Cloudera Runtime 7.3.1

Configuring and Using Ranger RMS (Hive-S3 ACL Sync)

Date published: 2020-07-28
Date modified: 2024-12-10

https://docs.cloudera.com/

Legal Notice

Contents

Ranger RMS - HIVE-S3 ACL Sync Overview

Ranger Resource Mapping Server (RMS) enables automatic translation of access policies from HIVE to S3. This feature is available only in AWS deployments.

About HIVE-S3 ACL Sync

It is common to have different workloads use the same data – some require authorizations at the table level (Apache Hive queries) and others at the underlying files (Apache Spark jobs). Unfortunately, in such instances you would have to create and maintain separate Ranger policies for both Hive and S3, that correspond to each other.

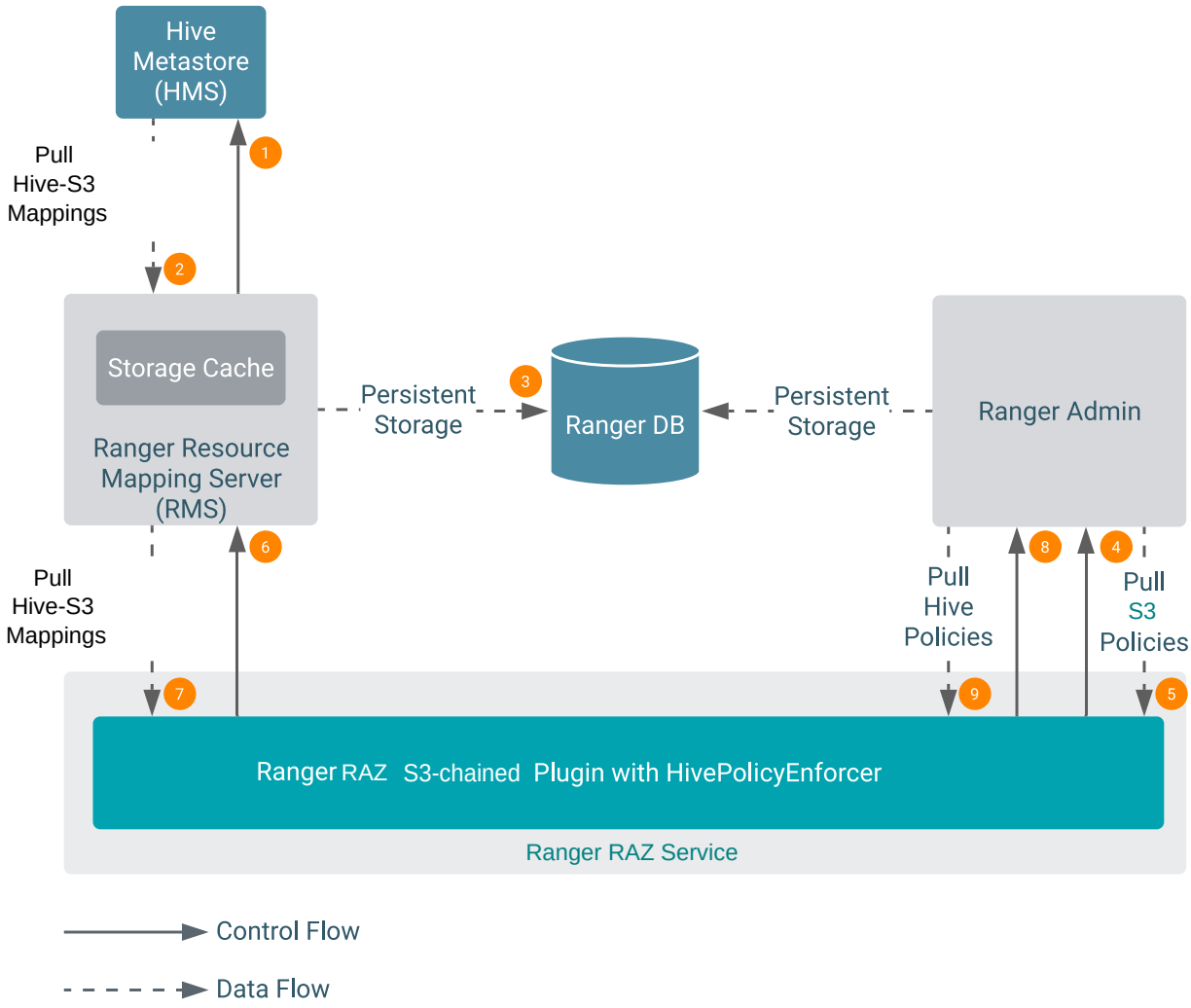
As a result, whenever a change is made on a Hive table policy, the data admin should make a consistent change in the corresponding S3 policy. Failure to do so could result in security and/or data exposure issues. Ideally the data admin would set a single table policy, and the corresponding file access policies would automatically be kept in sync along with access audits, referring to the table policy that enforced it.

Legacy CDH users had a feature called the Hive-HDFS ACL sync which had 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.

Prior to CDP 7.2.18, Ranger only supported manually managing Hive and S3 policies separately. Ranger RMS (Resource Mapping Server) allows you to authorize access to S3 locations using policies defined for Hive tables. RMS is the service that enables Hive-S3 Policy Sync.

RMS periodically connects to the Hive Metastore and pulls Hive metadata (database-name, table-name) to S3 file-name mapping. We have introduced a RAZ-chained plugin (running in the Ranger RAZ service) which has an additional HivePolicyEnforcer module. The RAZ-S3 plugin downloads Hive policies from Ranger Admin, along with the mappings from Ranger RMS. S3 access is determined by both S3 policies and Hive policies.

Figure 1: HIVE-S3 ACL Sync using Ranger RMS



Phase I (items 1-3 above)

Ranger RMS periodically connects to the HIVE Metastore and pulls HIVE metadata (database-name, table-name) to S3 file-name mapping.

Phase II (items 4-9 above)

The Ranger RAZ S3 Chained Plugin (running in the RAZ service) periodically pulls S3 policies from Ranger Admin. With the introduction of Ranger RMS, the Ranger RAZ S3 Chained Plugin (running in the RAZ service) has been extended with an additional HIVEPolicyEnforcer module. It now pulls down the HIVE-S3 mappings from RMS and HIVE Policies from Ranger Admin.

After phase II completes, the requested S3 access is determined in the RAZ service by the S3 and HIVE policies defined by the Ranger Administrator.

About database-level grants feature

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. Specifically, using Sentry, you could make grants at the HIVE database level and HDFS permissions would propagate to the database directory, and to all tables and partitions under it.

In RMS-S3, we have introduced the database-level grant feature also. Ranger Resource Mapping Server (RMS) willl allow you to create a database-level policy in HIVE and have these permissions propagate to the S3 locations and all tables under it. RMS is the service that enables HIVE-S3 ACL Sync.

RMS captures database metadata from the HIVE Meta Store (HMS). After the first, full-synchronization run, RMS downloads mappings for tables and databases present in the HMS.

Whenever you create a new database, RMS synchronizes metadata information from HMS and uses it to update the resource mapping file linking HIVE database resources to their corresponding S3 location. Any user with access permissions on a HIVE database automatically receives similar S3 file-level access permissions on the database's data files. Select/ Read access for any user in the database location is allowed through default HIVE policy for all-databases. This behavior is treated as _any access, which is similar to the HIVE command `show tables`. If a user has no HIVE policy which allows access to the database, then the user is denied access to the corresponding S3 location of that database. Without this feature, users will not be allowed to access the S3 location of a database even if the user had permission to access the database through a HIVE policy. The S3 to HIVE access type mappings follow:

Access Type mapping for S3 to HIVE for Database:

- _any=[_any]
- read=[_any]
- write=[create, drop, alter]

Access Type mapping for S3 to HIVE for Table:

- _any=[_any]
- read=[select]
- write=[update, alter]

If you create tables under a database but the S3 location of the corresponding table does not reside under the S3 location of that database (for example: table locations are external locations), the HIVE policies (database- name, table = *, column= *) translate into S3 access rules and allow the RAZ S3 chained plugin to enforce them. If the policy is created only for the database resource, the same access translates to the S3 location of that database only; not for the tables residing under that database.

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 S3 directory, all partitions of the table must have locations that are under the /warehouse/foo directory.
- In public cloud 7.2.18 or above, RMS service will be available only on AWS deployment for fresh install setup (not for upgrade scenario). A customer with this new RMS entitlement ENABLE_RMS_ON_DATALAKE should be able to create a cluster with RMS as a configurable option (`--enable-ranger-rms`) through a cdp cli command `create-aws-datalake`. When RMS is selected during cluster setup, customers will not be required to install & configure RMS separately.
- The Ranger RMS ACL-sync feature supports a single logical HMS, to evaluate S3 access via HIVE permissions. This is aligned with the Sentry implementation in CDH.
- Permissions granted on views (traditional and materialized) do not extend to S3 access. This is aligned with the Sentry implementation in CDH.
- RMS ACL sync is designed to work on a specific pair of S3<->Hive Ranger service. Ranger RMS supports only one pair of Hive and S3 services. By default cm_s3 is configured as source service and cm_hive as target service.
- If a Public Cloud Base deployment supports multiple logical HMS with a single Ranger, Ranger RMS (Hive-S3 ACL-Sync) works with only one logical HMS. Permissions granted on databases/tables in other logical HMS instances will not be considered to authorize S3 access.
- Ranger RAZ memory requirements must be increased, based on the number of HIVE table mappings downloaded to S3 Ranger plugin. Additionally, maintaining HIVE policies in memory cache will also require additional memory.

- In public cloud deployments, Ranger RMS service will be installed only on DataLake and it will use the same database as Ranger Admin to store mappings downloaded from HMS.
- Ranger RAZ service running in DataHub will have a S3 chained plugin and it will do the authorization based on the policies and mappings downloaded & stored into policy-cache directory of RAZ service. Even if the RMS service is stopped, authorization will continue to work based on the files available in the policy-cache directory.
- Metrics support for RMS with S3 is not added in the CDH-7.2.18 release.
- Expect Ranger RAZ CPU load to increase, due to additional access evaluation performed to enforce HIVE policies and periodic downloading and processing of the HIVE table mappings. The latter increase is proportional to the number of table mappings downloaded to HDFS Ranger plugin.
- When multiple databases are mapped to a single S3 location, and if a HIVE policy allows a user to access one database. Then, users will be able to access its S3 location and all other files & directories under it. This may include table or database directories of other databases and tables. However, users will not be able to access other databases or tables under it through Hive queries.

  For example,

  music_a, music_b, music_c are created at S3 path'/data'.

  Policy-A to allow 'sam' user 'all' access on resource = {database=music_a; table= * ; column= * ; }

  Now, 'sam' user will get all access on S3 path /data and files, directories under it. Therefore, 'sam' user will be able to access S3 location of tables under music_b and music_c databases as long as those locations reside under /data directory.

  However, 'sam' user will not be able to access music_b and music_c databases or any tables under these databases through Hive queries.

  Note: This is the expected behavior when the S3 location for different database/tables are designed in such a way that these entities share the same S3 path or HIVE database/table locations fall under another database S3 location.

Comparison with Sentry HDFS ACL sync

The Ranger RMS (Hive-S3 ACL-Sync) feature resembles the Sentry HDFS ACL Sync feature in the way it downloads and keeps track of the HIVE table to S3 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 Ranger RAZ side, allowing each installation the option of turning this feature on or off.

Understanding Ranger policies with RMS

Ranger RMS for S3 access evaluation workflow

At a high level, the Ranger RMS for S3 access evaluation workflow is as follows:

- Ranger policies for the S3 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. Otherwise, if there is an S3 policy allowing access, then the access is allowed.
  Requested S3 permission is mapped to Hive permissions as follows:

  S3 ‘read’ ==> Hive ‘select’

  S3 ‘write’ ==> Hive ‘update’ or ‘alter’

- 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 S3 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 S3 location is NOT explicitly denied access by a Ranger S3 policy.

Location does not correspond to a Hive table.

In this case, access will be granted only if a Ranger S3 policy allows access. The audit log will show which policy 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 S3 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 RAZ-S3 access request.

- Access will be granted. If the access was originally granted by S3 policy, the audit log will show Hive policy.

How to full sync the Ranger RMS database

About this task

Performing a full-sync in Ranger RMS truncates all the existing data from RMS tables in the Ranger database. The sync process initiates a fresh synchronization from the Hive Metastore (HMS). All available tables and database metadata download from HMS and persist into the Ranger database.

Before you begin

Have a (7.2.18) cluster with Ranger, Ranger RAZ, and Ranger RMS service up and running.

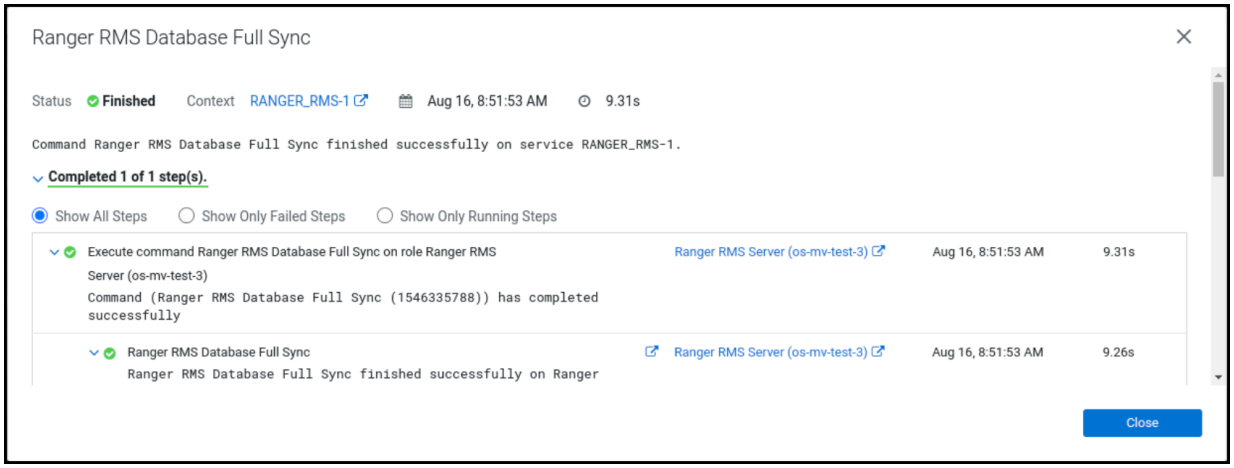
Note: You must have Admin privileges to perform the following steps.

Procedure

1. In Cloudera Manager Ranger_RMS Actions , click Stop.
2. In Ranger_RMS Actions , click Ranger RMS Database Full Sync.

3. Confirm that the command completed successfully, as shown in the following example:

Figure 2: Confirming successful Ranger RMS database full sync



Note: You can also confirm the text - '[I] Truncate operation executed successfully on Ranger RMS tables' from the stderr log of the command from Cloudera Manager UI.

4. In Cloudera Manager Ranger_RMS , click Start.

Configuring Ranger RMS (Hive-S3 ACL Sync)

Ranger Resource Mapping Server (RMS) is fully configured after installing in 7.2.18 public cloud. This topic provides further information about RMS configuration settings and workflows.

Important configuration information - Hive S3

- Ranger RMS enables S3 access via Ranger Hive policies. Ranger RMS is pre-configured with the names of S3 and Hive services (aka Repos). In your installation, there may be multiple Ranger services created for S3 and Hive. These can be seen from the Ranger Admin web UI. RMS ACL sync is designed to work on a specific pair of S3<->Hive- Ranger services. Therefore, it is important to identify those service names before Ranger RMS is installed. The default value for Ranger S3 service name is cm_s3, and for the Ranger Hive service the default name is cm_hive.
- After 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 (no-zone), as well as in all security-zones for the Hive service.
- In case of custom kerberos principal/user, ensure that the Hive service identified in the installation above allows the rangerrmsfoo0 (custom) user select access to all tables in all databases in default (no-zone), as well as in all security-zones for the Hive service.
- In public cloud, Ranger RMS by default tracks both external and managed tables in Hive. To configure Ranger RMS to track only external Hive tables, add the following configuration setting to Ranger RMS.

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

Note:

Locations behind managed tables are granted Read access only, even if the users have Write access at the Hive table level. However, this restriction is not enforced for the hive and impala users.

- To disable RMS for S3 authorization, go to Cloudera Manager Ranger-Ranger RAZ Configuration Advanced Configuration Snippet (Safety Valve) for ranger-raz-conf/ranger-raz-site.xml , then add empty values to the following settings:

```
ranger.raz.service-type.S3.chained.services =
ranger.raz.service-type.S3.chained.services.cm_hive.impl =
```

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 performing a “RMS Full Sync”. Please refer to the related topic, "How to full sync the Ranger RMS database" for specific steps.

Advanced configurations

S3 plugin-side configurations

- ranger.raz.service-type.s3.mapping.hive.authorize.with.only.chained.policies
  - true: Enforce strict Sentry semantics.
  - false: If there is no applicable Hive policy, let S3 determine access.
  - Default setting: false
- ranger.raz.service-type.s3.accesstype.mapping.read
  - A comma-separated list of hive access types that S3 "read" maps to.
  - Default setting: select
- ranger.raz.service-type.s3.accesstype.mapping.write
  - A comma-separated list of hive access types that S3 "write" maps to.
  - Default setting: update,alter
- ranger.raz.service-type.s3.privileged.user.names
  - Default setting: admin,dpprofiler,hue,beacon,hive,impala
  - **Note:** The comma-separated lists that you define for S3 privileged user names and service names are users that, based on default hive policies, have all access for all Hive resources. Therefore, for these users, checking Hive policies when they access storage locations which map to Hive resources is unnecessary, and may cause access violations if masking/row-filtering policies are configured for public group.
- ranger.raz.service-type.s3.mapping.source.download.interval
  - The time in milliseconds between mappings download requests from the S3 Ranger plugin to RMS.
  - Default setting: 30 seconds

    By default, the RMS plugin checks for new mapping downloads every 30 seconds, based on this configuration. If you have mapping data (found in the raz_cm_hive_resource_mapping.json file) of approximately 360MB file size; then performing this operation every 30 seconds could cause an excessive load on the NameNode. After enabling performance logs, we can observe that saveToCache takes 11 seconds and loadFromCache operations take 7 seconds to complete. The cacheing process takes approximately 18~19 seconds to complete, as shown in the following example performance logs:

    ```
    DEBUG org.apache.ranger.perf.resourcemapping.init: [PERF] RangerMappingR
    efresher.loadFromCache(serviceName=cm_hive): 7449
    DEBUG org.apache.ranger.perf.resourcemapping.init: [PERF] RangerMappin
    gRefresher.saveToCache(serviceName=cm_hive): 11787
    ```

    In this case, you should adjust the frequency of download RMS mappings to at least 18*2= 36 seconds. A more conservative value = 45 seconds. In this way, you can tune RMS configurations to optimize performance in the RAZ S3 chained plugin.

Hive service configuration

- ranger.plugin.audit.exclude.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: 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 S3 service name (default: cm_s3).
- 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: true
- ranger-rms.polling.notifications.frequency.ms
  - The time in milliseconds between polls from RMS to HMS for changes to tables.
  - Default setting: 30 seconds
- ranger-rms.supported.uri.scheme
  - A comma-separated list of uri schemes supported by RMS
  - Default setting : s3a

Ranger RMS (Hive-S3 ACL-Sync) Use Cases

This topic presents a few common use cases for Ranger RMS (Hive-S3 ACL-Sync).

Use Case 1: RMS Hive policies control access to a table's S3 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 storage location through the hdfs command.

Before setting up RMS:

If S3 policies configured through Ranger Admin allow access to the location for Customer table, access will be granted to "unixuser1". The audit log will have "ramger-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 S3 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 S3 command.

Before setting up RMS:

If S3 policies configureed through Ranger Admin allow access to the location of "Customer" table, access will be granted to ‘"unixuser1". Otherwise, access is denied.

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 S3 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 S3 policies configured through Ranger Admin allow access to the location of Customer table, access will be granted to "unixuser1". Otherwise, access is denied.

After setting up RMS:

Access will be denied by the masking policy.

Use Case 4: RMS Hive policies take precedence over S3 policies

Prerequisites:

1. Create a "Customer" Hive table under the default database.
2. Create a "unixuser1" user.
3. User "unixuser1" has a S3 policy allowing read access.
4. User "unixuser1" has a Hive 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 S3 policy.

After setting up RMS:

Access will be granted to the "unixuser1" user through the Hive policy. The audit log should display the same.