

Trino security and governance

Date published:

Date modified:

CLOUDEXERA

Legal Notice

© Cloudera Inc. 2026. 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 authorization for Trino Virtual Warehouses..... 4**
 - Authorizing the Hive user for Trino access to Ozone.....4
- Adding a user to Ranger service policies..... 5**
- cm_trino service user impersonation behavior..... 6**
- Using resource-based authorization policies for Trino..... 6**
- Using resource-based column masking for Trino.....7**
- Using row-level filtering for Trino.....8**

Ranger authorization for Trino Virtual Warehouses

Authorization for Trino is supported through Apache Ranger by default and you can create or update Ranger policies for specific resources and assign permissions to Trino users, groups, or roles. When a user submits a query to Trino, the system verifies the defined policies to ensure that the user has the necessary permissions to run queries.

Authorization is the process that checks user permissions to perform select operations, such as creating, reading, and writing data, as well as editing table metadata.

Trino supports the following Ranger features:

- Resource-based authorization policies
- Resource-based column masking
- Row-level filtering

Trino authorization service (cm_trino)

Ranger authorization for all the Trino supported connectors is offered through the cm_trino resource-based service.

For each resource specified in a Hive policy, an "allow" policy must also be configured in the cm_trino service to enable successful access control verification. You must be aware that there are two services where Hive authorization policies are defined and therefore you will notice two sets of audit logs; one from the cm_trino service and another from the HadoopSQL service (cm_hive). However, the cm_trino service allows for more restrictive permission as compared to HadoopSQL.

Access Request Audit

Audit log entries are created for requests, detailing the request information and the authorization outcome (allow or deny). These logs can be viewed through the Audits service in the Ranger Admin Web UI.

Authorizing the Hive user for Trino access to Ozone

Trino uses the Hive user to interact with the Ozone file system. To ensure secure data access, you must authorize the Hive user in your Ranger policies. This configuration allows Trino to perform authenticated read and write operations on Ozone data volumes and buckets.

Procedure

1. In Cloudera Management Console, click **Clusters Ranger Ranger Admin Web UI**.
2. Enter your username and password, and then click **Sign In**.
The Ranger Service Manager page for Cloudera services is displayed.
3. From the Service Manager page, click the cm_ozone service.
4. Edit the volume and bucket policies to add the hive user.
The Policy Details page is displayed.
5. On the Policy Details page, go to the Allow Condition section.
6. From the Select User drop-down menu, select hive.
7. Click **Save**.

Results

The hive user is added to the volume and bucket policies.

Adding a user to Ranger service policies

Learn how to add a logged-in user or resource owner, who is not already part of the required groups, to the relevant Ranger policies.

Procedure

1. In Cloudera Manager, click **Clusters Ranger Ranger Admin Web UI**, enter your username and password, and then click **Sign In**.
The **Ranger Service Manager** page for Cloudera services is displayed.
2. From the **Service Manager** page, click the **cm_trino** service.
3. In the **TRINO Policies** page of the **cm_trino** service, click **edit** against the following policies and include the logged-in user or resource owner in the **Allow Conditions**:
 - all – trino user
 - all – catalog, schema, table
 - all – catalog, schema
 - all – sysinfo
 - all – catalog, schema, procedure
 - all – catalog, schema, schemafunction
 - all - function
 - all - queryid
 - all - role
 - all – catalog, schema, table, column
 - all – catalog
 - all – catalog, sessionproperty
 - all – systemproperty
4. Click **Save** to apply the changes.
5. Return to the **Service Manager** page and click the **Hadoop SQL** service to modify the default policies.
6. In the **HIVE policies** page of the **Hadoop SQL** service, edit each of the following default Hive policies to include the logged-in user or resource owner in the **Allow Conditions**:
 - all - global
 - all - database, table, column
 - all - database, table
 - all - storage-type, storage-url
 - all - database
 - all - hiveservice
 - all - database, udf
 - all - url
7. Click **Save** to apply the changes.
8. Return to the **Service Manager** page and click the **cm_hdfs** service to modify the default policies.
9. In the **HDFS Policies** page of the **cm_hdfs** service, edit the following default policy to include the logged-in user or resource owner in the **Allow Conditions**:
 - all - path
10. Click **Save** to apply the changes.

Results

The logged-in user or resource owner now has the necessary permissions for the HDFS, Hadoop SQL, and Trino services.

cm_trino service user impersonation behavior

The Ranger Trino authorization service (cm_trino) is configured with a default security posture that is too permissive. Learn how you can review and reinforce these settings to prevent unauthorized user impersonation.

By default, the Ranger configuration parameter, ranger.default.policy.groups is populated with a specific administrative group (_c_ranger_admins_...). As a result, the default Ranger policy, "all - trino user" allows all users within this administrative group to impersonate any other user.

In Production environments, it is recommended that you perform the following recommendations to avoid a relaxed security posture and restrict user impersonation:

Service-level impersonation

Only specific technical service users (typically prefixed with srv_) require the ability to act on behalf of any user. You can modify the "all - trino user" Ranger policy to include only these specific users or groups.

User self-impersonation

Trino requires an explicit policy to allow an authenticated user to act on their own behalf. To ensure a secure posture, you can create a self-impersonation policy in the Ranger Trino service.

Under this policy, you must set the Trino User resource specifically to "{USER}" rather than using the wildcard (*). This is the key difference between a secure setup and the permissive "all - trino user" policy. Additionally, under the **Allow Conditions** pane, set the Select User to "{USER}" and grant impersonate permissions. This configuration ensures that each user is authorized to impersonate only themselves and cannot act on behalf of other users in the cluster.

Using resource-based authorization policies for Trino

The Apache Ranger access policy model consists of two major components: the specification of the resources a policy is applied to, such as Catalog, Schema, tables, columns, and so on; and the specification of access conditions for specific users and groups.

About this task

The following table lists the allowed permissions and the actions that they support in Trino:

Permission	Action
Select	Provides read access to an object (table, view, and materialized view)
Insert	Provides the ability to insert records into a table
Update	Provides the permission to modify column values in existing rows of a table
Delete	Provides the ability to remove records from a table
Create	Provides the permission to create an object (schema, table, view, and materialized view)
Alter	Provides the ability to: - Rename a schema - Rename a table or column - Add or drop a column - Set session property

Permission	Action
Drop	Provides the permission to drop an object (schema, table, view, and materialized view)  Note: For Hive connectors, the DROP SCHEMA statement with the CASCADE option also checks the table permissions before dropping the schema.
Use	Update the session to use the specified catalog and schema
Grant	Assign privileges to users on an object
Revoke	Deny privileges to users on an object
Show	Provides the ability to: - List the existing objects (catalogs, schemas, tables and views) - Show the SQL statement that creates a schema, table, or view
Excute	Provides the permission to run a SQL statement



Important: Currently, the Grant and Revoke permissions do not modify Ranger policy definitions.

Perform the following steps to set up the required authorization policies for Trino:

Procedure

1. In Cloudera Manager, click Clusters Ranger Ranger Admin Web UI , enter your username and password, and then click Sign In.
The **Ranger Service Manager** page for Cloudera services is displayed.
2. Click the cm_trino resource-based service to create or modify a Trino policy.
3. In the **Trino Policies** page, under the **Access** tab, click Add New Policy to create a new Trino policy.
The **Create Policy** modal displays controls for creating details for a new policy.
4. Enter a name for the policy and choose the Trino catalog, schema, tables, and columns that you want to give access to.
5. In the **Allow Conditions** pane, add the "trino" user, and then click **+** to choose the necessary permissions.
6. Click Save to save the policy.

Using resource-based column masking for Trino

You can use Apache Ranger dynamic resource-based column masking capabilities to protect sensitive data. You can set policies that mask or anonymize sensitive data columns (such as PII, PCI, and PHI) dynamically from the Trino query output. For example, you can mask sensitive data within a column to show only the first or last four characters.

You can create a masking policy from the **Masking** tab in the **Trino Policies** page. You can then set filters for specific users, groups, roles, and choose the masking options.

The following masking options are supported:

Mask Option	Description
Redact	Replace lowercase with 'x', uppercase with 'X', and digits with '0'
Partial mask: show last 4	Show last 4 characters; replace rest with 'x'

Mask Option	Description
Partial mask: show first 4	Show first 4 characters; replace rest with 'x'
Hash	Hash the value
Nullify	Replace with NULL
Date: show only year	Show only the year
Custom	Specify a custom masked value or expression

**Important:**

- The Hadoop SQL service (cm_hive) supports all data types for all column mask options, except for the "Hash" option.
- The Ranger Trino service (cm_trino) supports column masking only for varchar data types.

Using row-level filtering for Trino

Row-level filtering policies are similar to other Ranger access policies. Apache Ranger row-level filtering policy allows to set access policies for rows when reading from a table.

You can use Apache Ranger row-level filtering policies to set access policies for rows when reading from a table. You can set filters for specific users, groups, and conditions. For example, only employees from the sales department may access sales transaction records, payments, and related data.

You can create a row-level filter policy from the **Row Level Filter** tab in the **Trino Policies** page. You can then set filters for specific users, groups, roles, and specify a filter expression. The filter expression must be a valid WHERE clause for the table or view, such as `employeeId='123abc'`.



Note: You can add multiple filters and the filters are evaluated in the order listed in the policy. The filter at the top of the Row Filter Conditions list is applied first, then the second, then the third, and so on.