

Storing Data Using Ozone

Date published: 2020-04-24

Date modified: 2020-05-21



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

Introduction to Ozone.....	4
Ozone architecture.....	4
How Ozone manages read operations.....	5
How Ozone manages write operations.....	6
Managing storage elements by using the command-line interface.....	6
Commands for managing volumes.....	6
Assigning administrator privileges to users.....	8
Commands for managing buckets.....	9
Commands for managing keys.....	10
Using Ozone S3 Gateway to work with storage elements.....	12
URL schema for Ozone S3 Gateway.....	12
URL to browse Ozone buckets.....	12
REST endpoints supported on Ozone S3 Gateway.....	13
Mapping for an Ozone volume in Amazon S3 API.....	13
Examples of using the Amazon Web Services command-line interface for S3 Gateway.....	13
Working with Ozone File System.....	15
Setting up OzoneFS.....	15
Configuration updates for Spark to work with OzoneFS.....	16
Overview of the Ozone Manager in High Availability.....	16
Considerations for configuring High Availability on Ozone Manager.....	16
Ozone Manager nodes in High Availability.....	16
Read and write requests with Ozone Manager in High Availability.....	17
Working with the Recon web user interface.....	17
Access the Recon web user interface.....	17
Elements of the Recon web user interface.....	18
Overview page.....	18
DataNodes page.....	18
Pipelines page.....	19
Missing Containers page.....	20
Configuring Ozone to work with Prometheus.....	21

Introduction to Ozone

Apache Ozone is a scalable, redundant, and distributed object store optimized for big data workloads. Apart from scaling to billions of objects of varying sizes, applications that use frameworks like Apache Spark, Apache YARN and Apache Hive work natively on Ozone without any modifications. Ozone natively supports the S3 API and provides a Hadoop-compatible file system interface. Ozone is typically available in a CDP Private Cloud Base deployment.



Important: Apache Hadoop Ozone in CDP is available as Beta and is considered to be under development. Do not use Ozone in your production systems. If you have questions regarding Ozone, contact support by logging a case on the [Cloudera Support Portal](#).

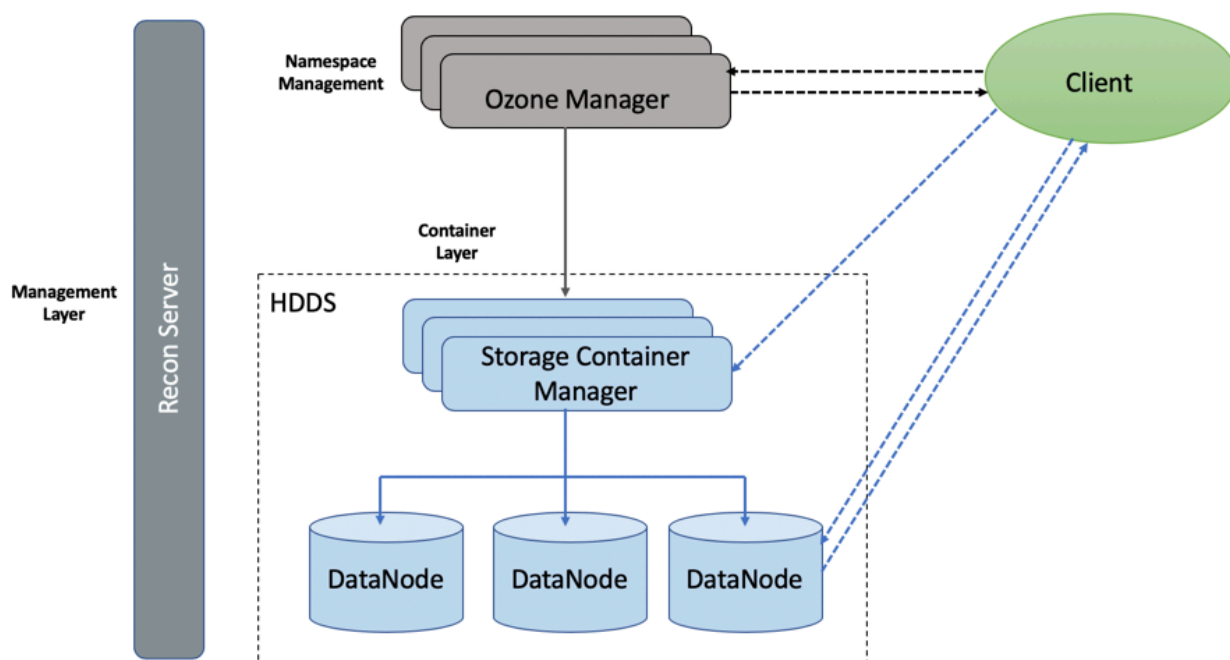
Ozone consists of three important storage elements: volumes, buckets, and keys. Each key is part of a bucket, which, in turn, belongs to a volume. Only an administrator can create volumes. Depending on their requirements, regular users can create buckets in volumes. Ozone stores data as keys inside these buckets.

When a client writes a key, Ozone stores the associated data on DataNodes in chunks called blocks. Therefore, each key is associated with one or more blocks. Within a DataNode, multiple unrelated blocks can reside in a storage container.

Ozone architecture

Ozone can be co-located with HDFS with single security and governance policies for easy data exchange or migration and also offers seamless application portability. Ozone has a scale-out architecture with minimal operational overheads. Ozone separates management of namespaces and storage, helping it to scale effectively. The Ozone Manager (OM) manages the namespaces while the Storage Container Manager (SCM) handles the containers.

The following diagram shows the components that form the basic architecture of Ozone:



Hadoop Distributed Data Store

Ozone is built on a highly available, replicated block storage layer called Hadoop Distributed Data Store (HDDS).

Blocks

Blocks are the basic unit of storage. In Ozone, each block is of 256 MB in size. A collection of blocks forms a storage container. The SCM allocates blocks inside storage containers for the client to store data.

Storage Containers

A storage container is a group of unrelated blocks managed together as a single entity. A container exists in a DataNode and is the basic unit of replication, with a capacity of 2 GB to 16 GB.

DataNodes

DataNodes contain storage containers comprising of data blocks. The SCM monitors DataNodes through heartbeats.

Ozone Manager

The Ozone Manager (OM) is the metadata manager for Ozone. The OM manages the following storage elements:

- The list of volumes for each user
- The list of buckets for each volume
- The list of keys for each bucket

The OM maintains the mappings between keys and their corresponding Block IDs. When a client application requests for keys to perform read and write operations, the OM interacts with the SCM for information about blocks relevant to the read and write operations, and provides this information to the client. In addition, the OM also handles metadata operations from the clients.

Ozone Manager

The Ozone Manager (OM) is a highly available namespace manager for Ozone.

OM manages the metadata for volumes, buckets, and keys. OM maintains the mappings between keys and their corresponding Block IDs. When a client application requests for keys to perform read and write operations, OM interacts with SCM for information about blocks relevant to the read and write operations, and provides this information to the client. In addition, OM also handles metadata operations from the clients.

Storage Container Manager

The Storage Container Manager performs multiple critical functions for an Ozone cluster.

SCM manages the addition and removal of DataNodes, and allocates storage containers and blocks. SCM also manages block collections, ensuring that the blocks maintain the required level of replication. SCM allocates blocks to clients through OM for read and write operations. In addition, SCM executes recovery actions when faced with DataNode or disk failures.

Pipelines

Pipelines determine the replication strategy for the blocks associated with a write operation.

Recon Server

Recon is the management interface for Ozone. Recon provides a unified management API for Ozone.

How Ozone manages read operations

The client requests the block locations corresponding to the key it wants to read. The Ozone Manager (OM) returns the block locations if the client has the required read privileges.

The following steps explain how Ozone manages read operations:

1. The client requests OM for block locations corresponding to the key to read.
2. OM checks the ACLs to confirm whether the client has the required privileges, and returns the block locations and the block token that allows the client to read data from the DataNodes.

3. The client connects to the DataNode associated with the returned Block ID and reads the data blocks.

How Ozone manages write operations

The client requests a block from the Ozone Manager (OM) to write a key. OM returns the Block ID and the corresponding DataNodes for the client to write data.

The following steps explain how Ozone manages write operations:

1. The client requests a block from OM to write a key. The request includes the key, the data to write, the pipeline type, and the replication count.
2. OM finds the blocks that match the request and returns them to the client.



Note: If security is enabled on the cluster, OM also provides a block token along with the block location to the client. The client uses the block token to connect to the DataNodes and send the command to write chunks of data.

3. The client connects to the DataNodes associated with the returned block information and writes the data.
4. After writing the data, the client updates the block information on OM.
5. OM records the associated key information.

Managing storage elements by using the command-line interface

The Ozone shell is the primary command line interface for managing storage elements such as volumes, buckets, and keys.

For more information about the various Ozone command-line tools and the Ozone shell, see <https://ozone.apache.org/docs/1.3.0/interface/cli.html>.

Commands for managing volumes

Depending on whether you are an administrator or an individual user, the Ozone shell commands enable you to create, delete, view, list, and update volumes. Before running these commands, you must have configured the Ozone Service ID for your cluster from the Configuration tab of the Ozone service on Cloudera Manager.

Creating a volume

Only an administrator can create a volume and assign it to a user. You must assign administrator privileges to users before they can create volumes. For more information, see [Assigning administrator privileges to users](#).

Command Syntax	<code>ozone sh volume create --quota=<volume capacity> --user=<username> URI</code>
Purpose	Creates a volume and assigns it to a user.

Arguments	<ul style="list-style-type: none"> • -q, quota: Specifies the maximum size the volume can occupy in the cluster. This is an optional parameter. • -u, user: The name of the user who can use the volume. The designated user can create buckets and keys inside the particular volume. This is a mandatory parameter. • URI: The name of the volume to create in the <prefix>://<Service ID>/<volumename> format.
Example	<pre>ozone sh volume create --quota=2TB - -user=usr1 o3://ozone1/vol1</pre> <p>This command creates a 2-TB volume named vol1 for user usr1. Here, ozone1 is the Ozone Service ID.</p>

Deleting a volume

Command Syntax	<pre>ozone sh volume delete URI</pre>
Purpose	Deletes the specified volume, which must be empty.
Arguments	URI: The name of the volume to delete in the <prefix>://<Service ID>/<volumename> format.
Example	<pre>ozone sh volume delete o3://ozone1/v ol2</pre> <p>This command deletes the empty volume vol2. Here, ozone1 is the Ozone Service ID.</p>

Viewing volume information

Command Syntax	<pre>ozone sh volume info URI</pre>
Purpose	Provides information about the specified volume.
Arguments	URI: The name of the volume whose details you want to view, in the <prefix>://<Service ID>/<volumename> format.
Example	<pre>ozone sh volume info o3://ozone1/ vol3</pre> <p>This command provides information about the volume vol3. Here, ozone1 is the Ozone Service ID.</p>

Listing volumes

Command Syntax	<pre>ozone sh volume list --user <username> URI</pre>
Purpose	Lists all the volumes owned by the specified user.

Arguments	<ul style="list-style-type: none"> -u, user: The name of the user whose volumes you want to list. URI: The Service ID of the cluster in the <prefix>://<Service ID>/ format.
Example	<pre>ozone sh volume list --user usr2 o3: //ozone1/</pre> <p>This command lists the volumes owned by user usr2. Here, ozone1 is the Ozone Service ID.</p>

Updating a volume

Command Syntax	<pre>ozone sh volume update --quota=<volume capacity> --user=<username> URI</pre>
Purpose	Updates the quota or changes the user associated with a specific volume.
Arguments	<ul style="list-style-type: none"> -q, quota: Specifies the maximum size the volume can occupy in the cluster. This is an optional parameter. -u, user: The name of the user who can use the volume. The designated user can create buckets and keys inside the particular volume. This is an optional parameter. URI: The name of the volume to update in the <prefix>://<Service ID>/<volumename> format.
Example	<pre>ozone sh volume update --quota=10TB o3://ozone1/vol4</pre> <p>This command updates the quota of volume vol4 to 10TB. Here, ozone1 is the Ozone Service ID.</p>

Assigning administrator privileges to users

You must assign administrator privileges to users before they can create Ozone volumes. You can use Cloudera Manager to assign the administrative privileges.

About this task

Procedure

1. On Cloudera Manager, go to the Ozone service.
2. Click the Configuration tab.
3. Search for the Ozone Service Advanced Configuration Snippet (Safety Valve) for ozone-conf/ozone-site.xml property.

Specify values for the selected properties as follows:

- Name: Enter ozone.administrators.
 - Value: Enter the ID of the user that you want as an administrator. In case of multiple users, specify a comma-separated list of users.
 - Description: Specify a description for the property. This is an optional value.
4. Enter a Reason for Change, and then click Save Changes to commit the change.

Commands for managing buckets

The Ozone shell commands enable you to create, delete, view, and list buckets. Before running these commands, you must have configured the Ozone Service ID for your cluster from the Configuration tab of the Ozone service on Cloudera Manager.

Creating a bucket

Command Syntax	<code>ozone sh bucket create URI</code>
Purpose	Creates a bucket in the specified volume.
Arguments	URI: The name of the bucket to create in the <prefix>://<Service ID>/<volumename>/<bucketname> format.
Example	<pre>ozone sh bucket create o3://ozone1/vol1/buck1</pre> <p>This command creates a bucket buck1 in the volume vol1. Here, ozone1 is the Ozone Service ID.</p>

Deleting a bucket

Command Syntax	<code>ozone sh bucket delete URI</code>
Purpose	Deletes the specified bucket, which must be empty.
Arguments	URI: The name of the bucket to delete in the <prefix>://<Service ID>/<volumename>/<bucketname> format.
Example	<pre>ozone sh bucket delete o3://ozone1/vol1/buck2</pre> <p>This command deletes the empty bucket buck2. Here, ozone1 is the Ozone Service ID.</p>

Viewing bucket information

Command Syntax	<code>ozone sh bucket info URI</code>
Purpose	Provides information about the specified bucket.
Arguments	URI: The name of the bucket whose details you want to view, in the <prefix>://<Service ID>/<volumename>/<bucketname> format.
Example	<pre>ozone sh bucket info o3://ozone1/vol1/buck3</pre> <p>This command provides information about bucket buck3. Here, ozone1 is the Ozone Service ID.</p>

Listing buckets

Command Syntax	<pre>ozone sh bucket list URI --length=<number_of_buckets> --prefix=<bucket_prefix> --start=<starting_bucket></pre>
Purpose	Lists all the buckets in a specified volume.
Arguments	<ul style="list-style-type: none"> -l, length: Specifies the maximum number of results to return. The default is 100. -p, prefix: Lists bucket names that match the specified prefix. -s, start: Returns results starting with the bucket <i>after</i> the specified value. URI: The name of the volume whose buckets you want to list, in the <prefix>://<Service ID>/<volumename>/ format.
Example	<pre>ozone sh bucket list o3://ozone1/vol2 --length=100 --prefix=buck --start=buck</pre> <p>This command lists 100 buckets belonging to volume vol2 and names starting with the prefix buck. Here, ozone1 is the Ozone Service ID.</p>

Commands for managing keys

The Ozone shell commands enable you to upload, download, view, delete, and list keys. Before running these commands, you must have configured the Ozone Service ID for your cluster from the Configuration tab of the Ozone service on Cloudera Manager.

Downloading a key from a bucket

Command Syntax	<pre>ozone sh key get URI <local_file_name></pre>
Purpose	Downloads the specified key from a bucket in the Ozone cluster to the local file system.
Arguments	<ul style="list-style-type: none"> URI: The name of the key to download in the <prefix>://<Service ID>/<volumename>/<bucketname>/<keyname> format. filename: The name of the file to which you want to write the key.
Example	<pre>ozone sh key get o3://ozone1/hive/jun/sales.orc sales_jun.orc</pre> <p>This command downloads the sales.orc file from the /hive/jun bucket and writes to the sales_jun.orc file present in the local file system. Here, ozone1 is the Ozone Service ID.</p>

Uploading a key to a bucket

Command Syntax	<pre>ozone sh key put URI <filename></pre>
Purpose	Uploads a file from the local file system to the specified bucket in the Ozone cluster.

Arguments	<ul style="list-style-type: none"> URI: The name of the key to upload in the <prefix>://<Service ID>/<volumename>/<bucketname>/<keyname> format. filename: The name of the local file that you want to upload. -r, --replication: The number of copies of the file that you want to upload.
Example	<pre>ozone sh key put o3://ozonel/hive/year/sales.orc sales_corrected.orc</pre> <p>This command adds the sales_corrected.orc file from the local file system as key to /hive/year/sales.orc on the Ozone cluster. Here, ozonel is the Ozone Service ID.</p>

Deleting a key

Command Syntax	<pre>ozone sh key delete URI</pre>
Purpose	Deletes the specified key from the Ozone cluster.
Arguments	URI: The name of the key to delete in the <prefix>://<Service ID>/<volumename>/<bucketname>/<keyname> format.
Example	<pre>ozone sh key delete o3://ozonel/hive/jun/sales_duplicate.orc</pre> <p>This command deletes the sales_duplicate.orc key. Here, ozonel is the Ozone Service ID.</p>

Viewing key information

Command Syntax	<pre>ozone sh key info URI</pre>
Purpose	Provides information about the specified key.
Arguments	URI: The name of the key whose details you want to view, in the <prefix>://<Service ID>/<volumename>/<bucketname>/<keyname> format.
Example	<pre>ozone sh key info o3://ozonel/hive/jun/sales_jun.orc</pre> <p>This command provides information about the sales_jun.orc key. Here, ozonel is the Ozone Service ID.</p>

Listing keys

Command Syntax	<pre>ozone sh key list URI --length=<number_of_keys> --prefix=<key_prefix> --start=<starting_key></pre>
Purpose	Lists the keys in a specified bucket.

Arguments	<ul style="list-style-type: none"> • -l, length: Specifies the maximum number of results to return. The default is 100. • -p, prefix: Returns keys that match the specified prefix. • -s, start: Returns results starting with the key <i>after</i> the specified value. • URI: The name of the bucket whose keys you want to list, in the <prefix>://<Service ID>/<volumename>/<bucketname>/ format.
Example	<pre>ozone sh key list o3://ozone1/hive/year/ --length=100 --prefix=key_prefix --start=day1</pre> <p>This command lists 100 keys belonging to the volume /hive/year/ and names starting with the prefix day, but listed after the value day1. Here, ozone1 is the Ozone Service ID.</p>

Using Ozone S3 Gateway to work with storage elements

Ozone provides S3 Gateway, a REST interface that is compatible with the [Amazon S3 API](#). You can use S3 Gateway to work with the Ozone storage elements.

In addition, you can use the [Amazon Web Services CLI](#) to use S3 Gateway.

After starting Ozone S3 Gateway, you can access it from the following link:

```
http://localhost:9878
```



Note: For the users or client applications that use S3 Gateway to access Ozone buckets on a secure cluster, Ozone provides the AWS access key ID and AWS secret key. See the Ozone security documentation for more information.

URL schema for Ozone S3 Gateway

Ozone S3 Gateway supports both the virtual host-style URL s3 bucket addresses (<http://bucketname.host:9878>) and the path-style addresses (<http://host:9878/bucketname>). By default, S3 Gateway uses path-style addresses.

If you want to use virtual host-style URLs, you must set the main domain name in `ozone-site.xml`.

```
<property>
  <name>ozone.s3g.domain.name</name>
  <value>s3g.internal</value>
</property>
```

URL to browse Ozone buckets

You can browse the contents of Ozone buckets by adding `?browser=true` to the bucket URL.

For example, you can use the following URL to browse the contents of the Ozone bucket buckoz1:

```
http://localhost:9878/buckoz1?browser=true
```



Important: You cannot use the Ozone S3 Gateway browser to view buckets created using the Ozone shell. Only the buckets created using the Ozone S3 Gateway are visible from the browser.

REST endpoints supported on Ozone S3 Gateway

In addition to the GET service operation, Ozone S3 Gateway supports various bucket and object operations that the Amazon S3 API provides.

The following table lists the supported Amazon S3 operations:

Operations on S3 Gateway

- GET service

Bucket operations

- GET Bucket (List Objects) Version 2
- HEAD Bucket
- DELETE Bucket
- PUT Bucket
- Delete multiple objects (POST)

Object operations

- PUT Object
- GET Object
- DELETE Object
- HEAD Object
- Multipart Upload (Except the listing of the current MultiPart Uploads)

Mapping for an Ozone volume in Amazon S3 API

The Amazon S3 REST API does not provide any entity that directly maps to an Ozone volume in the Ozone S3 Gateway. Instead, a bucket name defined in the S3 API maps to its corresponding Ozone volume/bucket location.

Consider the following example of a bucket b1 created using the S3 API:

```
aws s3api --endpoint-url http://localhost:9878 create-bucket --bucket=b1
```

You can view the bucket location using the following command:

```
ozone sh bucket path b1
```

Examples of using the Amazon Web Services command-line interface for S3 Gateway

You can use the Amazon Web Services (AWS) command-line interface (CLI) to interact with S3 Gateway and work with various Ozone storage elements.

Defining an alias for the S3 Gateway endpoint

Defining an alias for the S3 Gateway endpoint helps you in using a simplified form of the AWS CLI. The following example shows how you can define an alias for the S3 Gateway endpoint URL:

```
alias ozones3api='aws s3api --endpoint http://localhost:9878'
```

Examples of using the AWS CLI to work with the Ozone storage elements

The following examples show how you can use the AWS CLI to perform various operations on the Ozone storage elements. All the examples specify the alias `ozones3api`:

Operations	Examples
Creating a bucket	<pre>ozones3api create-bucket --bucket buck1</pre> <p>This command creates a bucket <code>buck1</code>.</p>
Adding objects to a bucket	<pre>ozones3api put-object --bucket buck1 --key Doc1 --body ./Doc1.md</pre> <p>This command adds the key <code>Doc1</code> containing data from <code>Doc1.md</code> to the bucket <code>buck1</code>.</p>
Listing objects in a bucket	<pre>ozones3api list-objects --bucket buck1</pre> <p>This command lists the objects in the bucket <code>buck1</code>. An example output of the command is as follows:</p> <pre>{ "Contents": [{ "LastModified": "2018-11-02T21:57:40.875Z", "ETag": "1541195860875", "StorageClass": "STANDARD", "Key": "Doc1", "Size": 2845 }, { "LastModified": "2018-11-02T22:36:23.358Z", "ETag": "1541198183358", "StorageClass": "STANDARD", "Key": "Doc2", "Size": 5615 }, { "LastModified": "2018-11-02T21:56:47.370Z", "ETag": "1541195807370", "StorageClass": "STANDARD", "Key": "Doc3", "Size": 1780 }] }</pre>
Downloading an object from a bucket	<pre>ozones3api get-object --bucket buck1 --key Doc1 ./Dpcl</pre> <p>This command downloads the key <code>Doc1</code> from the bucket <code>buck1</code> as a file <code>Dpcl</code>. An example output of the command is as follows:</p> <pre>{ "Content-Type": "application/octet-stream", "Content-Length": 2845, "ETag": "1541195860875", "Last-Modified": "2018-11-02T21:57:40.875Z", "Storage-Class": "STANDARD", "X-Amz-Server-Side-Encryption": "AES256" }</pre>

Working with Ozone File System

Ozone File System (OzoneFS) is a Hadoop-compatible file system. Applications such as Hive, Spark, YARN, and MapReduce run natively on OzoneFS without any modifications.

OzoneFS resides on a bucket in the Ozone cluster. All the files created through OzoneFS are stored as keys in that bucket. Any keys created in the particular bucket without using the file system commands are shown as files or directories on OzoneFS.

Setting up OzoneFS

Select the Ozone bucket to configure OzoneFS and add specific properties to core-site.xml.

Procedure

1. Select the Ozone bucket on which you want OzoneFS to reside.

If you do not have a designated volume or bucket for OzoneFS, create them using the required commands:

```
ozone sh volume create /volume
ozone sh bucket create /volume/bucket
```

2. Add the properties fs.o3fs.impl and fs.default.name to core-site.xml.

Adding these properties makes the bucket as the default file system for HDFS dfs commands and registers the o3fs file system type.

```
<property>
  <name>fs.o3fs.impl</name>
  <value>org.apache.hadoop.fs.ozone.OzoneFileSystem</value>
</property>
<property>
  <name>fs.defaultFS</name>
  <value>o3fs://bucket.volume</value>
</property>
```

3. Add the ozone-filesystem.jarfile to the classpath.

```
export HADOOP_CLASSPATH=/opt/ozone/share/hadoop/ozonefs/hadoop-ozone-fil
esystem.jar:$HADOOP_CLASSPATH
```

After setting up OzoneFS, you can run hdfs commands such as the following on Ozone:

- hdfs dfs -ls /
- hdfs dfs -mkdir /users

Now, applications such as Hive and Spark can run on this file system after some basic configuration changes.



Note: Any keys that are created or deleted in the bucket using methods other than OzoneFS are displayed as directories and files in OzoneFS.

Related Information

[Configuration updates for Spark to work with OzoneFS](#)

Configuration updates for Spark to work with OzoneFS

After setting up OzoneFS, you can make configuration updates specific to components such as Spark to ensure that they work with Ozone.

Configuration changes for Spark

To run Spark jobs with o3fs on a secure Kerberos-enabled cluster, ensure that you assign a valid URI by setting the value of the Spark Client Advanced Configuration Snippet (Safety Valve) property for the `spark.conf` or the `spark-defaults.conf` file through the Cloudera Manager web UI.

For example:

```
spark.yarn.access.hadoopFileSystems=o3fs://bucket1.voll.securehost1.example.com:9862
```

Related Information

[Setting up OzoneFS](#)

Overview of the Ozone Manager in High Availability

Configuring High Availability (HA) for the Ozone Manager (OM) enables you to run redundant Ozone Managers on your Ozone cluster and prevents the occurrence of a single point of failure in the cluster from the perspective of namespace management. In addition, Ozone Manager HA ensures continued interactions with the client applications for read and write operations.

Ozone Manager HA involves a leader OM that handles read and write requests from the client applications, and at least two follower OM, one of which can take over as the leader in situations such as the following:

- Unplanned events such as a crash involving the node that contains the leader OM.
- Planned events such as a hardware or software upgrade on the node that contains the leader OM.

Considerations for configuring High Availability on Ozone Manager

There are various factors that you must consider when configuring High Availability (HA) for the Ozone Manager (OM).

- You cannot enable HA on a CDP cluster that already has Ozone configured. You must remove the configured Ozone service and reconfigure it with HA. Therefore, ensure that you back up your Ozone data before configuring HA. Also, ensure that there are no jobs running on the cluster before configuring HA.



Note: For information about adding and deleting services using Cloudera Manager, see the following:

- [Adding a service](#)
- [Deleting services](#)
- To reconfigure Ozone, you must define the Ozone Service ID property.
- You must define the OM on at least three nodes so that one OM node is the leader and the remaining nodes are the followers. The OM nodes automatically elect a leader.

Ozone Manager nodes in High Availability

A High Availability (HA) configuration of the Ozone Manager (OM) involves one leader OM node and two or more follower nodes. The leader node services read and write requests from the client. The follower nodes closely keep

track of the updates made by the leader so that in the event of a failure, one of the follower nodes can take over the operations of the leader.

The leader commits a transaction only after at least one of the followers acknowledges to have received the transaction.

Read and write requests with Ozone Manager in High Availability

Read requests from the client applications are directed to the leader Ozone Manager (OM) node. After receiving an acknowledgement to its request, the client caches the details of the leader OM node, and routes subsequent requests to this node.

If repeated requests to the designated leader OM node start failing or fail with a *NonLeaderException*, it could mean that the particular node is no longer the leader. In this situation, the client must identify the correct leader OM node and reroute the requests accordingly.

The following command lists the OM leader node and the follower nodes in an Ozone cluster:

```
ozone admin om getserviceroles -id=<ozone service id>
```

In the case of write requests from clients, the OM leader services the request after receiving a quorum of acknowledgements from the follower.



Note: The read and write requests from clients could fail in situations such as a failover event or network failure. In such situations, the client can retry the requests.

Working with the Recon web user interface

Recon is a centralized monitoring and management service within an Ozone cluster that provides information about the metadata maintained by different Ozone components such as the Ozone Manager (OM) and the Storage Container Manager (SCM).

Recon keeps track of the metadata as the cluster is operational, and displays the relevant information through a dashboard and different views on the Recon web user interface. This information helps in understanding the overall state of the Ozone cluster.

The metadata that components such as the OM and the SCM maintain are quite different from one another. For example, the OM maintains the mapping between keys and containers in an Ozone cluster while the SCM maintains information about containers, DataNodes, and pipelines. The Recon web user interface provides a consolidated view of all these elements.

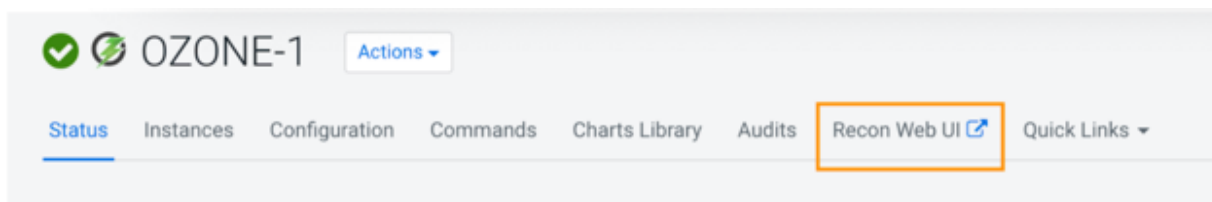
Access the Recon web user interface

You can launch the Recon web user interface from Cloudera Manager. Recon starts its HTTP server over port 9888 by default. The default port is 9889 when auto-TLS is enabled.

Procedure

1. Go to the Ozone service.

2. Click Recon Web UI.



The Recon web user interface loads in a new browser window.

Elements of the Recon web user interface

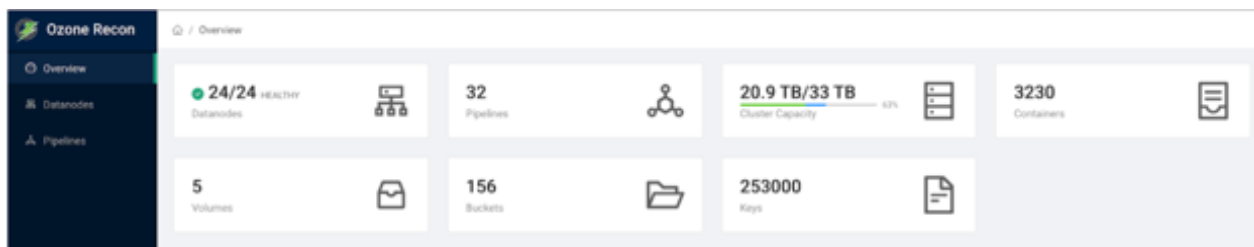
The Recon web user interface displays information about the Ozone cluster on the following pages: Overview, DataNodes, and Pipelines. In addition, a separate page displays information about any missing storage containers.

Overview page

The Overview page displays information about different elements on the Ozone cluster in the form of a consolidated dashboard. This page loads by default when you launch the Recon web user interface.



Note: Recon interacts with the Storage Container Manager (SCM), the DataNodes, and the Ozone Manager (OM) at specific intervals to update its databases and reflect the state of the Ozone cluster, and then populates the Overview page. Therefore, the information displayed on the Overview page might occasionally not be in synchronization with the current state of the Ozone cluster because of a time lag. However, Recon ensures that the information is eventually consistent with that of the cluster.



Recon displays the following information from the SCM and the DataNodes on the Overview page in the form of cards:

- Health of the DataNodes in the cluster. Clicking this card loads the DataNodes page.
- Number of pipelines involved in data replication. Clicking this card loads the Pipelines page.
- Capacity of the cluster. The capacity includes the amount of storage used by Ozone, by services other than Ozone, and any remaining storage capacity of the cluster.
- Number of storage containers in the SCM. If there are any missing containers reported, the Containers card is highlighted with a red border. You can then click the card to view more information about the missing containers on a separate page.

Recon displays following information from the Ozone Manager (OM) on the Overview page:

- Number of volumes in the cluster
- Total number of buckets for all the volumes in the cluster
- Total number of keys for all the buckets in the cluster

DataNodes page

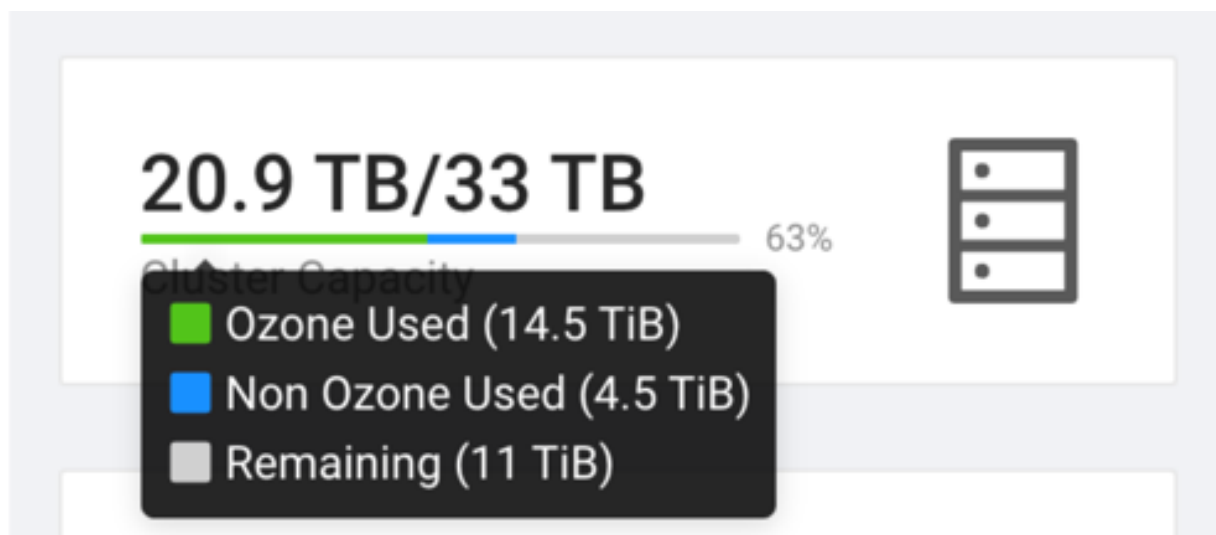
The DataNodes page displays information about the state of the DataNodes in a tabular format. You can load this page either by clicking the DataNodes tab on the left pane or the DataNodes card on the Overview page.

Status	Hostname	Storage Capacity	Last Heartbeat	Pipeline ID(s)	Containers
HEALTHY	adycl-1.adycl.root.hwx.site	151.4 MB + 29.9 GB / 251.9 GB 12%	Apr 8, 2020 10:16 AM	1fd2509b-4aba-4ae9-99d7-851b0870a2ed 8671df3e-d486-41e3-ae86-6bd9902053c	2
HEALTHY	adycl-2.adycl.root.hwx.site	151.4 MB + 24.4 GB / 251.9 GB 10%	Apr 8, 2020 10:16 AM	c3e72990-1684-4962-8146-4efec10f094 8671df3e-d486-41e3-ae86-6bd9902053c	2
HEALTHY	adycl-3.adycl.root.hwx.site	151.4 MB + 22.1 GB / 251.9 GB 9%	Apr 8, 2020 10:16 AM	1f2c47d3-554e-4716-b0cc-e3e70f53cd0b 8671df3e-d486-41e3-ae86-6bd9902053c	2

The following columns of the table provide details of the DataNodes:

- **Status:** The health status of the particular DataNode. The status can be either of the following:
 - **HEALTHY:** Indicates a normal functional DataNode.
 - **STALE:** Indicates that the SCM has not received a heartbeat from the DataNode for a certain period of time after the previous heartbeat.
 - **DEAD:** Indicates that the SCM has not received a heartbeat beyond a certain period of time since receiving the previous heartbeat. The time period beyond which the DataNode can be categorized as DEAD is configurable. The default value is five minutes. Until this threshold is reached, the DataNode is in a STALE state.
 - **DECOMMISSIONING:** Indicates that the DataNode is being decommissioned.
- **Hostname:** The cluster host that contains the particular DataNode.
- **Storage Capacity:** The storage capacity of the particular DataNode. The capacity information includes the amount of storage used by Ozone, by services other than Ozone, and any remaining storage capacity of the host.

Hovering your mouse pointer over a particular entry displays the detailed capacity information as a tool tip.

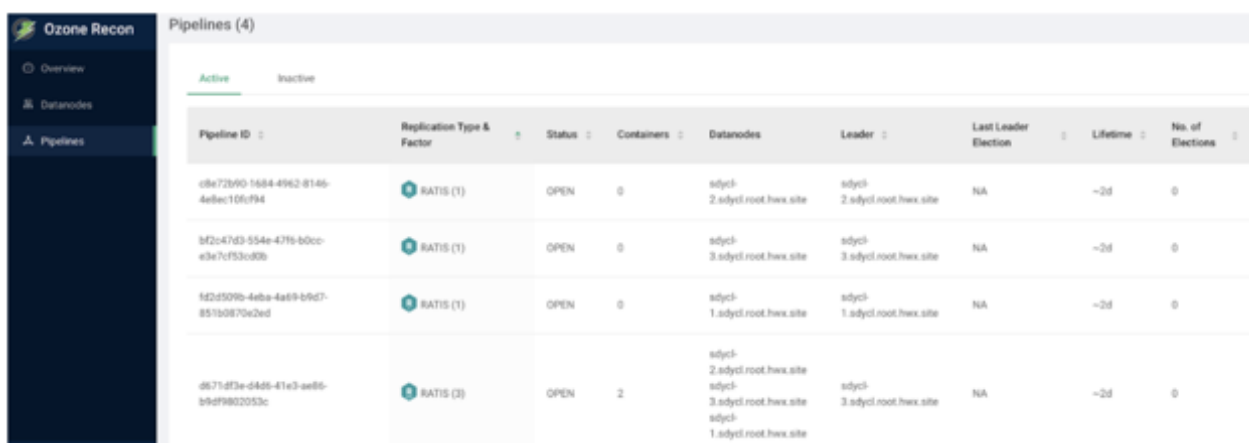


- **Last Heartbeat:** The timestamp of the last heartbeat sent by the particular DataNode to the SCM.
- **Pipeline ID(s):** The IDs of the pipelines to which the particular DataNode belongs.
- **Containers:** The number of storage containers inside the particular DataNode.

Pipelines page

The Pipelines page displays information about active pipelines including their IDs, the corresponding replication factors and the associated DataNodes. The page does not display any inactive pipelines.

An active pipeline is one that continues to participate in the replication process. In contrast, an inactive pipeline contains DataNodes that are dead or inaccessible, leading to the removal of its metadata from the Recon database, and eventually the destruction of the pipeline itself.



The screenshot shows the 'Pipelines (4)' page in the Ozone Recon web interface. The left sidebar has 'Overview', 'DataNodes', and 'Pipelines' (selected). The main area displays a table with 4 pipelines. The table has columns: Pipeline ID, Replication Type & Factor, Status, Containers, DataNodes, Leader, Last Leader Election, Lifetime, and No. of Elections. All pipelines are in 'OPEN' status. The first three pipelines have a replication factor of 1 (RATIS (1)), and the fourth has a replication factor of 3 (RATIS (3)).

Pipeline ID	Replication Type & Factor	Status	Containers	DataNodes	Leader	Last Leader Election	Lifetime	No. of Elections
c8e72b90-1684-4962-8146-4e8ec10fc934	RATIS (1)	OPEN	0	sdycl-2.sdycl.root.hwx.site	sdycl-2.sdycl.root.hwx.site	NA	-3d	0
3f2c47e3-554e-4716-b0cc-e3e7cf53c89b	RATIS (1)	OPEN	0	sdycl-3.sdycl.root.hwx.site	sdycl-3.sdycl.root.hwx.site	NA	-3d	0
1d2d509b-44ba-4a69-b9d7-851b0870a2ed	RATIS (1)	OPEN	0	sdycl-1.sdycl.root.hwx.site	sdycl-1.sdycl.root.hwx.site	NA	-3d	0
d671d73e-d4d5-41e3-ae85-b9d9802053c	RATIS (3)	OPEN	2	sdycl-2.sdycl.root.hwx.site sdycl-3.sdycl.root.hwx.site sdycl-1.sdycl.root.hwx.site	sdycl-3.sdycl.root.hwx.site	NA	-3d	0

The page displays Pipeline information in a tabular format. The following columns provide the required information:

- Pipeline ID(s): The ID of a particular pipeline.
- Replication Type & Factor: The type of replication and the corresponding replication factor associated with a particular pipeline. The replication types are Standalone and Ratis. Accordingly, the default replication factor is three for Ratis and one for Standalone.
- Status: Specifies whether the particular pipeline is open or closed.
- DataNodes: The DataNodes that are a part of the particular pipeline.
- Leader: The DataNode that is elected as the Ratis leader for the write operations associated with the particular pipeline.
- Lifetime: The period of time for which the particular pipeline is open.
- Last Leader Election: The timestamp of the last election of the leader DataNode associated with this pipeline.



Note: This field does not show any data for the current release.

- No. of Elections: The number of times the DataNodes associated with the pipeline have elected a leader.

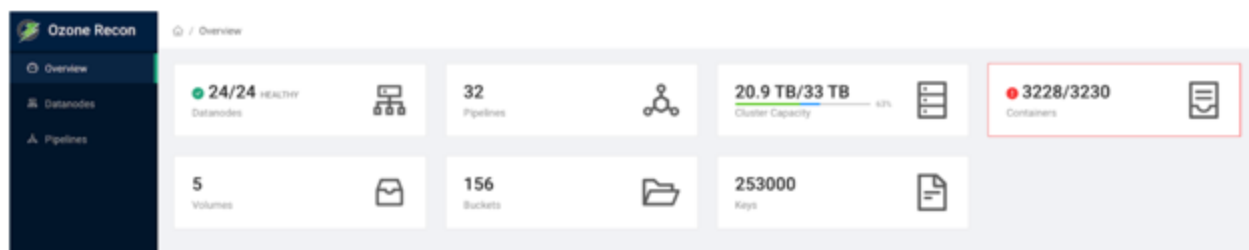


Note: This field does not show any data for the current release.

Missing Containers page

There can be situations when a storage container or its replicas are not reported in any of the DataNode reports to the SCM. Such containers are flagged as missing containers to Recon. Ozone clients cannot read any blocks that are present in a missing container.

The Containers card on the Overview page of the Recon web user interface is highlighted with a red border in the case of missing containers. Clicking the card loads the Missing Containers page.



Overview

Datanodes

Pipelines

Missing Containers

Missing Containers (2)

Container ID	No. of Keys	Datanodes	Pipeline ID	Missing Since																								
<div><div></div>1</div>	1235	<div><div></div>localhost1.storage.enterprise.com</div> <div><div></div>localhost3.storage.enterprise.com</div> <div><div></div>localhost5.storage.enterprise.com</div>	05e3d908-ff01-4ce6-ad75-f3ec79b0c7982	Jan 8, 2020 5:49 AM																								
<table><tr><th>Volume</th><th>Bucket</th><th>Key</th><th>Size</th><th>Date Created</th><th>Date Modified</th></tr><tr><td>vol-0-20448</td><td>bucket-0-12811</td><td>key-0-77505</td><td>10.2 kB</td><td>Nov 26, 2019 1:18 PM</td><td>Nov 26, 2019 1:18 PM</td></tr><tr><td>vol-0-20448</td><td>bucket-0-12811</td><td>key-21-64511</td><td>5.69 MB</td><td>Nov 26, 2019 1:19 PM</td><td>Nov 26, 2019 1:19 PM</td></tr><tr><td>vol-0-20448</td><td>bucket-0-12811</td><td>key-22-68104</td><td>189 kB</td><td>Nov 26, 2019 1:19 PM</td><td>Nov 26, 2019 1:19 PM</td></tr></table>					Volume	Bucket	Key	Size	Date Created	Date Modified	vol-0-20448	bucket-0-12811	key-0-77505	10.2 kB	Nov 26, 2019 1:18 PM	Nov 26, 2019 1:18 PM	vol-0-20448	bucket-0-12811	key-21-64511	5.69 MB	Nov 26, 2019 1:19 PM	Nov 26, 2019 1:19 PM	vol-0-20448	bucket-0-12811	key-22-68104	189 kB	Nov 26, 2019 1:19 PM	Nov 26, 2019 1:19 PM
Volume	Bucket	Key	Size	Date Created	Date Modified																							
vol-0-20448	bucket-0-12811	key-0-77505	10.2 kB	Nov 26, 2019 1:18 PM	Nov 26, 2019 1:18 PM																							
vol-0-20448	bucket-0-12811	key-21-64511	5.69 MB	Nov 26, 2019 1:19 PM	Nov 26, 2019 1:19 PM																							
vol-0-20448	bucket-0-12811	key-22-68104	189 kB	Nov 26, 2019 1:19 PM	Nov 26, 2019 1:19 PM																							
1-3 of 3 keys <div><div></div><div>1</div><div></div></div>																												
<div><div></div>2</div>	1356	<div><div></div>localhost1.storage.enterprise.com</div> <div><div></div>localhost3.storage.enterprise.com</div> <div><div></div>localhost5.storage.enterprise.com</div>	04a5d908-ff01-4ce6-ad75-f3ec73dfc8a2	Jan 8, 2020 5:51 AM																								

1-2 of 2 missing containers

1

10 / page

The page displays information about missing containers in a tabular format. The following columns provide the required information:

- **Container ID:** The ID of the storage container that is reported as missing due to the unavailability of the container and its replicas. Expanding the + sign next to a Container ID displays the following additional information:
 - **Volume:** The name of the volume to which the particular key belongs.
 - **Bucket:** The name of the bucket to which the particular key belongs.
 - **Key:** The name of the key.
 - **Size:** The size of the key.
 - **Date Created:** The date of creation of the key.
 - **Date Modified:** The date of modification of the key.
- **No of Keys:** The number of keys that were a part of the particular missing container.
- **DataNodes:** A list of DataNodes that had a replica of the missing storage container. Hovering your mouse pointer on the information icon shows a tool tip with the timestamp when the container replica was first and last reported on the DataNode.

Container ID	No. of Keys	Datanodes	Pipeline ID	Missing Since
1	1235	localhost1.storage.enterprise.com localhost3.storage.enterprise.com localhost5.storage.enterprise.com	05e3d908-ff01-4ce6-ad75-f3ec79b0c7982	Jan 8, 2020 5:49 AM
2	1356	localhost1.storage.enterprise.com localhost3.storage.enterprise.com localhost5.storage.enterprise.com	04a5d908-ff01-4ce6-ad75-f3ec73dfc8a2	Jan 8, 2020 5:51 AM

Configuring Ozone to work with Prometheus

You can configure your Ozone cluster to enable [Prometheus](#) for real time monitoring of the cluster.

About this task

To enable Prometheus to work on your Ozone cluster, use Cloudera Manager to add the Ozone Prometheus role instance.

Procedure

1. In Cloudera Manager, go to the Ozone service.
2. Add the Ozone Prometheus role instance to the Ozone service.

For more information about adding role instances using Cloudera Manager, see [Adding a role instance](#).



Note: If you do not see Ozone Prometheus in the list of role instances to configure, it means that the role instance is not configured correctly. In this situation, the Prometheus logs (/var/log/hadoop-ozone/ozone-prometheus.log) on the Prometheus instance host show a FileNotFoundException error.

3. Start the Ozone Prometheus role instance.

For information about starting role instances using Cloudera Manager, see [Starting, stopping, and restarting role instances](#).

After starting the role instance, the Prometheus Web UI quick link is added to the Ozone Prometheus page on Cloudera Manager.

4. Click the Prometheus Web UI quick link to launch the web user interface on a separate browser window.
The metrics drop-down list displays various metrics from the Ozone daemons.
5. Select any metric from the drop-down list or enter the name of a metric and click Execute.
Click the Graph or Console tab to view further details.