Cloudera Runtime 7.1.5

Cruise Control

Date published: 2020-04-24
Date modified: 2020-12-02

https://docs.cloudera.com/

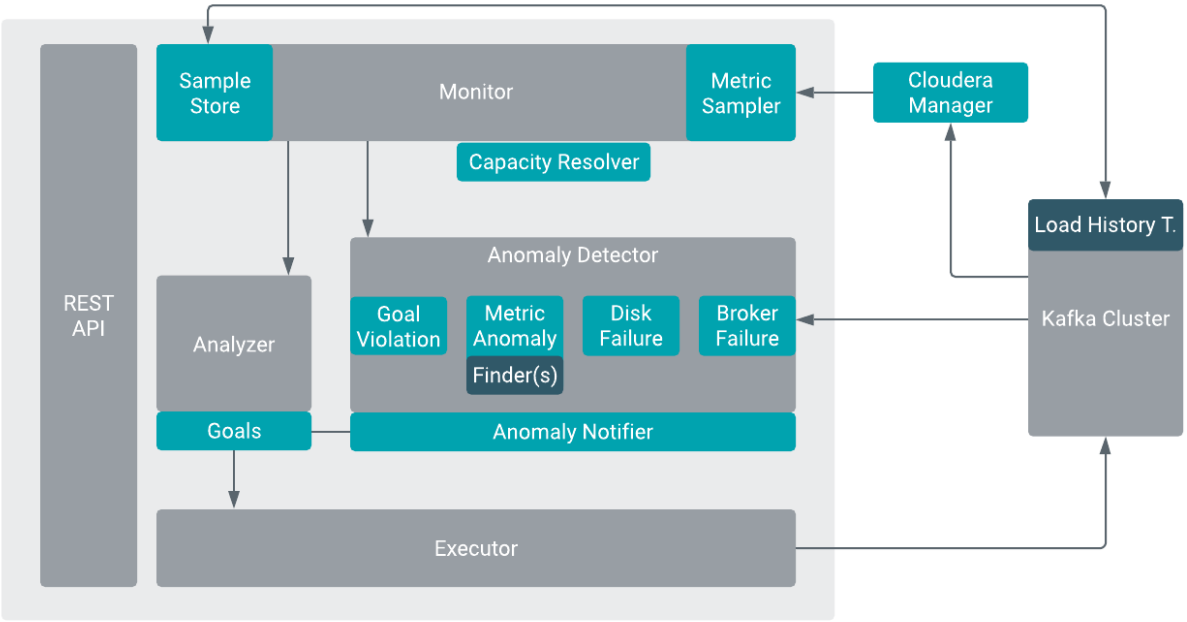
Legal Notice

Contents

Kafka cluster load balancing using Cruise Control

You can use Cruise Control as a load balancing component in large Kafka installations to automatically balance the partitions based on specific conditions for your deployment. The elements in the Cruise Control architecture are responsible for different parts of the rebalancing process that uses Kafka metrics and optimization goals.

The following illustration shows the architecture of Cruise Control.



Load Monitor

Generates a cluster workload model based on standard Kafka metrics and resource metrics to utilize disk, CPU, bytes-in rate and bytes-out rate. Feeds the cluster model into Anomaly Detector and Analyzer.

Analyzer

Generates optimization proposals based on optimization goals provided by the user, and cluster workload model from Load Monitor. Hard goals and soft goals can be set. Hard goals must be fulfilled, while soft goals can be left unfulfilled if hard goals are reached. The optimization fails if the hard goal is violated by optimization results.

Anomaly Detector

Responsible for detecting the following anomalies:

Anomaly	Cause	Result
Broker failure	Non-empty broker crashes or leaves a cluster.	Cruise Control fixes the cluster by removing the failed brokers.
Goal violations	Optimization is violated.	Cruise Control automatically analyzes the workload and executes the optimization proposals, if self-healing is enabled.
Disk failure	Non-empty disk dies.	Cruise Control moves all the offline replicas to healthy brokers, if self-healing is enabled.

Anomaly	Cause	Result
Metric anomaly	Anomalies in the collected metrics.	May first demote, and if the anomaly persists, remove slow brokers depending on the self.healing.slow.brokers.removal.enabled configuration.

Executor

Carries out the optimization proposals and it can be safely interrupted when executing proposals. The executions are always resource-aware processes.

How Cruise Control retrieves metrics

Cruise Control creates metric samples using the retrieved raw metrics from Kafka. The metric samples are used to set up the cluster workload model for the Load Monitor. When deploying Cruise Control in a CDP environment, Cloudera Manager executes the process of retrieving the metrics from Kafka to Cruise Control.

In Load Monitor, the Metric Fetcher Manager is responsible for coordinating all the sampling tasks: the Metric Sampling Task, the Bootstrap Task and the Linear Model Training Task.

Each sampling task is carried out by a configured number of Metric Fetcher threads. Each Metric Fetcher thread uses a pluggable Metric Sampler to fetch samples. Each Metric Fetcher is assigned with a few partitions in the cluster to get the samples. The metric samples are organized by the Metric Sample Aggregator that puts each metric sample into a workload snapshot according to the timestamp of a metric sample.

The cluster workload model is the primary output of the Load Monitor. The cluster workload model reflects the current replica assignment of the cluster and provides interfaces to move partitions or replicas. These interfaces are used by the Analyzer to generate optimization solutions.

The Sample Store stores the metric and training samples for future use.

With the metric sampler, you can deploy Cruise Control to various environments and work with the existing metric system.

When you use Cruise Control in the Cloudera environment, HttpMetricsReporter reports metrics to the Cloudera Manager time-series database. As a result, the Kafka metrics can be read using the Cloudera Manager.