

# Atlas Use Case: Controlling Data Access Using Tags

Date published: 2019-09-23

Date modified: 2020-10-13



# 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.

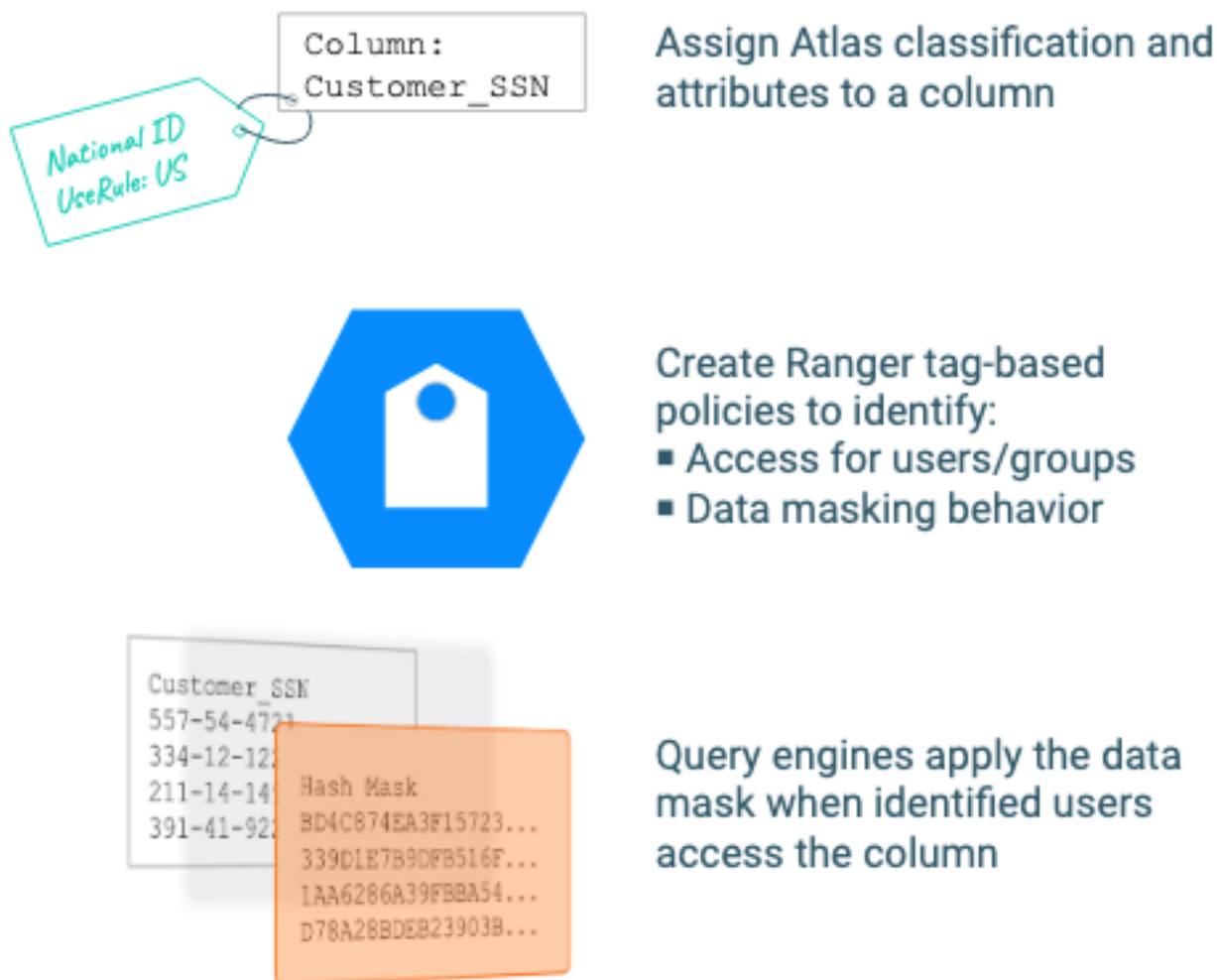
# Contents

<b>Atlas classifications drive Ranger policies.....</b>	<b>4</b>
<b>When to use Atlas classifications for access control.....</b>	<b>5</b>
<b>How tag-based access control works.....</b>	<b>5</b>
<b>Examples of controlling data access using classifications.....</b>	<b>6</b>

## Atlas classifications drive Ranger policies

Ranger policies can use tags to identify data; Atlas classifications are pulled into Ranger as tags.

You can use Atlas classifications to drive data access control through Ranger. Ranger offers both resource-based and tag-based access control policies. Using metadata tags rather than specific resource names gives you flexibility and allows access controls to apply immediately to new data assets without requiring administrator intervention.



You can use Atlas classifications to control user access to data assets by using Atlas classifications to define Ranger tag-based access-control policies. Ranger tag-based policies ensure that services such as Hive and Impala control user and group access to specific data assets. Ranger policies can have services apply masks to column data, so users see results such as partial data or anonymized data. To make this work, you define classifications in Atlas; associate the classifications with data assets, including databases, tables, views, and columns; then define policies that act on the data assets tagged with the Atlas classifications.

Some of the ways you can use classifications include:

- Add attributes to Atlas classifications to define separate behaviors for separate contexts. For example, you could mark columns as “National ID” and apply a policy based on that information. You could add an attribute to the “National ID” classification that describes which the rules to apply to the display of the national ID, such as “Apply Rules From”: “EU” or “Apply Rules From”: “JPN”). Ranger policies can use the attribute value to apply different mask patterns to the data.

- Atlas lineage can propagate classifications from one column to columns created later from the same data. When classifications are propagated, the Ranger policies built on these classifications apply to the new location of the data. No intervention is required to ensure that the access controls on the original data are applied to the new copy.
- If Ranger is set up to deny access to new data except for the owner, you can use tags to reveal this data (access only to classified data).

### Related Information

[When to use Atlas classifications for access control](#)

[How tag-based access control works](#)

[Examples of controlling data access using classifications](#)

## When to use Atlas classifications for access control

Resource-based and tag-based policies are useful in different ways.

Ranger provides resource-based policies and tag-based policies. The following table provides some examples of when you would choose one type of policy over the other:

### Resource-based Policies

Control access to data assets per service type (multiple policies for each data asset)

Control access to entire databases

Control access to long-lived tables

Control access to well-known columns in specific tables, which don't change over time

### Tag-based Policies

Control access to data assets across all service types

Control access to columns in source tables that users can copy or transform to other tables

Control access to data until it is reviewed/classified by setting an validity date

### Related Information

[Ranger tag-based policies](#)

[Resource-based services and policies](#)

## How tag-based access control works

Do some prep in Atlas to make tags available for creating Ranger policies.

Follow these steps to set up tag-based access control in your environment:

1. Determine what data to control, who it's controlled for, and how you want to control it.

If you know the data characteristics but there isn't a reliable column name for the data or if you want to reveal parts of the data to some users, assign a classification to the column and set a tag-based policy in Ranger to apply masks to the data.

- Same resource across multiple services. Set tag-based policies in Ranger. Note that resource-based policies apply to a single service.
- Entire databases. Set resource-based policies in Ranger.
- Tables. Set resource-based policies in Ranger.
- Columns. Tagging columns in Atlas and then creating tag-based policies in Ranger allows you to control access to this data even as it is transformed into other tables.

2. Create classifications in Atlas that describe triggers for when data should be controlled.

3. Assign the classifications to the Atlas data assets.
4. Create “tag based policies” in Ranger.
5. Use Hue or Zeppelin to validate that the policies work as expected.

**Related Information**[Ranger tag-based policies](#)[Resource-based services and policies](#)[When to use Atlas classifications for access control](#)[Examples of controlling data access using classifications](#)

## Examples of controlling data access using classifications

Some of the ways you can use classifications to control access to data.

Use classifications to control data:

- Validity period or expiration date.
- Sensitive data masking
- Segregating access privileges by department or region

**Related Information**[Adding a tag-based PII policy](#)[Default EXPIRES ON tag policy](#)[Dynamic tag-based column masking in Hive with Ranger policies](#)