Cloudera Operational Database on High Performance Storage

Date published: 2020-08-14
Date modified: 2025-03-17

https://docs.cloudera.com/

Legal Notice

Contents

Overview

Cloudera Operational Database supports both high performance storage and Amazon S3 or Azure Data Lake Gen2 storage through the abfs connector as a storage layer. While the default option is cloud storage, Cloudera Operational Database also provides you an option to choose high performance storage.

You can use high performance storage either in an on-premises or a cloud environment to maintain low-latency and high-throughput for your application. If your applications are latency sensitive, Cloudera recommends that you use high performance storage for your Cloudera Operational Database databases. The average read and write latency for Cloudera Operational Database on a high performance storage is 500% better than the Cloudera Operational Database that uses S3 storage for all the workloads.

Note: Cloudera Operational Database on GCP environment supports only high performance storage.

HDFS volume type

Hadoop Distributed File System (HDFS) is a Java-based file system for storing large volumes of data. Designed to span large clusters of commodity servers, HDFS provides scalable and reliable data storage.

An HDFS cluster contains the following main components: a NameNode and DataNodes. The NameNode manages the cluster metadata that includes file and directory structures, permissions, modifications, and disk space quotas. The file content is split into multiple data blocks, with each block replicated at multiple DataNodes.

For more information, see *HDFS Overview*.

While creating a Cloudera Operational Database database, you have the following two options for storing your data when you select HDFS as the storage type.

- HDD (Hard Disk Drives): HDD is a cost-efficient option if you want to access the data frequently. This is a preferred storage option for applications that are not very sensitive to latency.
- SSD (Solid State Drives): SSDs provide more storage volume, speed, and efficiency; however, this storage option could be expensive compared to HDD.

For more information on the difference between SSD and HDD, see Amazon documentation.

Worker nodes

Worker nodes store the data on your Cloudera Operational Database cluster. The general responsibilities of the worker nodes on your Cloudera Operational Database cluster include processing the data stored in the cluster and managing the network traffic. The worker nodes also ensure smooth operations between the applications across multiple Cloudera Operational Database clusters.

When you create a Cloudera Operational Database cluster, the minimum and maximum number of worker nodes vary for different storage types.

- Micro duty: Minimum node count: 1. Maximum node count: 5.
- Light duty: Minimum node count: 3. Maximum node count: 100.
- Heavy duty: Minimum node count: 3. Maximum node count: 800.

Related Information

HDFS Overview

Storage types

Learn about the supported storage types of the Cloudera Operational Database.

Cloudera Operational Database supports the following storage types when creating a database. These storage types can be specified as part of the --storage-type parameter.

Syntax: `–storage-type` [*CLOUD_WITH_EPHEMERAL*][*CLOUD*][*HDFS*]

You can deploy the Cloudera Operational Database using three scale types namely HEAVY DUTY, LIGHT DUTY, and MICRO DUTY.

Performance characteristics

The Cloudera Operational Database uses different instance types on the worker nodes based on the selected scale and storage types. The following are the high-level descriptions of each storage type used by the Cloudera Operational Database.

- Cloud Storage with Caching

  --storage-type CLOUD_WITH_EPHEMERAL

  - This storage configuration utilizes ephemeral storage alongside the block storage provided by the underlying cloud provider. In this setup, the Cloudera Operational Database prioritizes storing as much data as possible in ephemeral storage to ensure faster access. If certain data is unavailable in ephemeral storage, it is retrieved from the slower block storage. The database achieves optimal performance when all data resides in ephemeral storage. For more information, see *Ephemeral Storage*.
  - In this configuration, each worker node is equipped with 1.6TB of ephemeral storage. To achieve optimal performance, the cluster size must be planned such that all data fits within the available ephemeral storage.
  - In this configuration, an initial cost is incurred to warm up the ephemeral cache, during which all data is cached into the ephemeral storage at cluster startup. This cache warming process ensures that the cluster operates at peak efficiency once the cache is fully populated.
  - Cache warming also mitigates the impact of AWS S3 throttling, which limits the number of calls allowed per second to the cloud storage.
  - Once the cache is fully warmed up, this configuration delivers twice the performance of HDFS while achieving a lower total cost of ownership (TCO).
  - This configuration is particularly beneficial for use cases that require strong read performance, albeit at a slightly higher cost. A common scenario is using this configuration for a production cluster that handles both read and write workloads while still achieving a 40% cost reduction compared to an HDFS-based cluster.
- Cloud Storage

  --storage-type CLOUD

  - This storage type relies solely on the block storage provided by the underlying cloud provider. It is more suitable for scenarios where read performance is not a priority.
  - In this configuration, the absence of ephemeral storage means that all read requests are served directly from the slower block storage. As a result, this configuration typically performs slower than other storage types.
  - This configuration is particularly useful for use cases that involve heavy write workloads with minimal read workloads.
  - By eliminating ephemeral storage, you can rely on cloud storage and achieve a 25% cost savings compared to an HDFS-based cluster.
  - This configuration is ideal for use as a disaster recovery (DR) cluster, handling writes from less critical applications.
- HDFS

  --storage-type HDFS

  - The Cloudera Operational Database leverages the Hadoop Distributed File System (HDFS) to store large volumes of data. HDFS offers scalable and reliable storage by utilizing clusters of commodity servers. For more information, see *HDFS Overview*.
  - HDFS relies on costlier EBS-HDD storage and requires three times the actual storage space to accommodate data replication, making it more expensive from a total cost of ownership (TCO) perspective.

For more information, see this article, *Cloudera Operational Database Performance Benchmarking: Comparing HDFS and Cloud Storage*.

Instance types

You can retrieve the list of supported Cloudera Operational Database instance types by using the `list-supported-instance-types` CLI command. This command provides details about the instance types supported for specific combinations of cloud platform, scale type, and storage type. Additionally, you can apply filters based on instance groups and architecture to narrow down the results.

The following is an example of the command.

`cdp opdb list-supported-instance-types --cloud-platform` *AZURE* `--storage-type` *CLOUD_WITH_EPHEMERAL* `--scale-type` *MICRO* `--instance-group` *WORKER* `--architecture` *X86_64*

For more information on instance types, see the following Cloudera Operational Database release notes:

- *Enhancements to the create-database command*
- *A new CLI command to get the list of supported instance types*

Related Information

Ephemeral Storage
HDFS Overview
Cloudera Operational Database (COD) Performance Benchmarking: Comparing HDFS and Cloud Storage
Enhancements to the create-database command
A new CLI command to get the list of supported instance types

Using high performance storage

You can create a Cloudera Operational Database database on high performance storage through CDP CLI beta version.

Procedure

Provide the following API request to create a Cloudera Operational Database database with high performance storage from CDP CLI beta version 0.9.40.

```
$ cdp opdb create-database --environment-name foo --database-name bar --use-hdfs
```

```
--use-hdfs | --no-use-hdfs (boolean) - Controls whether hbase root be deployed on hdfs
```

Related Information

Installing CDP CLI beta

Installing CDP CLI beta

To use Cloudera Operational Database on high performance storage through CDP CLI, you must download and install CDP CLI beta version 0.9.40.

Before you begin

Do not install both the regular and beta CLIs in the same Python environment, as they use the same entry points and will therefore conflict. Either use a separate virtual environment or uninstall the cdpcli first before installing cdpcli-beta.

Procedure

1. Install Python if needed, as described in CDP CLI documentation for your platform.

   Do not install CDP CLI. Instead, proceed to step 2.
2. Run the following to install CDP CLI beta:

   ```
   pip3 install cdpcli-beta
   ```
3. Configure access keys as described in Generating an API access key

Related Information

Using high performance storage

Autoscaling

Cloudera Operational Database with cloud storage type uses high performance storage to store Write-Ahead Log (WALs); however Cloudera Operational Database with high performance storage type uses both WALs and HFiles.

Cloudera Operational Database provides autoscaling for both the storage types. Your database automatically autoscales whenever the high performance storage reaches 60% of its buffer capacity to accommodate the growth in the data.