cedrusdata

Описание функциональных характеристик CedrusData

ООО «Кверифай Лабс»

ОГРН 1217800163790

ИНН 7811766769

КПП 781101001

Введение

CedrusData это высокопроизводительная распределенная платформа для сквозного анализа всех данных предприятия в облаке и on-premise через единую точку доступа с SQL интерфейсом.

CedrusData основана на распределенном SQL-движке Trino (https://trino.io/) и включает дополнительный функционал управления и мониторинга (в том числе в облачных инфраструктурах), улучшения производительности, профессиональную документацию и поддержку.

Данный документ содержит описание функциональных характеристик CedrusData.

1. Сценарии использования

Для выполнения SQL-запросов CedrusData подключается ко внешним источникам данных, после чего производит чтение, запись и объединение информации из одного или нескольких источников.

Типичными сценариями применения CedrusData являются:

- Работа с озерами данных. Например, пользователь может подключиться к озеру данных, которое хранит данные в файлах в распределенной файловой системе (например, HDFS) или облачном хранилище (например, Yandex Object Storage), и запустить SQL-запрос к этим данным, или же осуществить запись новых данных.
- Объединение данных из разных источников (виртуализация). Это позволяет пользователям работать со всеми данными организации через единый SQL-интерфейс. Например, пользователь может запустить SQL-запрос, который прочитает данные из корпоративного хранилище данных (например, Greenplum), объединит их с данными из озера данных (например, Hive Metastore), обогатит их справочной информацией из операционной СУБД (например, Postgres), после чего запишет результат в витрину в другой системе (например, ClickHouse).

2. Работа с данными

2.1. Источники данных

CedrusData поддерживает следующие популярные источники данных:

- Озера данных (data lakes) под управлением Hive Metastore и Apache Iceberg.
- Хранилища данных: Greenplum, ClickHouse, Apache Druid, Apache Pinot.
- Реляционные СУБД: Postgres, MySQL, Oracle, SQL Server, MariaDB.
- Нереляционные источники: Cassandra, MongoDB, Redis, Kafka.

2.2. Операции над данными

CedrusData поддерживает следующие типы SQL операций для работы с данными:

- Операции SELECT: чтение данных из одного или нескольких источников с использованием стандартного синтаксиса SQL.

- DML-операции: вставка (`INSERT`), обновление (`UPDATE`) или удаление (`DELETE`) данных.
- DDL-операции: создание (`CREATE`), изменение (`ALTER`) и удаление (`DROP`) объектов в источниках (`SCHEMA`, `TABLE`, `VIEW`, `FUNCTION`, `ROLE`).

2.3. Запуск SQL-запросов

Для запуска SQL-запросов пользователи должны подключиться к кластеру CedrusData одним из двух способов:

- С помощью утилиты командной строки Trino CLI - подходит для изучения и тестирования функционала продукта.
- С помощью JDBC драйвера - наиболее распространенный сценарий, который позволяет подключаться к CedrusData любым программам и утилитам, которые поддерживают технологию JDBC, включая средства визуализации (например, Apache Superset), технологии работы с данными (например, Apache Spark, Pandas), среды разработки (например, Jupyter Notebook, DBeaver), а также собственные приложения пользователя.

3. Кластер CedrusData

Основной задачей CedrusData является выполнение SQL-запросов к большим объемам данных, хранящихся во внешних источниках. Для этого CedrusData предоставляет возможность запуска одного или нескольких процессов, называемых **узлами**, которые работают над решением общих задач.

Узлы CedrusData могут выполнять разные задачи в зависимости от назначенной им роли:

- **Coordinator** - принимает SQL-запросы от пользователя, координирует выполнение SQL-запросов в кластере, отдает результаты SQL-запросов пользователю. Кластер CedrusData может иметь один или несколько coordinator узлов.
- **Worker** - выполняет отдельные части SQL-запроса на основе команд от coordinator и пересылает результаты другим узлам. Кластер CedrusData может иметь один или несколько worker узлов.

Совокупность узлов называется **кластером**. Для начала работы с данными, пользователь должен запустить кластер CedrusData, который содержит как минимум один coordinator узел и как минимум один worker узел.

По мере изменения профиля нагрузки пользователь может добавлять или удалять узлы из кластера, тем самым обеспечивая гибкое масштабирование вычислительной мощности кластера. Например, в периоды пиковых нагрузок, кластер CedrusData может содержать десятки узлов, в том время как в выходные дни размер кластера может быть уменьшен до минимальных размеров.

4. Развертывание CedrusData

CedrusData реализует shared storage (serverless) архитектуру, в которой кластер CedrusData осуществляет только обработку данных (чтение, запись), в то время как за хранение данных отвечают источники. Благодаря этому кластер CedrusData может быть легко развернут как на конкретных мощностях (on-premise), так и в облаке, без необходимости копирования и перемещения существующих данных.

При использовании CedrusData on-premise пользователь может запускать узлы CedrusData как вручную из дистрибутива (архива), так с использованием Docker-образа CedrusData.

При развертывании CedrusData в облаке пользователь может использовать Docker-образ CedrusData, а также интегрировать его с другими облачными технологиями, такими как Kubernetes, Terraform, Helm.