Cloudera Runtime 7.3.2

Configuring Cruise Control

Date published: 2019-08-22
Date modified: 2026-03-31

https://docs.cloudera.com/

Legal Notice

Contents

Setting capacity estimations and goals..... 4
Configuring capacity estimations..... 4
Configuring goals..... 5
Example of Cruise Control goal configuration..... 6
Multi-level rack-aware distribution goal..... 7

Configuring Metrics Reporter in Cruise Control..... 7

Adding self-healing goals to Cruise Control in Cloudera Manager..... 8

Setting capacity estimations and goals

Cruise Control rebalancing works using capacity estimations and goals. You need to configure the capacity estimates based on your resources, and set the goals for Cruise Control to achieve the Kafka partition rebalancing that meets your requirements.

When configuring Cruise Control, you need to make sure that the Kafka topics and partitions, the capacity estimates, and the proper goals are provided so the rebalancing process works as expected.

You can find the capacity estimate and goal configurations at the following location in Cloudera Manager:

1. Access Cloudera Manager for the Cruise Control configurations.
   a. Navigate to Cloudera Management Console Environments , and select the environment where you have created your cluster.
   b. Select Cloudera Manager from the services.
   c. Select Clusters Cruise Control .
2. Click Configuration.
3. Select Main from the Filters.

Configuring capacity estimations

The values for capacity estimation needs to be provided based on your available resources for CPU and network. Beside the capacity estimation, you also need to provide information about the broker and partition metrics. You can set the capacity estimations and Kafka properties in Cloudera Manager.

For the rebalancing, you need to provide the capacity values of your resources. These values are used for specifying the rebalancing criteria for your deployment. The following capacity values must be set:

Capacity	Description
capacity.default.cpu	100 by default
capacity.default.network-in	Given by the internet provider
capacity.default.network-out	

Note: For the capacity estimates, the disk capacity value is also needed. However, Cruise Control automatically retrieves the disk capacity value from the kafka_log_directory_disk_total_space Kafka metric.

The optimizers in Cruise Control use the network incoming and outgoing capacities to define a boundary for optimization. The capacity estimates are generated and read by Cruise Control. A capacity.json file is generated when Cruise Control is started. When a new broker is added, Cruise Control uses the default broker capacity values. However, in case disk related goals are used, Cruise Control must be restarted to load the actual disk capacity metrics of the new broker.

The following table lists all the configurations that are needed to configure Cruise Control specifically to your environment:

Note: The security settings are not listed in the table below.

Configuration	Description
num.metric.fetchers	Parallel threads for fetching metrics from the Cloudera Manager database
partition.metric.sample.store.topic	Storing Cruise Control metrics
broker.metric.sample.store.topic	Storing Cruise Control metircs

Configuration	Description
partition.metrics.window.ms	Time window size for partition metrics
broker.metrics.window.ms	Time window size for broker metrics
num.partition.metrics.windows	Number of stored partition windows
num.broker.metrics.windows	Number of stored broker windows

Configuring goals

After setting the capacity estimates, you can specify which goals need to be used for the rebalancing process in Cloudera Manager. The provided goals are used for the optimization proposal of your Kafka cluster.

Procedure

1. Access Cloudera Manager for the Cruise Control configurations.
   a) Navigate to Cloudera Management Console Environments , and select the environment where you have created your cluster.
   b) Select the Streams Messaging cluster from the list of Cloudera Data Hub clusters.
   c) Select Cloudera Manager from the services.
   d) Select Clusters Cruise Control .
   e) Click Configuration.
2. Search for goals using the search bar.
   The list of goals are displayed based on the goal sets.
3. Add goals using the property name to the Default, Supported, Hard, Self-healing and Anomaly detection lists based on your requirements, and click Save Changes.

   The following table lists the goals that can be used:

Goal	Property name	Description
`RackAwareDistributionGoal`	com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareDistributionGoal	As long as replicas of each partition can achieve a perfectly even distribution across the racks, this goal lets placement of multiple replicas of a partition into a single rack.
`ReplicaCapacityGoal`	com.linkedin.kafka.cruisecontrol.analyzer.goals.ReplicaCapacityGoal	Attempt to make all the brokers in a cluster to have less than a given number of replicas.
`CapacityGoals`	com.linkedin.kafka.cruisecontrol.analyzer.goals.DiskCapacityGoal	Goals that ensure the broker resource utilization is below a given threshold for the corresponding resource.
	com.linkedin.kafka.cruisecontrol.analyzer.goals.NetworkInboundCapacityGoal	
	com.linkedin.kafka.cruisecontrol.analyzer.goals.NetworkOutboundCapacityGoal	
	com.linkedin.kafka.cruisecontrol.analyzer.goals.CpuCapacityGoal	
`ReplicaDistributionGoal`	com.linkedin.kafka.cruisecontrol.analyzer.goals.ReplicaDistributionGoal	Attempt to make all the brokers in a cluster to have a similar number of replicas.
`PotentialNwOutGoal`	com.linkedin.kafka.cruisecontrol.analyzer.goals.PotentialNwOutGoal	A goal that ensures the potential network output (when all the replicas become leaders) on each of the brokers do not exceed the broker's network outbound bandwidth capacity.

Goal	Property name	Description
ResourceDistributionGoals	com.linkedin.kafka.cruisecontrol.analyzer.goals.DiskUsageDistributionGoal	Attempt to make the resource utilization variance among all the brokers are within a certain range. This goal does not do anything if the cluster is in a low utilization mode (when all the resource utilization of each broker is below a configured percentage.) This is not a single goal, but consists of the following separate goals for each of the resources.
	com.linkedin.kafka.cruisecontrol.analyzer.goals.NetworkInboundUsageDistributionGoal	
	com.linkedin.kafka.cruisecontrol.analyzer.goals.NetworkOutboundUsageDistributionGoal	
	com.linkedin.kafka.cruisecontrol.analyzer.goals.CpuUsageDistributionGoal	
TopicReplicaDistributionGoal	com.linkedin.kafka.cruisecontrol.analyzer.goals.TopicReplicaDistributionGoal	Attempt to make the replicas of the same topic evenly distributed across the entire cluster.
LeaderReplicaDistributionGoal	com.linkedin.kafka.cruisecontrol.analyzer.goals.LeaderReplicaDistributionGoal	Attempt to make all the brokers in a cluster to have the similar number of leader replicas.
LeaderBytesInDistributionGoal	com.linkedin.kafka.cruisecontrol.analyzer.goals.LeaderBytesInDistributionGoal	Attempt to make the leader bytes in rate on each host to be balanced.
PreferredLeaderElectionGoal	com.linkedin.kafka.cruisecontrol.analyzer.goals.PreferredLeaderElectionGoal	Attempt to make the first replica in the replica list leader replica of the partition for all topic partitions.
MinTopicLeadersPerBrokerGoal	com.linkedin.kafka.cruisecontrol.analyzer.goals.MinTopicLeadersPerBrokerGoal	Ensures that each alive broker has at least a certain number of leader replica of each topic in a configured set of topics
KafkaAssignerGoals[1]	com.linkedin.kafka.cruisecontrol.analyzer.kafkaassigner.KafkaAssignerDiskUsageDistributionGoal	A goal that ensures all the replicas of each partition are assigned in a rack aware manner.
	com.linkedin.kafka.cruisecontrol.analyzer.kafkaassigner.KafkaAssignerEvenRackAwareGoal	Attempt to make all the brokers in a cluster to have the similar number of replicas

Example of Cruise Control goal configuration

By default, Cruise Control is configured with a set of Default, Supported, Hard, Self-healing and Anomaly detection goals in Cloudera Manager. The default configurations can be changed based on what you would like to achieve with the rebalancing.

The following example details how to configure Cruise Control to achieve the following:

- Find dead/failed brokers and create an anomaly to remove load from them (self.healing.broker.failure.enabled)
- Move load back to the brokers when the brokers are available again (self.healing.goal.violation.enabled and added goals)
- Prevent too frequent rebalances to reduce cluster costs (incremented thresholds, reduced self.healing.goals set)
- Have an always balanced cluster from the replicas and leader replicas point of view
- Not enable every type of self-healing methods if it is not required (only two type of self-healing is enabled)

Configurations that need to be added to the Cruise Control Server Advanced Configuration Snippet (Safety Valve) for cruisecontrol.properties property:

- self.healing.goal.violation.enabled=true
- self.healing.broker.failure.enabled=true
- self.healing.exclude.recently.removed.brokers=false

Configurations that need to be set (and available explicitly among properties):

- anomaly.notifier.class=com.linkedin.kafka.cruisecontrol.detector.notifier.SelfHealingNotifier

[1] These goals are used to make Cruise Control behave like a Kafka assigner tool. These goals will be picked up if kafka_assigner parameter is set to true in the corresponding request (for example, with the rebalance request as shown in the Cruise Control documentation).

- replica.count.balance.threshold=1.25
- leader.replica.count.balance.threshold=1.25

Goals that need to be added to Hard goals:

- com.linkedin.kafka.cruisecontrol.analyzer.goals.ReplicaDistributionGoal
- com.linkedin.kafka.cruisecontrol.analyzer.goals.LeaderReplicaDistributionGoal

Goals that need to be added to Self-healing goals:

- com.linkedin.kafka.cruisecontrol.analyzer.goals.ReplicaDistributionGoal
- com.linkedin.kafka.cruisecontrol.analyzer.goals.LeaderReplicaDistributionGoal

Goals that need to be added to Anomaly detection goals:

- com.linkedin.kafka.cruisecontrol.analyzer.goals.ReplicaDistributionGoal
- com.linkedin.kafka.cruisecontrol.analyzer.goals.LeaderReplicaDistributionGoal

Other configurations can remain as set by default.

Multi-level rack-aware distribution goal

You can use the MultiLevelRackAwareDistributionGoal to ensure rack awareness on a higher level than for the standard rack aware goal for Kafka clusters using Cruise Control.

The MultiLevelRackAwareDistributionGoal behaves differently than the default RackAwareGoal or RackAwar eDistributionGoal in Cruise Control. The standard goals have lighter requirements on rack awareness, and always optimize based on the current state of the cluster and with the priority on making all replicas come back online.

This means that in case a network partition failure occurs, and a data center goes offline, a Cruise Control rebalance operation using a standard rack-aware goal ignores the data center that is not working, and moves replicas around as if there were one fewer data center in the cluster. For example, if a Kafka cluster has three data centers and one goes offline, the standard goals are not aware of the existence of the third data center, and act as if only two data centers are used in the cluster.

The MultiLevelRackAwareDistributionGoal acts differently in the following aspects:

- Handles rack IDs as multi-level rack IDs, respecting the hierarchy of racks when distributing replicas
- Keeps track of the whole state of the cluster with caching previous states to make sure that all racks are visible
- Prioritizes multi-level rack awareness guarantees over bringing all replicas back online

In the same failure situation, where one data center is offline out of three, the multi-level rack-aware goal is still aware of the existence of the third data center. This means that the offline replicas are not moved from the third data center if the migration violates the multi-level rack awareness guarantees. The goal allows optimizations to pass even in the presence of offline replicas, which can be configured with cloudera.multi.level.rack.awareness.ensure.no.offlin e.replicas property. If the cloudera.multi.level.rack.awareness.ensure.no.offline.replicas is set to true, the goal causes the rebalance operation to fail if the replicas would stay offline after the optimizations are implemented.

Configuring Metrics Reporter in Cruise Control

You can choose between using the default Cruise Control Metrics Reporter or using the Cloudera Manager Metrics Reporter for fetching metrics in Cruise Control. Cloudera recommends using the Cloudera Manager solution with light installation, and the default solution with heavy installations of Kafka deployments.

Procedure

1. Access Cloudera Manager for the Cruise Control configurations.
   a) Navigate to Cloudera Management Console Environments , and select the environment where you have created your cluster.
   b) Select the Streams Messaging cluster from the list of Cloudera Data Hub clusters.
   c) Select Cloudera Manager from the services.
   d) Select Clusters Cruise Control .
   e) Click Configuration.
2. Search for Metrics Reporter.
3. Select Cloudera Manager metrics reporter or Cruise Control metrics reporter based on your requirements.
4. Click Save changes.
5. Click on Action Restart next to the Cruise Control service name to restart Cruise Control.

Adding self-healing goals to Cruise Control in Cloudera Manager

As self-healing is enabled by default for Cruise Control, you only need to specify the actions Cruise Control should take when detecting anomaly types by providing self-healing goals in Cloudera Manager.

Procedure

1. Access Cloudera Manager for the Cruise Control configurations.
   a) Navigate to Cloudera Management Console Environments , and select the environment where you have created your cluster.
   b) Select the Streams Messaging cluster from the list of Cloudera Data Hub clusters.
   c) Select Cloudera Manager from the services.
   d) Select Clusters Cruise Control .
   e) Click Configuration.
2. Search for Self-Healing Goals.
3. Add the required self-healing goals to the corresponding field.
4. Click Save changes.
5. Click on Action > Restart next to the Cruise Control service name to restart Cruise Control.