

Cloudera Runtime 7.1.6

Configuring Ozone security

Date published: 2020-04-24

Date modified: 2021-03-16

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

Using Ranger with Ozone.....	4
Kerberos configuration for Ozone.....	4
Security tokens in Ozone.....	4
Kerberos principal and keytab properties for Ozone service daemons.....	5
Securing DataNodes.....	7
Configure S3 credentials for working with Ozone.....	7
Configure Transparent Data Encryption for Ozone.....	8

Using Ranger with Ozone

You can use Apache Ranger to secure access to your Ozone data. For Ozone to work with Ranger, you can use Cloudera Manager and enable the Ranger service instance with which you want to use Ozone.

Procedure

1. Go to the Ozone service.
2. Click the Configuration tab.
3. Search for ranger_service.
4. Select the RANGER service checkbox.
5. Enter a Reason for Change, and then click Save Changes to save the property changes.
6. Restart the cluster.

What to do next

Set up policies in Ranger for the users to have the right access permissions to the various Ozone objects such as buckets and volumes.

When using Ranger to provide a particular user with read/write permissions to a specific bucket, you must configure a separate policy for the user to have read access to the volume in addition to policies configured for the bucket.

For example,

- user/group -> Allow Read on Volume=testvol, Bucket=testbucket1.
- user/group -> Allow All (including delete on) on Volume=testvol, Bucket=testbucket1, Keys=*

To disallow users to delete volumes or buckets:

- user/group -> Allow Read on Volume=testvol, Bucket=testbucket1. Deny Delete on Volume=testvol, Bucket=testbucket1
- user/group -> Allow All (including delete on) on Volume=testvol, Bucket=testbucket1, Keys=*

Further, if Infra-Solr is managed by Ranger, the Ozone Manager principal (om) must have access to Infra-Solr. You can provide access to the Ozone Manager principal by adding om to the RANGER_AUDITS_COLLECTION Solr collection for cm_solr on Ranger.

Related Information

[Apache Ranger Authorization](#)

Kerberos configuration for Ozone

Ozone depends on Kerberos to make the clusters secure. To enable security in an Ozone cluster, you must set the parameters ozone.security.enabled to true and hadoop.security.authentication to kerberos.

Security tokens in Ozone

Ozone issues delegation and block tokens to users or client applications authenticated with the help of Kerberos such that they can perform specified operations against the cluster, as if they have kerberos tickets.

Delegation tokens

Delegation tokens allow a user or client application to impersonate a users kerberos credentials. This token is based on verification of kerberos identity and is issued by the Ozone Manager. Delegation tokens are enabled by default when security is enabled.

Block tokens

Block tokens allow a user or client application to read or write a block. This ensures that only users or client applications with the required permissions can read or write to blocks in DataNodes. Block tokens are issued to authenticated clients and signed by Ozone Manager. They are validated by the DataNode using the certificate or public key of the issuer (Ozone Manager).

S3 tokens

Users or client applications accessing Ozone using S3 APIs with S3 credential tokens. These tokens are also enabled by default when security is enabled.

 **Note:** Ozone supports the AWS Signature Version 4 protocol.

Kerberos principal and keytab properties for Ozone service daemons

For the Kerberos-authenticated users or client applications to access Ozone, each of the Ozone components requires a Kerberos service principal name and a corresponding kerberos keytab file. You must set the corresponding in ozone-site.xml.

The following are the properties for the Kerberos service principal and the keytab file that you must set for the different Ozone components:

Storage Container Manager (SCM) properties

Property	Description
hdds.scm.kerberos.principal	The SCM service principal. You can specify this value, for example, in the following format: <code>scm/_HOST@REALM.COM</code>
hdds.scm.kerberos.keytab.file	The keytab file that the SCM daemon uses to log in as its service principal.
hdds.scm.http.auth.kerberos.principal	The service principal of the SCM HTTP server.
hdds.scm.http.auth.kerberos.keytab	The keytab file that the SCM HTTP server uses to log in as its service principal.

Ozone Manager (OM) properties

Property	Description
ozone.om.kerberos.principal	The Ozone Manager service principal. You can specify this value, for example, in the following format: <code>om/_HOST@REALM.COM</code>

Property	Description
ozone.om.kerberos.keytab.file	The keytab file that the Ozone Manager daemon uses to log in as its service principal.
ozone.om.http.auth.kerberos.principal	The service principal of the Ozone Manager HTTP server.
ozone.om.http.auth.kerberos.keytab	The keytab file that the Ozone Manager HTTP server uses to log in as its service principal.

S3 Gateway properties

Property	Description
ozone.s3g.kerberos.principal	The S3 Gateway principal. You can specify this value, for example, in the following format: <code>s3g/_HOST@REALM.COM</code>
ozone.s3g.kerberos.keytab.file	The keytab file used by S3Gateway daemon to login as its service principal. The principal name is configured with ozone.s3g.kerberos.principal.
ozone.s3g.http.auth.kerberos.principal	The server principal used by Ozone S3 Gateway server. This is typically set to <code>HTTP/_HOST@REALM.TLD</code> . The SPNEGO server principal begins with the prefix <code>HTTP/</code> by convention.
ozone.s3g.http.auth.kerberos.keytab	The keytab file used by the S3 Gateway server to login as its service principal.

Recon properties

Property	Description
ozone.recon.kerberos.principal	The Recon web user service principal. You can specify this value, for example, in the following format: <code>recon/_HOST@REALM.COM</code>
ozone.recon.kerberos.keytab.file	The keytab file used by the Recon web user.
ozone.recon.http.auth.kerberos.principal	The server principal used by Ozone Recon server. This is typically set to <code>HTTP/_HOST@REALM.TLD</code> . The SPNEGO server principal begins with the prefix <code>HTTP/</code> by convention.
ozone.recon.http.auth.kerberos.keytab	The keytab file for HTTP Kerberos authentication in Recon.



Note: Ozone *does not support* authentication based on Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO) for http endpoints in the current CDP release.

Securing DataNodes

You can secure Ozone DataNodes by creating keytab files on each of the DataNodes. You must ensure that certain properties are configured in hdfs-site.xml to provide the Kerberos-authenticated users or client applications with access to the DataNodes.

Configure the following parameters in hdfs-site.xml to enable DataNode access:

Property	Description
dfs.datanode.kerberos.principal	The DataNode service principal. You can specify this value, for example, in the following format: dn/_HOST@REALM.COM
dfs.datanode.kerberos.keytab.file	The keytab file that the DataNode daemon uses to log in as its service principal.
hdds.datanode.http.auth.kerberos.principal	The service principal of the DataNode http server.
hdds.datanode.http.auth.kerberos.keytab	The keytab file that the DataNode http server uses to log in as its service principal.

Certificate request and approval

When a DataNode boots up, it creates a private key and sends an DataNode identity certificate request to Storage Container Manager (SCM). If the DataNode has a Kerberos keytab, SCM trusts the Kerberos credentials and automatically issues a certificate.

Configure S3 credentials for working with Ozone

For the users or client applications that use S3 APIs to access Ozone buckets, Ozone provides the AWS access key ID and AWS secret key. You can add the access key ID and secret key in the AWS config file for Ozone to ensure that a particular user or client application can get automatic access to the Ozone buckets.

Before you begin

The user or the client application accessing Ozone must be authenticated against your cluster's KDC.

Procedure

- Display the access key ID and the secret key for a particular user or client application.

```
ozone s3 getsecret
```

The command communicates with Ozone, validates the user credentials through Kerberos, and generates the AWS credentials. These credentials are printed on the screen.



Note: These S3 credentials are like Kerberos passwords that give complete access to the Ozone buckets.

- Add the generated credentials to the AWS config file.

The following example shows how you can add the credentials.

```
aws configure set default.s3.signature_version s3v4
aws configure set aws_access_key_id ${accessId}
aws configure set aws_secret_access_key ${secret}
aws configure set region us-west-1
```

Configure Transparent Data Encryption for Ozone

Transparent Data Encryption (TDE) allows data on the disks to be encrypted-at-rest and automatically decrypted during access. For Ozone, you can enable TDE at the bucket-level.

Before you begin

- The Key Management Server must be installed and running. Ozone uses the same Key Management Server as HDFS.
- If you are using Ranger Key Management Server, you must ensure that the Ranger KMS admin user has assigned the om service user with the Get Metadata and Generate EEK permissions.

Further, you must ensure that any user attempting to create an encryption key has the required permissions in Ranger.

For information about adding or editing permissions on Ranger, see [Add or edit permissions](#).

Procedure

1. Create a bucket encryption key.

```
hadoop key create enck1
```

This command creates an encryption key for the bucket you want to protect. After the key is created, Ozone can use that key when you are reading and writing data into a bucket.

2. Assign the encryption key to a bucket.

The following example shows how you can assign the key enck1 to the bucket encbucket1:

```
ozone sh bucket create -k enck1 /vol/encbucket1
```

After you run this command, all data written to encbucket1 will be encrypted using encKey. During the read process, the client applications interact with the Key Management Server to read the key and decrypt it.

The encryption of data is completely transparent to users and client applications.