

Configuring and Using Hive-HDFS ACL Sync

Date published: 2020-10-12

Date modified:



Legal Notice

© Cloudera Inc. 2025. 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

Ranger Hive-HDFS ACL Sync Overview.....	4
Configure High Availability for Hive-HDFS ACL Sync.....	5
Configure Hive-HDFS ACL Sync.....	12
Hive-HDFS ACL Sync Use Cases.....	13
Hive-HDFS ACL Sync Reference.....	15

Ranger Hive-HDFS ACL Sync Overview

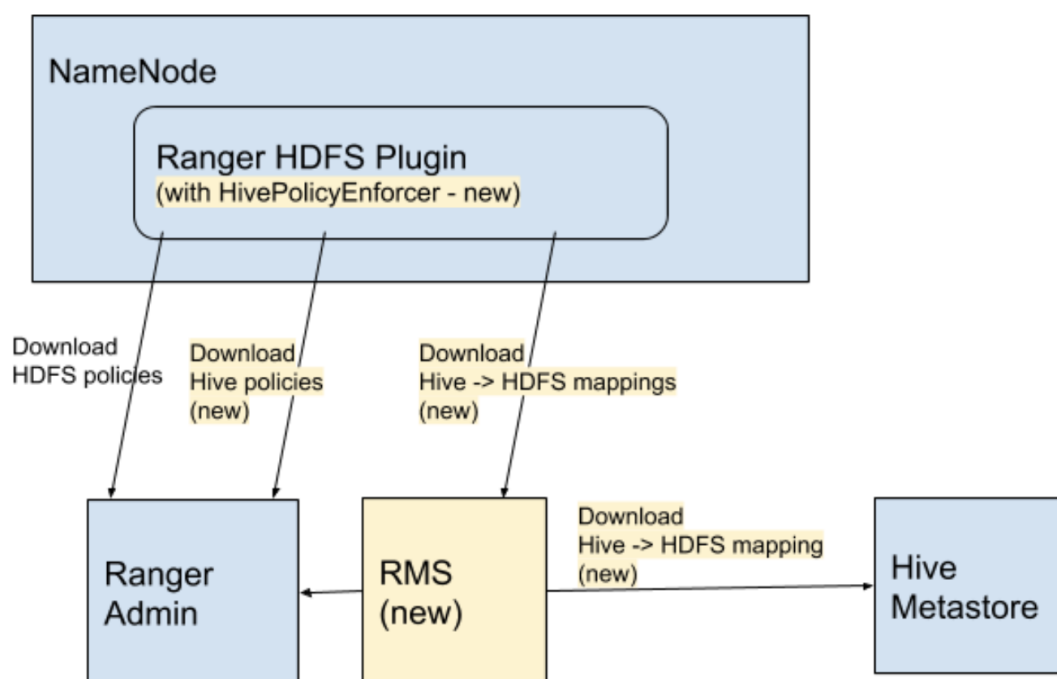
Ranger Resource Mapping Server (RMS) enables automatic translation of access policies from Hive to HDFS.

About Hive-HDFS ACL Sync

Legacy CDH users used Hive policies in Apache Sentry that automatically linked Hive permissions with HDFS ACLs. This was especially convenient for external table data used by Spark or Hive.

Previously, Ranger only supported managing Hive and HDFS policies separately. Ranger RMS (Resource Mapping Server) allows you to authorize access to HDFS directories and files using policies defined for Hive tables. RMS is the service that enables Hive-HDFS ACL Sync.

RMS periodically connects to the Hive Metastore and pulls Hive metadata (database-name, table-name) to HDFS file-name mapping. The Ranger HDFS Plugin (running in the NameNode) has been extended with an additional HivePolicyEnforcer module. The HDFS plugin downloads Hive policies from Ranger Admin, along with the mappings from Ranger RMS. HDFS access is determined by both HDFS policies and Hive policies.



Ranger RMS Assumptions and Limitations

- All partitions of a table are assumed to be under the location specified for the table. Therefore, table permissions will not authorize access to partitions that store data outside the location specified for the table. For example, if a table is located in a `/warehouse/foo` HDFS directory, all partitions of the table must have locations that are under the `/warehouse/foo` directory.
- The Ranger RMS service is not set up automatically when a CDP Private Cloud Base cluster is deployed. You must install and configure Ranger RMS separately.
- Ranger policies should be configured (with `rangerms` user access) before RMS is started and runs the first sync from the Hive Metastore (HMS).
- The Ranger RMS ACL-sync feature supports a single logical HMS, to evaluate HDFS access via Hive permissions. This is aligned with the Sentry implementation in CDH.

- Permissions granted on views (traditional and materialized) do not extend to HDFS access. This is aligned with the Sentry implementation in CDH.
- If a Private Cloud Base deployment supports multiple logical HMS with a single Ranger, Ranger ACL-sync works with only one logical HMS. Permissions granted on databases/tables in other logical HMS instances will not be considered to authorize HDFS access.

Comparison with Sentry HDFS ACL sync

The RMS ACL Sync feature resembles the Sentry HDFS ACL Sync feature in the way it downloads and keeps track of the Hive table to HDFS location mapping.

It differs from Sentry in the way it completely and transparently supports all features that Ranger policies express. Therefore, support for tag-based policies, security-zones, masking and row-filtering and audit logging is included with this implementation.

Also, the feature is enabled or disabled by a simple configuration on the HDFS side, allowing each installation the option of turning this feature on or off.

Related Information

[Installing Ranger RMS](#)

Configure High Availability for Hive-HDFS ACL Sync

Use the following steps to configure high availability for the Ranger Resource Mapping Server (RMS) and Hive-HDFS ACL Sync.

Procedure

1. In Cloudera Manager, select Ranger RMS, then select Actions > Add Role Instances.

The screenshot shows the Cloudera Manager interface for the 'Cluster 1' Ranger RMS configuration. The left sidebar contains navigation links: Clusters, Hosts, Diagnostics, Audits, Charts, Replication, Administration, and Private Cloud. The main content area displays the 'Ranger RMS' configuration page with tabs for Status, Instances, and Configuration. The 'Status' tab is active, showing 'Health Tests' (Ranger RMS Server Health), 'Status Summary' (Ranger RMS Server and Hosts), and 'Health History'. The 'Actions' dropdown menu is open, showing options: Start, Restart, Stop, Add Role Instances (highlighted), Rename, and Enter Maintenance Mode. The 'Charts' section on the right shows 'Informational Events' and 'Important Events and Alerts'.

2. On the Add Role Instances page, click Select hosts.

The screenshot shows the 'Assign Roles' page in Cloudera Manager. The page title is 'Add Role Instances to Ranger RMS'. The page has two steps: 1. Assign Roles and 2. Review Changes. The 'Assign Roles' step is active. The page content includes the text 'You can specify the role assignments for your new roles here.' and 'You can also view the role assignments by host. View By Host'. Below this, there is a section for 'Ranger RMS Server x 1' with a 'Select hosts' button highlighted by a red box.

- On the selected hosts page, select a backup Ranger RMS host. A Ranger RM (RR) icon appears in the Added Roles column for the selected host. Click OK to continue.

2 Hosts Selected

Select hosts for a new or existing role. The host list is filtered to remove hosts that are not valid candidates; these include hosts that are unhealthy, members of other clusters, or have an incompatible version of the software installed on them.

Enter hostnames: host01, host[01-10], IP addresses or rack. [Search](#)

Tip: Click the first checkbox, hold down the Shift key and click the last checkbox to select a range.

☐	Hostname ↑	IP Address	Rack	Cores	Physical Memory	Existing Roles	Added Roles
☒	dhoyle714-1.dhoyle714.root.hwx.site	172.17.0.10	/default	80	251.6 GiB	AS, CCS, G, HB..., RS, DN, G, ID, KB, KC, KG, M, G, LS, RA, RT, RU, SRS, G, G, SM..., SM..., SR..., SR..., G, NM, ZS	RR...
☐	dhoyle714-2.dhoyle714.root.hwx.site	172.17.0.13	/default	32	251.6 GiB	M, B, NN, NF..., SNN, G, HMS, G, HS2, LB, HS, KTR, ICS, ISS, KB, KC, LHBI, TS, G, AP, CS, LM, DM, SM	

[Cancel](#) [OK](#)

- The Add Role Instances page is redisplayed with the new backup host. Click Continue.

CDP deployment from 2020-Sep-11 06:10

Add Role Instances to Ranger RMS

- Assign Roles
- Review Changes

Assign Roles

You can specify the role assignments for your new roles here.

You can also view the role assignments by host. [View By Host](#)

Ranger RMS Server x (1 + 1 New)

dhoyle714-1.dhoyle714.root.hwx.site

[Back](#) [Continue](#)

- CDEF Deployment from 2020-Sep-11 06:10

Back Continue

6. The new role instance appears on the Ranger RMS page.

CLUSTERA
Manager

Search

Clusters

Hosts

Diagnostics

Audits

Charts

Replication

Administration

Private Cloud New

Cluster 1

Ranger RMS

Actions

StatusInstancesConfigurationCommandsCharts LibraryAuditsQuick Links

Search

Filters

Last Updated: Sep 28, 3:21:58 PM UTC

Filters

Actions for Selected

Add Role Instances

Role Groups

STATUS

Good Health1

Stopped1

COMMISSION STATE

MAINTENANCE MODE

RACK ID

ROLE GROUP

ROLE TYPE

STATE

HEALTH TEST

	Status	Role Type	State	Hostname	Commission State	Role Group
☐	✓	Ranger RMS Server	Started	10.10.10.10	Commissioned	Ranger RMS Server Default Group
☐	⊙	Ranger RMS Server	Stopped	10.10.10.10	Commissioned	Ranger RMS Server Default Group

1 - 2 of 2

7. In Cloudera Manager, select Ranger RMS, then click Configuration.
- a) Select the Ranger RMS Server HA checkbox. In the Ranger RMS Server IDs box, add a comma-separated list of IDs for each RMS server.

The screenshot shows the Cloudera Manager configuration interface for Ranger RMS. The 'Ranger RMS Server HA' checkbox is checked. The 'Ranger RMS Server IDs' field contains 'id1,id2'. The 'Ranger RMS Server Default Group' field contains '/ranger-rms'. The 'Ranger RMS Server HA' checkbox is checked. The 'Ranger RMS Server IDs' field contains 'id1,id2'. The 'Ranger RMS Server Default Group' field contains '/ranger-rms'.

- b) Use the Add (+) icons for the Ranger RMS Server Advanced Configuration Snippet (Safety Valve) for ranger-rms-conf/ranger-rms-site.xml property to add entries for each RMS host with its corresponding server ID.
- ranger-rms.server.address.id1=<hostname of RMS server for id1>:8383
 - ranger-rms.server.address.id2=<hostname of RMS server for id2>:8383



Note: If SSL is enabled, use port 8484.

Ranger RMS Server Advanced Configuration Snippet (Safety Valve) for ranger-rms-conf/ranger-rms-site.xml

Ranger

Name

Value

Desc

Name

Value

8. Click Save Changes, then Click the Restart icon.
9. On the Stale Configurations page, click Restart Stale Services.
10. On the Restart Stale Services page, select the Re-deploy client configuration checkbox, then click Restart Now.
11. A progress indicator page appears while the services are being restarted. When the services have restarted, click Finish.

Related Information

[Installing Ranger RMS](#)

Configure Hive-HDFS ACL Sync

Ranger Resource Mapping Server (RMS) should be fully configured after installation. This topic provides further information about RMS configuration settings and workflows.

Important configuration information

- Ranger RMS enable HDFS access via Ranger Hive policies. Ranger RMS must be configured with the names of HDFS and Hive services (AKA Repos). In your installation, there may be multiple Ranger services created for HDFS and Hive. These can be seen from the Ranger Admin web UI. RMS ACL sync is designed to work on a specific pair of HDFS and Hive Ranger services. Therefore, it is important to identify those service names before Ranger RMS is installed. These names should be configured during the installation of Ranger RMS. The default value for Ranger HDFS service name is `cm_hdfs`, and for the Ranger Hive service the default name is `cm_hive`.
- Before starting the Ranger RMS installation, ensure that the Hive service identified in the installation above allows the `rangerrms` user select access to all tables in all databases in default, as well as in all security-zones for the Hive service.
- By default, Ranger RMS tracks only external tables in Hive. To configure Ranger RMS to also track managed Hive tables, add the following configuration setting to Ranger RMS.

```
ranger-rms.HMS.map.managed.tables=true
```

- In Cloudera Manager, select HDFS > Configuration > HDFS Service Advanced Configuration Snippet (Safety Valve) for `ranger-hdfs-security.xml`, then confirm the following settings:

```
ranger.plugin.hdfs.chained.services = cm_hive  
ranger.plugin.hdfs.chained.services.cm_hive.impl = org.apache.ranger.chain  
edplugin.hdfs.hive.RangerHdfsHiveChainedPlugin
```



Note: If any of these configurations are changed after Ranger RMS is started and has synchronized with Hive Metastore, the only way to have Ranger RMS use a new configuration is by following these steps:

1. Stop Ranger RMS.
2. Log in to the Ranger RMS database and run delete from `x_rms_mapping_provider`; to remove the only row from this table.
3. Start Ranger RMS.

On restart, Ranger RMS will resynchronize all data from the Hive Metastore. This may take a significant amount of time depending on the number of Hive tables in the Hive Metastore.

Understanding Ranger policies with RMS

At a high level, the Ranger RMS workflow is as follows:

- Ranger policies for the HDFS service are evaluated. If any policy explicitly denies access, access is denied.
- Ranger checks to see if the accessed location maps to a Hive table.

- If it does, Hive policies are evaluated for the mapped Hive table. If there is an HDFS policy allowing access, access is allowed. Otherwise, the default HDFS ACLs determine the access.
- Requested HDFS permission is mapped to Hive permissions as follows:
 - HDFS 'read' ==> Hive 'select'
 - HDFS 'write' ==> Hive 'update' or 'alter'
 - HDFS 'execute' ==> Any Hive permission
- If there is no Hive policy that explicitly allows access to the mapped table, access is denied, otherwise access is allowed.

Appropriate tag policies are considered both during HDFS access evaluation and if needed, during Hive access evaluation phases. Also, one or more log records are generated to indicate which policy, if any, made the access decision.

The following scenarios illustrate how the access permissions are determined. All scenarios assume that the HDFS location is NOT explicitly denied access by a Ranger HDFS policy.

- Location does not correspond to a Hive table.
 - In this case, access will be granted only if a Ranger HDFS policy allows access or HDFS native ACLs allow access. The audit log will show which policy (or Hadoop-acl) made the decision.
- Location corresponds to a Hive table.
 - A Ranger Hive policy explicitly denied access to the mapped table for any of the accesses derived from the original HDFS request.
 - Access will be denied by Hive policy.
 - There is no matching Ranger Hive policy.
 - Access will be denied. Audit log will not specify the policy.
 - Ranger policy masks some columns in the mapped table.
 - Access will be denied. Audit log will show Hive masking policy.
 - Mapped Hive table has a row-filter policy
 - Access will be denied. Audit log will show Hive Row-filter policy.
 - A Ranger Hive policy allows access to the mapped table for the access derived from the original HDFS access request.
 - Access will be granted. If the access was originally granted by HDFS policy, the audit log will show HDFS policy.

Related Information

[Installing Ranger RMS](#)

Hive-HDFS ACL Sync Use Cases

This topic presents a few common use cases for Hive-HDFS ACL Sync.

Use Case 1: RMS Hive policies control access to a table's HDFS directories

Prerequisites:

1. Create a "Customer" Hive table under the default database.
2. Create a "unixuser1" user.
3. User "unixuser1" does not have any policy to allow it access to table "Customer".
4. User "unixuser1" tries to access the Hive data through the hdfs command.

Before setting up RMS:

If HDFS ACLs allow access to the location for Customer table, access will be granted to "unixuser1". The audit log will have "hadoop-acl" as the access enforcer.

After setting up RMS:

Access will not be granted to user "unixuser1". The audit log will not specify denying policy.

Use Case 2: RMS Hive policies propagate tag-based access control on tables to HDFS directories

Prerequisites:

1. Create a "Customer" Hive table under the default database.
2. Create a "unixuser1" user.
3. The tag "SPECIAL_ACCESS" is associated with the "Customer" table.
4. A policy for the tag "SPECIAL_ACCESS" provides Hive select access to "unixuser1".
5. User "unixuser1" tries to read the Hive data through the hdfs command.

Before setting up RMS:

If HDFS ACLs allow access to the location for "Customer" table, access will be granted to "unixuser1". The audit log will have "hadoop-acl" as the access enforcer.

After setting up RMS:

Access will be granted by tag-based policy for "SPECIAL_ACCESS".

Use Case 3: RMS Hive policies propagate tag-based masking on tables and denies access to HDFS directories

Prerequisites:

1. Create a "Customer" Hive table under the default database.
2. Create a "unixuser1" user.
3. The tag "SPECIAL_ACCESS" is associated with the "Customer" table.
4. A policy for the tag "SPECIAL_ACCESS" provides Hive select access to "unixuser1".
5. A masking policy for the "Customer" table is set up so that for "unixuser1" a column "SSN" is redacted.
6. User "unixuser1" tries to read the Hive data through the hdfs command.

Before setting up RMS:

If HDFS ACLs allow access to the location for Customer table, access will be granted to "unixuser1". The audit log will have "hadoop-acl" as the access enforcer.

After setting up RMS:

Access will be denied by the masking policy.

Use Case 4: RMS Hive policies take precedence over HDFS policies

Prerequisites:

1. Create a "Customer" Hive table under the default database.
2. Create a "unixuser1" user.
3. User "unixuser1" has a HDFS policy allowing read access.
4. User "unixuser1" does not have any policy to allow it access to the "Customer" table.
5. User "unixuser1" tries to access the Hive data through the hdfs command.

Before setting up RMS:

Access will be granted by the Ranger HDFS policy.

After setting up RMS:

Access will not be granted to the "unixuser1" user. The audit log will not specify a denying policy.

Hive-HDFS ACL Sync Reference

This topic provides reference information for Hive-HDFS ACL Sync.

Debugging common issues

- To reload Ranger RMS mapping from scratch, perform the following steps.
 1. Stop Ranger RMS.
 2. Log in to the Ranger RMS database and run delete from `x_rms_mapping_provider`; to remove the only row from this table.
 3. Start Ranger RMS.

RMS tables

The following tables are used by Ranger RMS to persist mappings:

- `X_rms_mapping_provider`
- `X_rms_notification`
- `X_rms_resource_mapping`
- `X_rms_service_resource`

Advanced configurations

HDFS plugin side configurations

- `ranger.plugin.hdfs.mapping.hive.authorize.with.only.chained.policies`
 - `true`: Enforce strict Sentry semantics.
 - `false`: If there is no applicable Hive policy, let HDFS determine access.
 - Default setting: `true`
- `ranger.plugin.hdfs.accesstype.mapping.read`
 - A comma-separated list of hive access types that HDFS "read" maps to.
 - Default setting: `select`
- `ranger.plugin.hdfs.accesstype.mapping.write`
 - A comma-separated list of hive access types that HDFS "write" maps to.
 - Default setting: `update,alter`
- `ranger.plugin.hdfs.accesstype.mapping.execute`
 - A comma-separated list of hive access types that HDFS "execute" maps to.
 - Default setting: `_any`
- `ranger.plugin.hdfs.mapping.source.download.interval`
 - The time in milliseconds between mappings download requests from the HDFS Ranger plugin to RMS.
 - Default setting: 30 seconds

Hive service configuration

- `ranger.plugin.audit.excluder.users`
 - This configuration, added in the Hive service-configs, lists the users whose access to Hive or Hive Metastore does not generate audit records. There may be a large number of audit records created when "rangerrms" makes requests to the Hive Metastore when downloading Hive table data. By adding the "rangerrms" user to the comma-separated list of users in this configuration, such audit records will not be generated.

RMS side configurations

Note that changes to any of these requires that RMS is stopped, all rows are deleted from RMS database table "x_rms_mapping_provider" and RMS is restarted. On restart, RMS downloads complete table data from RMS, which may take a significant amount of time depending on the number of tables in HMS.

- ranger-rms.HMS.source.service.name
 - The Ranger HDFS service name (default: cm_hdfs).
- ranger-rms.HMS.target.service.name
 - The Ranger Hive service name (default: cm_hive).
- ranger-rms.HMS.map.managed.tables
 - true – Track managed and external tables.
 - false – Track only external tables.
 - Default setting: false
- ranger-rms.polling.notifications.frequency.ms
 - The time in milliseconds between polls from RMS to HMS for changes to tables.
 - Default setting: 30 seconds