

Cloudera Runtime 7.1.5

## Storing Data Using Ozone

Date published: 2020-04-24

Date modified: 2020-11-30

**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

<b>Managing storage elements by using the command-line interface.....</b>	<b>4</b>
Commands for managing volumes.....	4
Assigning administrator privileges to users.....	5
Commands for managing buckets.....	6
Commands for managing keys.....	7
<b>Using Ozone S3 Gateway to work with storage elements.....</b>	<b>9</b>
URL schema for Ozone S3 Gateway.....	9
Volume for buckets created using Ozone S3 Gateway.....	10
URL to browse Ozone buckets.....	10
REST endpoints supported on Ozone S3 Gateway.....	10
Examples of using the Amazon Web Services command-line interface for S3 Gateway.....	11
<b>Working with Ozone File System (o3fs).....</b>	<b>13</b>
Setting up o3fs.....	13
<b>Working with ofs.....</b>	<b>14</b>
Volume and bucket management using ofs.....	14
Key management using ofs.....	15
<b>Ozone configuration options to work with CDP components.....</b>	<b>16</b>
Configuration options for Spark to work with o3fs.....	16
Configuration options to store Hive managed tables on Ozone.....	16
<b>Overview of the Ozone Manager in High Availability.....</b>	<b>17</b>
Considerations for configuring High Availability on Ozone Manager.....	17
Ozone Manager nodes in High Availability.....	17
Read and write requests with Ozone Manager in High Availability.....	17
<b>Working with the Recon web user interface.....</b>	<b>18</b>
Access the Recon web user interface.....	18
Elements of the Recon web user interface.....	18
Overview page.....	18
DataNodes page.....	19
Pipelines page.....	20
Missing Containers page.....	21
<b>Configuring Ozone to work with Prometheus.....</b>	<b>22</b>

# 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://hadoop.apache.org/ozone/docs/1.0.0/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=&lt;volumeCapacity&gt; --user=&lt;username&gt; URI</code>
Purpose	Creates a volume and assigns it to a user.
Arguments	<ul style="list-style-type: none"> <li>-q, quota: Specifies the maximum size the volume can occupy in the cluster. This is an optional parameter.</li> <li>-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.</li> <li>URI: The name of the volume to create in the &lt;prefix&gt;://&lt;Service ID&gt;/&lt;volumename&gt; format.</li> </ul>
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	<code>ozone sh volume delete URI</code>
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/vol2</pre> <p>This command deletes the empty volume vol2. Here, ozone1 is the Ozone Service ID.</p>

## Viewing volume information

Command Syntax	<code>ozone sh volume info URI</code>
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	<code>ozone sh volume list --user &lt;username&gt; URI</code>
Purpose	Lists all the volumes owned by the specified user.
Arguments	<ul style="list-style-type: none"> <li>-u, user: The name of the user whose volumes you want to list.</li> <li>URI: The Service ID of the cluster in the &lt;prefix&gt;://&lt;Service ID&gt;/ format.</li> </ul>
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	<code>ozone sh volume update --quota=&lt;volumeCapacity&gt; --user=&lt;username&gt; URI</code>
Purpose	Updates the quota or changes the user associated with a specific volume.
Arguments	<ul style="list-style-type: none"> <li>-q, quota: Specifies the maximum size the volume can occupy in the cluster. This is an optional parameter.</li> <li>-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.</li> <li>URI: The name of the volume to update in the &lt;prefix&gt;://&lt;Service ID&gt;/&lt;volumename&gt; format.</li> </ul>
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	<code>ozone sh bucket list URI --length=&lt;number_of_buckets&gt; --prefix=&lt;bucket_prefix&gt; --start=&lt;starting_bucket&gt;</code>
Purpose	Lists all the buckets in a specified volume.
Arguments	<ul style="list-style-type: none"> <li>-l, length: Specifies the maximum number of results to return. The default is 100.</li> <li>-p, prefix: Lists bucket names that match the specified prefix.</li> <li>-s, start: Returns results starting with the bucket <i>after</i> the specified value.</li> <li>URI: The name of the volume whose buckets you want to list, in the &lt;prefix&gt;://&lt;Service ID&gt;/&lt;volumename&gt; format.</li> </ul>
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	<code>ozone sh key get URI &lt;local_file_name&gt;</code>
Purpose	Downloads the specified key from a bucket in the Ozone cluster to the local file system.

Arguments	<ul style="list-style-type: none"> <li>URI: The name of the key to download in the &lt;prefix&gt;://&lt;Service ID&gt;/&lt;volumename&gt;/&lt;bucketname&gt;/&lt;keyname&gt; format.</li> <li>filename: The name of the file to which you want to write the key.</li> </ul>
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	<code>ozone sh key put URI &lt;filename&gt;</code>
Purpose	Uploads a file from the local file system to the specified bucket in the Ozone cluster.
Arguments	<ul style="list-style-type: none"> <li>URI: The name of the key to upload in the &lt;prefix&gt;://&lt;Service ID&gt;/&lt;volumename&gt;/&lt;bucketname&gt;/&lt;keyname&gt; format.</li> <li>filename: The name of the local file that you want to upload.</li> <li>-r, --replication: The number of copies of the file that you want to upload.</li> </ul>
Example	<pre>ozone sh key put o3://ozone1/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, ozone1 is the Ozone Service ID.</p>

## Deleting a key

Command Syntax	<code>ozone sh key delete URI</code>
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://ozone1/hive/jun/sales_duplicate.orc</pre> <p>This command deletes the sales_duplicate.orc key. Here, ozone1 is the Ozone Service ID.</p>

## Viewing key information

Command Syntax	<code>ozone sh key info URI</code>
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://ozone1/hive/jun/sales_jun.orc</pre> <p>This command provides information about the sales_jun.orc key. Here, ozone1 is the Ozone Service ID.</p>

### Listing keys

Command Syntax	<pre>ozone sh key list URI --length=&lt;number_of_keys&gt; --prefix=&lt;key_prefix&gt; --start=&lt;starting_key&gt;</pre>
Purpose	Lists the keys in a specified bucket.
Arguments	<ul style="list-style-type: none"> <li>-l, length: Specifies the maximum number of results to return. The default is 100.</li> <li>-p, prefix: Returns keys that match the specified prefix.</li> <li>-s, start: Returns results starting with the key <i>after</i> the specified value.</li> <li>URI: The name of the bucket whose keys you want to list, in the &lt;prefix&gt;://&lt;Service ID&gt;/&lt;volumename&gt;/&lt;bucketname&gt;/ format.</li> </ul>
Example	<pre>ozone sh key list o3://ozone1/hive/year/ --length=100 --prefix=&lt;key_prefix&gt; --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>
```

## Volume for buckets created using Ozone S3 Gateway

Ozone S3 Gateway allows access to buckets through a single volume. The administrator must create this volume before buckets can be added using S3 Gateway.

Consider the following example:

```
ozone sh volume create s3v
```

Given this setting, all buckets created using the Ozone S3 Gateway will be placed under the volume named s3v.

You can configure the name of the default volume using the ozone.s3g.volume.name property in ozone-site.xml, as mentioned in the following example:

```
<property>
  <name>ozone.s3g.volume.name</name>
  <value>s3v</value>
</property>
```

## URL to browse Ozone buckets

You can browse the contents of the 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
```

To browse the contents of an already existing bucket using the Ozone S3 Gateway, create a symbolic link to that bucket. For example, if you want to use S3 Gateway to access a bucket buckoz2 located in vol2, create the link as specified before accessing the bucket:

```
ozone sh bucket link /vol2/buckoz2 /s3v/buck2
```

## 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)

## 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:

The examples assume that you have created the default volume as explained in [Default volume for Ozone buckets](#).

Operations	Examples
Creating a bucket	<pre data-bbox="866 276 1454 333">ozones3api create-bucket --bucket buck1</pre> <p data-bbox="858 367 1192 390">This command creates a bucket buck1.</p>
Adding objects to a bucket	<pre data-bbox="866 439 1454 496">ozones3api put-object --bucket buck1 --key Doc1 --body ./Doc1.md</pre> <p data-bbox="858 530 1470 578">This command adds the key Doc1 containing data from Doc1.md to the bucket buck1.</p>
Listing objects in a bucket	<pre data-bbox="866 623 1454 658">ozones3api list-objects --bucket buck1</pre> <p data-bbox="858 713 1470 762">This command lists the objects in the bucket buck1. An example output of the command is as follows:</p> <pre data-bbox="866 798 1454 1755"> {   "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 data-bbox="866 1826 1454 1883">ozones3api get-object --bucket buck1 --key Doc1 ./Dpc1</pre> <p data-bbox="858 1917 1445 1966">This command downloads the key Doc1 from the bucket buck1 as a file Dpc1. An example output of the command is as follows:</p> <pre data-bbox="866 2002 1454 2092"> {   "ContentType": "application/octet-stream",   "ContentLength": 2845,</pre>

# Working with Ozone File System (o3fs)

The Ozone File System (o3fs) is a Hadoop-compatible file system. Applications such as Hive, Spark, YARN, and MapReduce run natively on o3fs without any modifications.

The Ozone File System resides on a bucket in the Ozone cluster. All the files created through o3fs 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 o3fs.

## Setting up o3fs

Select the Ozone bucket to configure o3fs and add specific properties to core-site.xml.

### Procedure

1. Select the Ozone bucket on which you want o3fs to reside.

If you do not have a designated volume or bucket for o3fs, 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.OzoneServiceId</value>  
</property>
```

3. Add the ozone-filesystem-hadoop3.jar to the classpath.

```
export HADOOP_CLASSPATH=/opt/ozone/share/ozonefs/lib/hadoop-ozone-filesystem-hadoop3-*.jar:$HADOOP_CLASSPATH
```



**Note:** With Hadoop 2.x, use the hadoop-ozone-filesystem-hadoop2-\*.jar.



After setting up o3fs, 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 o3fs are displayed as directories and files in o3fs.

### Related Information

[Configuration options for Spark to work with o3fs](#)

# Working with ofs

The ofs file system is a flat layout file system that allows Ozone clients to access all the volumes and buckets under a single root. Client applications such as Hive, Spark, YARN, and MapReduce run natively on ofs without any modifications.

## Volume and bucket management using ofs

When using ofs, Ozone administrators and users can perform various volume and bucket operations with the help of the Hadoop shell commands such as creating volumes and buckets and using ACLs on the volumes and buckets.

### Creating volumes and buckets

Ozone administrators can create directories under the root and first-level directories using the Hadoop shell. Creating a directory under the root is equivalent to creating an Ozone volume. Creating a directory under a first-level directory is equivalent to creating a bucket. In addition, Ozone users can create buckets under volumes to which they have the write access.

In the following example, you create a volume named volume1 using the -mkdir command of the Hadoop shell:

```
ozone fs -mkdir ofs://ozservice1/volume1/
```

The equivalent Ozone command to create a volume is as follows:

```
ozone sh volume create o3://ozservice1/volume1/
```

Similarly, the Hadoop shell command for creating a bucket is as follows:

```
ozone fs -mkdir ofs://ozservice1/volume1/bucket1/
```



**Note:** If you use the -mkdir -p command to create volumes and buckets that do not exist, Ozone creates the specified volumes and buckets.

### Using the /tmp directory

The ofs root contains a special tmp volume mount for backward compatibility with legacy Hadoop applications that use the /tmp/ directory. To use the volume mount, the Ozone administrator must first create a tmp volume and set its Access Control List (ACL) to ALL. This administrator needs to perform this process once for every cluster.

The following example shows how to create the tmp volume and assign it the required ACLs:

```
ozone sh volume create tmp  
ozone sh volume setacl tmp -al world::a
```

After the administrator has created the tmp volume, each user must initialize their respective tmp bucket once. The following example shows how to initialize the tmp bucket.

```
ozone fs -mkdir ofs://ozservice1/tmp/
```

The user can then write to the /tmp/ bucket just as they would to a regular bucket.

### Using ACLs on volumes and buckets

You must consider the following when setting Access Control Lists (ACLs) on Ozone volumes and buckets:

- Setting ACLs on a first-level directory except /tmp/ is the same as setting ACLs on a volume.
- Setting ACLs on a second-level directory is the same as setting ACLs on a bucket.
- The ACLs on the /tmp/ directory are the same as those on the bucket from which the /tmp/ directory is mapped.

For example, if you map ofs:///tmp/ from o3fs:///tmp/<tmp-bucket-for-current-user>/, the ACLs on ofs:///tmp/<tmp-bucket-for-current-user>/ are the same as those on o3fs:///tmp/bucket1/.



**Note:** The name of a user's bucket under the /tmp/ volume is the MD5 hash of the username.

- You cannot set ACLs on the root (/) because it is only a logical root.

### Renaming volumes and buckets

The ofs file system does not support renaming of volumes and buckets. Any attempt to rename a volume or a bucket results in an exception. You can only rename directories inside a bucket.

For example, ofs supports renaming of ofs:///volume1/bucket1/dir1 to ofs:///volume1/bucket1/dir2.

## Key management using ofs

When using ofs, Ozone administrators and users can perform various operations on Ozone keys with the help of the Hadoop shell commands such as creating keys, recursively listing keys, and renaming keys in a bucket.

### Creating keys

You must consider the following when creating Ozone keys using ofs:

- You cannot create files (keys) under the root or the first-level directory (volume) except in the /tmp/ directory.
- You can add keys to the second-level directory (bucket) or lower-level directories.

### Recursively listing keys

You must consider the following when using the ls -R command to recursively list Ozone keys under volumes and buckets:

Running the ls -R command...	Recursively lists the following...
For a bucket	All the keys that belong to the particular bucket
For a volume	All the buckets that belong to the specified volume and the keys that belong to each bucket
At the root	All the volumes under the root, all the buckets that belong to each volume, and all the keys that belong to each bucket

### Renaming keys

You can rename only the keys that belong to a bucket. The ofs file system does not allow you to rename the keys across volumes or buckets.

For example, ofs allows renaming of the key ofs:///volume1/bucket1/key1.txt to ofs:///volume1/bucket1/key2.txt. However, ofs:///volume1/bucket1/key1.txt cannot be renamed to ofs:///volume1/bucket2/key11.txt.

# Ozone configuration options to work with CDP components

There are specific options that you must configure to ensure that other CDP components such as Spark and Hive work with Ozone.

In the case of Spark, you must update a specific configuration property to run Spark jobs with o3fs on a secure Kerberos-enabled cluster. Similarly, for Hive, you must configure the values of specific properties to store Hive managed tables on Ozone.

## Configuration options for Spark to work with o3fs

After setting up o3fs, you can make configuration updates specific to components such as Spark to ensure that they work with Ozone.

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.vol1.securehost1.example.com:9862
```

### Related Information

[Setting up o3fs](#)

## Configuration options to store Hive managed tables on Ozone

If you want to store Hive managed tables with ACID properties on Ozone, you must configure specific properties in `hive-site.xml`.

You can consider either of the following options to store Hive managed tables with ACID support on Ozone:

- Set the value of the `hive.metastore.warehouse.dir` property to point to the path of the Ozone directory where you want to store the Hive tables.
- Set the value of the `metastore.warehouse.tenant.colocation` property to true. You can then set the `MANAGEDLOCATION` of your Hive database to point to an Ozone directory so that the Hive tables can reside at the specified location.



**Note:** Dynamic partitioning in Hive with the default settings can generate an unexpected load on the filesystem when bulk loading data into tables because Hive creates a number of files for every partition.

To avoid this issue, consider updating the following properties and tuning them further based on your requirements: `hive.optimize.sort.dynamic.partition` and `hive.optimize.sort.dynamic.partition.threshold`.

From a filesystem perspective, the recommended values are as follows:

- `hive.optimize.sort.dynamic.partition = true`
- `hive.optimize.sort.dynamic.partition.threshold = 0`

If you notice that some queries are taking a longer time to complete or failing entirely (usually noticed in large clusters), you can choose to revert the value of `hive.optimize.sort.dynamic.partition.threshold` to "-1". The performance issue is related to [HIVE-26283](#).

# 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 OMs, 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).

- OM HA is automatically enabled when you set up Ozone on a CDP cluster with at least three nodes as OM hosts.
- 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.

The following command lists the OM leader node and the follower nodes:

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

## 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:

```
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.

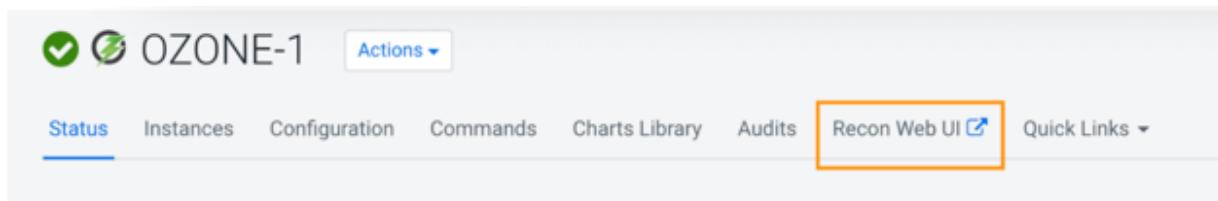
The metadata that components such as the OM and the SCM maintain are quite different from one another. For example, the OM maintains the mapping between keys and containers in an Ozone cluster while the SCM maintains information about containers, DataNodes, and pipelines. The Recon web user interface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

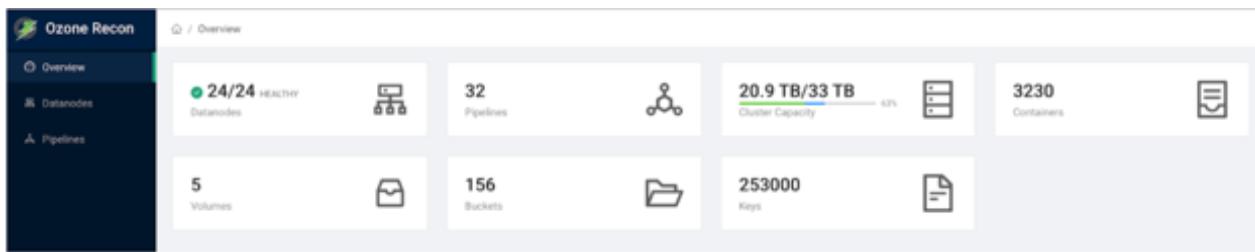
The Recon web user interface displays information about the Ozone cluster on the following pages: Overview, DataNodes, and Pipelines. In addition, a separate page displays information about any missing storage containers.

### 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.

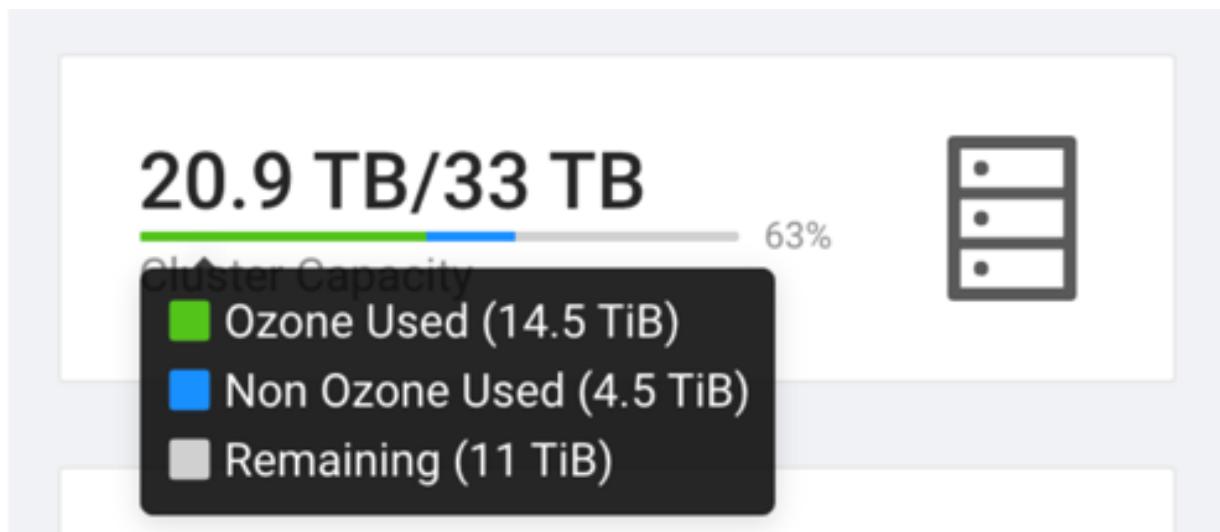
The screenshot shows the Ozone Recon DataNodes page. On the left is a dark sidebar with navigation links: Overview, Datanodes, and Pipelines. The main area has a light gray background with a table titled "Datanodes (3)". The table columns are: Status, Hostname, Storage Capacity, Last Heartbeat, Pipeline ID(s), and Containers. The data rows are:

Status	Hostname	Storage Capacity	Last Heartbeat	Pipeline ID(s)	Containers
HEALTHY	sdyc1-1.sdycl.root.hwx.site	151.4 MiB + 29.9 GiB / 251.9 GiB	10% Apr 8, 2020 10:16 AM	f62d5090-4eba-4a69-b9d7-851b0870e2ed d671df3e-d465-41e3-aed6-b9df9802053c	2
HEALTHY	sdyc1-2.sdycl.root.hwx.site	151.4 MiB + 24.4 GiB / 251.9 GiB	10% Apr 8, 2020 10:16 AM	cbe72b90-1684-4f62-8146-4e8ec10fcf94 d671df3e-d465-41e3-aed6-b9df9802053c	2
HEALTHY	sdyc1-3.sdycl.root.hwx.site	151.4 MiB + 22.1 GiB / 251.9 GiB	9% Apr 8, 2020 10:16 AM	f2c47fd3-554e-47f6-b0cc-e3e7f53c0d6 d671df3e-d465-41e3-aed6-b9df9802053c	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.

Pipelines (4)									
Active		Inactive							
Pipeline ID	Replication Type & Factor	Status	Containers	DataNodes	Leader	Last Leader Election	Lifetime	No. of Elections	
cfe72b90-1684-4962-8146-4e8ec10fcf94	RATIS (1)	OPEN	0	sdyc1-2.sdyc1.root.hwx.site	sdyc1-2.sdyc1.root.hwx.site	NA	-2d	0	
bfc2c47d3-554e-47ff-b0cc-a37e793cd9b	RATIS (1)	OPEN	0	sdyc1-3.sdyc1.root.hwx.site	sdyc1-3.sdyc1.root.hwx.site	NA	-2d	0	
f62d509b-4eb0-4a69-b9d7-857a0870e2ed	RATIS (1)	OPEN	0	sdyc1-1.sdyc1.root.hwx.site	sdyc1-1.sdyc1.root.hwx.site	NA	-2d	0	
d671-df3e-d4d6-41e3-ae86-b9df9802053c	RATIS (3)	OPEN	2	sdyc1-2.sdyc1.root.hwx.site sdyc1-3.sdyc1.root.hwx.site sdyc1-1.sdyc1.root.hwx.site	sdyc1-3.sdyc1.root.hwx.site	NA	-2d	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.

The screenshot shows two views of the Ozone Recon interface. The top view is the 'Overview' page, which includes a sidebar with 'Overview', 'Datanodes', and 'Pipelines'. The main area displays various metrics: 24/24 HEALTHY Datanodes, 32 Pipelines, 20.9 TB/33 TB Cluster Capacity (43%), and 3228/3230 Containers (highlighted with a red border). Below these are cards for Volumes (5), Buckets (156), and Keys (253000). The bottom view is the 'Missing Containers' page, also with a sidebar. It shows 'Missing Containers (2)'. Each entry has columns for Container ID, No. of Keys, Datanodes, Pipeline ID, and Missing Since. Below this is a detailed table for each container, showing Volume, Bucket, Key, Size, Date Created, and Date Modified. Navigation controls like page numbers and a '10 / page' dropdown are at the bottom.

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-f011-4ce6-ad75-f3ec799cc79e2	Jan 8, 2020 5:49 AM			
		vol-0-20448 vol-0-20448 vol-0-20448	bucket-0-12811 bucket-0-12811 bucket-0-12811	key-0-77905 key-21-64511 key-22-69104	10.2 kB 5.69 MB 189 kB	Nov 26, 2019 1:18 PM Nov 26, 2019 1:19 PM Nov 26, 2019 1:19 PM	Nov 26, 2019 1:18 PM Nov 26, 2019 1:19 PM Nov 26, 2019 1:19 PM
2	1356	localhost1.storage.enterprise.com localhost3.storage.enterprise.com localhost5.storage.enterprise.com	04a5cf908-f011-4ce6-ad75-f3ec73df0fa2	Jan 8, 2020 5:51 AM			

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.

Missing Containers (2)				
Container ID	No. of Keys	Datanodes	Pipeline ID	Missing Since
+ 1	First Report Time: Jan 8, 2020 5:49 AM Last Report Time: Jan 8, 2020 6:07 AM	localhost1.storage.enterprise.com localhost3.storage.enterprise.com localhost5.storage.enterprise.com	05e3d908-ff01-4ce6-ad75-f3ec79bcc7982	Jan 8, 2020 5:49 AM
+ 2	1356	localhost1.storage.enterprise.com localhost3.storage.enterprise.com localhost5.storage.enterprise.com	04a5d908-ff01-4ce6-ad75-f3ec73dfcfa2	Jan 8, 2020 5:51 AM

1/2 of 2 missing containers < > 1 10 / page ▾

## 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 FileNotFound 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.