modal.

- Click Copy to Clipboard to copy the configuration to the clipboard.
- Click Export to file to download the configuration as a JSON file.



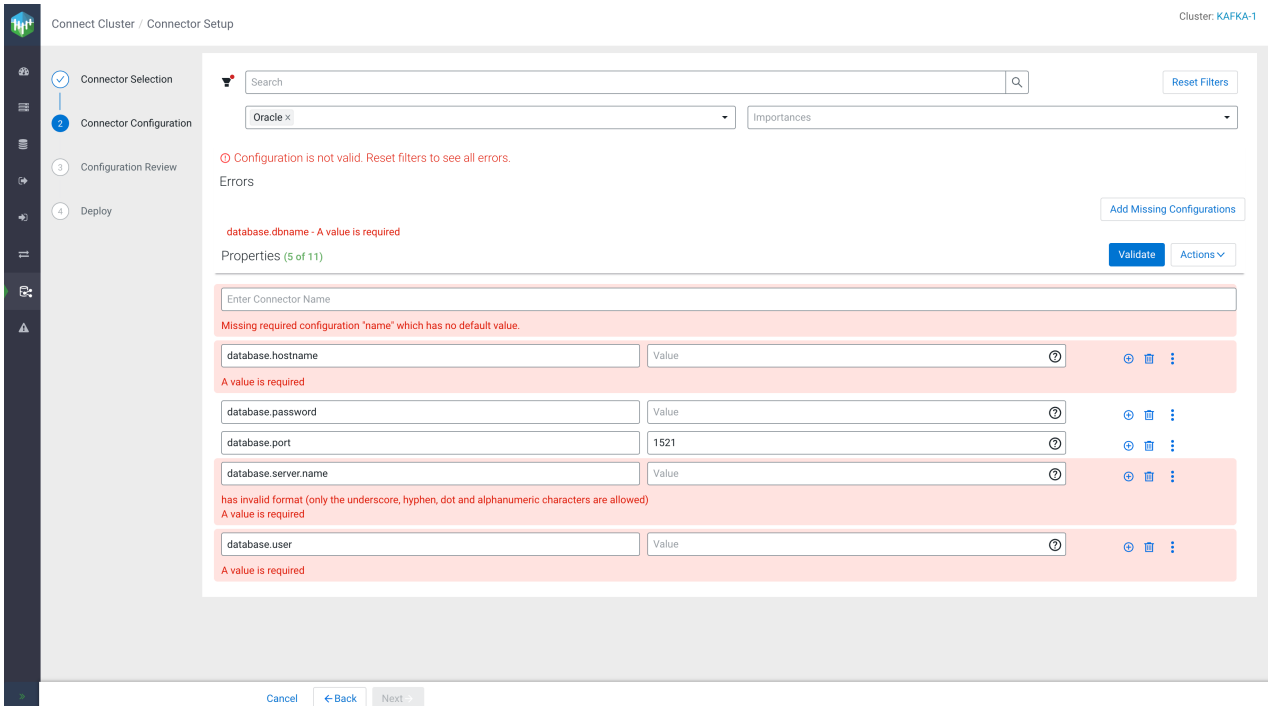
Validating configurations

Before you can deploy or redeploy a connector, you must validate the configuration. This is done using Validate. If Streams Messaging Manager finds any errors in the configuration, the properties that contain errors are highlighted in red, and an error message with the details regarding the configuration issue is displayed.

If the configuration is missing a mandatory property, the name of the missing property is displayed in an **Errors** section on the top of the page. You can add missing mandatory properties by clicking Add missing configurations. This option adds both property keys and values. Values are populated using the following logic.

- If the connector was previously deployed and you are updating the configuration, the default value used will be the value that the connector was last deployed with.
- If you are configuring a new connector, the default value from the sample configuration is used.
- If you are configuring a new connector and there is no sample configuration or the sample configuration does not contain the property, the value defined in the `ConfigDef` class of the connector is used.

If the connector configuration is filtered when you run the validation and Streams Messaging Manager finds errors with properties that are filtered, an error message is displayed on the top of the page notifying you that not all validation errors are visible. In a case like this, click Reset Filters to view all validation errors.



Related Information

Connectors

Streams Messaging Reference

The Kafka Connect UI

Kafka Connect Secrets Storage

Configuring flow.snapshot for Stateless NiFi connectors

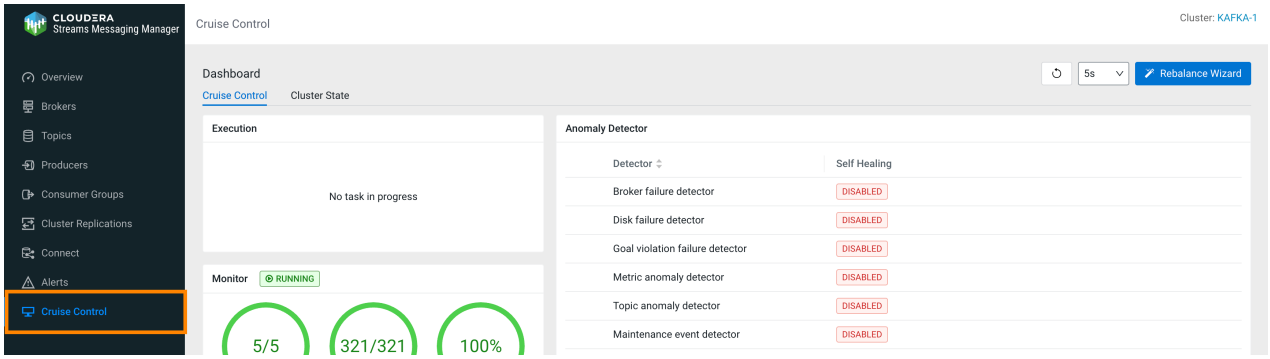
Managing and monitoring Kafka rebalancing with Cruise Control

Get started with Cruise Control in Streams Messaging Manager. Cruise Control is automatically integrated with Streams Messaging Manager. It provides user-friendly dashboards for monitoring the rebalancing process and state of the Kafka cluster. You can also manage and customize the rebalancing process, based on the configurations in Cloudera Manager.

Cruise Control dashboard in Streams Messaging Manager UI

Learn about the Cruise Control section in the Streams Messaging Manager UI, which you can use to monitor and manage the rebalancing process of Kafka clusters.

The Cruise Control UI in Streams Messaging Manager allows you to monitor the state and operations of the Cruise Control service, including the latest state of the Kafka cluster known to Cruise Control. The Rebalance Wizard allows admins to start rebalance operations, specifying goals, anomaly detectors and estimations that should be used for generating the rebalance proposals. You can access the Cruise Control UI from the left navigation pane of Streams Messaging Manager. Clicking on loads the **Cruise Control Dashboard**.

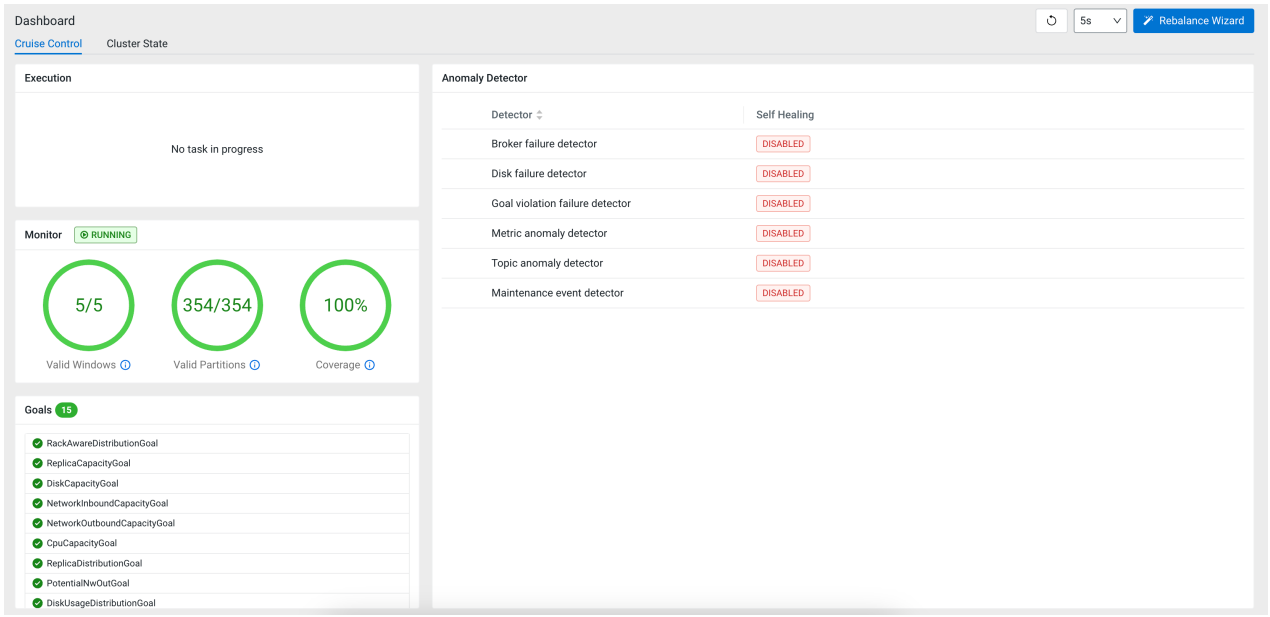


Note: In case Cruise Control is not available in the left navigation pane, review the instruction in *Authorizing users to access Cruise Control in Streams Messaging Manager*.

Cruise Control Dashboard

The **Cruise Control Dashboard** provides an overview of the Cruise Control state, including the following information:

- Any operations that are currently being executed by Cruise Control and their progress
- Status of internal Cruise Control metrics (valid windows, valid partitions and coverage)
- List of goals configured in Cruise Control and their status (ready or not ready)
- List of anomaly detectors and their status (disabled or enabled)



You can manually refresh the **Dashboard** using ↻, or you can set a refresh interval by selecting one of the options available with ⌄.

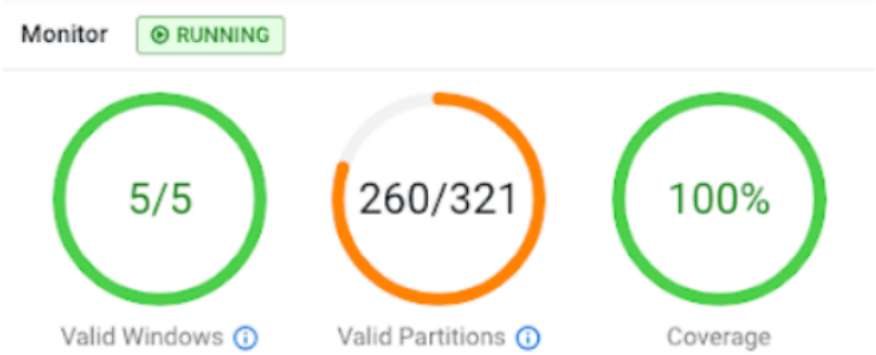
Execution

The **Execution** tile of the **Dashboard** allows you to track and monitor the tasks being executed by Cruise Control (including rebalancing tasks that are triggered by the enabled goals or other operations started through the REST API). If the goals are fulfilled and there is no ongoing execution, the No task in progress message is shown. Only users with Admin level can stop the execution process.



Monitor

The **Monitor** tile always shows the ratio of **Valid Windows**, **Valid Partitions**, and **Coverage** of the Kafka cluster.



Valid windows

The metrics that Cruise Control collects about brokers and partitions in a Kafka cluster are aggregated in windows of time. A window represents a time range. There is a minimum number of metric samples that a window should have to be considered "valid" by Cruise Control. Only valid windows can be used for balance proposals and anomaly detection.

The validity of a window is determined based on valid brokers, valid partitions and overall valid metric samples count. The parameters of this mechanism can be configured in Cloudera Manager with the following Cruise Control configuration properties:

Property	Description
num.broker.metrics.windows	The maximum number of broker windows the Load Monitor keeps. Each window covers a time window defined by the broker.metrics.window.ms configuration property.
broker.metrics.window.ms	The size of the window in milliseconds to aggregate the Kafka broker metrics. The window must be greater than the value of the metric.sampling.interval.ms configuration property. metric.sampling.interval.ms is the duration until metrics are collected and sampled. When an interval is ready, another interval starts and tries to collect the metrics from the brokers.
min.samples.per.broker.metrics.window	The minimum number of metric samples a valid broker window should have. If a broker does not have enough samples in a broker window, the broker is removed from the window due to insufficient data.

Property	Description
max.allowed.extrapolations.per.broker	The maximum number of extrapolations allowed for each broker. A broker is considered as invalid if the total number extrapolations in all the windows goes above the maximum allowed number. Load Monitor contains information about the status of the linear regression model, which is used to estimate the CPU utilization, number of valid windows and partitions. If there are flawed brokers, not all windows have metrics. Cruise Control extrapolates these metric values.

Valid partitions

Cruise Control aggregates metrics about each partition in the connected Kafka cluster. There is a minimum number of metric samples a valid partition should have. The validity of a partition is determined based on the valid metric samples count. The mechanism can be configured in Cloudera Manager with the following Cruise Control configuration properties:

Property	Description
num.partition.metrics.windows	The maximum number of partition windows the Load Monitor keeps. Each window covers a time window defined by the partition.metrics.window.ms configuration property.
partition.metrics.window.ms	The size of the window in milliseconds to aggregate the Kafka partition metrics. The window must be greater than the value of the metric.sampling.interval .ms configuration property. metric.sampling.interval.ms is the duration until metrics are collected and sampled. When an interval is ready, another interval starts and tries to collect the metrics from the brokers.
min.samples.per.partition.metrics.window	The minimum number of metric samples a valid partition window should have. If a partition does not have enough samples in a partition window, the topic of the partition is removed from the window due to insufficient data.
max.allowed.extrapolations.per.partition	The maximum number of extrapolations allowed for each partition. A partition is considered as invalid if the total number extrapolations in all the windows goes above this number. Load Monitor contains information about the status of the linear regression model, which is used to estimate the CPU utilization, number of valid windows and partitions. If there are flawed partitions, not all windows have metrics. Cruise Control extrapolates these metric values.

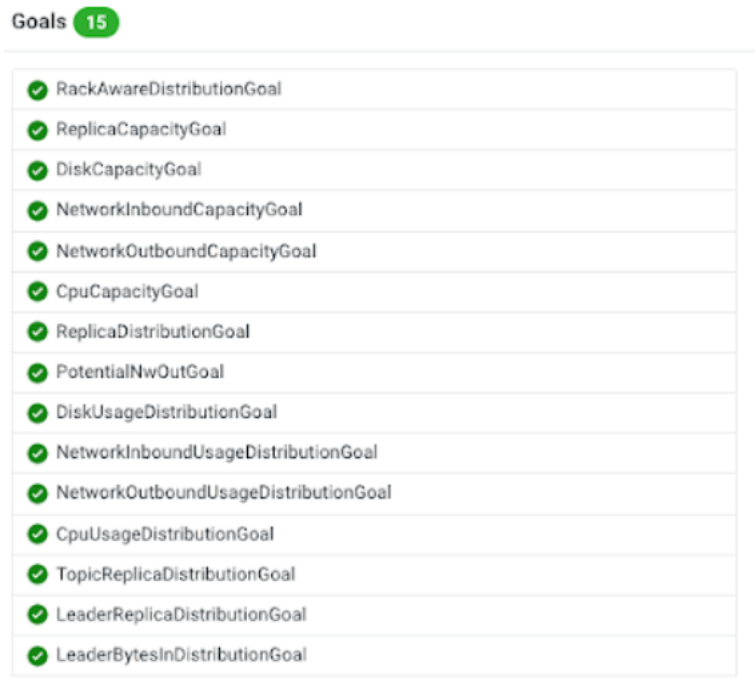
For more information on the configuration properties, see the Cruise Control upstream documentation

Coverage

The coverage of this cluster model through monitored partitions ratio, showing how representative the cluster is.

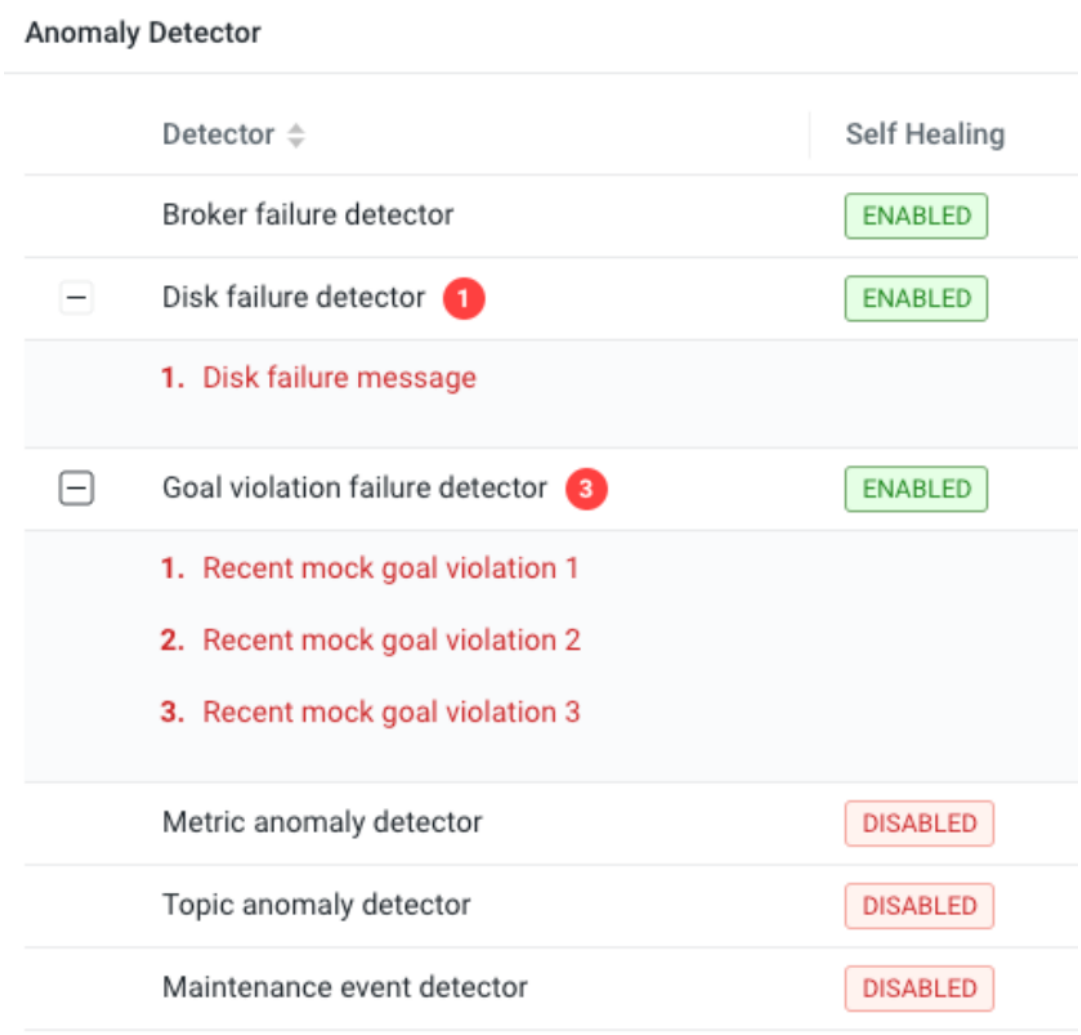
List of Goals

The **Goals** tile shows the list of enabled goals and indicates if the goals have been fulfilled or not with checkmarks. The list of goals can be configured in Cloudera Manager. For more information, see *Setting capacity estimates and goals*.



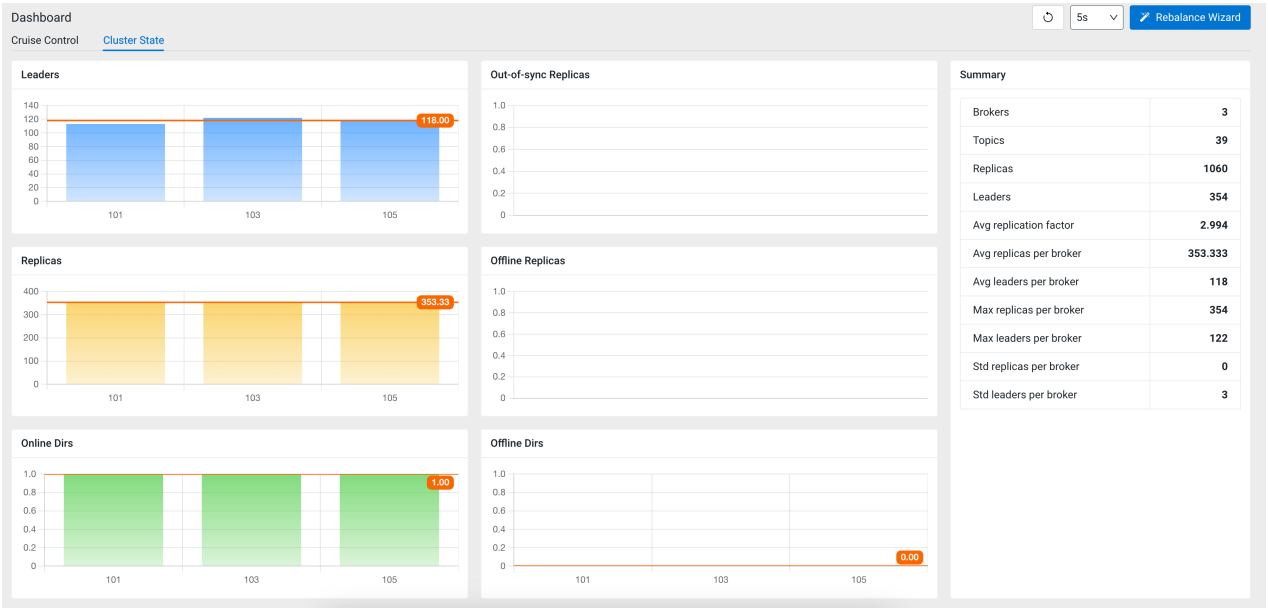
List of Anomaly Detectors

The **Anomaly Detector** tile shows the list of anomaly detectors and whether self-healing is enabled for the given detector. If there are any anomalies found by the detectors, the anomalies are listed under the corresponding detector.



Cluster State Dashboard

The **Cluster State Dashboard** provides an overview of the Kafka cluster specific metrics related to leaders, replicas, and directories. The metrics shown on the **Dashboard** are based on the number of leaders, online, offline and out-of-sync replicas, online and offline directories within the brokers available on the Kafka cluster. You can use these metrics to monitor the Kafka cluster state.



In the **Summary** section, you can review the general information of the number of brokers, topics, replicas and leaders within the Kafka cluster, and more metrics that show the average, maximum and standard ratio of leaders per broker and replicas per broker.

Related Information

Setting capacity estimations and goals

Authorizing users to access Cruise Control in Streams Messaging Manager

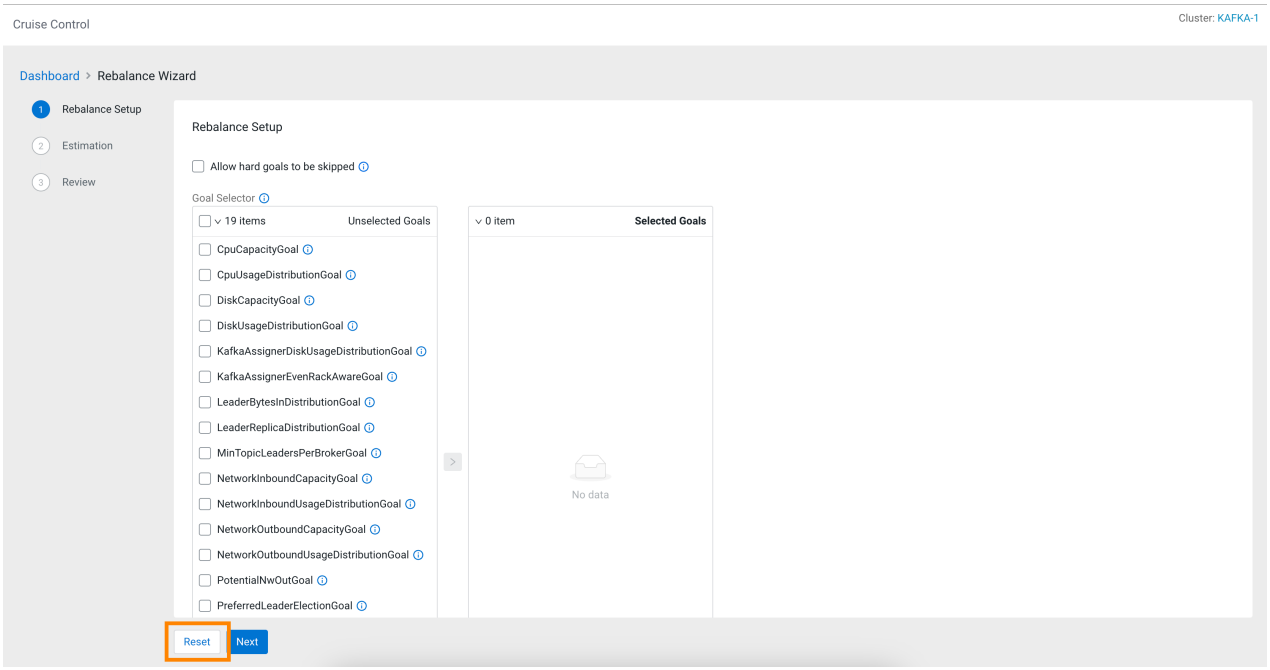
Using the Rebalance Wizard in Cruise Control

You can start and manage the rebalancing process of the Kafka cluster using the Rebalance Wizard.

About this task

You can use the Rebalance Wizard, located beside the **Dashboard**, used to trigger the rebalancing process of the Kafka cluster. You must have Admin-level permission to use the Rebalance Wizard.

To restore the default settings, you can use Reset in the Rebalance Wizard, and go back to the Dashboard using the Dashboard link.



Procedure

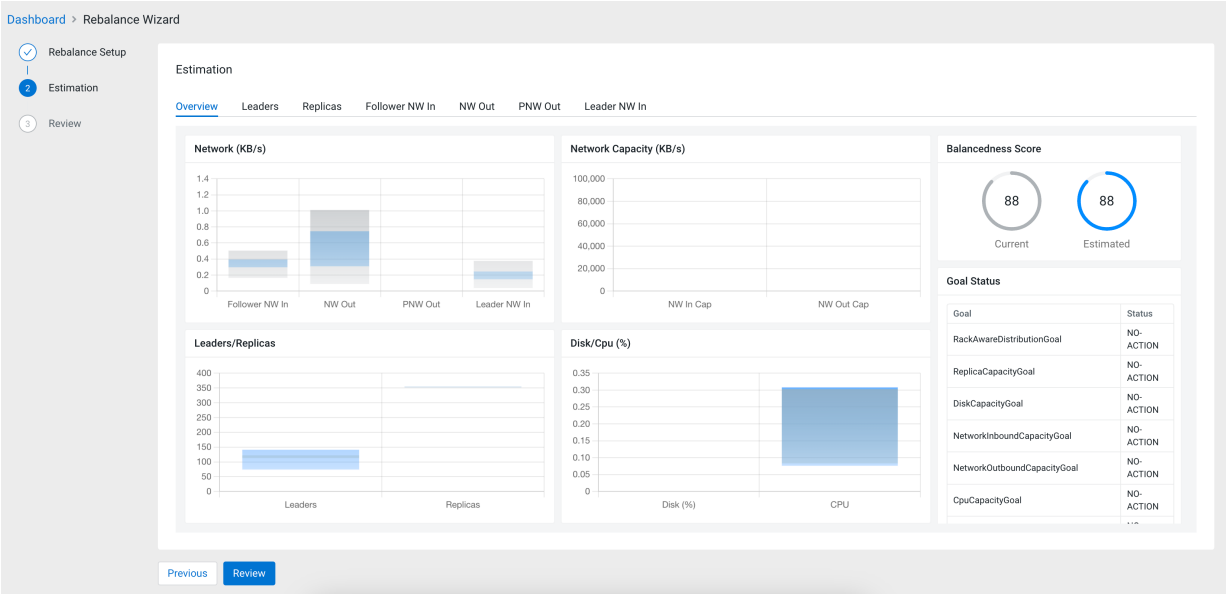
1. Click Rebalance Wizard on the **Dashboard**.
2. Configure the Rebalance Setup to determine how the rebalance request is executed.

   The configurations can be used to customize the rebalancing process based on your requirements. If the configurations are not changed and the fields are empty, Cruise Control completes the rebalancing process with the default settings. The rebalance setup and goal list is based on the configuration in Cloudera Manager.

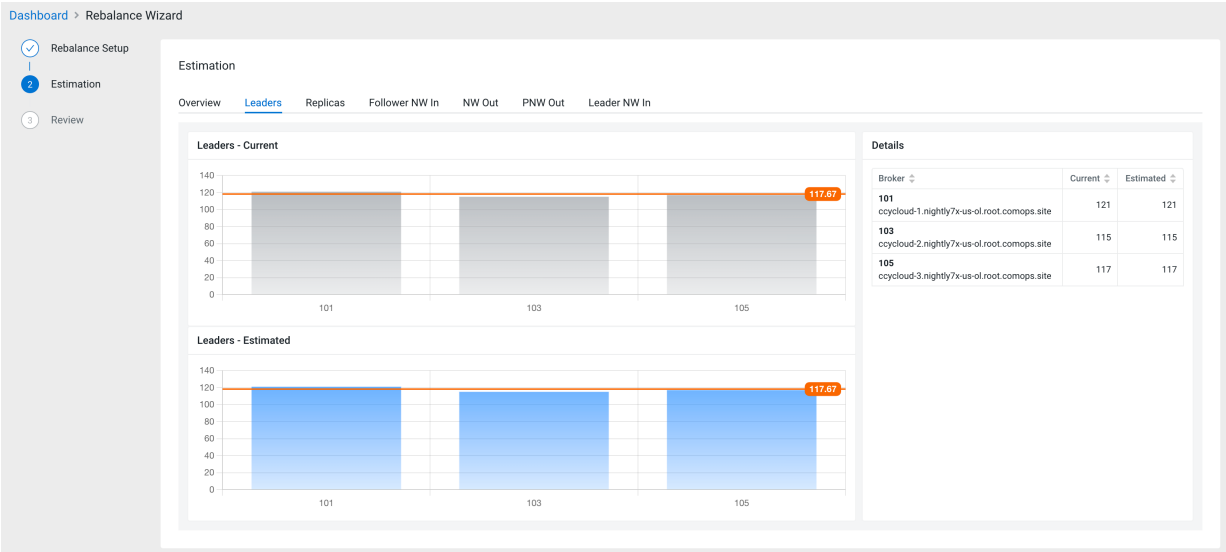
   Tip: Cloudera recommends to configure the following configurations even if the other configurations stay as default:

   - Disk rebalance to ensure that the disks or the entire brokers are rebalanced
   - Excluded topics to run rebalance only for those topics that are explicitly selected
   - Destination brokers to run rebalance only for a group of brokers
3. Click Next to generate the proposal.

   Note: If the proposal generation fails, an error message is displayed with the incorrect configuration. After correcting the configurations, you can move forward to **Estimation**.
4. Review the **Estimation** of the rebalancing proposal.



The **Overview** tab contains diagrams about the Network, Network Capacity, Leaders and Replicas ratio and Disk CPU. The **Balancedness Score** shows how efficiently the resources are balanced by comparing the current state and the estimated state that was configured during the **Rebalance Setup**. You can also review the list of goals and the set actions that should be triggered when the specific goal is violated under **Goal Status**. The **Movements** table gives you an overview of the estimated movement to achieve the ideal rebalancement.

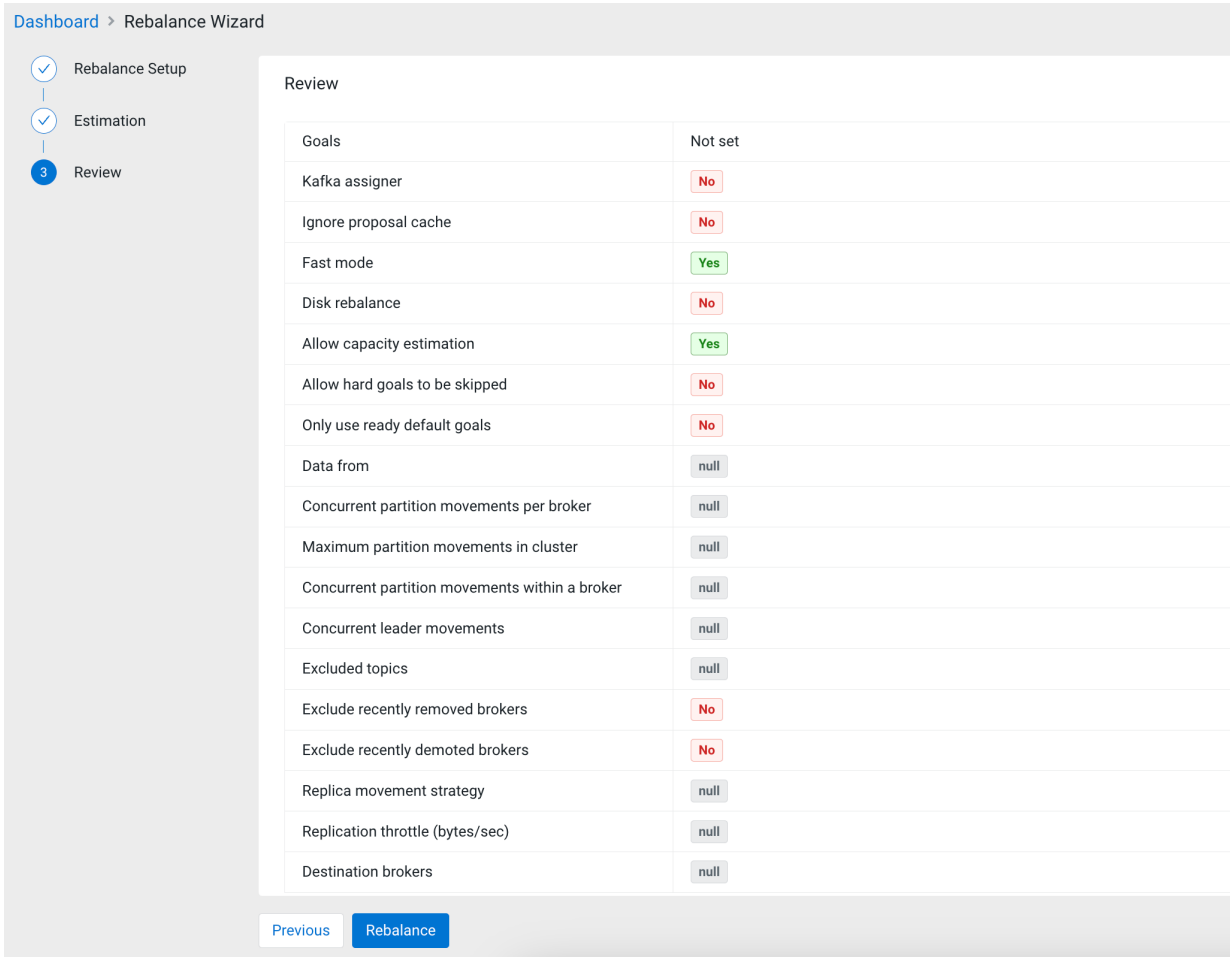


The **Leaders**, **Replicas**, **Follower NW in**, **NW Out**, **PNW Out** and **Leader NW in** tabs compare the current state and the estimated state after applying the new rebalancing configured in the **Rebalance Setup**. On each tab, the **Details** table shows the raw data that is visualized on the bar chart.

As the goal of the Estimation step is to review and assess the result of the rebalancing process, the configurations can be set again at the Rebalance Setup by clicking Previous.

5. Click Review if the estimation overview showed the desired results.

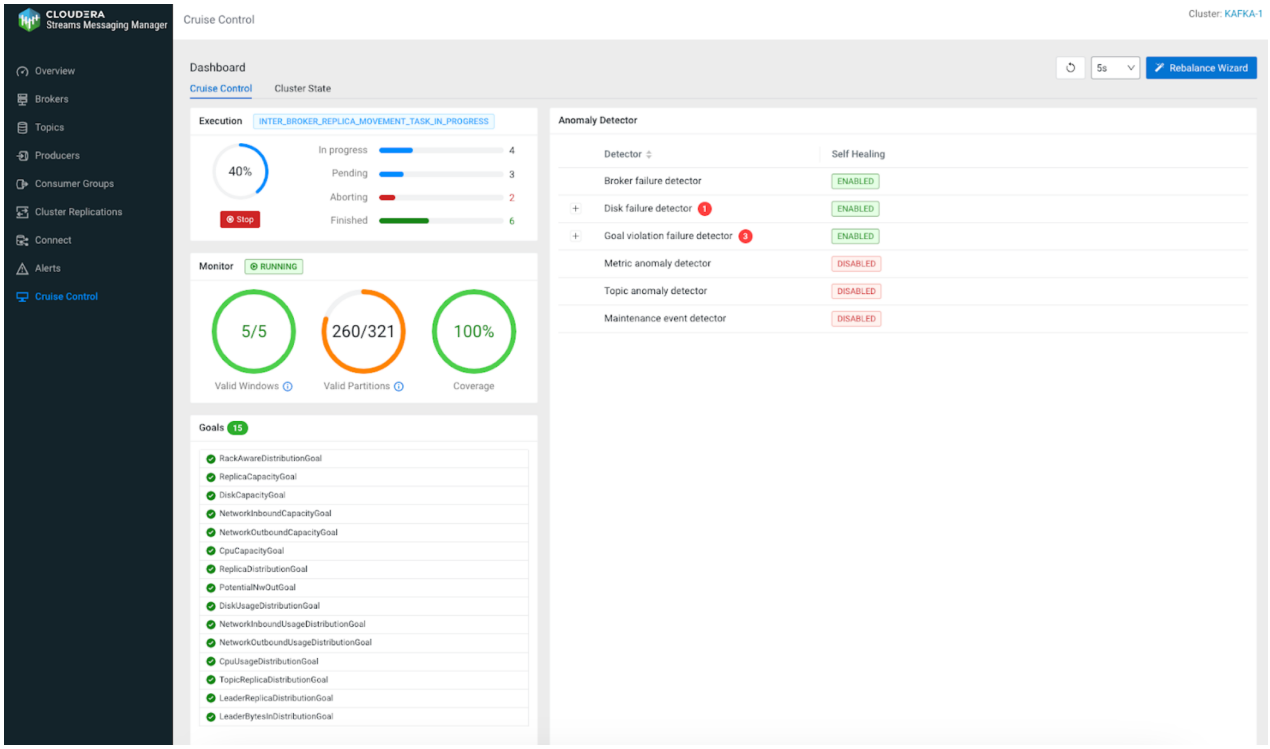
   The **Review** shows the specified configurations as the list of goals for the rebalance request. At this step, you still have the opportunity to change the rebalancing setup.



6. Click Rebalance to start the rebalancing process if the configurations are correct.
7. Click Rebalance again to confirm the rebalancing process.

Results

The rebalancing process is triggered based on the configurations provided. You are redirected to the **Cruise Control Dashboard**. The execution of the rebalancing process can be stopped anytime using the Stop button on the **Execution** tile, if you have Admin level permissions.


CLOUDERA
Streams Messaging Manager
Cruise Control
Cluster: KAFKA-1
Overview
Brokers
Topics
Producers
Consumer Groups
Cluster Replications
Connect
Alerts
Cruise Control
Dashboard
Cruise Control
Cluster State
5s
Rebalance Wizard
Execution
INTER_BROKER_REPLICA_MOVEMENT_TASK_IN_PROGRESS
40%
Stop
In progress 4
Pending 3
Aborting 2
Finished 6
Monitor
RUNNING
5/5
Valid Windows
260/321
Valid Partitions
100%
Coverage
Goals 15
RackAwareDistributionGoal
ReplicaCapacityGoal
DiskCapacityGoal
NetworkInboundCapacityGoal
NetworkOutboundCapacityGoal
CpuCapacityGoal
ReplicaDistributionGoal
PotentialNwOutGoal
DiskUsageDistributionGoal
NetworkInboundUsageDistributionGoal
NetworkOutboundUsageDistributionGoal
CpuUsageDistributionGoal
TopicReplicaDistributionGoal
LeaderReplicaDistributionGoal
LeaderBytesInDistributionGoal
Anomaly Detector
Detector
Self Healing
Broker failure detector ENABLED
Disk failure detector 1 ENABLED
Goal violation failure detector 3 ENABLED
Metric anomaly detector DISABLED
Topic anomaly detector DISABLED
Maintenance event detector DISABLED