

Concepts and Use Cases

Date published: 2022-02-17

Date modified: 2023-11-30

CLOUDERA

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

Cloudera Operational Database service in the public cloud.....	4
Cloudera Operational Database quick start.....	5
COD benefits.....	5
Operational Database use cases.....	6
Autoscaling in public cloud environments.....	7
COD integration with RAZ.....	9
Checking RAZ for S3 and ADLS Gen2.....	10
Ephemeral Storage.....	10
HBase Store File Tracking.....	11
Store File Tracking usage.....	11
COD migration.....	11
Restrictions in Store File Tracking.....	12
Multi-AZ deployment on COD.....	12
Advantages of Multi-AZ deployments.....	13
How to enable Multi-AZ deployment on COD.....	13
Troubleshooting Multi-AZ deployments.....	14
Unable to join the cluster automatically.....	14
OMID service fails to recover.....	14
Restrictions in Multi-AZ deployment on COD.....	15

Cloudera Operational Database service in the public cloud

Cloudera Operational Database (COD) is a service that runs on the Cloudera Data Platform (CDP). COD enables you to create a new operational database with a single click and auto-scales based on your workload.

Cloudera Operational Database delivers a real-time, always available, scalable operational database that serves traditional structured data alongside unstructured data within a unified operational and warehousing platform.

Cloudera Operational Database is powered by Apache HBase and Apache Phoenix. In Cloudera Operational Database, you use Apache HBase as a datastore with HDFS and/or S3 providing the storage infrastructure. You have the choice to either develop applications using one of the native Apache HBase API, or you can use Apache Phoenix for data access. Apache Phoenix is a SQL layer that provides a programmatic ANSI SQL interface. It works on top of Apache HBase, and it makes it possible to handle data using standard SQL queries and Apache Phoenix commands. You can use Cloudera Operational Database in the public cloud or on-premises.

You can access COD from the Cloudera CDP console: <https://console.cdp.cloudera.com/>. The COD experience is a tile that you can click on your CDP console home screen.



Related Information

[Cloudera Operational Database overview](#)

[Cloudera Operational Database quick start](#)

[Create a database using COD](#)

[Compiling applications for COD](#)

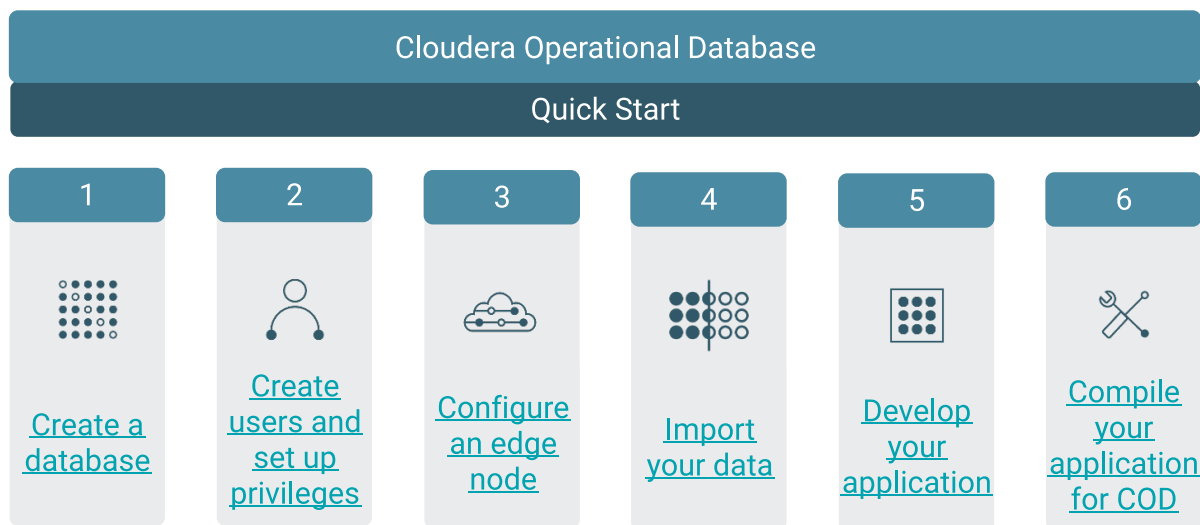
[Cloudera Operational Database](#)

Cloudera Operational Database quick start

In this quickstart, we would like to guide you through the necessary steps to build and manage your applications using Cloudera Operational Database (COD).

Before you start, ensure that you have created a CDP environment. For more information, see *Getting started as an admin*.

The following graphics represents the top tasks hierarchically that you need to perform to kick start with COD.



rect 26, 274, 120, 338 [Creating a database using COD](#)

rect 159, 254, 254, 353 [Managing user access and authorization](#)

rect 296, 255, 389, 349 [COD edge node overview](#)

rect 429, 273, 526, 343 [Importing and restoring data into COD database](#)

rect 563, 266, 664, 352 [COD supported languages](#)

rect 703, 254, 801, 355 [Client connectivity information for compiling against COD](#)

Related Information

[Cloudera Operational Database service in the public cloud](#)

[Getting started as an admin](#)

[AWS environments](#)

[Azure environments](#)

[GCP environments](#)

[Managing user access and authorization](#)

[Compiling applications for COD](#)

[COD benefits](#)

COD benefits

You deploy COD on a public cloud infrastructure that provides you with capabilities and flexibility that your on-premises hardware sometimes cannot offer. Using an existing CDP environment, you can quickly create an operational database with a single click. You can launch a database with the durable and consistent storage technology you may already be familiar with if you have used CDH or HDP, but with none of the legacy complexity.

Data access

Access data stored in COD with the Apache HBase Java API, the Phoenix JDBC driver, or the Phoenix thin client JDBC driver. You can also use other CDP components and experiences to help you with data ingress.

Cloudera Shared Data Experience (SDX) support

Cloudera SDX Data Lake provides COD with common security, auditing, and lineage capabilities used by other CDP experiences. You can pause and resume your COD instances for Research and Development environments to optimize cloud cost.

Auto-scaling

Auto-scaling means that the capabilities of your database can grow to automatically handle increased load against your database, and shrink automatically to reduce your costs without sacrificing availability. COD continuously monitors the services and periodically collects all the metrics. COD meets all the latency requirements by analyzing the collected metrics. COD also fine-tunes itself to improve performance over time.

COD monitors the underlying the services and automatically scales up or down the services to ensure that the latency and RPC metrics are met. COD enables you to create a cluster quickly and has auto-scaling to help you with your different workload requirements. For more information, see [Autoscaling in public cloud environments](#) on page 7.

Auto-healing

COD continuously monitors the clusters and if any failure scenario is encountered, COD automatically heals and repairs them. For example, COD monitors the instances on the cluster in a CDP environment, and if any instance is broken or missing, COD recreates the instances.

Related Information

[Cloudera Operational Database overview](#)

[Cloudera Operational Database quick start](#)

[HDFS overview](#)

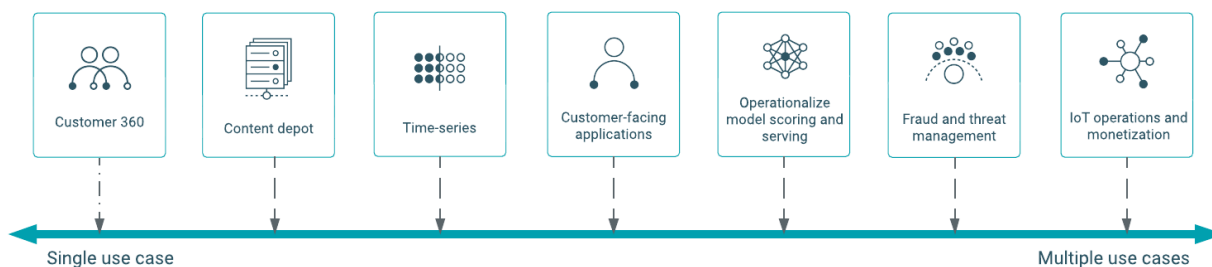
[Introduction to Data Lakes](#)

[Introduction to Data Access](#)

[Shared Data Experience](#)

Operational Database use cases

You can use the Cloudera Operational Database service in online transaction processing (OLTP) use cases and other low-latency and high-throughput application use cases.



Customer 360

- Address the far-reaching effects of the shift in consumer expectations by enabling a holistic view of your business and your customers, from all products, systems, devices, and interaction channels
- Deliver a consistent, personalized, context specific, and relevant experience
- Build churn prediction models to identify at-risk customers and proactively target them with retention programs

Content Depot

- Ensure that all users can access data because of high concurrency and low latency
- Build data-based applications that distribute custom, easy-to-digest information across your organization

Time-series

Include real-time data and analysis into decision points across your organization

Customer-facing applications

- Enable serving analytics on mobile and web applications directly to end-customers
- Use as a key-value store for applications

Operationalize model scoring and serving

- Build and score models on operational data for prevention, optimization, prescription, and prediction
- Increase conversion rate of cross-sell and upsell opportunities
- Predict credit-worthiness and lifetime customer value

Fraud and threat management

Perform fraud model serving and detection

IoT - Operation and monetization

- Leverage IoT to evolve or change your business model and operations for greater efficiencies
- Provide an up-to-the-minute picture of the status of the fleet through real-time monitoring, alerting, and diagnosis
- Deliver economic value by enabling new business models
- Increase conversion rate of cross-sell and upsell opportunities

Operational excellence

- Achieve operational excellence by reducing the total cost of ownership (TCO), improving efficiency, and eliminating threats
- Decrease network downtime using predictive maintenance enabled by active collection and monitoring of network data
- Optimize equipment performance and costs using real-time IoT analytics

Autoscaling in public cloud environments

You can scale up or down the resources allocated to your database automatically by ensuring that an autoscaling event is triggered if pre-defined acceptable latency policy is not met. This ensures predictable database performance.

An autoscaling event is triggered if the rolling average value of the combined metrics reaches a pre-defined threshold value.

Autoscaling does the following to ensure that you have predictable database performance when scaling up or down:

- Gradual increase and decrease to the number of nodes (cluster does not shrink or expand drastically).
- Sufficient cooldown period between scaling up or down to avoid frequent change in cluster size.

You can see the results of scale-up or scale-down on the [Databases Charts](#) page. This page shows you graphs for concurrent clients and RPC latency that provides information about how COD is scaling up or down.

The scaling metric responds to the increase or decrease in the nodes, but there may be cases where increasing or decreasing the number of nodes does not have an effect because of the kind of workload that is running or if there are limited regions for a table.

The autoscaling framework got extended and besides the latency-based metrics, the database can scale based on HDFS disk utilization, latency, average region count per RegionServer, and bucket cache usage for cloud with ephemeral storage type of COD.

For more information, see *Fast autoscaling in COD*.

To check which autoscaling parameters are set for the COD cluster, you can use the `describe-database` CDP CLI command.

```
cdp opdb describe-database --environment-name [CDP ENVIRONMENT NAME] --database-name [DATABASE NAME]
```

To define the autoscaling, you can use the `update-database` CDP CLI command and define the required parameters in the `autoScalingConfig` option. Following is a sample command.

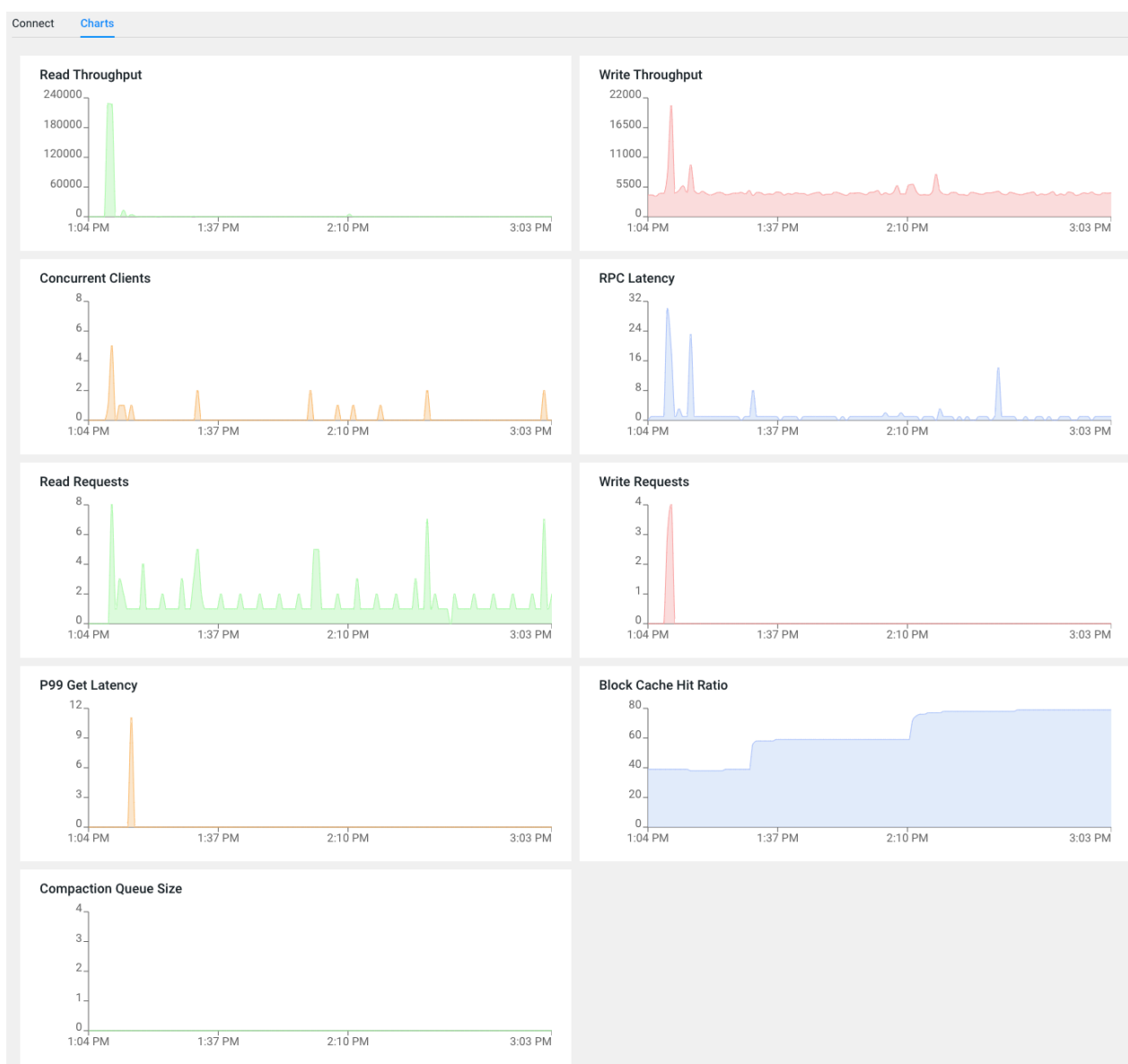
```
cdp opdb update-database --environment-name [CDP ENVIRONMENT NAME] --database-name [DATABASE NAME] --auto-scaling-parameters "minWorkersForDatabase=<min_worker_nodes>,maxWorkersForDatabase= <max_worker_nodes>"
```

The minimum and maximum nodes are limited with the `minWorkersForDatabase` and `maxWorkersForDatabase` autoscaling configurations.



Important: If the `minimumBlockCacheGB` is set to a non-default value and if the number of worker nodes required to support this amount of cache exceeds the `maxWorkersForDatabase` configured while creating the COD cluster, the `maxWorkersForDatabase` is automatically adjusted to 1.5 times the required number of worker nodes.

For more information, see *describe-database* CDP CLI command.



Related Information

[Monitor COD](#)

[Fast autoscaling in COD](#)

[CDP CLI command](#)

[Ephemeral Storage](#)

COD integration with RAZ

CDP Public Cloud defaults to using cloud storage, which introduces new challenges around managing data access across teams and individual users. The Ranger Authorization Service (RAZ) addresses these challenges, enabling users to have the equivalent fine-grained and audit capabilities in Apache Ranger they used with HDFS files in an on-prem or IaaS deployment.

The Ranger Authorization Service (RAZ) is a fine grained authorization service for cloud storage. As a regular individual user or as an HBase user, you can limit the authorization levels in the cloud storage to a directory level.

COD supports RAZ integration from the Runtime version 7.2.11.0. You can grant fine-grained access to directories. COD integration with RAZ is supported for AWS and Azure.



Note: COD integration with RAZ is not supported for GCP.

Checking RAZ for S3 and ADLS Gen2

To verify if RAZ is enabled for S3 and ADLS Gen2, perform the following steps:

1. Select the environments summary page.
2. Ensure that the Ranger Authorization for S3 is Enabled under Fine-grained access control.
3. For Azure environment, perform the following step:
 - Ensure that the Ranger Authorization for ADLS Gen2 is Enabled under Fine-grained access control

Once RAZ is enabled, all the requests for cloud storage will be routed through RAZ. RAZ checks if it should allow or deny access to objects in the object store based on the defined policies.

To know more about, RAZ requirements for AWS see, AWS requirements for [RAZ-enabled AWS environment](#).

To register a RAZ-enabled AWS environment, see [Using CDP UI to register RAZ-enabled AWS environment](#).

To register a RAZ-enabled Azure environment, see [Registering a RAZ-enabled Azure environment](#).

Ephemeral Storage

COD now supports NVMe (Non-volatile Memory Express) based cache that significantly improves the performance when you deploy COD with object storage.

Accessing cloud storage is generally slower than block storage, as data in such deployments is not local to HBase. Our benchmarks show that COD over cloud storage performs, on average, five times slower than COD clusters using HDFS on block storage with HDD. To address this penalty, COD uses ephemeral storage (NVMe disks available with all major cloud providers) to deploy RegionServers with file base bucket cache on the ephemeral volumes. With all data cached locally in the ephemeral disks, COD on cloud storage can achieve performance parity compared to COD with HDFS on block storage, but at a lower cost.

Once you enable the entitlement for your tenant, COD automatically configures the newly created cluster for this feature. It also allows the auto-scale functionality based on the available cache space. For more information, see *Auto-scaling in public cloud environments*. No additional configuration is needed.

AWS

COD supports a 1.6 TB NVMe based cache that significantly improves the performance when you deploy COD with S3.

Azure

The NVMe-based cache is now enabled on Azure too. It uses a high-performance AMD instance type with a 2 TB NVMe disk for the worker nodes.

GCP

COD supports a 3 TB NVMe-based cache that uses local SSD for ephemeral storage on GCP. This significantly enhances the performance when you deploy COD on GCP.

Related Information

[Autoscaling in public cloud environments](#)

[COD on cloud storage with ephemeral cache](#)

HBase Store File Tracking

Cloudera Operational Database (COD) supports *Store File Tracking (SFT)* as a separate, pluggable layer to handle storefile life cycle, and includes the FILE based built-in implementation that avoids internal file rename or move operations while managing the storefiles.

Cloudera has worked with the Apache HBase project to deliver the first version of this feature through [HBASE-26067](#), and has delivered this feature as a part of CDP.

When using S3 for HBase data, COD can dynamically scale the number of workers based on the compute resources required, rather than the workers required to host the data in HDFS. To deliver this ability to you in a reasonable timeframe, Cloudera built HBOSS. This feature is the next evolution of HBase using S3 which no longer requires the HBOSS solution. The SFT feature for HBase with S3 prevents unwanted I/O due to renames on S3. With HDFS, a rename is a constant-time operation, but on S3 a rename requires a full copy of the file. Because of this, using S3 doubles the I/O costs for HBase operations like compactions, flushes, and snapshot-based operations. The SFT feature removes the reliance on renames from HBase internal functions that handle user data, which benefits file systems that lack atomic directory rename operation, such as S3.

You can set SFT at the HBase service level using the `hbase.store.file-tracker.impl` property in `hbase-site.xml` file or at the table or column family level by configuring `TableDescriptor`.

Store File Tracking usage

Learn how to use the *Store File Tracking (SFT)* on your COD cluster.

When you create a COD cluster on an AWS environment using S3 storage, the new cluster automatically has the SFT feature enabled. To use this feature, you can create a new COD database using `cdpcli-beta`.

The following is an example of the `create-database` command.

```
cdp opdb create-database --environment-name myEnvironment --database-name myDatabase
```

COD databases are created and the provided feature flag results in all HBase and Phoenix tables using the SFT feature. If you want to use the HDFS storage for your COD cluster and use the `--use-hdfs` option while creating the COD database, the SFT feature is not enabled because this feature is not relevant when HDFS is being used.



Note: If you try to create a COD database for a CDP environment on Azure or GCP, the call fails because this feature is only applicable to AWS.

You do not need to recompile or update code in your applications for COD databases which have this feature enabled. Cloudera recommends that you update applications to the same version of HBase and Phoenix client libraries to match the version of their COD database.

Related Information

[COD migration](#)

COD migration

You can migrate HBase and Phoenix tables from CDH or HDP to COD CDP Public Cloud where the *Store File Tracking (SFT)* is enabled.

When you are migrating from CDH or HDP to an SFT enabled COD CDP Public Cloud, ensure that you create an SFT enabled COD database on the target cluster. For more details on creating a COD database, see [Store File Tracking usage](#).

To migrate from CDH or HDP to an SFT enabled COD CDP Public Cloud, see [HBase Migration through Replication Manager](#).

Even after restoring the snapshots on the SFT enabled COD CDP Public Cloud, the resulting tables do not load the correct SFT configurations automatically; SFT considers the DEFAULT tracker, which still relies on the temporary directories and renames.

You must convert the DEFAULT tracker to the FILE tracker in each of the restored table configurations by defining the MIGRATION tracker at the table level configurations. Consider the following example to switch the tracker from DEFAULT to FILE in a table configuration.

```
alter 'my-table', CONFIGURATION => {'hbase.store.file-tracker.impl' => 'MIGRATION',  
'hbase.store.file-tracker.migration.src.impl' => 'DEFAULT',  
'hbase.store.file-tracker.migration.dst.impl' => 'FILE'}
```

Once all table regions are online, disable the MIGRATION by setting `hbase.store.file-tracker.migration.dst.impl` value as `hbase.store.file-tracker.impl`. In the above example, this can be represented as.

```
alter 'my-table', CONFIGURATION => {'hbase.store.file-tracker.impl' => 'FILE'  
'}'
```

Related Information

[Store File Tracking usage](#)

[HBase Migration through Replication Manager](#)

Restrictions in Store File Tracking

Know the current restrictions and caveats while using the *Store File Tracking (SFT)* feature.

The following are the current restrictions and caveats.

- There is no automated opt-in to this feature for already existing databases. Deploy a new database using the above directions.
- The existing COD databases which are upgraded to Cloudera Runtime 7.2.14 do not use the SFT feature. These databases continue to use HBOSS to safely interact with S3.

Multi-AZ deployment on COD

Learn about Multi-AZ (Multiple Availability Zones) deployments and how Cloudera Operational Database (COD) ensures high availability and fault tolerance using Multi-AZ deployments.

How Multi-AZ deployment enables HBase high availability



Important: Multi-AZ for COD is currently supported only on Amazon Web Services (AWS) and Microsoft Azure environments. However, it is supported on an Azure environment as a technical preview and is considered under development.

To understand the Multi-AZ deployments on COD, it is important to know how a cloud provider is configured across the globe and how it provides the availability of services regardless of your location. A cloud provider is made up of a number of regions, which are physical locations around the world. Each region has multiple availability zones (AZs), which act as failure domains, preventing small outages from affecting entire regions. Most regions are home to two to three AZs, with each AZ providing adequate redundancy within a given region. An AZ is represented by a region code followed by a letter identifier; for example, *us-west-1a*, *us-west-1b*.

However, this redundancy is only applied to the storage layer (S3) and does not exist for virtual machines used to build up your workload. If the AZ where your server instances reside is having an outage, your services cease to function, because the entire compute infrastructure is offline.

A Multi-AZ deployment means that compute infrastructure for HBase's master and region servers are distributed across multiple AZs ensuring that when a single availability zone has an outage, only a portion of Region Servers is impacted and clients automatically switch over to the remaining servers in the available AZs. Similarly, the backup master (assuming the primary master was in the AZ, which is having an outage) automatically takes the role of the failing master because it is deployed in a separate AZ from the primary master server. The entire process is automatic and requires no setup and management from an user. Such a deployment works to ensure an application does not suffer an outage due to the loss of a single AZ.

To understand more on the Multi-AZ support on COD, see this blog, *High Availability (Multi-AZ) for Cloudera Operational Database*.

Related Information

[High Availability \(Multi-AZ\) for Cloudera Operational Database](#)

Advantages of Multi-AZ deployments

Understand the single and multiple availability zones that Cloudera Operational Database (COD) supports along with the advantages of deploying Multi-AZ.

COD in a single availability zone

Without multiple availability zone based deployments, CDP Control Plane deploys COD and Data Hubs into a single subnet, meaning that each of them has VMs spread across a single availability zone.



Note: On AWS every subnet is related to a single availability zone.

COD in a multiple availability zone

If you choose to deploy your COD Environment and Data Hubs across multiple availability zones (multi-AZ), each of these components could be spread across two or more availability zones, providing high availability, resilience, and fault tolerance.

Table 1: Key advantages

Advantages	Descriptions
High availability	Supports high availability for your application with automatic database failover. It ensures that you do not lose any data and zero manual intervention.
Resilience	Ensures that the data on your standby database instance is up-to-date with the primary.
Fault tolerance	Enhances availability by deploying a standby instance in a second AZ, and achieve fault tolerance in the event of an AZ or database instance failure.

How to enable Multi-AZ deployment on COD

Understand the entitlements that you need to access the Multi-AZ on COD and also how you can disable Multi-AZ on your COD environment.

To use this feature on an AWS or Azure environment, you must first contact Cloudera Support or your Cloudera account team to enable the following entitlements in CDP Public Cloud.

- On an AWS environment, you need the CDP_CB_AWS_NATIVE entitlement.
- On an Azure environment, you need both CDP_CB_AZURE_MULTIAZ and CDP_AZURE_DATABASE_FLEXIBLE_SERVER entitlements.

Once the entitlement is enabled, COD supports multiple availability while creating databases, by default.

To disable multiple availability on COD, you can provide the `--disable-multi-az` option while creating a new COD database using `cdplici-beta`.

For example, `cdp opdb create-database --environment myEnvironment --database myDatabase --disable-multi-az`

If you provide this option for a CDP environment on GCP, the call fails because this feature is only applicable to AWS and Azure (as a technical preview).

Troubleshooting Multi-AZ deployments

Troubleshoot various scenarios which you might encounter while deploying Multi-AZ on your COD environment.

Unable to join the cluster automatically

Condition

HBase Region Servers do not join the cluster automatically after the availability zones are recovered.

Cause

The availability zones and servers are offline for too long and the Master and Region Server processes are stopped.

Solution

Procedure

1. Log in to Cloudera Manager.
2. Restart the Master and Region Server processes.

OMID service fails to recover

Condition

OMID service is failing to recover after the availability zones are down.

Cause

Root cause of this problem is yet to be identified.

Solution

Procedure

1. Log in to Cloudera Manager.
2. Restart the OMID service.

Restrictions in Multi-AZ deployment on COD

Know the restrictions while deploying multiple availability zones on your CDP Operational Database (COD) environment.

The following are the current restrictions:

1. Cloudera Manager currently does not support High Availability, therefore if the AZ containing the gateway node is down, the Cloudera Manager services are unavailable.
2. COD auto-scaling does not work during an AZ outage. When an AZ outage occurs, the Data Hub identifies that multiple nodes are down and goes into an unhealthy state. COD cannot initiate auto-scaling during an unhealthy state.
3. If the node containing the YARN Resource Manager (RM) is down during an AZ outage, the MapReduce jobs related to the YARN services might be impacted, because the RM is not distributed.
4. While performing Multi-AZ failover testing, you might encounter longer failover times because of the Ranger RAZ services, which applies authorization logic in front of cloud storage systems. When this failover does not happen promptly, HBase is unable to perform any operations that require communication with the cloud storage system. Cloudera continues to evaluate this issue in later releases.