

Scaling Streams Messaging Clusters

Date published: 2019-12-16

Date modified: 2021-10-25



Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Scaling Kafka brokers.....	4
Scaling up Kafka brokers.....	4
Scaling down Kafka brokers.....	5
Troubleshooting.....	6
The downscale operation fails with decommission failed.....	6

Scaling Kafka brokers

The Kafka brokers provisioned with the Light and Heavy duty versions of the Streams Messaging cluster definitions can be scaled. This is done by adding or removing nodes from the host groups containing Kafka brokers. During a scaling operation Cruise Control automatically rebalances partitions on the cluster. Before scaling Kafka brokers, Cloudera recommends that you review these notes regarding scaling operations.

Version limitations

Scaling brokers is only supported on:

- Clusters newly provisioned with Cloudera Runtime 7.2.12 or higher.
- Clusters upgraded from Cloudera Runtime 7.2.12 or higher to a newer version.

Not all Kafka host groups are scalable

Streams Messaging clusters running Cloudera Runtime 7.2.12 or higher have two host groups of Kafka broker nodes. These are the **Core_broker** and **Broker** host groups. During an upscale or downscale operation, new broker nodes are added to or removed from the **Broker** host group. The **Core_broker** group contains a core set of brokers and is not scalable. For more information regarding the Streams Messaging cluster templates, see *Streams Messaging cluster layout*.

Downscale operations ensure that the cluster remains healthy

During a downscale operation there are a number of automated safety checks to ensure that the cluster stays healthy at all times and no data is lost when brokers are decommissioned. For example:

- Downscale operations are only carried out if sufficient resources are available. Otherwise they fail.
- The partitions of a decommissioned broker are automatically moved to other brokers.
- A broker is only decommissioned and removed from the broker's host group if there is no load on that broker.

Related Information

[Streams Messaging cluster layout](#)

Scaling up Kafka brokers

Learn how you can upscale Kafka brokers in a Streams Messaging cluster.

About this task

The Kafka brokers in a Streams Messaging cluster can be upscaled. During an upscale operation, broker nodes are added to the **Broker** host group. Newly commissioned brokers are automatically detected by Cruise Control, a rebalance is triggered once they become available, and data is moved to the new brokers.

Before you begin

- Ensure that you have reviewed [Scaling Kafka broker nodes](#).
- Ensure that the cluster, its hosts, and all its services are healthy.
- Ensure that the Kafka brokers are commissioned and running.
- Ensure that the Cruise Control service is running.

Procedure

1. Log in to the CDP web interface.
2. Go to Management ConsoleData Hub Clusters.

3. Find and select the Data Hub cluster you want to scale.
4. Click Actions Resize.
5. Select the **Broker** host group.
6. Set the desired size of the host group.
You can enter a number or use the plus and minus buttons to adjust the number of nodes.
7. Click Resize.
8. Wait until the new nodes become available.
9. Access the Cloudera Manager instance managing the cluster.
10. Restart all services that have stale configurations.

Results

The cluster is upscaled. The number of Kafka broker nodes available in the **Broker** host group is equal to the configured number of nodes. Data is automatically moved to the new brokers by Cruise Control.

Scaling down Kafka brokers

Learn how you can downscale Kafka brokers in a Streams Messaging cluster.

About this task

The Kafka brokers in a Streams Messaging cluster can be downscaled. During a downscale operation, broker nodes are decommissioned from the **Broker** host group. Data and partitions of the decommissioned nodes are automatically moved to other nodes.

There are two methods that can be used to downscale a cluster. You can either have Data Hub choose which nodes to decommission or manually specify which nodes should be decommissioned. The following steps walk you through both methods.

Before you begin

- Ensure that you have reviewed [Scaling Kafka broker nodes](#).
- Ensure that the cluster, its hosts, and all its services are healthy.
- Ensure that the Kafka brokers are commissioned and running.
- Ensure that the Cruise Control service is running.
- Using the Cruise Control REST API, verify that Cruise Control is reporting that all partitions are healthy.



Important: An access level (admin, user or viewer) must be set for the user calling the API endpoint in Cruise Control. For more information, see [Cruise Control REST API endpoints](#).

This can be done by calling the state API endpoint and ensuring the following in the response:

- numValidPartitions is equal to numTotalPartitions
- monitoringCoveragePct is 100.0

API call example:

```
https://[***MY-DATA-HUB-CLUSTER.COM**]/cdp-proxy-api/cruise-control/kafkacruisecontrol/state?json=true
```

You can find the API base URL on the Endpoints tab of the **Cluster Details** page. Management Console Data Hub Clusters [***YOUR CLUSTER***] Endpoints.

- Do not restart Kafka or Cruise Control during a downscale operation.
- Do not create new partitions during a downscale operation.

Procedure

1. Log in to the CDP web interface.
2. Go to Management ConsoleData Hub Clusters.
3. Find and select the Data Hub cluster you want to scale.
4. Decommission nodes.

You can either let Data Hub choose which nodes to decommission or you can select the nodes that you want to have decommissioned.

For Let Data Hub choose which nodes to decommission

- a. Click ActionsResize.
- b. Select the Broker host group.
- c. Set the desired size of the host group.

You can enter a number or use the plus and minus buttons to adjust the number of nodes.

- d. Click Resize.



Note: Data Hub chooses which nodes to decommission based on how suitable it is to remove a specific node. As a result, the nodes selected by Data Hub might not be the nodes that have the highest node ID.

For Manually specify which nodes to decommission

- a. Go to Hardware.
- b. Scroll down to the **Broker** host group.
- c. Select the nodes you want to remove.
- d. Click the Delete (trash bin) icon.
- e. Click Yes to confirm deletion.

5. Access the Cloudera Manager instance managing the cluster.
6. Restart all services that have stale configurations.

Results

The cluster is downscaled. The number of Kafka broker nodes available in the **Broker** host group is equal to the configured number of nodes. Partitions are automatically moved from the decommissioned brokers. Once no load is left on the broker, the broker is fully decommissioned and removed from the **Broker** host group.

Troubleshooting

Learn about the common issues that you might encounter when scaling Kafka broker nodes in Streams Messaging Clusters.

The downscale operation fails with decommission failed

Condition

The downscale operation fails with the Decommission failed error.

Cause

This issue can be caused by either of the following:

- The broker is not available.
- Cruise Control is reporting unhealthy partitions

- A manually initiated rebalance is running.
- An insufficient amount of brokers would remain after the downscale operation to host all replicas.

This only affects deployments where there is a topic with at least four-way replication set up.

Solution

Procedure

1. Back up the data disks of the failed node.
This can be done by creating a backup of `/hadoopfs/fs[***DATA VOL NUMBER***/kafka`.
2. Access the Cloudera Manager instance managing the nodes that you want to decommission.
3. Select the Kafka service and go to Instances.
4. Select the Kafka broker you want to decommission.
5. Click Actions for Selected Recommission and start.
6. Wait until the process is finished and the cluster is in a healthy (green) state.
7. In Management Console, navigate to the Data Hub cluster containing the node.
8. Go to Hardware.
9. Select the node and click the Repair (wrench) icon next to the host group.



Important: Do not select the Repair action from the upper right hand Actions dropdown. This action attempts to repair all unhealthy nodes in the selected host group. Under certain circumstances, the node that you want to repair might not be recognized as unhealthy. As a result, the repair operation might not be carried out on the affected node. Ensure that you initiate a repair on the affected nodes using the Repair (wrench) icon.