Cloudera Runtime 7.2.13

Ranger Authorization

Date published: 2020-07-28
Date modified: 2021-12-13

https://docs.cloudera.com/

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 to Provide Authorization in CDP....5

Ranger plugin overview....5
Ranger Hive Plugin....5
Ranger Kafka Plugin....7

Ranger special entities....8

Enabling Ranger HDFS plugin manually on a Data Hub....9

Ranger Policies Overview....11
Ranger tag-based policies....12
Tags and policy evaluation....13
Ranger access conditions....14

Using the Ranger Console....16
Accessing the Ranger console....17
Ranger console navigation....18

Resource-based Services and Policies....21
Configuring resource-based services....21
Configure a resource-based service: Atlas....22
Configure a resource-based service: HBase....23
Configure a resource-based service: HDFS....25
Configure a resource-based service: HadoopSQL....27
Configure a resource-based service: Kafka....29
Configure a resource-based service: Knox....31
Configure a resource-based service: NiFi....33
Configure a resource-based service: NiFi Registry....35
Configure a resource-based service: Solr....37
Configure a resource-based service: YARN....39
Configuring resource-based policies....41
Configure a resource-based policy: Atlas....42
Configure a resource-based policy: HBase....43
Configure a resource-based policy: HDFS....46
Configure a resource-based policy: HadoopSQL....48
Configure a resource-based storage handler policy: HadoopSQL....52
Configure a resource-based policy: Kafka....56
Configure a resource-based policy: Knox....57
Configure a resource-based policy: NiFi....59
Configure a resource-based policy: NiFi Registry....61
Configure a resource-based policy: S3....63
Configure a resource-based policy: Solr....65

Configure a resource-based policy: YARN ... 67
Wildcards and variables in resource-based policies ... 69
Preloaded resource-based services and policies ... 70
Importing and exporting resource-based policies ... 76
Import resource-based policies for a specific service ... 78
Import resource-based policies for all services ... 80
Export resource-based policies for a specific service ... 83
Export all resource-based policies for all services ... 84
Row-level filtering and column masking in Hive ... 86
Row-level filtering in Hive with Ranger policies ... 86
Dynamic resource-based column masking in Hive with Ranger policies ... 90
Dynamic tag-based column masking in Hive with Ranger policies ... 94

Tag-based Services and Policies ... 98
Adding a tag-based service ... 98
Adding tag-based policies ... 99
Using tag attributes and values in Ranger tag-based policy conditions ... 102
Adding a tag-based PII policy ... 104
Default EXPIRES ON tag policy ... 108
Importing and exporting tag-based policies ... 111
Import tag-based policies ... 113
Export tag-based policies ... 115

Create a time-bound policy ... 117

Create a Hive authorizer URL policy ... 119

Showing Role|Grant definitions from Ranger HiveAuthorizer ... 122

Ranger Security Zones ... 123
Security Zones Administration ... 123
Security Zones Example Use Cases ... 124
Adding a Ranger security zone ... 126

Administering Ranger Reports ... 131
View Ranger reports ... 132
Search Ranger reports ... 132
Export Ranger reports ... 133

Using Ranger client libraries ... 134

Using session cookies to validate Ranger policies ... 135

Configure optimized rename and recursive delete operations in Ranger Ozone plugin ... 136

Using Ranger to Provide Authorization in CDP

Apache Ranger manages access control through a user interface that ensures consistent policy administration across Cloudera Data Platform (CDP) components. Security administrators can define security policies at the database, table, column, and file levels, and can administer permissions for specific LDAP-based groups or individual users. Rules based on dynamic conditions such as time or geolocation can also be added to an existing policy rule. The Ranger authorization model is pluggable and can be easily extended to any data source using a service-based definition.

Once a user has been authenticated, their access rights must be determined. Authorization defines user access rights to resources. For example, a user may be allowed to create a policy and view reports, but not allowed to edit users and groups. You can use Ranger to set up and manage access to Hadoop services.

Ranger enables you to create services for specific resources (HDFS, HBase, Hive, etc.) and add access policies to those services. Ranger security zones enable you to organize service resources into multiple security zones. You can also create tag-based services and add access policies to those services. Using tag-based policies enables you to control access to resources across multiple components without creating separate services and policies in each component. You can also use Ranger TagSync to synchronize the Ranger tag store with an external metadata service such as Apache Atlas.

Note:

You can configure authorization using the Ranger UI, REST APIs, or client libraries. For more information about:

- Ranger REST APIs, see see https://ranger.apache.org/apidocs/index.html.
- Ranger client libraries, see Using Ranger client libraries.

Ranger plugin overview

Ranger enforces authorization using a plugin model.

Ranger at the core has a centralized web application, which consists of the policy administration. These policies are enforced within the Hadoop ecosystem using lightweight Ranger Java plugins. These plugins run as part of the same process as the namenode (HDFS), HiveServer2(Hive), HiveMetaStore, HBase server (Hbase), Kafka, Solr, NiFi, Raz, RazS3, ADLS, Yarn and Knox server (Knox). Plugins are enabled by default for each of these components except (Solr) and can be disabled individually, using Cloudera Manager.

Ranger plugins exist in the path of the user request. Each plugin decides whether to allow or deny user requests for accessing. Each plugin also collects and stores the access request details as access audit log records.

Ranger plugins enforce the policies defined in the policy database. Ranger Admin users can create a policy for a specific set of resources and assign a specific set of permissions to a specific set of users, groups and roles. Ranger admin users manage policies using the Ranger Admin Web UI.

Ranger policies are independent from native permissions (os permission). Ranger uses native permissions to authorize user access in the case that an applicable Ranger policy does not exist in the policy database.

Ranger Hive Plugin

Describes how the Ranger Hive plugin enforces authorization.

Ranger Hive Plugin is enabled in HiveServer2 which helps in storage-based authorization and SQL-standard authorization. In storage-based authorization when a new table is created by running CREATE TABLE statement in Beeline, which will submit query to HiveServer2 for processing, and before HiveServer2 is able to run the query, it will check the policy cache file and make sure the user who submits the query has the appropriate permission to perform the task. Once the authorization passes, a query is submitted and a table created.

Upon successful creation of the new table, two things will be triggered by Ranger's Hive plugin:

1. Sends audit event to Solr and/or HDFS
2. Sends Kafka event to topic "ATLAS_HOOK", to record that a new entity has been created, so effectively Ranger's Hive Plugin is the producer for "ATLAS_HOOK" topic in Kafka

SQL standard authorization provides grant/revoke functionality at database, table level. When a grant command is executed in beeline it updates/creates a policy for that user and when revoke is executed the user is added in the deny condition of the policy.

Ranger Hive Plugin Enforcement Example

Prerequisite

1. Create a database, table, column in hive service and also insert some data into it with hive user.
   - create database vehicle;
   - create table vehicle.cars(car_id int, car_name string, car_color string, car_price int);"
   - insert into vehicle.cars(car_id, car_name, car_color, car_price) VALUES (1,'car1','color1',100000), (2,'car2','color2',200000), (3,'car3','color3',300000), (4,'car4','color4',400000);
   - select * from vehicle.cars;
2. Create external user 'externaluser1'

Access Enforcement steps

1. Let's try to access the vehicle.cars table using 'externaluser1'.

   'externaluser1' will be denied access, because 'externaluser1' lacks permission to access the vehicle.cars table.
2. Lets create a policy in ranger-hive for the user:
   - Resource : [database=vehicle, table=cars, column=*]
   - allow policy item : [user='externaluser1', permission=select]
3. Let's try to access the vehicle.cars table using 'externaluser1'.

   'externaluser1' will be allowed access, because 'externaluser1' now has permission to access the vehicle.cars table.
4. You can check the logs related to these actions, using Ranger Admin Web UI Access Audit tab.

Masking Enforcement steps

Suppose you don't want to show the car_price to 'externaluser1' user so we can mask the data of that column for that user.

1. Lets create a masking policy in ranger-hive for the user:
   - Resource : [database=vehicle, table=cars, column=car_price]
   - allow policy item : [user='externaluser1', permission=select, Select Masking Option=Partial mask: show last 4]
2. Now let's try to access the vehicle.cars table using 'externaluser1'

   'externaluser1' will see the car_price - only last 4 digits - because 'externaluser1' has masked access to vehicle.cars table.

Row Enforcement steps

Suppose you don't want to show the only one row to 'externaluser1' user so we can do it using the row filter policy.

1. Lets create a masking policy in ranger-hive for the user:
   - Resource : [database=vehicle, table=cars]
   - allow policy item : [user='externaluser1', permission=select, Row Level Filter=car_color = 'color4']
2. Now let's try to access the vehicle.cars table using 'externaluser1'.

   'externaluser1' will see only the row whose car_color is 'color4'.

Table 1: Hive Commands to Ranger Permission Mapping

Permission	Action
SELECT	Gives read access to an object.
CREATE	Hive Create Table statement is used to create table.
UPDATE	Gives the ability to run update queries on an object (table).
ALTER	You can rename the table and column of existing Hive tables. You can add a new column to the table. Rename Hive table column. Add or drop table partition. Add Hadoop archive option to Hive table.
DROP	DROP TABLE command in the hive is used to drop a table inside the hive.
INDEX	An Index is nothing but a pointer on a particular column of a table. Creating an index means creating a pointer on a particular column of a table.
LOCK	Is used to lock the table.
Read	Read data from HDFS using hdfs or other cloud locations.
Write	Export Data to a location in hdfs or other cloud locations.
ReplAdmin	ReplAdmin privilege is related to REPL DUMP and REPL LOAD commands.
Service Admin	Enable hive ranger plugin to isolate various admin operations, in this case "Kill Query". "Service Admin" won't be able to do DATABASE / TABLE / COLUMN operations as this will all be taken care by the existing DATABASE/TABLE/COLUMN level permission model.
Temporary UDF Admin	Temporary UDF Admin is needed for creating UDFs.
Refresh	Refresh is used by only impala.
ALL	This is for all the permission mentioned above.

Ranger Kafka Plugin

Describes how the Ranger Kafka plugin enforces authorization.

Ranger Kafka plugin is enabled in master.

Ranger Kafka Plugin Enforcement Example

Prerequisite

1. Create external user 'externaluser3'

Access Enforcement steps

1. Let's try to create a topic and send some data using ‘externaluser3’, he will be denied as he doesn't have permission to create it.
2. Lets create a policy in ranger-hive for the user
   - Resource : [Topic=topictest01]
   - allow policy item : [user='externaluser3', permission=publish, consume, describe, create]
3. Let's try to create a topic and send some data using ‘externaluser3’, he will be allowed as he gets permission to access it.
4. You can check the logs related to these actions, using Ranger Admin Web UI Access Audit tab.

Table 2: Kafka Commands to Ranger Permission Mapping

Permission	Action
Resource = topic	
Publish, Describe, Create	To produce topic and publish
Describe, Create	To describe topic
Describe	sending message to topic
Publish	To publish topic
Consume	To read data (consume)
Describe	To list topic
Configure	To alter config of topic
Delete	To delete topic
Describe Config	To describe config of topic
Alter Config	To alter config
Resource = consumergroup	
Describe	To describe topic
Consume	To consume topic
Resource = cluster	
Create	To create topic
Describe	To describe topic
Idempotent Write	To write idempotently
Resource = transactionid	
Describe, Publish	To publish and describe

Ranger special entities

Ranger in CDP has specific, internal groups and entities that affect user authorization and access to all services in CDP.

In addition to any users, group, roles and permissions that you define using Ranger, you must understand the following Ranger special entities:

"public" group

A special, internal group within Ranger that consists of all users, including future users. Membership is implicit and automatic.

Note: All users belong to "public" group. Any policies granted to this group provide access to everyone.

The following, default policies give permissions to members of group "public":

- all - database > public > create permission
- default database tables columns > public > create permission
- Information_schema database tables columns > public > select permission

You can remove “public” from these default policies to further restrict user access, based on your security requirements.

{OWNER} special entity

A special Ranger entity attached to a user based on their actions. For example, when a user "bob" creates a table, "bob" becomes the {OWNER} of that table and would get any permissions provided to {OWNER} on that table across all the policies. The following default policies have permissions for {OWNER}:

- all - database, table, column > {OWNER} > all permissions
- all - database, table > {OWNER} > all permissions
- all - database, udf > {OWNER} > all permissions
- all - database > {OWNER} > all permissions

Although not recommended, you can modify access to special entity {OWNER}, based on your security requirements. Removing the default {OWNER} permissions may require adding additional, specific policies for each object owner, which may increase your policy management operational burden.

Enabling Ranger HDFS plugin manually on a Data Hub

How to enable an HDFS plugin for Ranger, service-wide, on a Data Hub using Cloudera Manager.

About this task

The Ranger HDFS plug-in helps to centralize HDFS authorization policies. Apache Ranger plugins validate the access of a user against the authorization policies defined in the Apache Ranger policy administration server, and stored in the HDFS service instance, also called a repository. When you enable the Ranger HDFS plugin and an HDFS service user attempts access, Ranger checks whether a policy exists granting or denying the user access. If no policy exists, Ranger defaults to use the native permissions model in HDFS. Access control rules configured through this combination of Ranger HDFS plugin and native file system permissions apply.

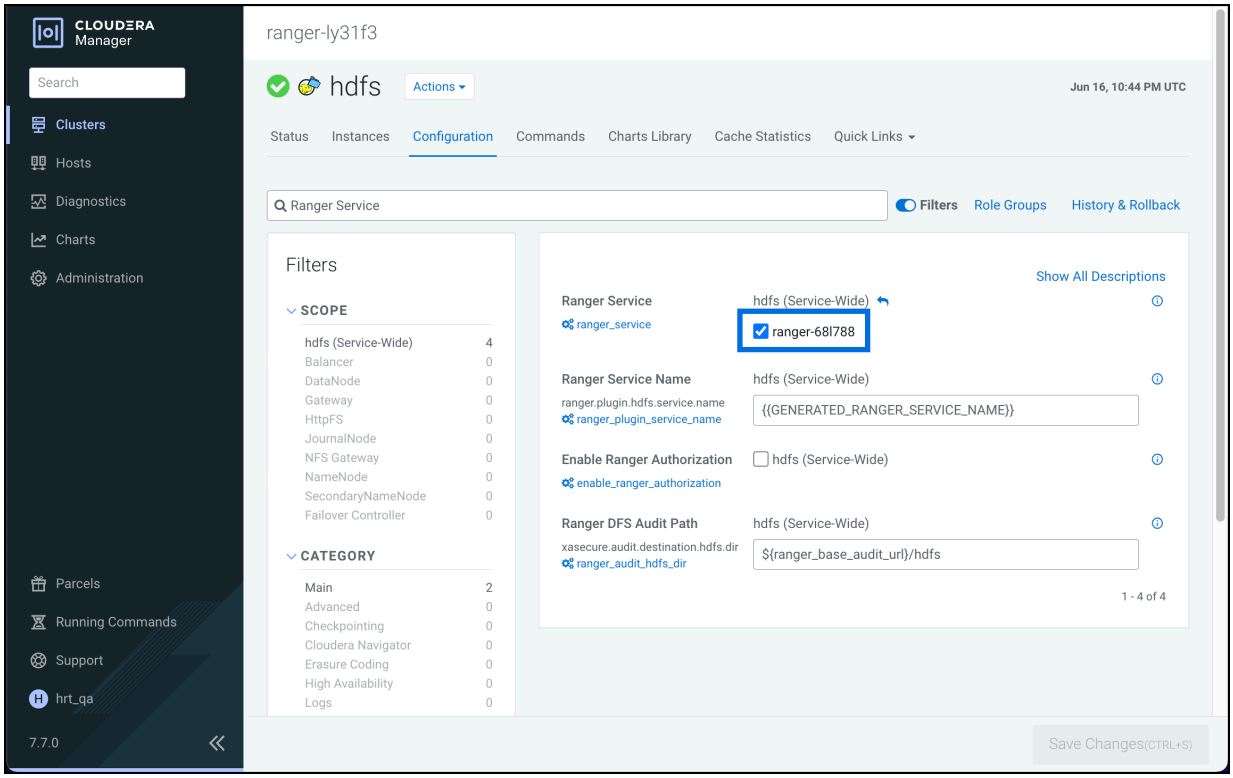
To enable users define Ranger authorization polices, using an HDFS service plugin:

Procedure

1. In a DataHub, go to Cloudera Manager HDFS Configuration .

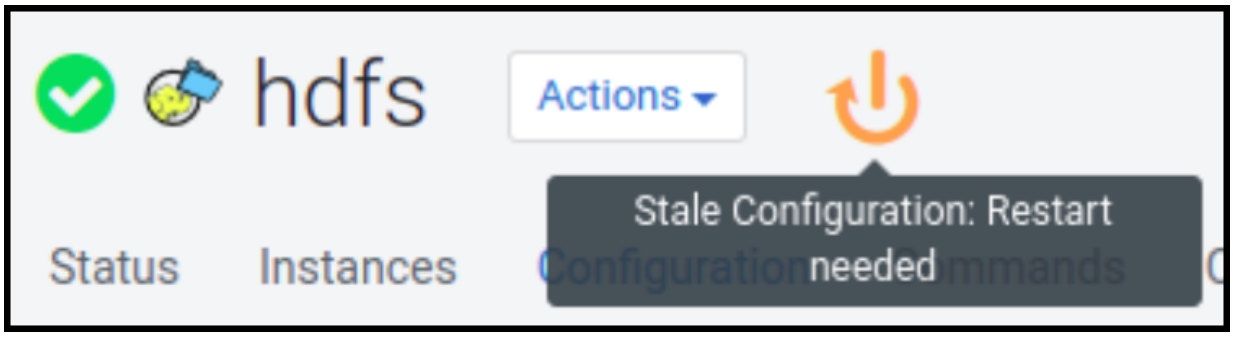
2. In Search, type Ranger Service, then click the box to enable the hdfs (service-wide) parameter for Ranger Service.

Figure 1: Enabling the HDFS Ranger plugin parameter on a Data Hub



A stale configuration icon displays for the hdfs service.

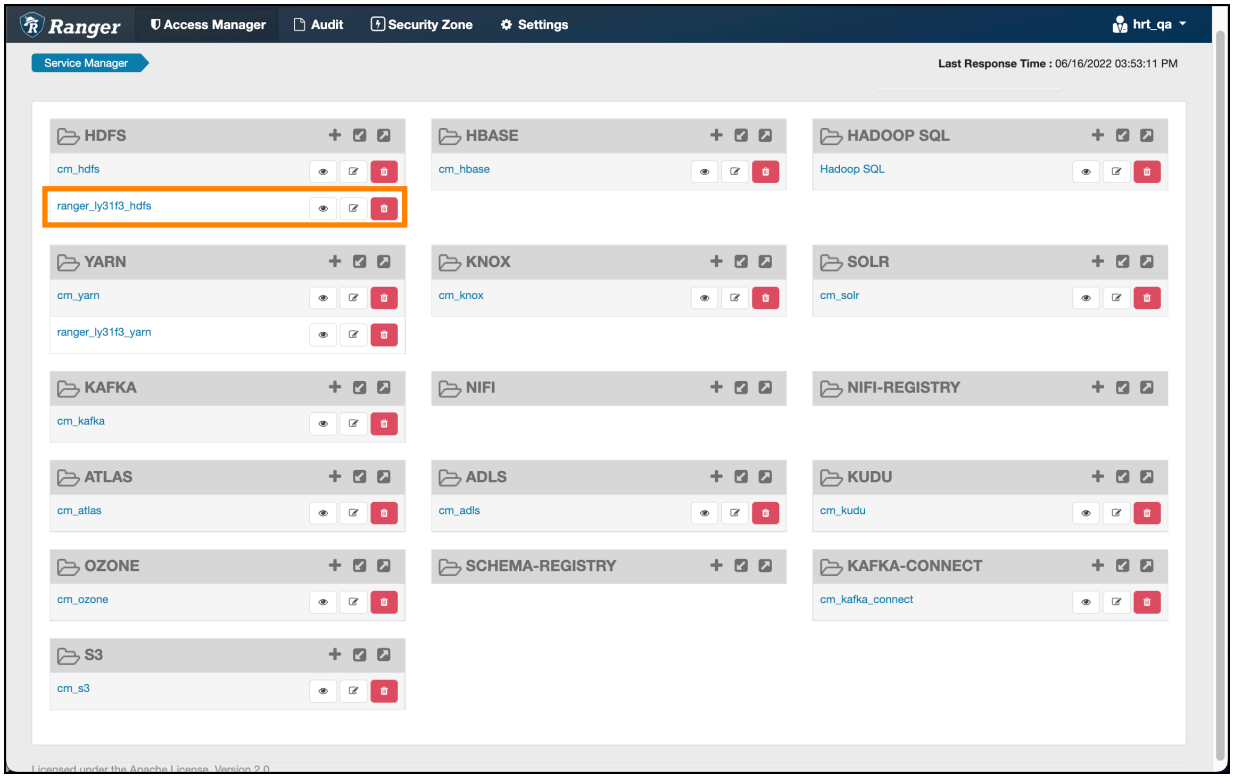
Figure 2: Result of adding a new configuration parameter



3. Before restarting Hdfs service for stale configurations, choose HDFS Actions Create Ranger Repository . After progress completes, close the Create Repository dialog.
4. Now proceed to restart the Hdfs service. Click HDFS Actions Restart . After progress completes, close the Restart dialog.
5. On the Data Lake, log in to Ranger.
6. Go to Admin Web UI Access Manager .

7. In Service Manager HDFS , confirm that (DataHub cluster name)_hdfs appears.

Figure 3: Confirming HDFS plugin added

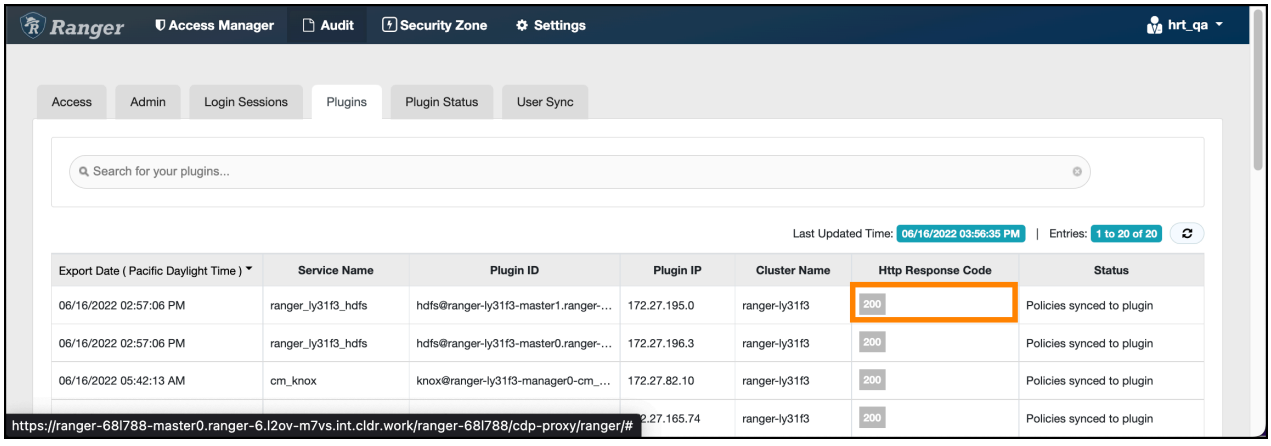


8. Go to Audit Plugins .

Results

Confirm that the Http response code for the Ranger Hdfs plugin enabled on the DataHub Hdfs service displays 200 (successful).

Figure 4: Confirming successful http response



Ranger Policies Overview

Ranger has two types of policies: resource-based and tag-based.

Resource-based policies

Ranger enables you to configure resource-based services (HDFS, HBase, Hive, etc.) and add access policies to those services.

Tag-based policies

Ranger enables you to create tag-based services and add access policies to those services.

Ranger tag-based policies

Ranger enables you to create tag-based services and add access policies to those services.

Tag-Based Policies Overview

- An important feature of Ranger tag-based authorization is the separation of resource-classification from access-authorization. For example, resources (HDFS file/directory, Hive database/table/column etc.) containing sensitive data such as social security numbers, credit card numbers, or sensitive health care data can be tagged with PII/PCI/PHI – either as the resource enters the Hadoop ecosystem or at a later time. Once a resource is tagged, the authorization for the tag would be automatically enforced, thus eliminating the need to create or update policies for the resource.
- Using tag-based policies also enables you to control access to resources across multiple Hadoop components without creating separate services and policies in each component.
- Tag details are stored in a tag store. Ranger TagSync can be used to synchronize the tag store with an external metadata service such as Apache Atlas.

Tag Store

Details of tags associated with resources are stored in a tag store. Apache Ranger plugins retrieve the tag details from the tag store for use during policy evaluation. To minimize the performance impact during policy evaluation (in finding tags for resources), Apache Ranger plugins cache the tags and periodically poll the tag store for any changes. When a change is detected, the plugins update the cache. In addition, the plugins store the tag details in a local cache file – just as the policies are stored in a local cache file. On component restart, the plugins will use the tag data from the local cache file if the tag store is not reachable.

Apache Ranger plugins download the tag details from the store managed by Ranger Admin. Ranger Admin persists the tag details in its policy store and provides a REST interface for the plugins to download the tag details.

Tags

Ranger Tags can have attributes. Tag attribute values can be used in Ranger tag-based policies to influence the authorization decision.

For example, to deny access to a resource after a specific date:

1. Add the EXPIRES_ON tag to the resource.
2. Add an exipry_date tag attribute and set its value to the expiry date.
3. Create a Ranger policy for the EXPIRES_ON tag.
4. Add a condition in this policy to deny access when the date specified the in expiry_date tag attribute is later than the current date.

Note that the EXPIRES_ON tag policy is created as the default policy in tag service instances.

TagSync

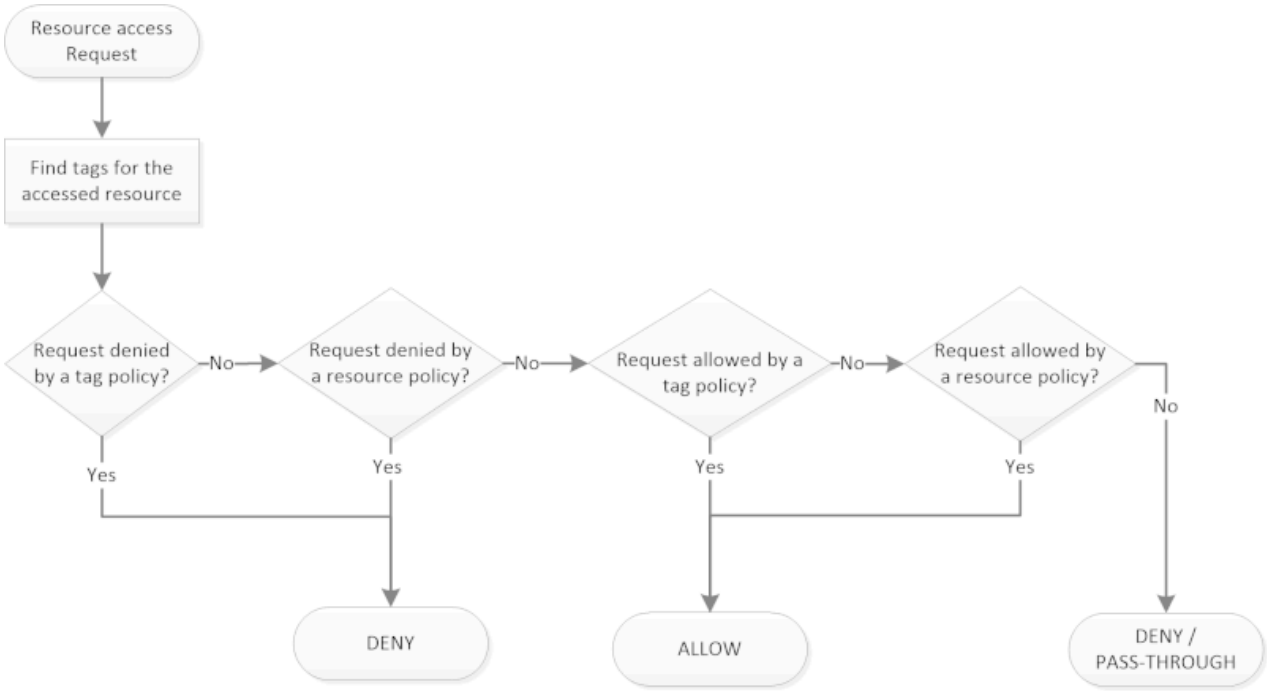
Ranger TagSync is used to synchronize the tag store with an external metadata service such as Apache Atlas. TagSync is a daemon process similar to the Ranger UserSync process.

Ranger TagSync receives tag details from Apache Atlas via change notifications. As tags are added to, updated, or deleted from resources in Apache Atlas, Ranger TagSync receives notifications and updates the tag store.

Tags and policy evaluation

When authorizing an access request, an Apache Ranger plugin evaluates applicable Ranger policies for the resource being accessed. The following diagram shows the details of the policy evaluation flow. More details on the steps in this workflow are provided in the subsequent sections.

Apache Ranger Policy Evaluation Flow with Tags



Apache Ranger Policy Evaluation Flow with Tags

Finding Tags

Apache Ranger supports a service to register context enrichers, which are used to update context data to the access request.

The Ranger Tag service, which is part of the tag-based policies feature, adds a context enricher named RangerTagEnricher. This context enricher is responsible for finding tags for the requested resource and adding the tag details to the request context. This context enricher keeps a cache of the available tags; while processing an access request, it finds the tags applicable for the requested resource and adds the tags to the request context. The context enricher keeps the cache updated by periodically polling Ranger Admin for changes.

Evaluating Tag-Based Policies

Once the list of tags for the requested resource is found, the Apache Ranger policy engine evaluates the tag-based policies applicable to the tags. If a policy for one of these tag results in a deny, access will be denied. If none of the tags are denied, and if a policy allows for one of the tags, access will be allowed. If there is no result for any tag, or if there are no tags for the resource, the policy engine will evaluate the resource-based policies to make the authorization decision.

Using Tags in Conditions

Apache Ranger allows the use of custom conditions while evaluating authorization policies. The Apache Ranger policy engine makes various request details – such as user, groups, resource, and context – available to the conditions. Tags in the request context, which are added by the enricher, are available to the conditions and can be used to influence the authorization decision.

The default policy in tag service instances, the EXPIRES_ON tag, uses such condition to check to see if the request date is later than the value specified in tag attribute expiry_date. This default policy does not work unless an EXPIRES_ON tag has been created in Atlas.

Related Information

Apache Ranger Wiki> Context Enrichers

Ranger access conditions

The Apache Ranger access policy model consists of two major components: specification of the resources a policy is applied to, such as HDFS files and directories, Hive databases. tables. and columns, HBase tables, column-families, and columns, and so on; and the specification of access conditions for specific users and groups

Allow Deny and Exclude Conditions

Apache Ranger supports the following access conditions:

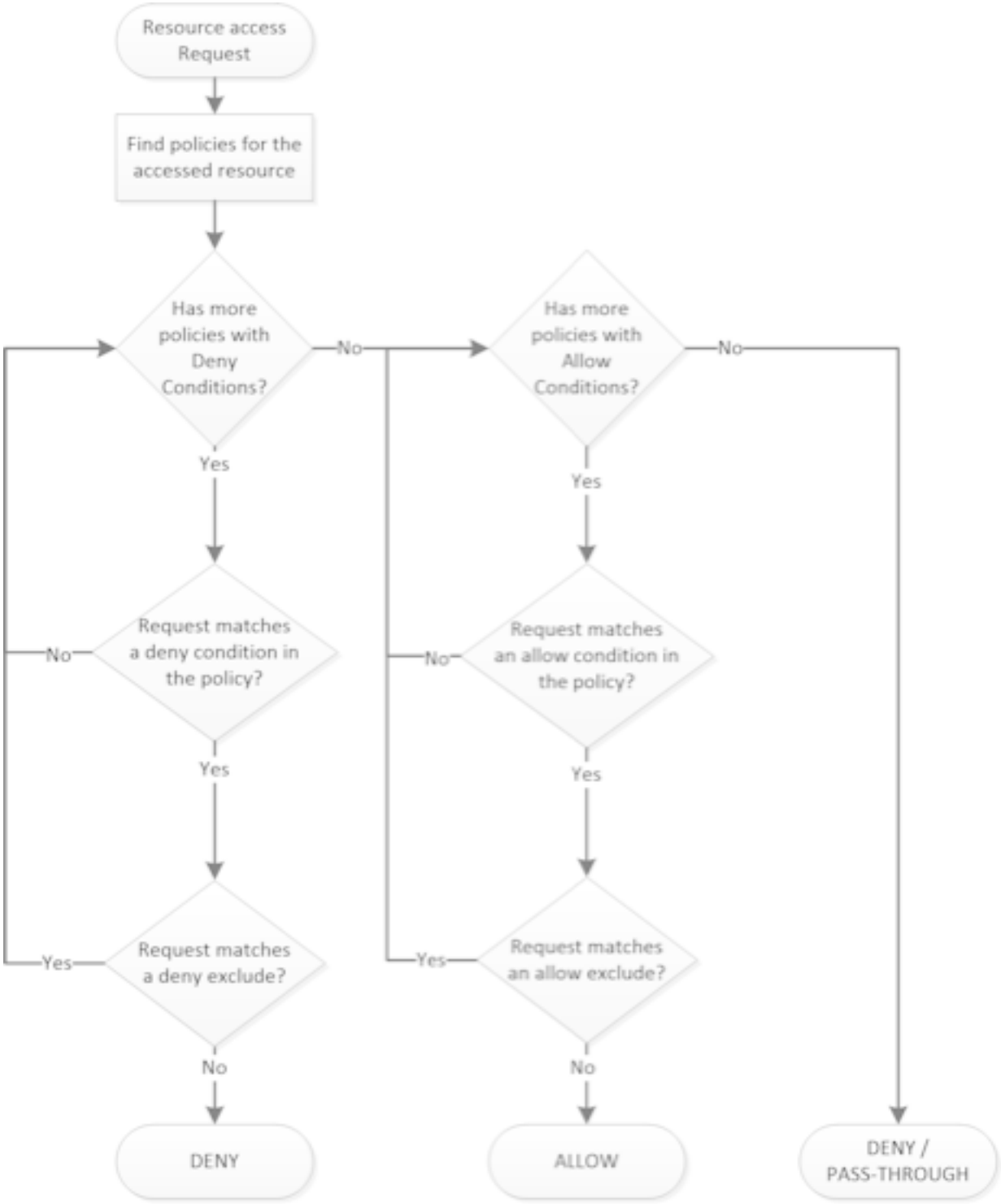
- Allow
- Exclude from Allow
- Deny
- Exclude from Deny

These access conditions enable you to set up fine-grained access control policies.

For example, you can allow access to a "finance" database to all users in the "finance" group, but deny access to all users in the "interns" group. Let's say that one of the members of the "interns" group, "scott", needs to work on an assignment that requires access to the "finance" database. In that case, you can add an Exclude from Deny condition that will allow user "scott" to access the "finance" database. The following image shows how this policy would be set up in Apache Ranger:

Policy Evaluation of Access Conditions

Apache Ranger policies are evaluated in a specific order to ensure predictable results (if there is no access policy that allows access, the authorization request will typically be denied). The following diagram shows the policy evaluation work-flow:



Apache Ranger Policy Evaluation Flow

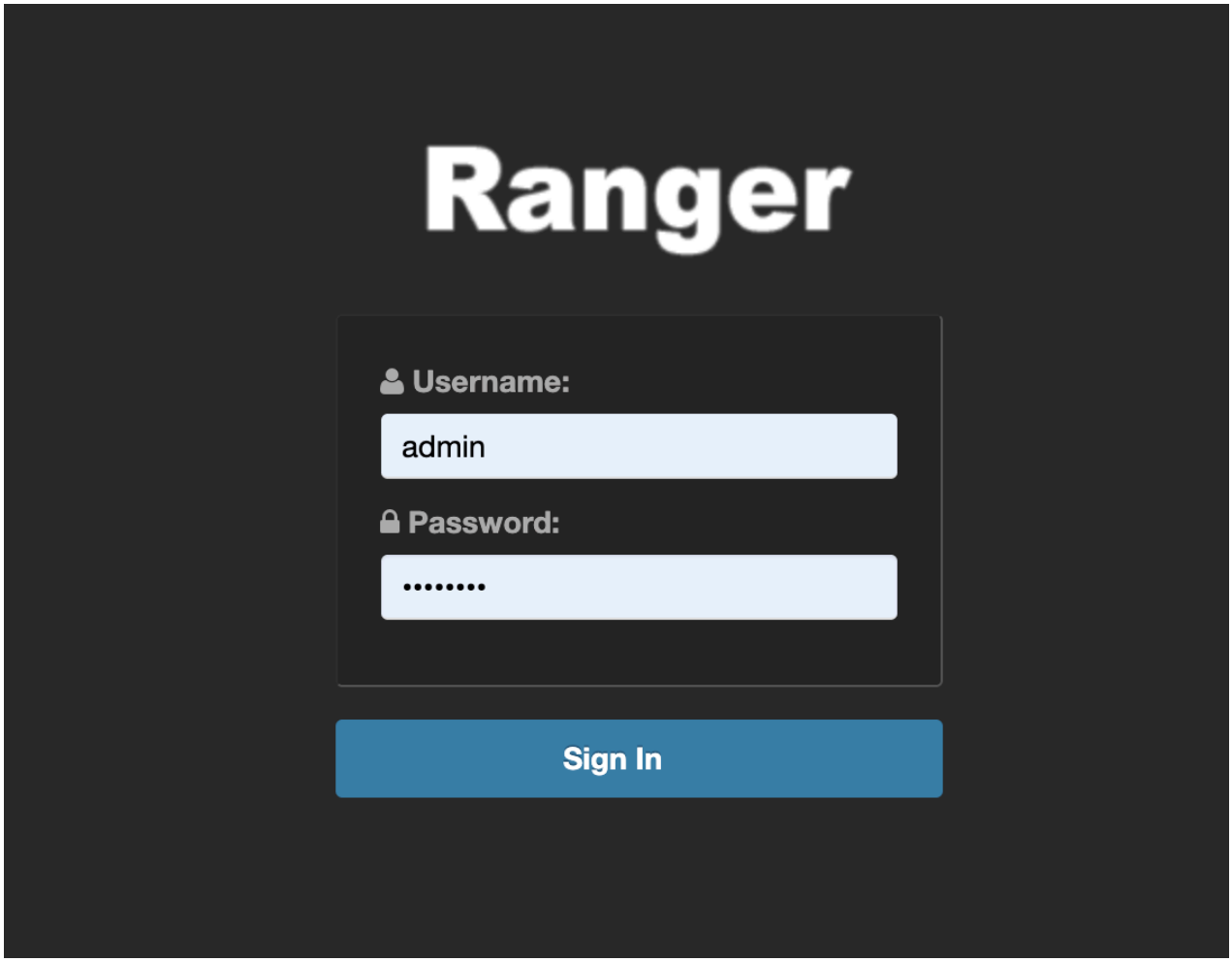
Using the Ranger Console

This chapter contains an overview of the Ranger console.

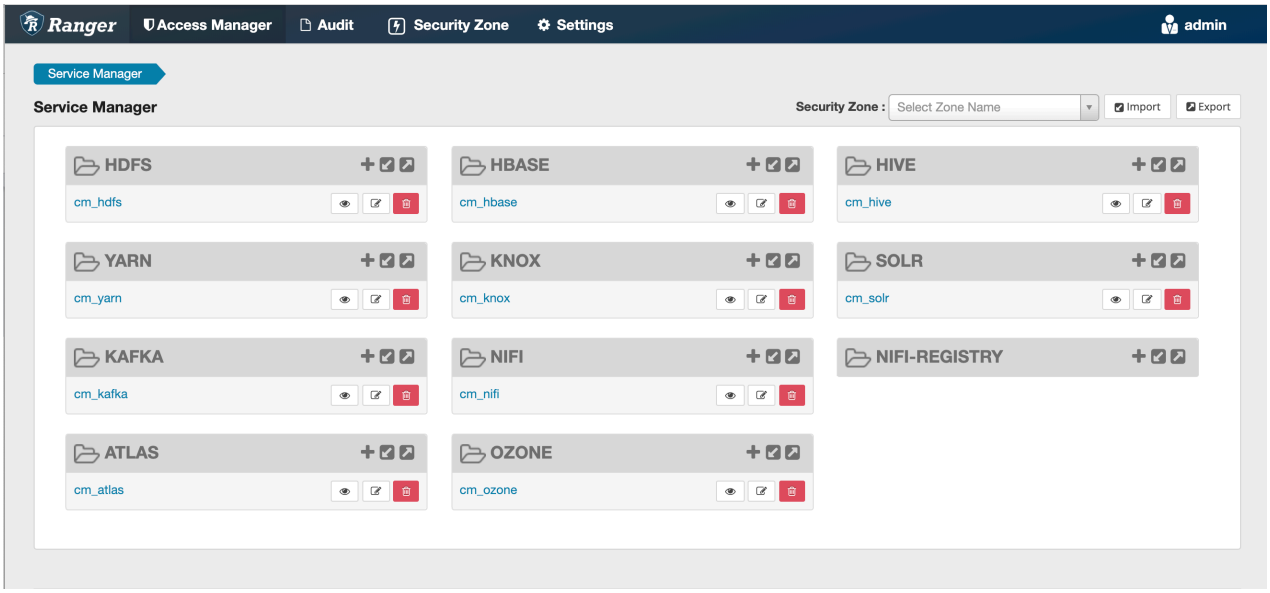
Accessing the Ranger console

How to access the Ranger console.

To access the Ranger Console, click the Ranger Admin web UI link, enter your user name and password, then click Sign In.



Ranger Console Home Page

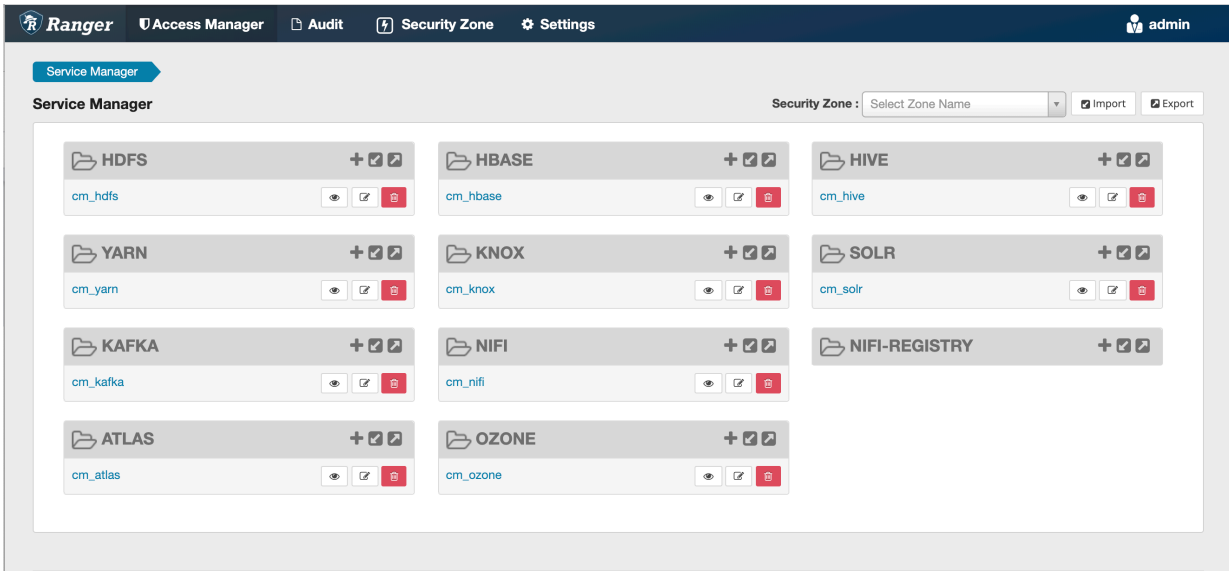


After you log in, your user name is displayed at the top right of the Ranger Console.

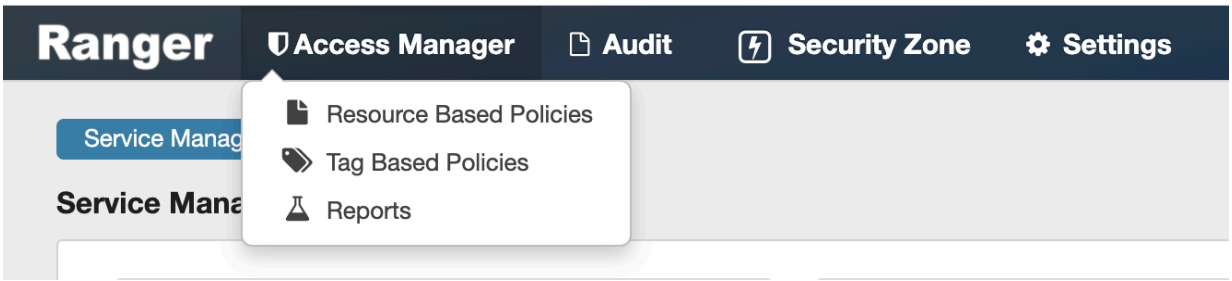
Ranger console navigation

Explains the basic Ranger console/GUI.

- The Service Manager for Resource Based Policies page is displayed when you log in to the Ranger Console. You can use this page to create services for Hadoop resources (HDFS, HBase, Hive, etc.) and add access policies to those resources.

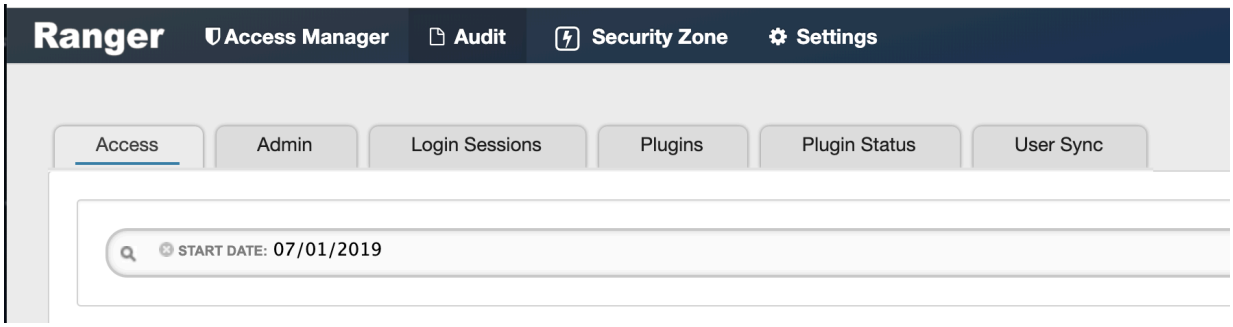


Clicking Access Manager in the top menu opens the Service Manager for Resource Based Policies page, and also displays a submenu with links to Resource Based Policies, Tag Based Policies, and Reports (this submenu is also displayed when you pass the mouse over the Access Manager link).

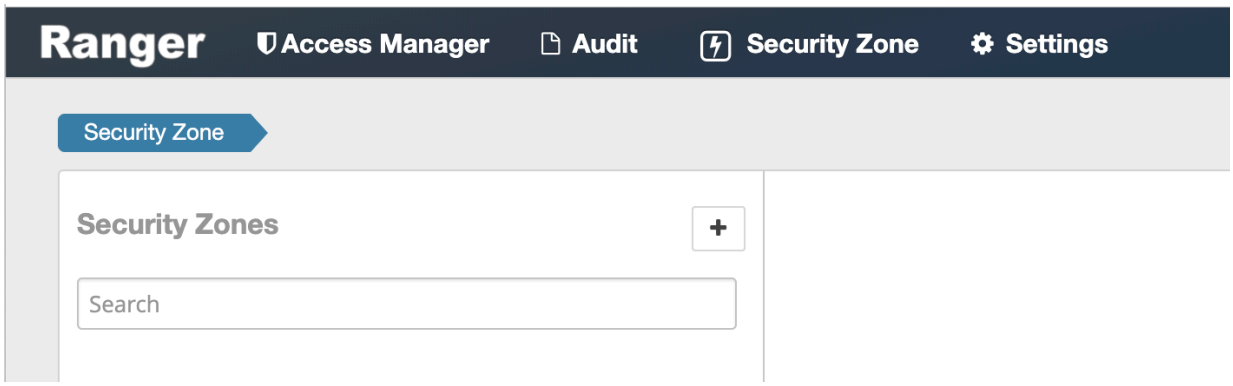


- Access Manager > Resource Based Policies -- Opens the Service Manager for Resource Based Policies page. You can use this page to create services for resources (HDFS, HBase, Hive, etc.) and add access policies to those services.
- Access Manager > Tag Based Policies -- Opens the Service Manager for Tag Based Policies page. You can use this page to create tag-based services and add access policies to those services. Using tag-based policies enables you to control access to resources across multiple components without creating separate services and policies in each component.
- Access Manager > Reports -- Opens the Reports page. You can use this page to generate user access reports for resource and tag-based policies based on search criteria such as policy name, resource, group, and user name.

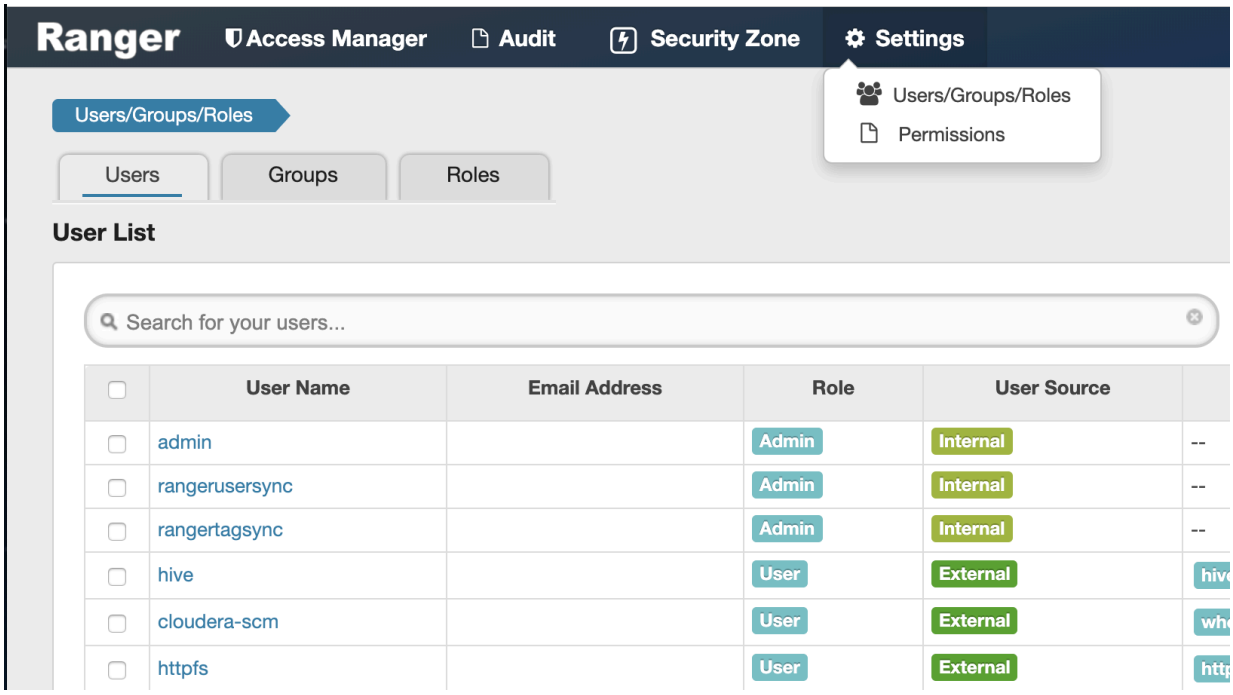
- Audit -- You can use the Audit page to monitor user activity at the resource level, and also to set up conditional auditing based on users, groups, or time. The Audit page includes the Access, Admin, Login Sessions, Plugins, Plugin Status, and User Sync tabs.



- Security Zone -- Lets you organize resource and tag-based services and policies into separate security zones. You can assign one or more administrators for each security zone. Security zone administrators can then create and update policies for their security zone.



- Settings -- Enables you to manage and assign policy permissions to users and groups. Clicking or passing the mouse over Settings displays a submenu with links to the Users/Groups/Roles and Permissions pages.



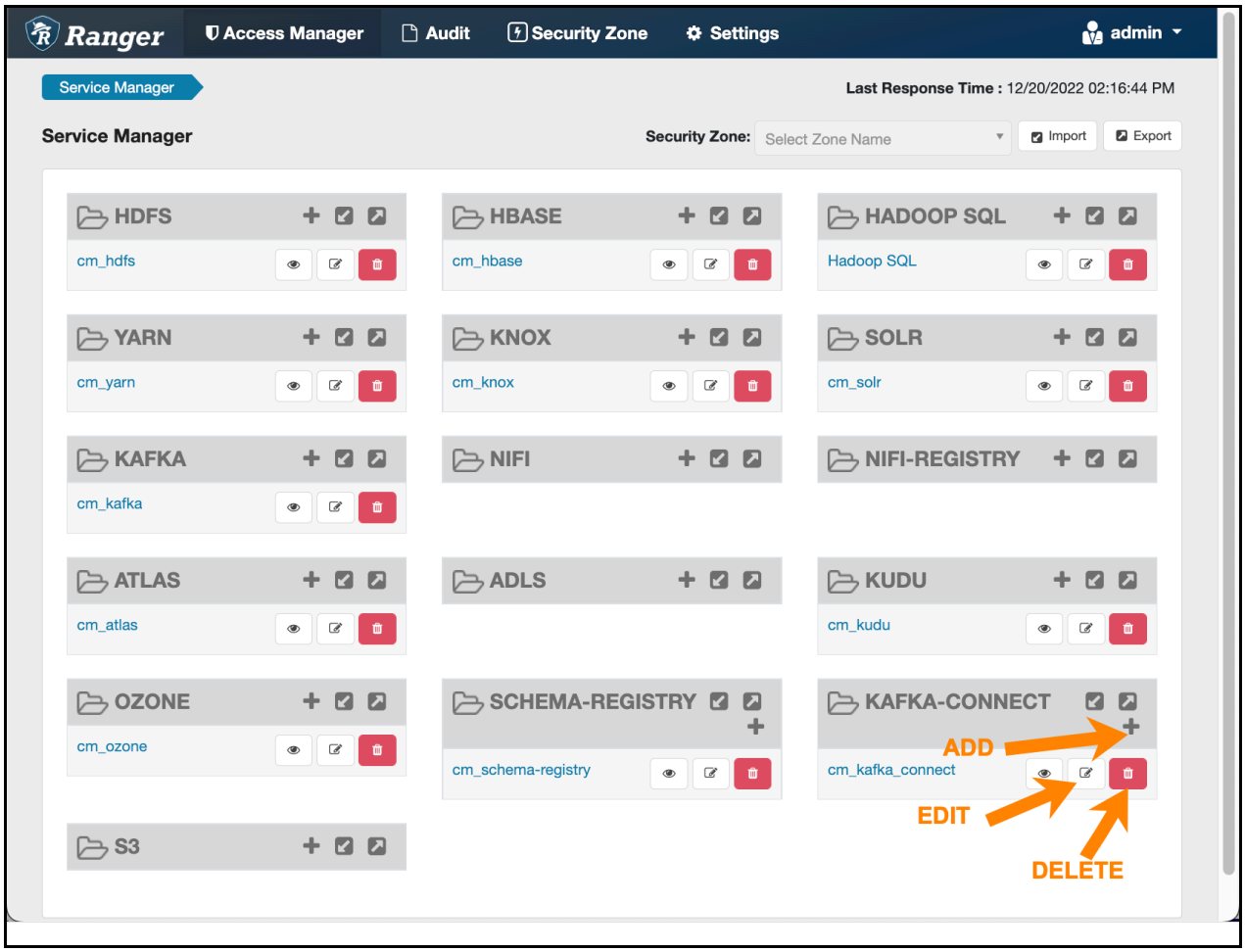
Resource-based Services and Policies

Ranger enables you to configure resource-based services for Hadoop components (e.g. HBase, Kafka, Storm, etc.) and add access policies to those services.

Configuring resource-based services

The Service Manager for Resource Based Policies page displays when you log in to the Ranger Admin Web UI. You can also access this page by selecting Access Manager Resource Based Policies . You can use this page to create services for Hadoop resources (HDFS, HBase, HadoopSQL, etc.) and add access policies to those resources.

- To add a new resource-based service, click the Add icon ( ) in the applicable box on the Service Manager page. Enter the required configuration settings, then click Add.
- To edit a resource-based service, click the Edit icon ( ) at the right of the service. Edit the service settings, then click Save to save your changes.
- To delete a resource-based service, click the Delete icon ( ) at the right of the service. Deleting a service also deletes all of the policies for that service.



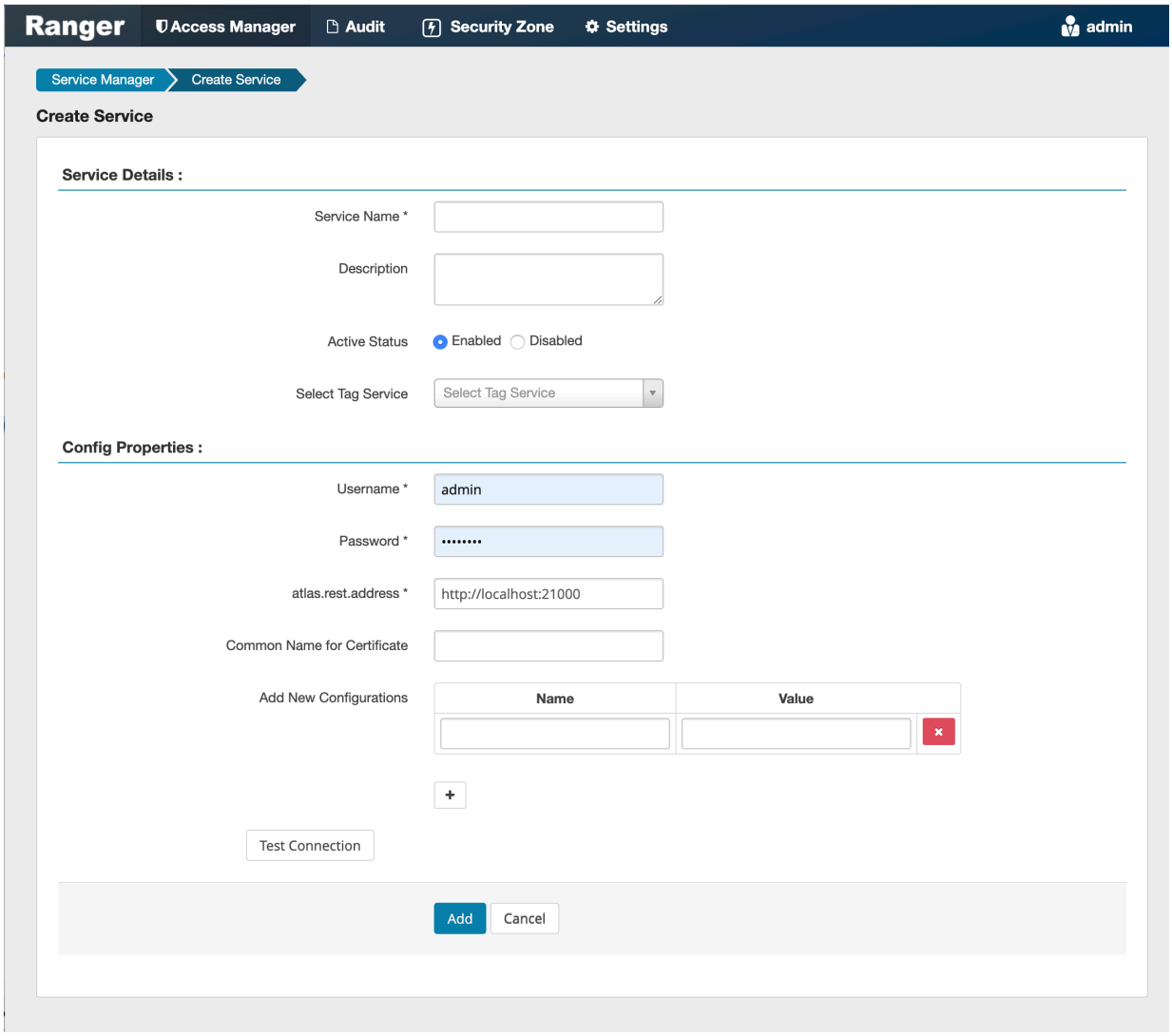
Configure a resource-based service: Atlas

How to add an Atlas service.

Procedure

1. On the Service Manager page, click the Add icon ( ) next to Atlas.

   The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 3: Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Field name	Description
Select Tag Service	Select a tag-based service to apply the service and its tag-based policies to Atlas.

Table 4: Configuration Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
atlas.rest.address	Atlas host and port: : http://atlas_host_FQDN:21000.
Common Name For Certificate	The name of the certificate. This field is interchangeably named Common Name For Certificate and Ranger Plugin SSL CName in Create Service pages.
Add New Configurations	Add any other new configuration(s).

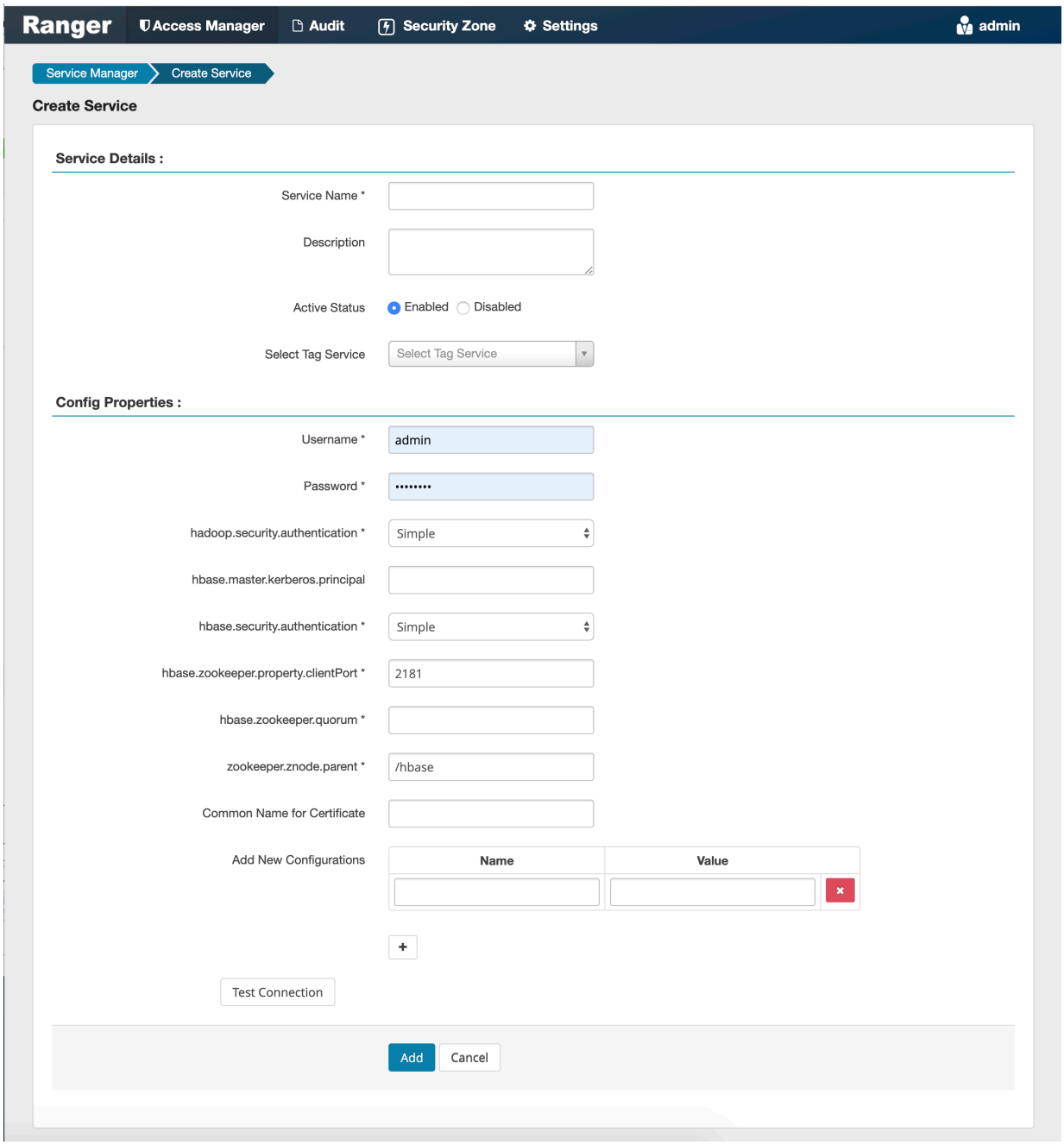
3. Click Test Connection.
4. Click Add.

Configure a resource-based service: HBase

How to add an HBase service.

Procedure

1. On the Service Manager page, click the Add icon ( ) next to HBase.
The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 5: Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.

Field name	Description
Active Status	Enabled or Disabled.
Select Tag Service	Select a tag-based service to apply the service and its tag-based policies to HBase.

Table 6: Configuration Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
hadoop.security.authorization	The complete connection URL, including port and database name. (Default port: 10000.) For example, on the sandbox, jdbc:hive2://sandbox:10000/.
hbase.master.kerberos.principal	The Kerberos principal for the HBase Master. (Required only if Kerberos authentication is enabled.)
hbase.security.authentication	As noted in the hadoop configuration file hbase-site.xml.
hbase.zookeeper.property.clientPort	As noted in the hadoop configuration file hbase-site.xml.
hbase.zookeeper.quorum	As noted in the hadoop configuration file hbase-site.xml.
zookeeper.znode.parent	As noted in the hadoop configuration file hbase-site.xml.
Common Name for Certificate	The name of the certificate. This field is interchangeably named Common Name For Certificate and Ranger Plugin SSL CName in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click Test Connection.
4. Click Add.

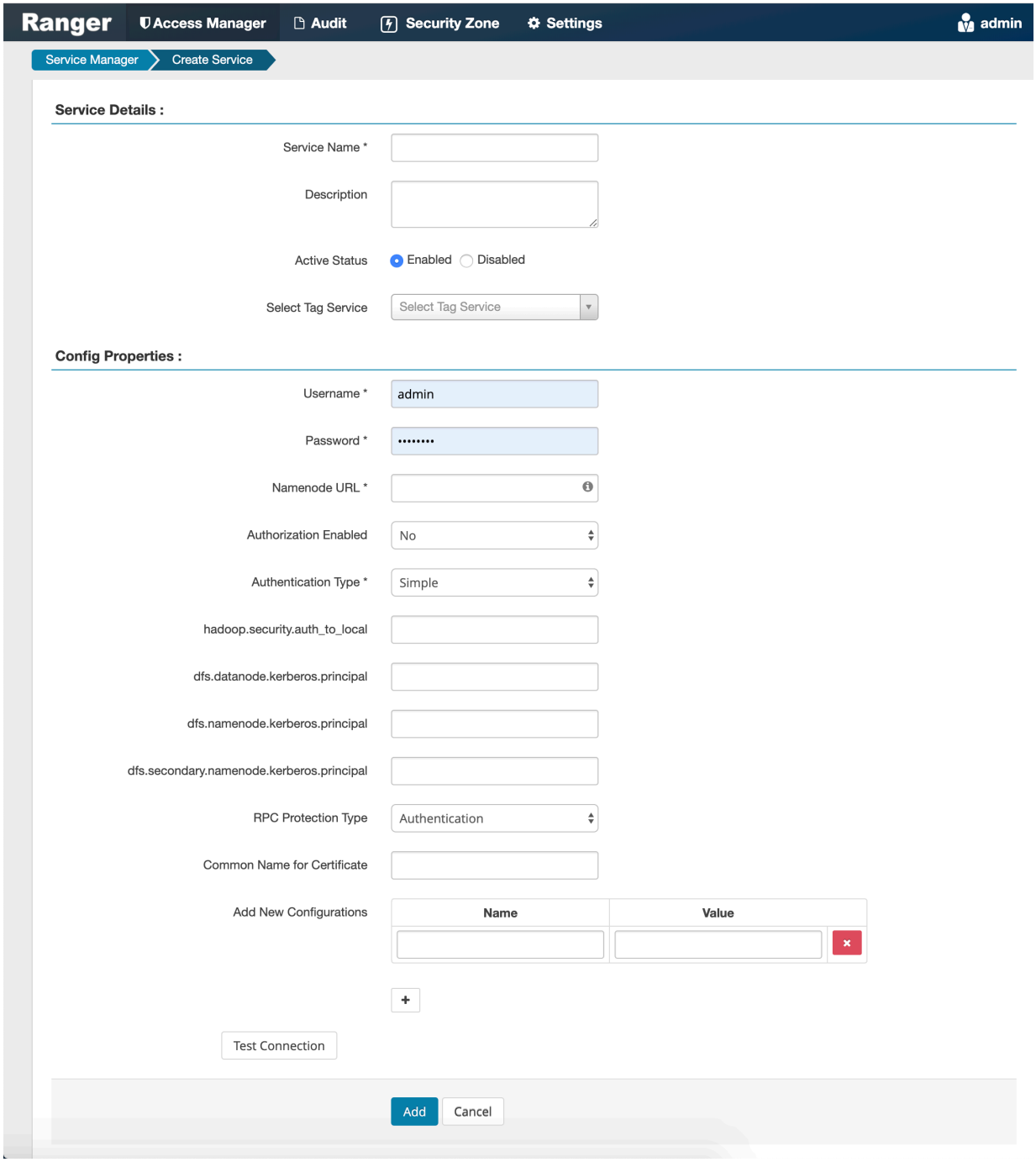
Configure a resource-based service: HDFS

How to add an HDFS service.

Procedure

1. On the Service Manager page, click the Add icon ( ) next to HDFS.

   The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 7: Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.

Field name	Description
Description	A description of the service.
Active Status	Enabled or Disabled.
Select Tag Service	Select a tag-based service to apply the service and its tag-based policies to HDFS.

Table 8: Configuration Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
NameNode URL	hdfs://NAMENODE_FQDN:8020 The location of the Hadoop HDFS service, as noted in the hadoop configuration file core-site.xml OR (if this is a HA environment) the path for the primary NameNode. This field was formerly named fs.defaultFS.
Authorization Enabled	Authorization involves restricting access to resources. If enabled, user need authorization credentials.
Authentication Type	The type of authorization in use, as noted in the hadoop configuration file core-site.xml; either simple or Kerberos. (Required only if authorization is enabled). This field was formerly named hadoop.security.authorization.
hadoop.security.auth_to_local	Maps the login credential to a username with Hadoop; use the value noted in the hadoop configuration file, core-site.xml.
dfs.datanode.kerberos.principal	The principal associated with the datanode where the service resides, as noted in the hadoop configuration file hdfs-site.xml. (Required only if Kerberos authentication is enabled).
dfs.namenode.kerberos.principal	The principal associated with the NameNode where the service resides, as noted in the hadoop configuration file hdfs-site.xml. (Required only if Kerberos authentication is enabled).
dfs.secondary.namenode.kerberos.principal	The principal associated with the secondary NameNode where the service resides, as noted in the hadoop configuration file hdfs-site.xml. (Required only if Kerberos authentication is enabled).
RPC Protection Type	Only authorised user can view, use, and contribute to a dataset. A list of protection values for secured SASL connections. Values: Authentication, Integrity, Privacy
Common Name For Certificate	The name of the certificate. This field is interchangeably named Common Name For Certificate and Ranger Plugin SSL CName in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click Test Connection.
4. Click Add.

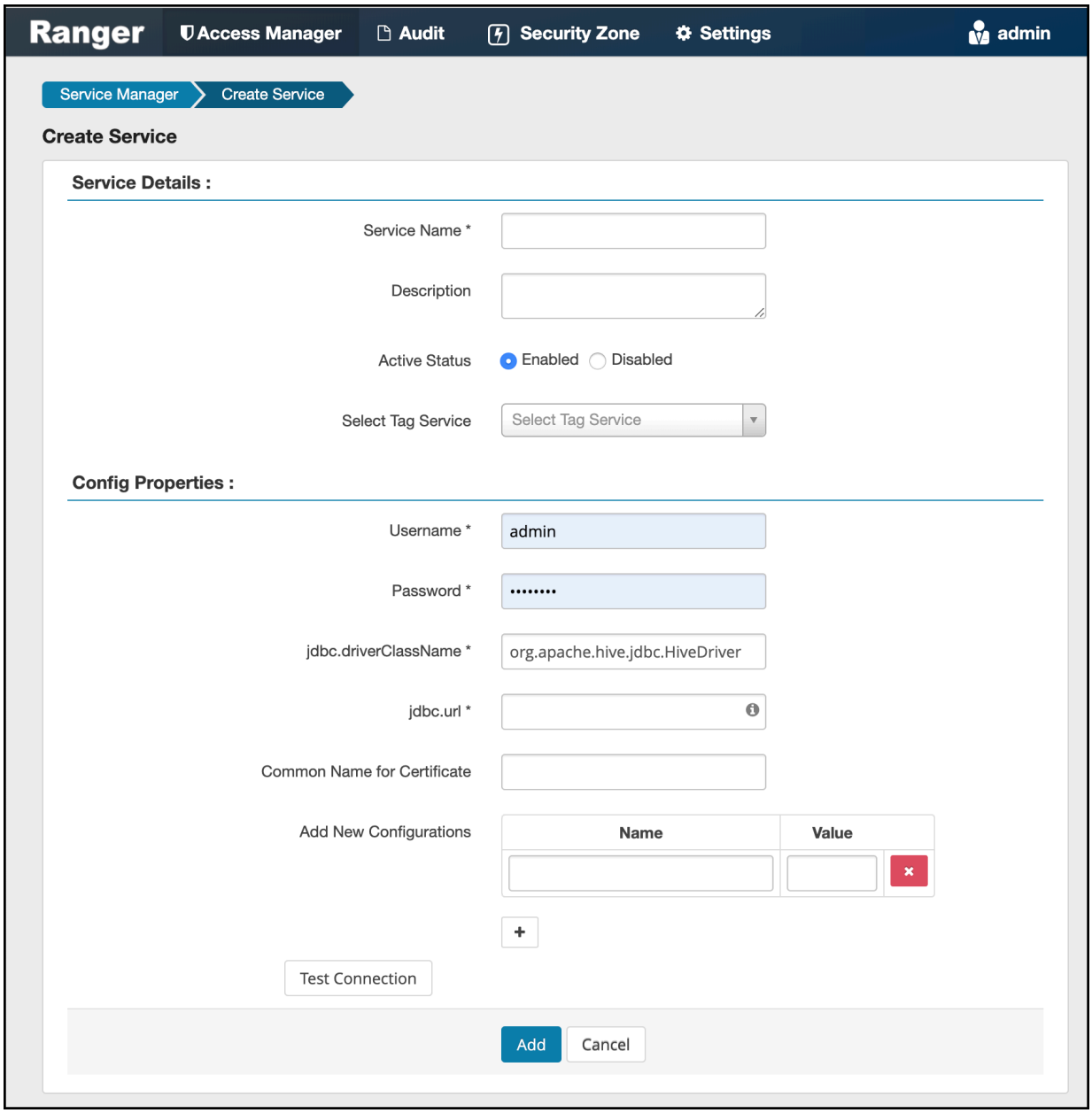
Configure a resource-based service: HadoopSQL

How to add a HadoopSQL service.

Procedure

1. On Service Manager, click Add ( ) next to HadoopSQL.

   Create Service appears.



2. On Create Service, enter the following information:

Table 9: Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Field name	Description
Select Tag Service	Select a tag-based service to apply the service and its tag-based policies to Hive.

Table 10: Configuration Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
jdbc.driver ClassName	The full classname of the driver used for Hive connections. Default: org.apache.hive.jdbc.HiveDriver
jdbc.url	The complete connection URL, including port and database name. (Default port: 10000.) For example, on the sandbox, jdbc:hive2://sandbox:10000/.
Common Name For Certificate	The name of the certificate. This field is interchangeably named Common Name For Certificate and Ranger Plugin SSL CName in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click Test Connection.
4. Click Add.

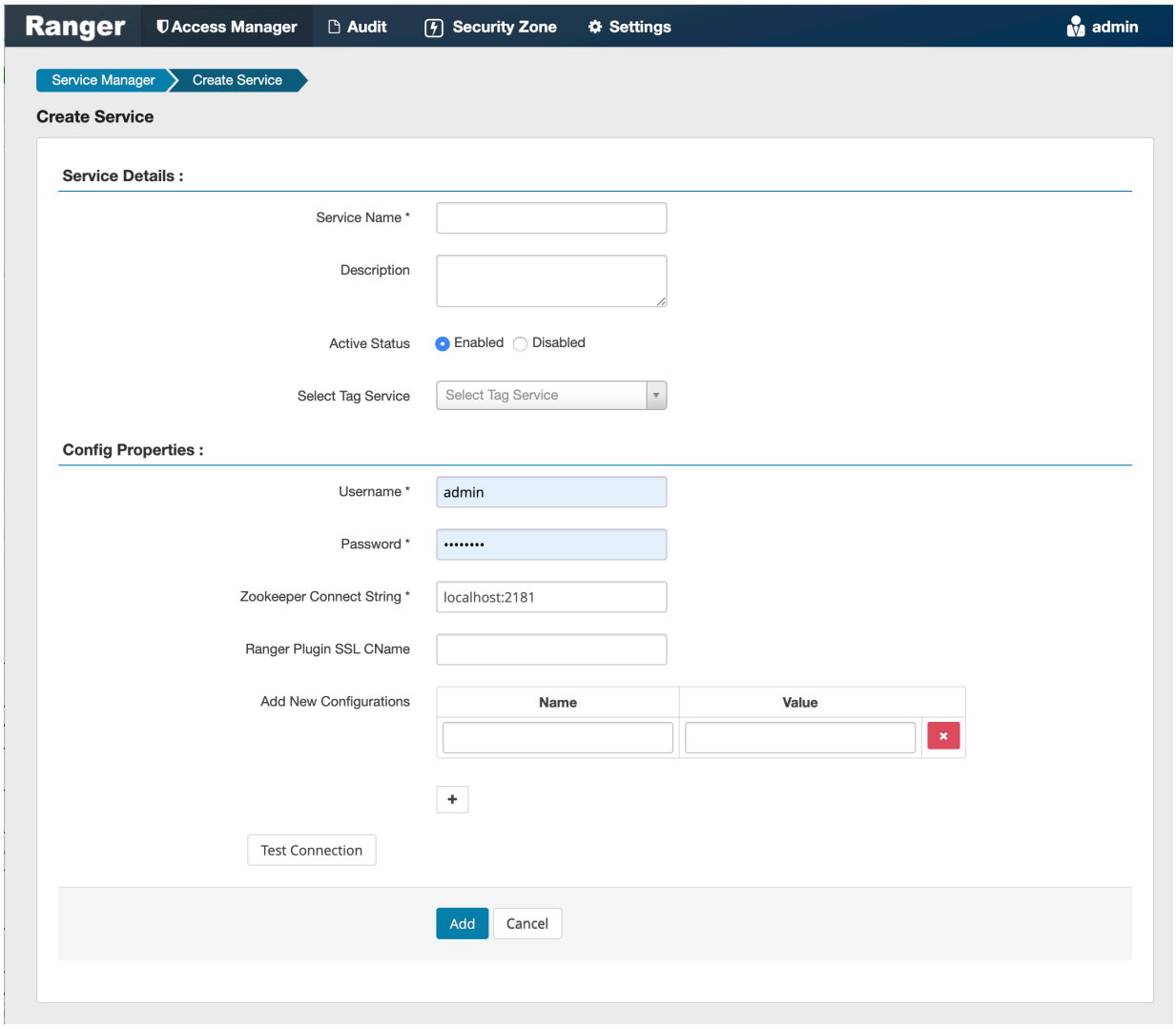
Configure a resource-based service: Kafka

How to add a Kafka service.

Procedure

1. On the Service Manager page, click the Add icon ( ) next to Kafka.

   The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 11: Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Field name	Description
Select Tag Service	Select a tag-based service to apply the service and its tag-based policies to Kafka.

Table 12: Configuration Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
ZooKeeper Connect String	Defaults to localhost:2181 (Provide FQDN of zookeeper host : 2181).
Ranger Plugin SSL CName	Provide common.name.for.certificate which is registered with Ranger (in Wire Encryption environment). This field is interchangeably named Common Name For Certificate and Ranger Plugin SSL CName in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click Test Connection.
4. Click Add.

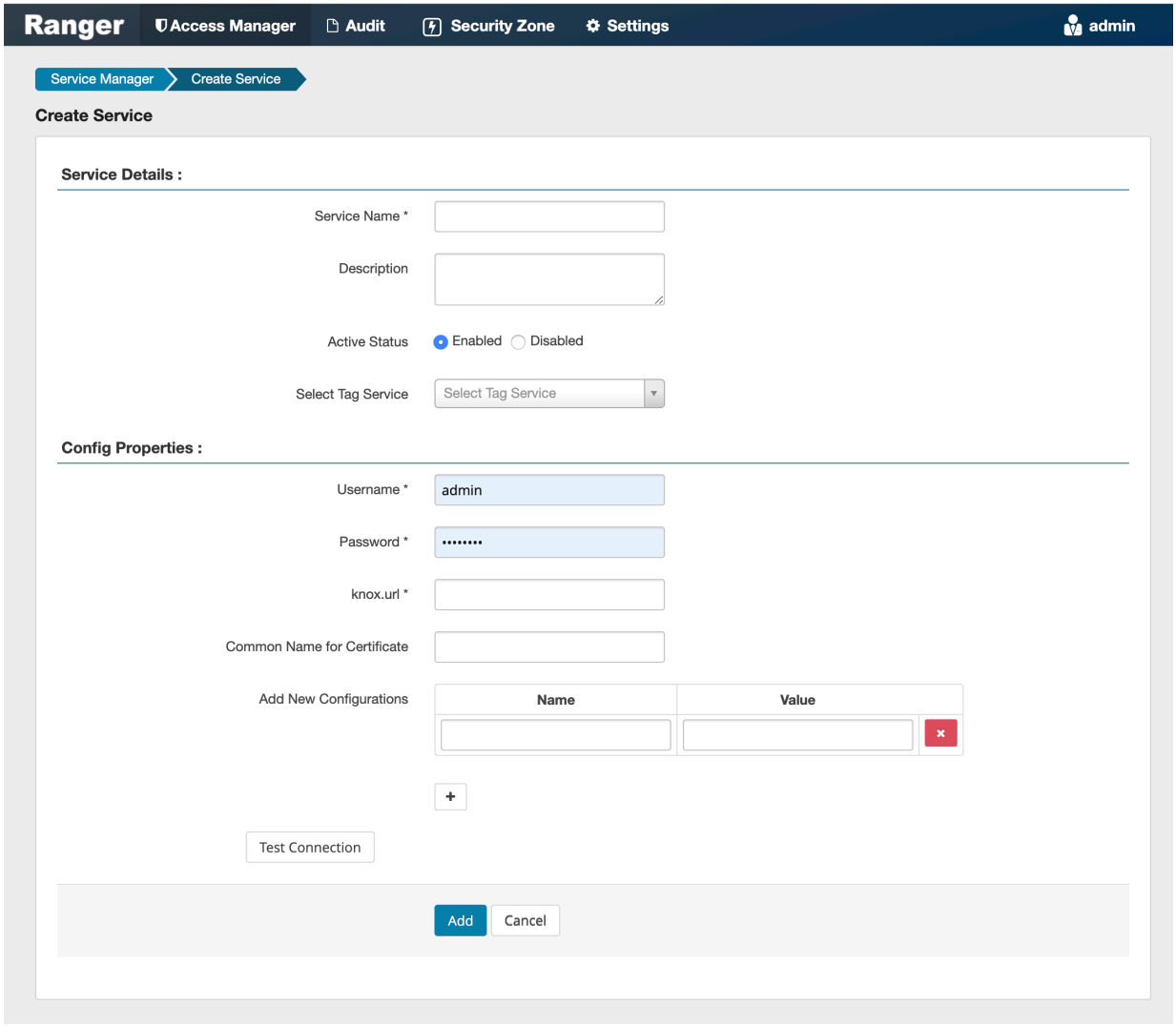
Configure a resource-based service: Knox

How to add a Knox service.

Procedure

1. On the Service Manager page, click the Add icon ( ) next to Knox.

   The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 13: Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Field name	Description
Select Tag Service	Select a tag-based service to apply the service and its tag-based policies to Knox.

Table 14: Configuration Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
knox.url	The Gateway URL for Knox.
Common Name For Certificate	The name of the certificate. This field is interchangeably named Common Name For Certificate and Ranger Plugin SSL CName in Create Service pages.
Add New Configurations	Add any other new configuration(s).

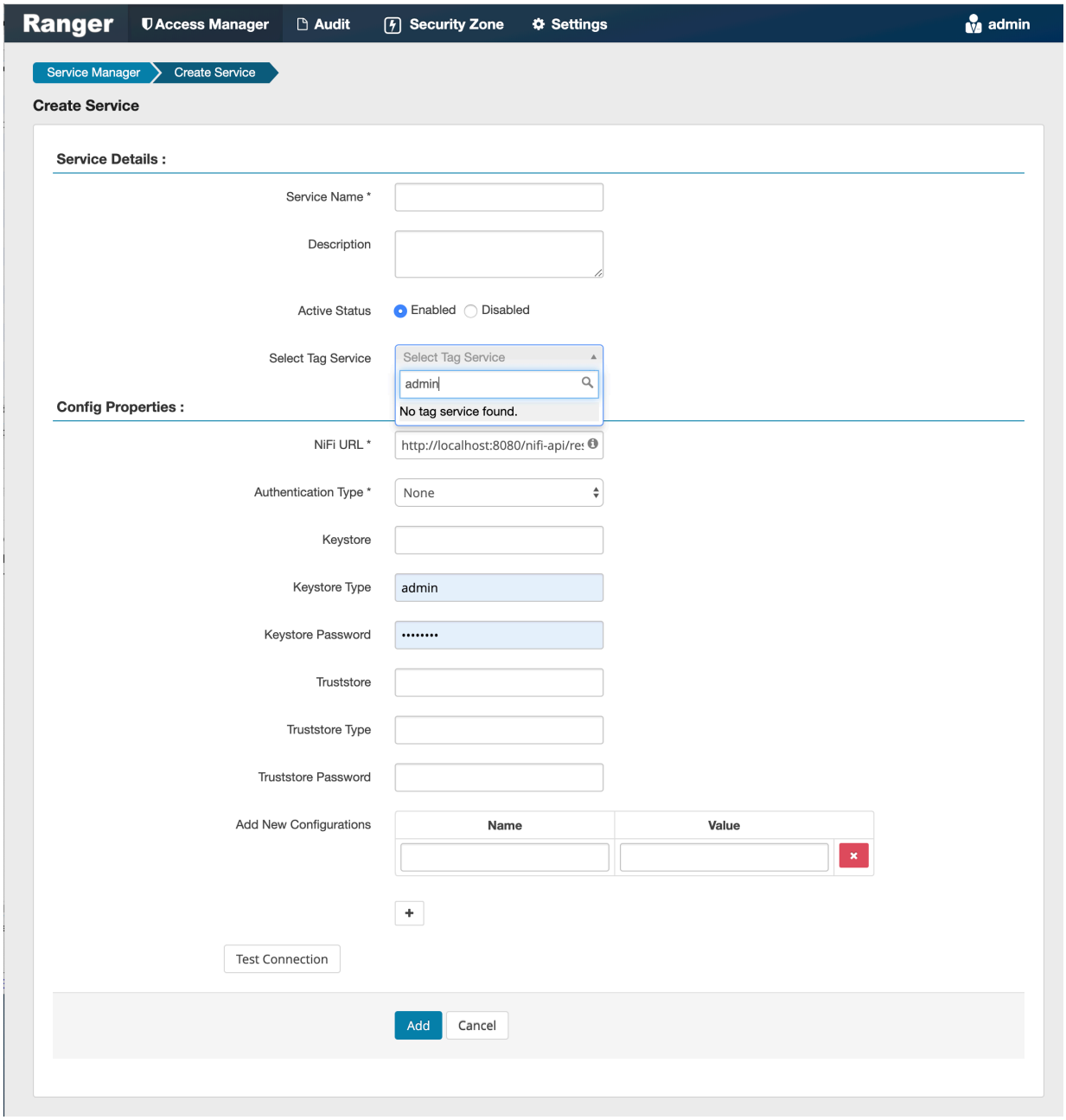
3. Click Test Connection.
4. Click Add.

Configure a resource-based service: NiFi

How to add a NiFi service.

Procedure

1. On the Service Manager page, click the Add icon ( ) next to NiFi.
The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 15: Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Field name	Description
Select Tag Service	Select a tag-based service to apply the service and its tag-based policies to NiFi.

Table 16: Configuration Properties

Field name	Description
NiFi URL	The complete NiFi host URL.
Authentication Type	None or SSL.
Keystore	The keystore to use when Ranger makes an https connection to NiFi. This keystore contains the certificate that represents the Ranger server.
Keystore Type	The keystore type (JKS or PKCS12).
Keystore Password	The keystore password.
Truststore	The truststore to use when Ranger makes an https connection to NiFi. This truststore contains the public key of the certificate authority that signed the NiFi server certificates.
Truststore Type	The truststore type (JKS or PKCS12).
Truststore Password	The truststore password.
Add New Configurations	Add any other new configuration(s).

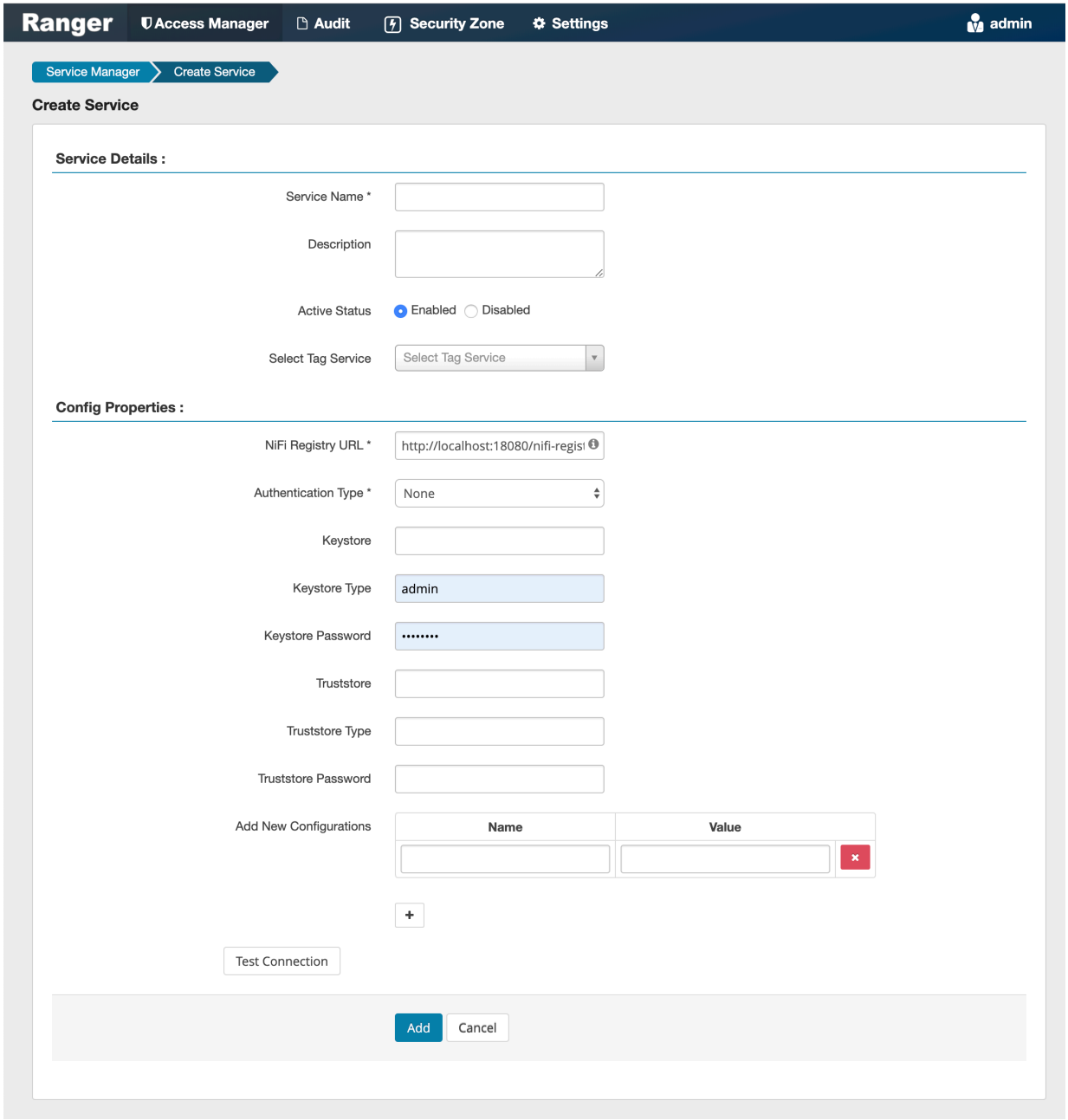
3. Click Test Connection.
4. Click Add.

Configure a resource-based service: NiFi Registry

How to add a NiFi Registry service.

Procedure

1. On the Service Manager page, click the Add icon ( ) next to NiFi Registry.
The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 17: Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Field name	Description
Select Tag Service	Select a tag-based service to apply the service and its tag-based policies to NiFi.

Table 18: Configuration Properties

Field name	Description
NiFi Registry URL	The complete NiFi Registry URL.
Authentication Type	None or SSL.
Keystore	The keystore to use when Ranger makes an https connection to the NiFi Registry. This keystore contains the certificate that represents the Ranger server.
Keystore Type	The keystore type (JKS or PKCS12).
Keystore Password	The keystore password.
Truststore	The truststore to use when Ranger makes an https connection to the NiFi Registry. This truststore contains the public key of the certificate authority that signed the NiFi server certificates.
Truststore Type	The truststore type (JKS or PKCS12).
Truststore Password	The truststore password.
Add New Configurations	Add any other new configuration(s).

3. Click Test Connection.
4. Click Add.

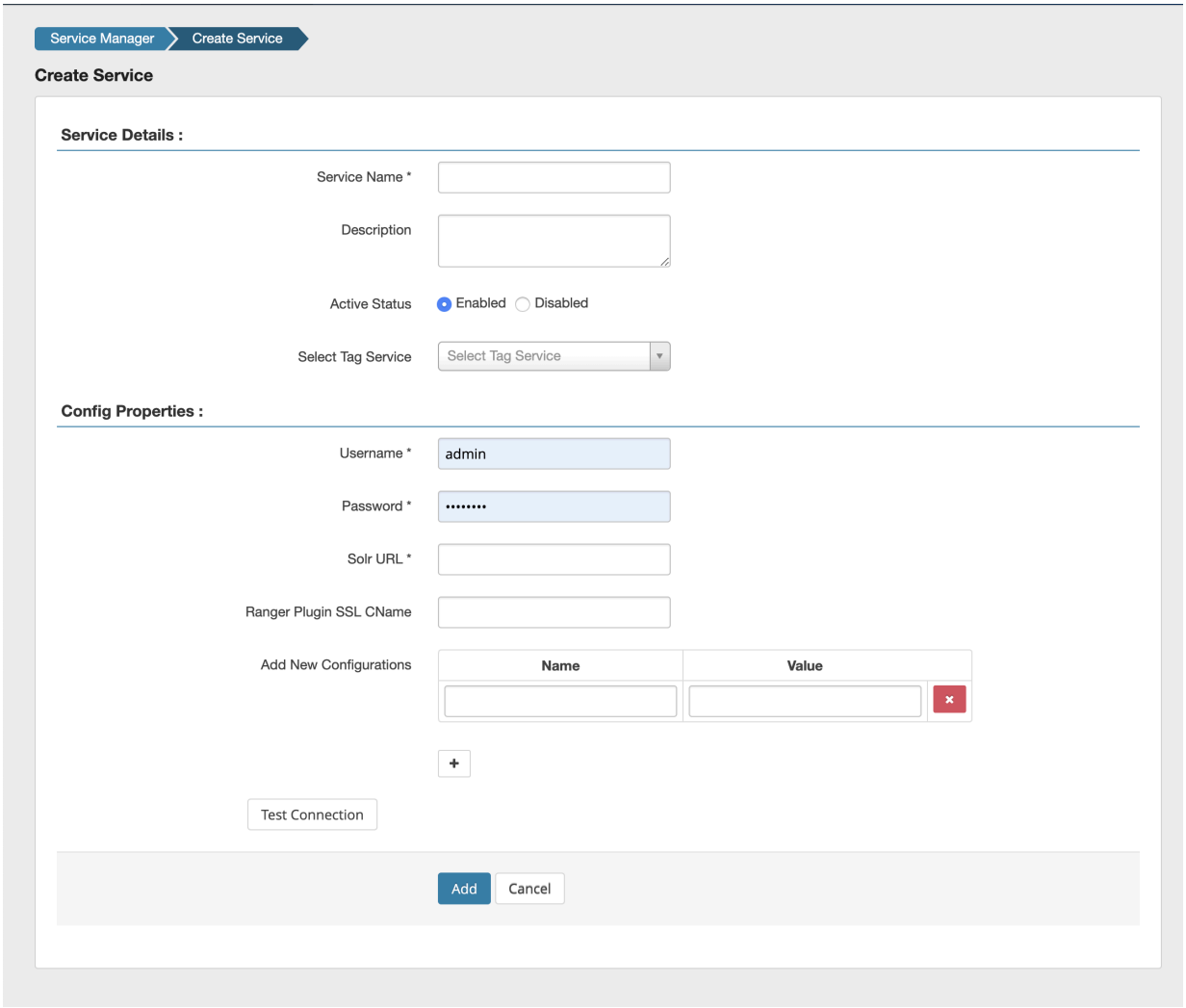
Configure a resource-based service: Solr

How to add a Solr service.

Procedure

1. On the Service Manager page, click the Add icon ( ) next to Solr.

   The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 19: Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Field name	Description
Select Tag Service	Select a tag-based service to apply the service and its tag-based policies to Solr.

Table 20: Configuration Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
Solr URL	http://Solr_host:8983
Ranger Plugin SSL CName	Provide common.name.for.certificate which is registered with Ranger (in Wire Encryption environment). This field is interchangeably named Common Name For Certificate and Ranger Plugin SSL CName in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click Test Connection.
4. Click Add.

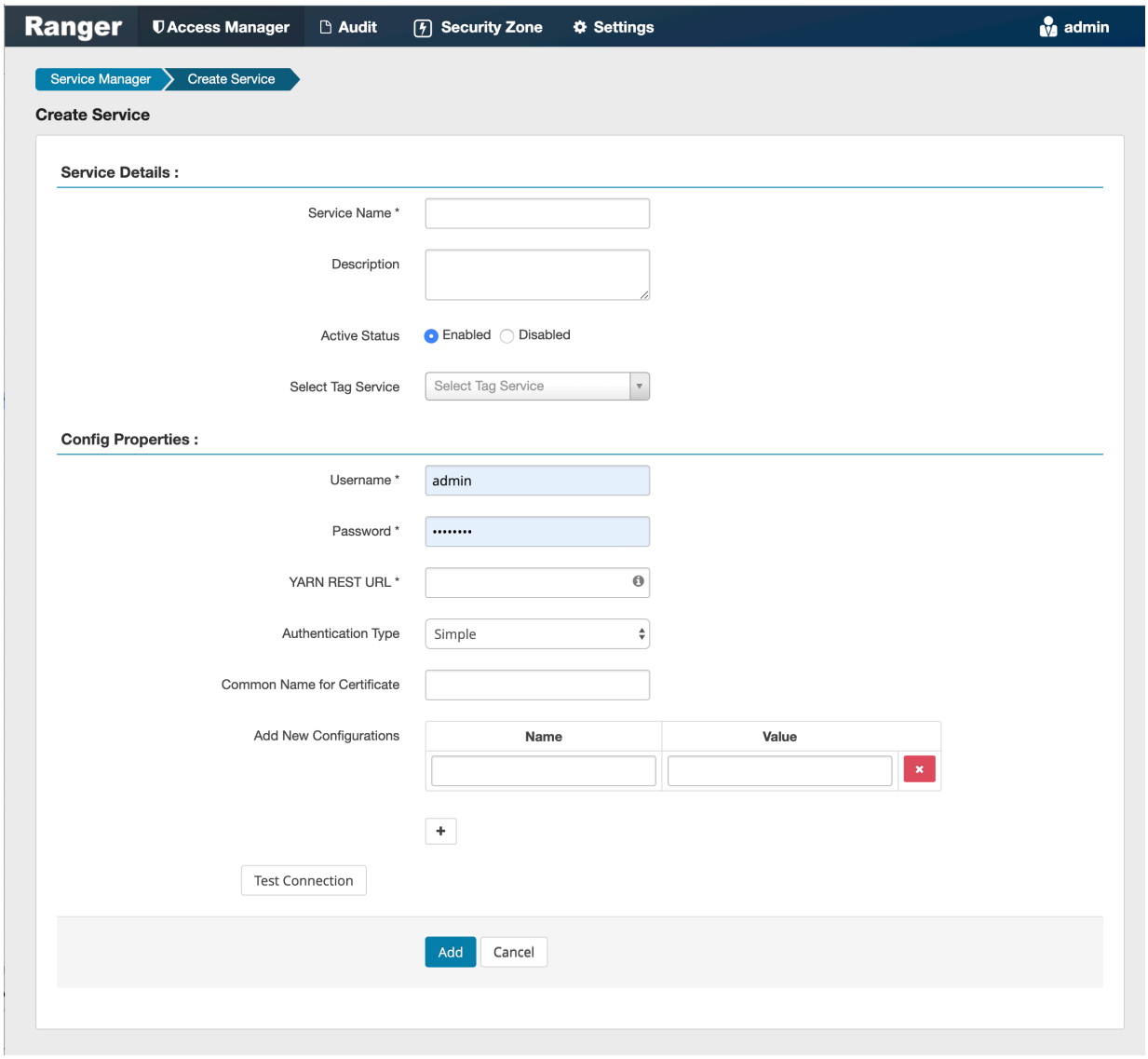
Configure a resource-based service: YARN

How to add a YARN service.

Procedure

1. On the Service Manager page, click the Add icon ( ) next to YARN.

   The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 21: Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Field name	Description
Select Tag Service	Select a tag-based service to apply the service and its tag-based policies to YARN.

Table 22: Configuration Properties

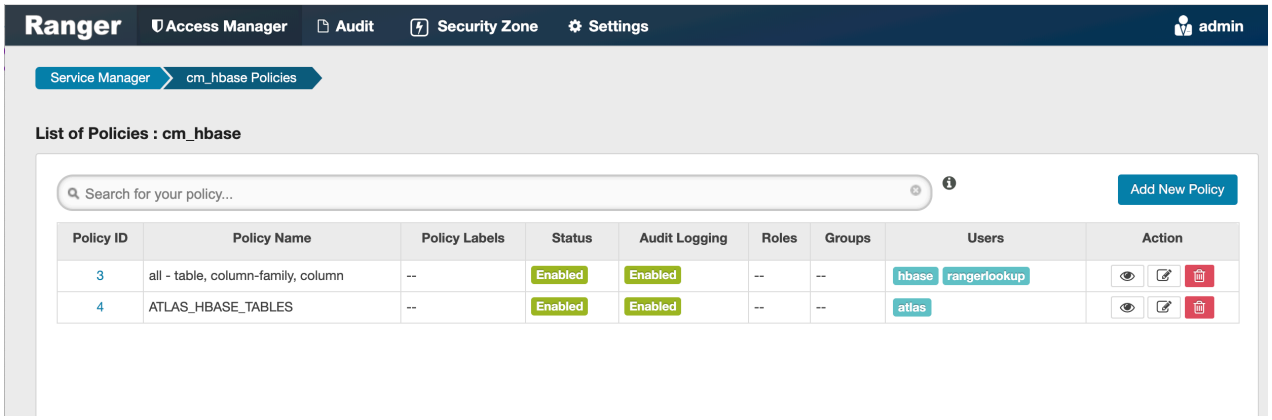
Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
YARN REST URL	Http or https://RESOURCEMANAGER_FQDN:8088.
Authentication Type	The type of authorization in use, as noted in the hadoop configuration file core-site.xml; either simple or Kerberos. (Required only if authorization is enabled). This field was formerly named hadoop.security.authorization.
Common Name For Certificate	The name of the certificate. This field is interchangeably named Common Name For Certificate and Ranger Plugin SSL CName in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click Test Connection.
4. Click Add.

Configuring resource-based policies

To view the policies associated with a service, click the service name on the Resource Based Policies Service Manager page. The policies for that service will be displayed in a list, along with a search box.

- To add a new resource-based policy to the service, click Add New Policy.
- To edit a resource-based policy, click the Edit icon ( ) for the service. Edit the policy settings, then click Save to save your changes.
- To delete a resource-based policy, click the Delete icon ( ) for the service.



Related Information

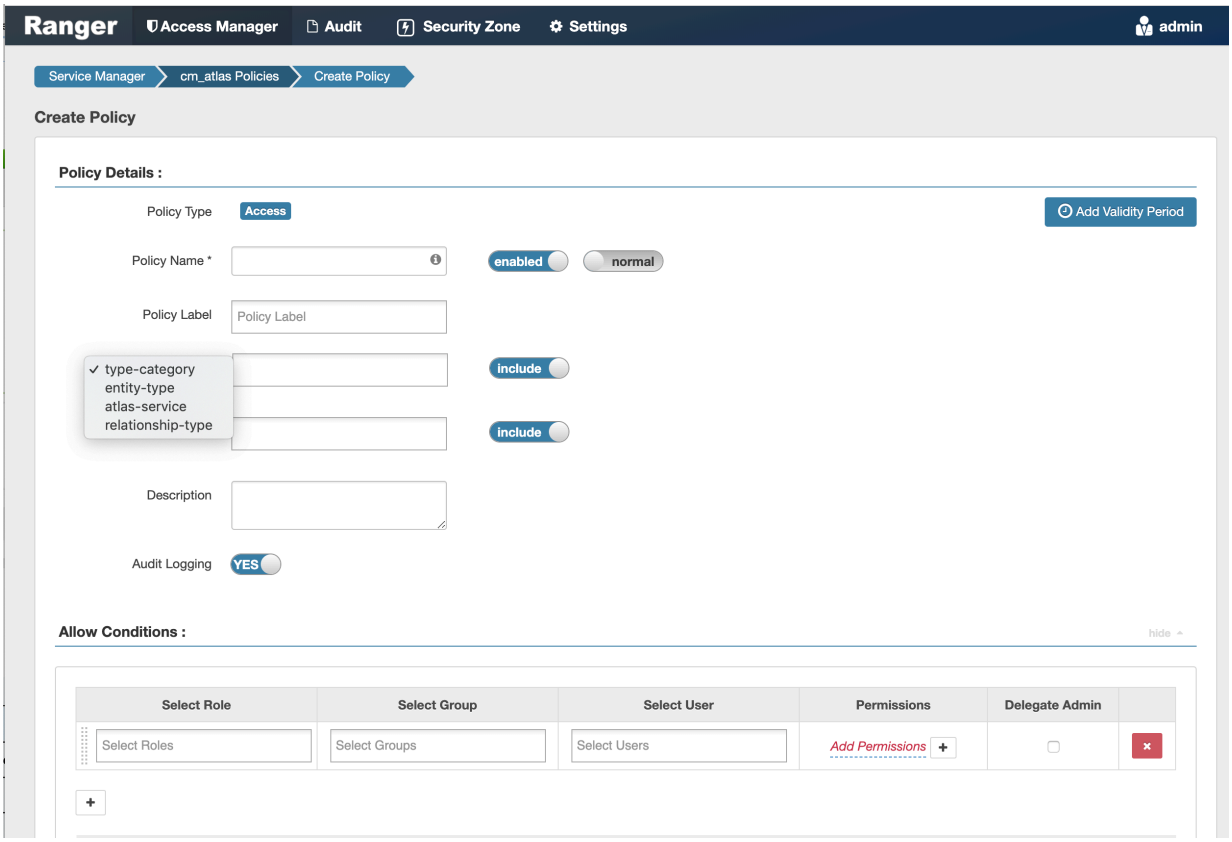
Importing and exporting resource-based policies

Configure a resource-based policy: Atlas

How to add a new policy to an existing Atlas service.

Procedure

1. On the Service Manager page, select an existing Atlas service.
   The List of Policies page appears.
2. Click Add New Policy.
   The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 23: Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
type-category	Select type-category, entity-type, atlas-service, or relationship-type.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.

Field	Description
Add Validity Period	Specify a start and end time for the policy.

Table 24: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Permissions	Add or edit permissions: Create Type, Update Type, Delete Type, Select/Deselect All.
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

Related Information

Wildcards and variables in resource-based policies

Configure a resource-based policy: HBase

How to add a new policy to an existing HBase service.

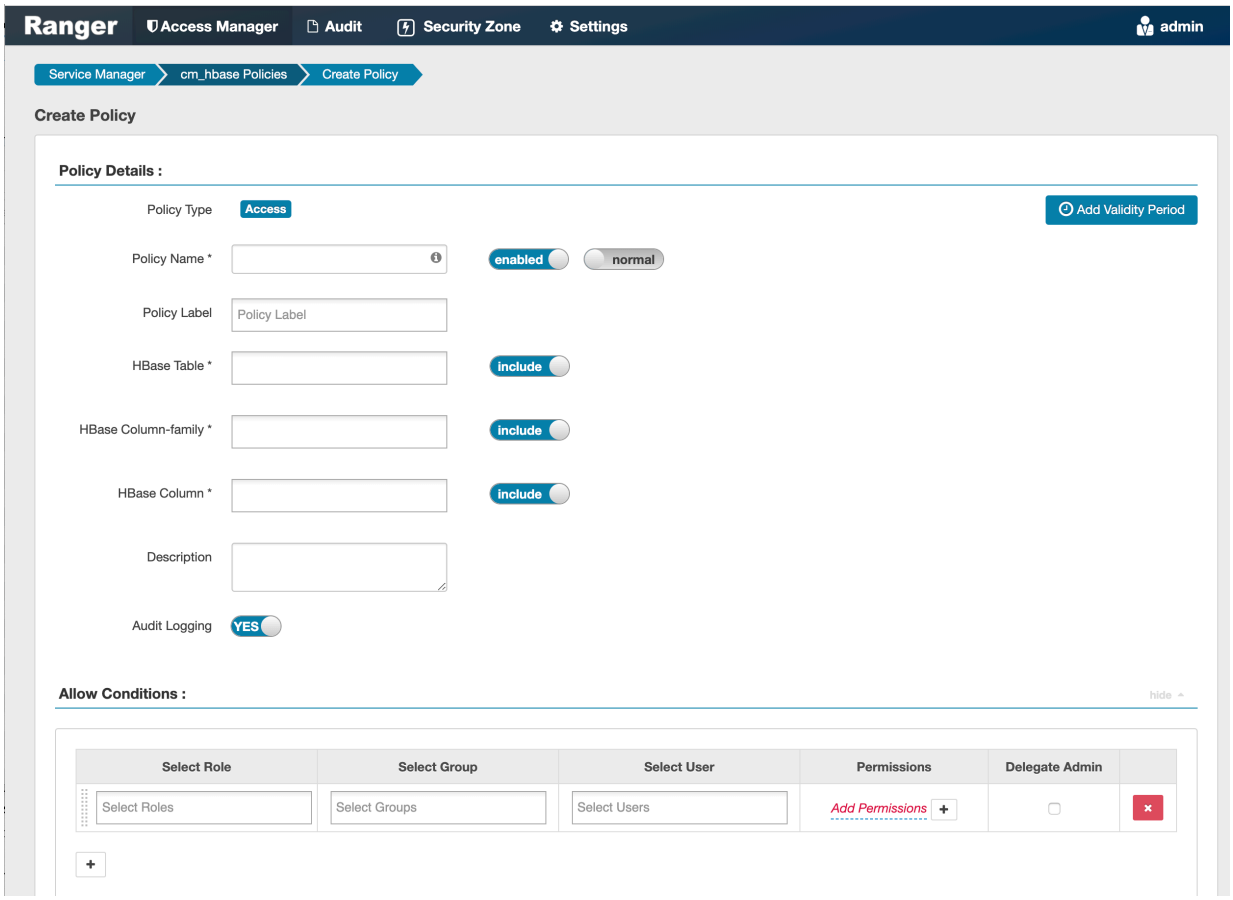
Procedure

1. On the Service Manager page, select an existing HBase service.

   The List of Policies page appears.

2. Click Add New Policy.

   The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 25: Policy Details

Label	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
HBase Table	Select the appropriate database. Multiple databases can be selected for a particular policy. This field is mandatory. **Note:** You can define a namespace in the HBase table field. Valid formats for a namespace-specific, HBase policy include: <namespace><colon> <namespace><colon><tablePrefix>*
HBase Column-family	For the selected table, specify the column families to which the policy applies.
HBase Column	For the selected table and column families, specify the columns to which the policy applies.
Description	(Optional) Describe the purpose of the policy.

Label	Description
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.
Add Validity Period	Specify a start and end time for the policy.

Table 26: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Permissions	Add or edit permissions: Read, Write, Create, Admin, Select/ Deselect All.
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

What to do next

Provide User Access to HBase Database Tables from the Command Line

HBase provides the means to manage user access to HBase database tables directly from the command line. The most commonly-used commands are:

- GRANT

  Syntax:

  ```
  grant '<user-or-group>','<permissions>','<table>
  ```

  For example, to create a policy that grants user1 read/write permission on the table usertable, the command would be:

  ```
  grant 'user1','RW','usertable'
  ```

  The syntax is the same for granting CREATE and ADMIN rights.
- REVOKE

  Syntax:

  ```
  revoke '<user-or-group>','<usertable>'
  ```

  For example, to revoke the read/write access of user1 to the table usertable, the command would be:

  ```
  revoke 'user1','usertable'
  ```

  Note:

  Unlike Hive, HBase has no specific revoke commands for each user privilege.

Related Information

Wildcards and variables in resource-based policies

Configure a resource-based policy: HDFS

How to add a new policy to an existing HDFS service.

About this task

Through configuration, Apache Ranger enables both Ranger policies and HDFS permissions to be checked for a user request. When the NameNode receives a user request, the Ranger plugin checks for policies set through the Ranger Service Manager. If there are no policies, the Ranger plugin checks for permissions set in HDFS.

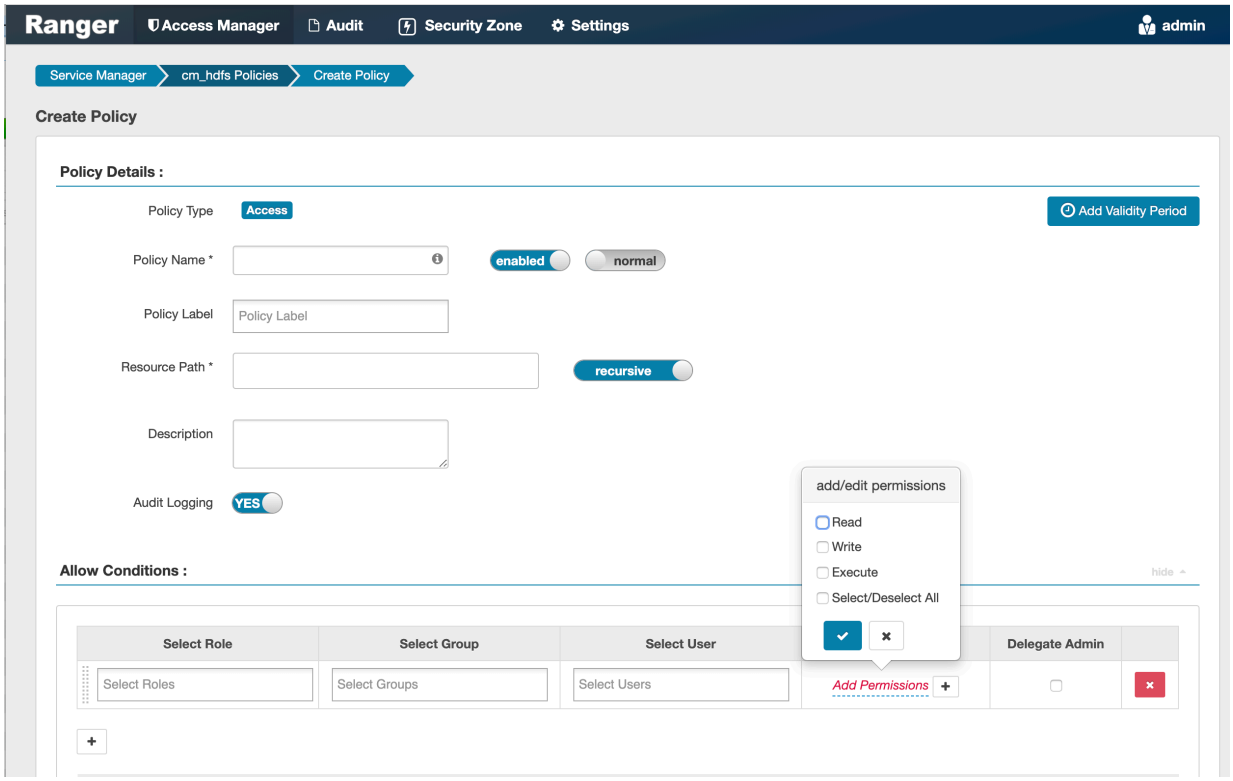
We recommend that permissions be created at the Ranger Service Manager, and to have restrictive permissions at the HDFS level.

Procedure

1. On the Service Manager page, select an existing HDFS service.

   The List of Policies page appears.

2. Click Add New Policy.
   The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 27: Policy Details

Field	Description
Policy Name	Enter a unique name for this policy. The name cannot be duplicated anywhere in the system.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
Resource Path	Define the resource path for the policy folder/file. The default recursive setting specifies that the resource path is recursive; you can also specify a non-recursive path.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.

Field	Description
Add Validity Period	Specify a start and end time for the policy.

Table 28: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Permissions	Add or edit permissions: Read, Write, Execute, Select/Deselect All.
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

Related Information

Wildcards and variables in resource-based policies

Configure a resource-based policy: HadoopSQL

How to add a new policy to an existing Hive service.

Procedure

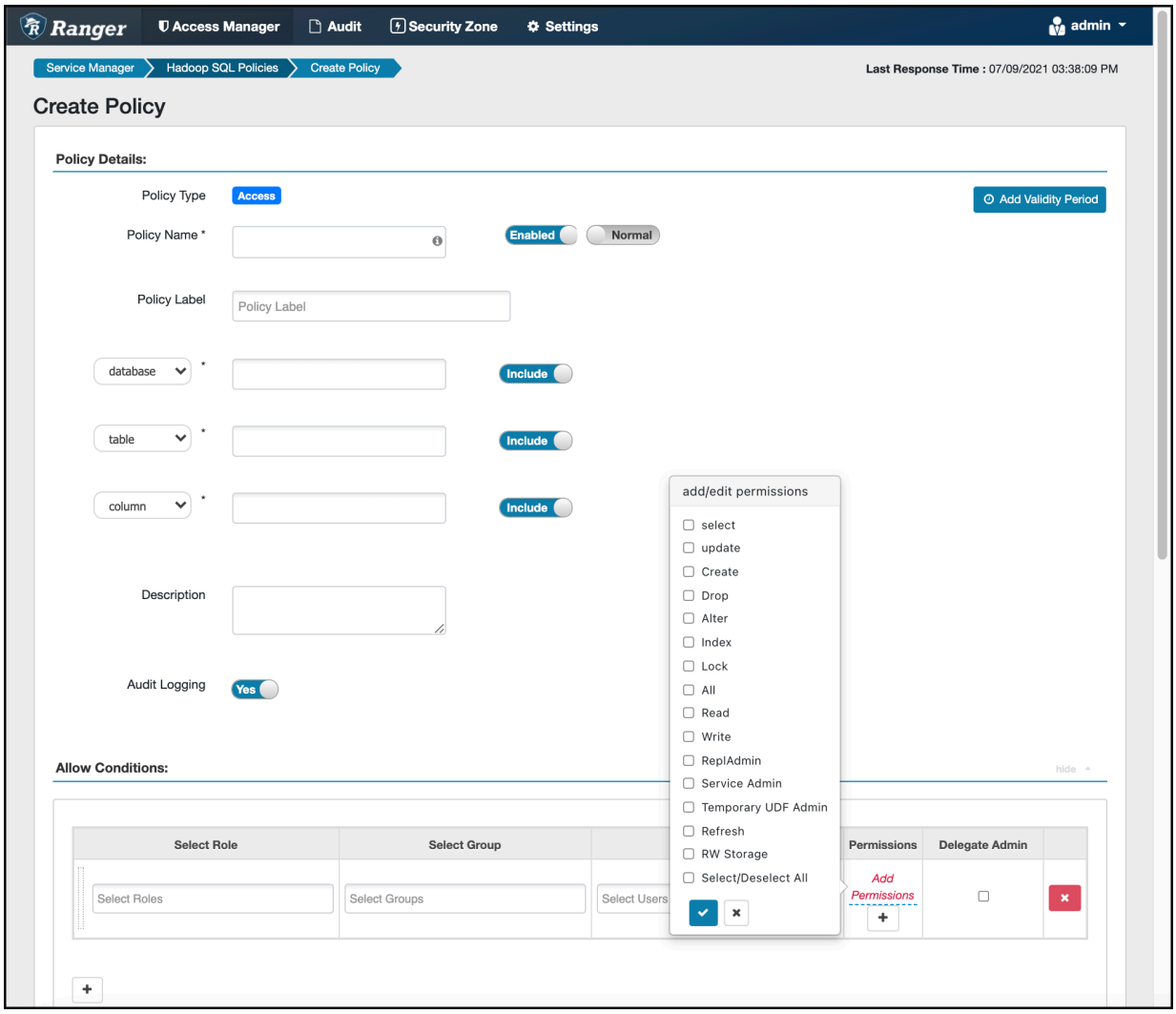
1. On the Service Manager page, select an existing HadoopSQL service.

   The List of Policies page appears.

   Note: Service_name remains cm_hive. Display name is HadoopSQL.

2. Click Add New Policy.

   The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 29: Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory. The policy is enabled by default.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
Database	Type in the applicable database name. The autocomplete feature displays available databases based on the entered text. Include is selected by default to allow access. Select Exclude to deny access..

Field	Description
table/udf	Specifies a table-based or UDF-based policy. Select table or udf, then type in the applicable table or UDF name. The autocomplete feature displays available tables based on the entered text. Include is selected by default to allow access. Select Exclude to deny access.
column	Type in the applicable column name. The autocomplete feature displays available columns based on the entered text. Include is selected by default to allow access. Select Exclude to deny access.
URL	Specify the cloud storage path (for example s3a://dev-admin/demo/campaigns.txt) where the end-user permission is needed to read/write the Hive data from/to a cloud storage path. Permissions: READ operation on the URL permits the user to perform HiveServer2 operations which use S3 as data source for Hive tables. WRITE operation on the URL permits the user to perform HiveServer2 operations which write data to the specified S3 location.
URI	Hive INSERT OVERWRITE queries require a Ranger URI policy to allow write operations, even if the user has write privilege granted through HDFS policy. Failure to specify this field will result in the following error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied: user [jdoe] does not have [WRITE] privilege on [/tmp/*] (state=42000,code=40000) Example value: /tmp/*
Description	(Optional) Describe the purpose of the policy.
Hive Service Name	hiveservice is used only in conjunction with Permissions=Service Admin. Enables a user who has Service Admin permission in Ranger to run the kill query API: kill query <queryID> . Supported value: *. (Required)
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.
Add Validity Period	Specify a start and end time for the policy.

Table 30: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.

Label	Description
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Permissions	Add or edit permissions: Select, Update, Create, Drop, Alter, Index, Lock, All, ReplAdmin, Service Admin, Temp UDF Admin, Refresh, RW Storage, Select/Deselect All. Service Admin is used in conjunction with Hive Service Name and the kill query API: kill query <queryID> .
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

What to do next

Provide User Access to Hive Database Tables from the Command Line

Hive provides the means to manage user access to Hive database tables directly from the command line. The most commonly-used commands are:

- GRANT

  Syntax:

  ```
  grant <permissions> on table <table> to user <user or group>;
  ```

  For example, to create a policy that grants user1 SELECT permission on the table default-hivesmoke22074, the command would be:

  ```
  grant select on table default.hivesmoke22074 to user user1;
  ```

  The syntax is the same for granting UPDATE, CREATE, DROP, ALTER, INDEX, LOCK, ALL, and ADMIN rights.
- REVOKE

  Syntax:

  ```
  revoke <permissions> on table <table> from user <user or group>;
  ```

  For example, to revoke the SELECT rights of user1 to the table default.hivesmoke22074, the command would be:

  ```
  revoke select on table default.hivesmoke22074 from user user1;
  ```

  The syntax is the same for revoking UPDATE, CREATE, DROP, ALTER, INDEX, LOCK, ALL, and ADMIN rights.

Related Information

Wildcards and variables in resource-based policies

Configure a resource-based storage handler policy: HadoopSQL

How to configure a policy that allows authorized users to create data tables using storage-handlers.

About this task

Ranger includes “storage-type” and “storage-url” resources in HadoopSQL Service that support only the “RW Storage ” permission. Ranger authorizes users to create or alter tables against this resource policy. Users are allowed to create/alter the table in the respective storage if they have the required “RW Storage” permission on the resource representing the storage-type and storage-url .

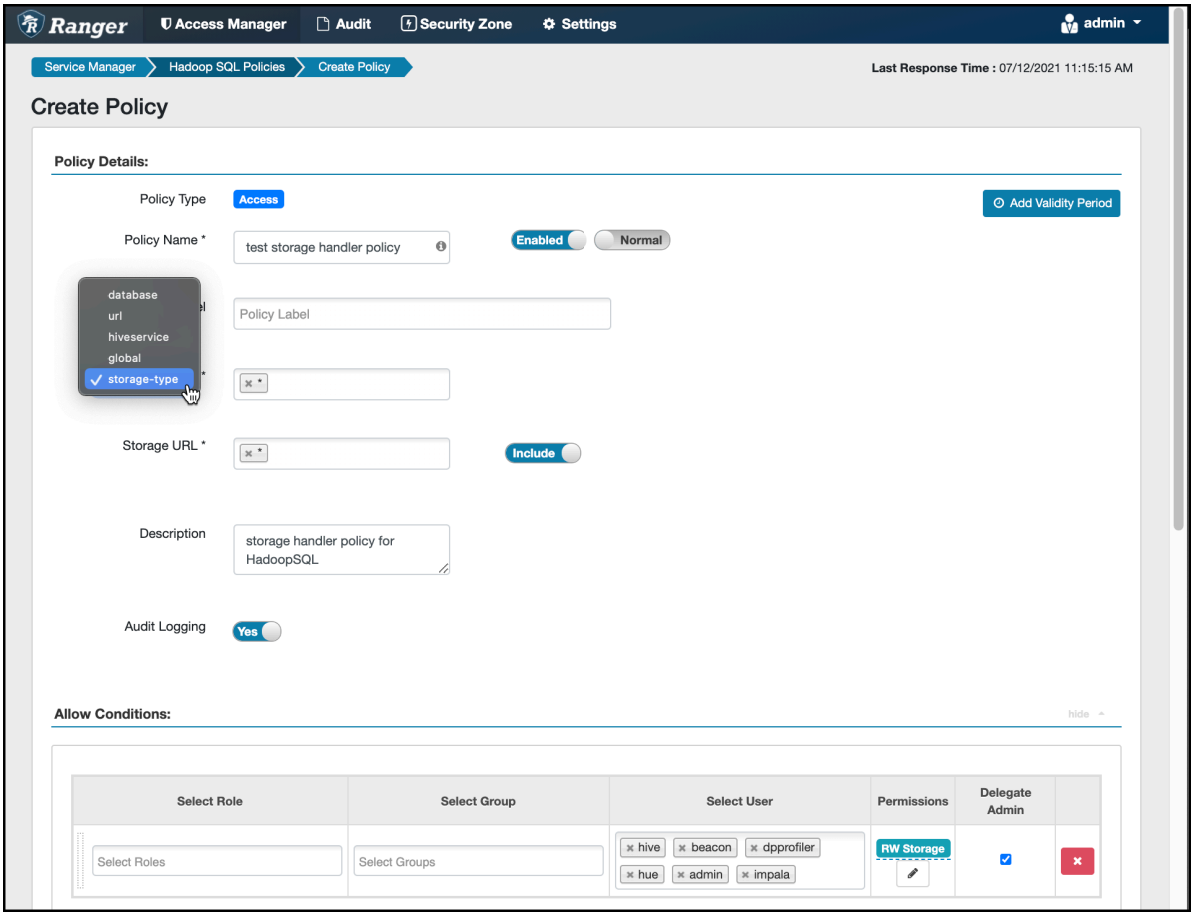
Procedure

1. Select the HadoopSQLservice on the Service Manager page.

   The List of Policies HadoopSQL page appears.

   Note: Service_name remains cm_hive. Display name is HadoopSQL.

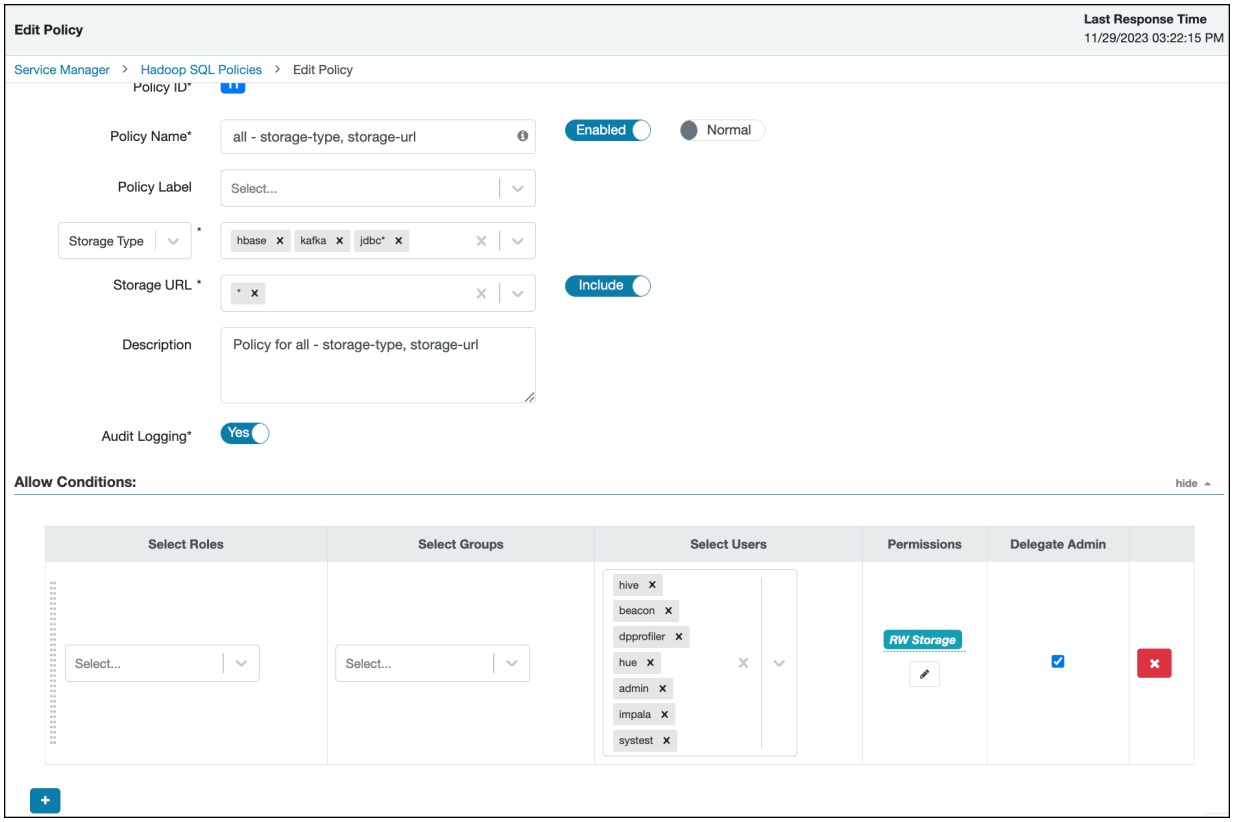
2. Select Add New Policy to create a new policy.
   a) Within Create Policy, select storage-type as shown in the following example:



   b) Complete the required* fields.
   c) Under Allow Conditions, select users and add the RW Storage permission, as shown in the preceeding example.
   d) Scroll to the bottom of Create Policy, then click Add.

3. To configure an existing policy named all - storage-type, storage-url, click Edit.

   The Edit Policy page appears.



4. Complete the Edit Policy page as shown in the preceding example using the follow policy detail descriptions:

Table 31: Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory. The policy is enabled by default.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.
storage-type	Type in the applicable storage type. * allows athorizes users to create any table in the spcified storage type..
storage url	Type in the applicable storage URL. * allows athorizes users to create any table in the spcified storage URL. Select Exclude to deny access.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).

Field	Description
Add Validity Period	Specify a start and end time for the policy.

Table 32: Allow Conditions

Label	Description
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Permissions	Add or edit permissions: RW Storage, You can assign read and select permissions to rangerlookup user.
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

Example

Example StorageHandler Policy Definitions and Use Cases:

HBase StorageHandler policy:

Storage Type: hbase

Storage URL: hbase-cluster:port/hbase-table

Storage create table command:

```
CREATE [EXTERNAL] table foo(…)
STORED BY‘org.apache.hadoop.hive.hbase.HBaseStorageHandler’
TBLPROPERTIES (‘hbase.table.name’ = ‘bar’);
```

```
e.g:
CREATE TABLE hive_hbase_test_1(key int, value string) STORED BY
 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPR
OPERTIES ("hbase.columns.mapping" = "cf:string", "hbase.table.na
me" = "hbase_test_1");
```

Iceberg StorageHandler policy:

Storage type: iceberg

Storage URL: DBname/Table* , or

Storage URL: DBname/*

JDBC StorageHandler policy:

Storage Type: jdbc:mysql

Storage URL: mysql-host:port/DBname/Table , or

Storage URL: mysql-host:port/DBname/*

Note:

Policy and table definitions must be in sync regarding the port definition, even for default port numbers. For example, if port number 3306 is defined in the policy for mysql and this port number is left out from the URL as default value for the JDBC Driver, you must use the same reference as defined in the policy when creating the external table.

Using an explicit table name allows only to reference that specific table with hive.sql.table while using * allows not only to reference any tables from the database but also allows you to write a custom query against this database, for example using hive.sql.query.

Storage create table command:

```
CREATE [EXTERNAL] TABLE student_jdbc
(
  name string,
  age int,
  gpa double
)
STORED BY 'org.apache.hive.storage.jdbc.JdbcStorageHandler'
TBLPROPERTIES (
    "hive.sql.database.type" = "MYSQL",
    "hive.sql.jdbc.driver" = "com.mysql.jdbc.Driver",
    "hive.sql.jdbc.url" = "jdbc:mysql://localhost/sample",
    "hive.sql.dbcp.username" = "hive",
    "hive.sql.dbcp.password" = "hive",
    "hive.sql.table" = "STUDENT",
    "hive.sql.dbcp.maxActive" = "1"
);
```

Kafka StorageHandler policy:

Storage Type: kafka

Storage URL: bootstrap-server:port/kafka-topic

Phoenix StorageHandler policy:

Storage Type: phoenix

Storage URL: phoenix-cluster:port/table-name

Storage create table command:

```
CREATE [EXTERNAL] TABLE phoenix_table (
                              s1 string,
                              i1 int,
                              f1 float,
                              d1 double
                              )
                              STORED BY 'org.apache.phoenix.hive.
PhoenixStorageHandler'
                              TBLPROPERTIES (
                              "phoenix.table.name" = "phoenix_table
",
                              "phoenix.zookeeper.quorum" = "localho
st",
                              "phoenix.zookeeper.znode.parent" = "/
hbase",
                              "phoenix.zookeeper.client.port" =
"2181",
                              "phoenix.rowkeys" = "s1, i1",
                              "phoenix.column.mapping" = "s1:s1,
 i1:i1, f1:f1, d1:d1",
```

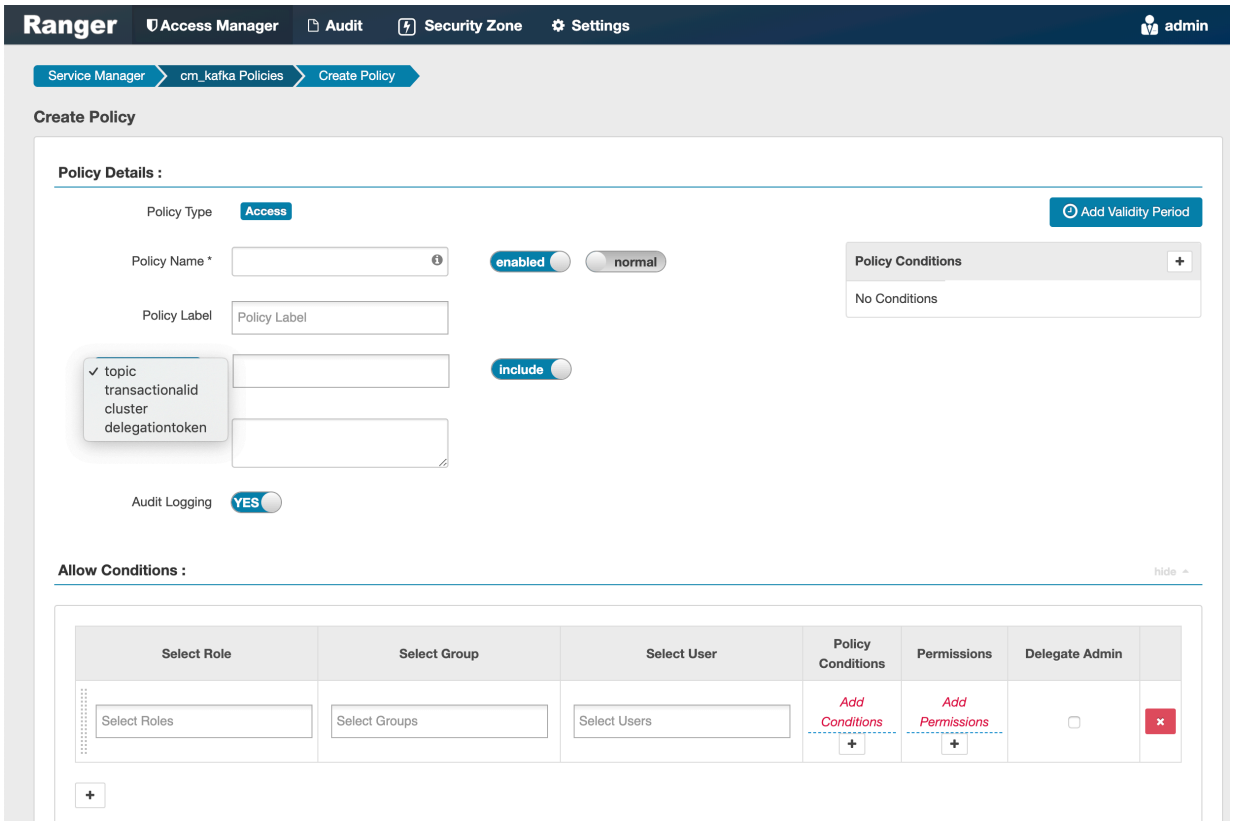
```
                                "phoenix.table.options" = "SALT_BUCKE
TS=10, DATA_BLOCK_ENCODING='DIFF'"
                                );
```

Configure a resource-based policy: Kafka

How to add a new policy to an existing Kafka service.

Procedure

1. On the Service Manager page, select an existing Kafka service.
   The List of Policies page appears.
2. Click Add New Policy.
   The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 33: Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.
Topic	Kafka resource type. A topic is a category or feed name to which messages are published.

Field	Description
Transactional ID	Kafka resource type, uniquely identifies producers in a persistent way.
Cluster	Kafka resource type.
Delegation Token	Kafka resource type for authentication.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Add Validity Period	Specify a start and end time for the policy.
Policy Conditions (applied at the policy level)	Click the + icon, then specify an IP address range.

Table 34: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Policy Conditions (applied at the item level)	Specify an IP address range.
Permissions	Add or edit permissions: Publish, Consume,Configure,Describe, Create, Delete, Describe Configs, Alter Configs, Select/Deselect All.
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

Related Information

Wildcards and variables in resource-based policies

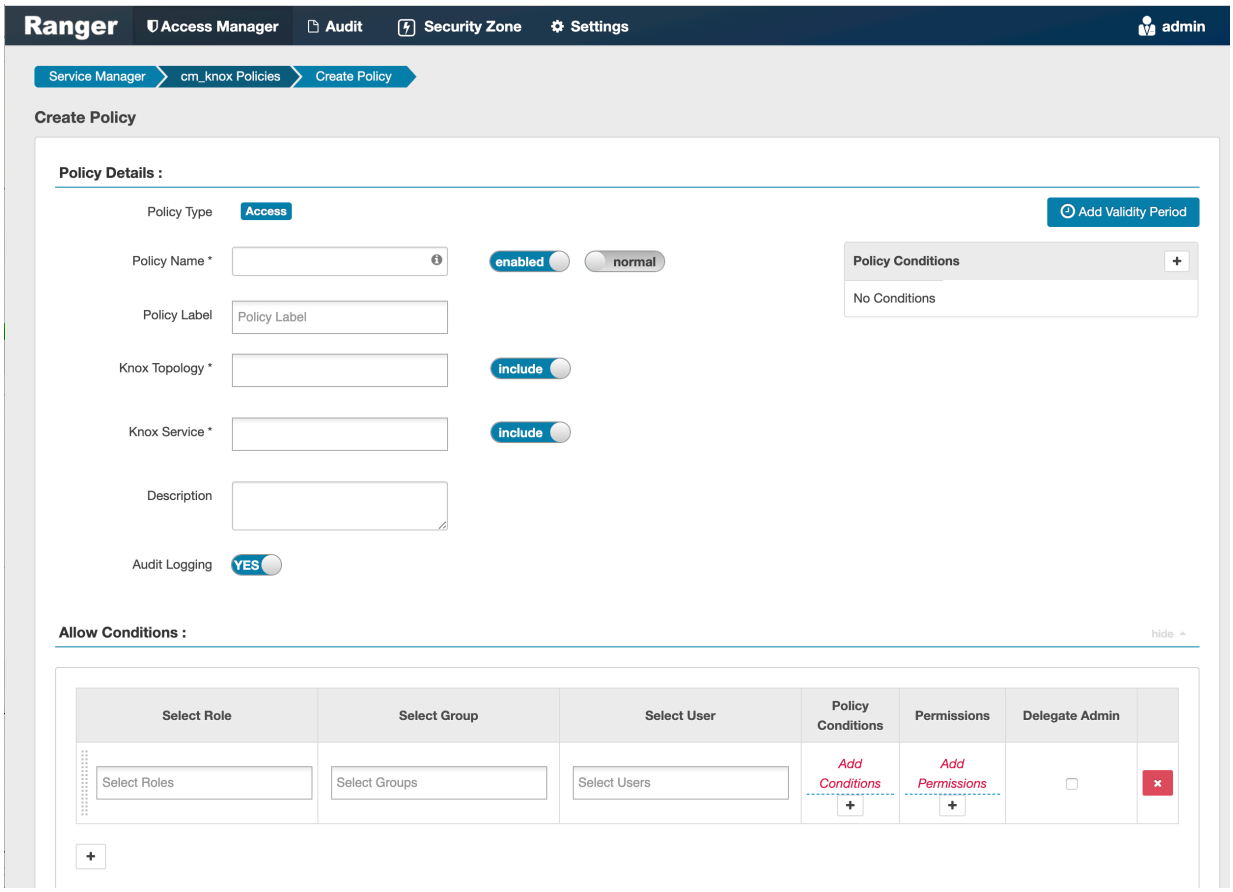
Configure a resource-based policy: Knox

How to add a new policy to an existing Knox service.

Procedure

1. On the Service Manager page, select an existing Knox service.
   The List of Policies page appears.

2. Click Add New Policy.
The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 35: Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
Knox Topology	Enter an appropriate Topology Name.
Knox Service	Enter an appropriate Service Name.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.
Add Validity Period	Specify a start and end time for the policy.

Field	Description
Policy Conditions (applied at the policy level)	Click the + icon, then specify an IP address range.

Table 36: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Policy Conditions (applied at the item level)	Specify an IP address range.
Permissions	Add or edit permissions: Allow
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

Since Knox does not provide a command line methodology for assigning privileges or roles to users, the User and Group Permissions portion of the Knox Create Policy form is especially important.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

Related Information

Wildcards and variables in resource-based policies

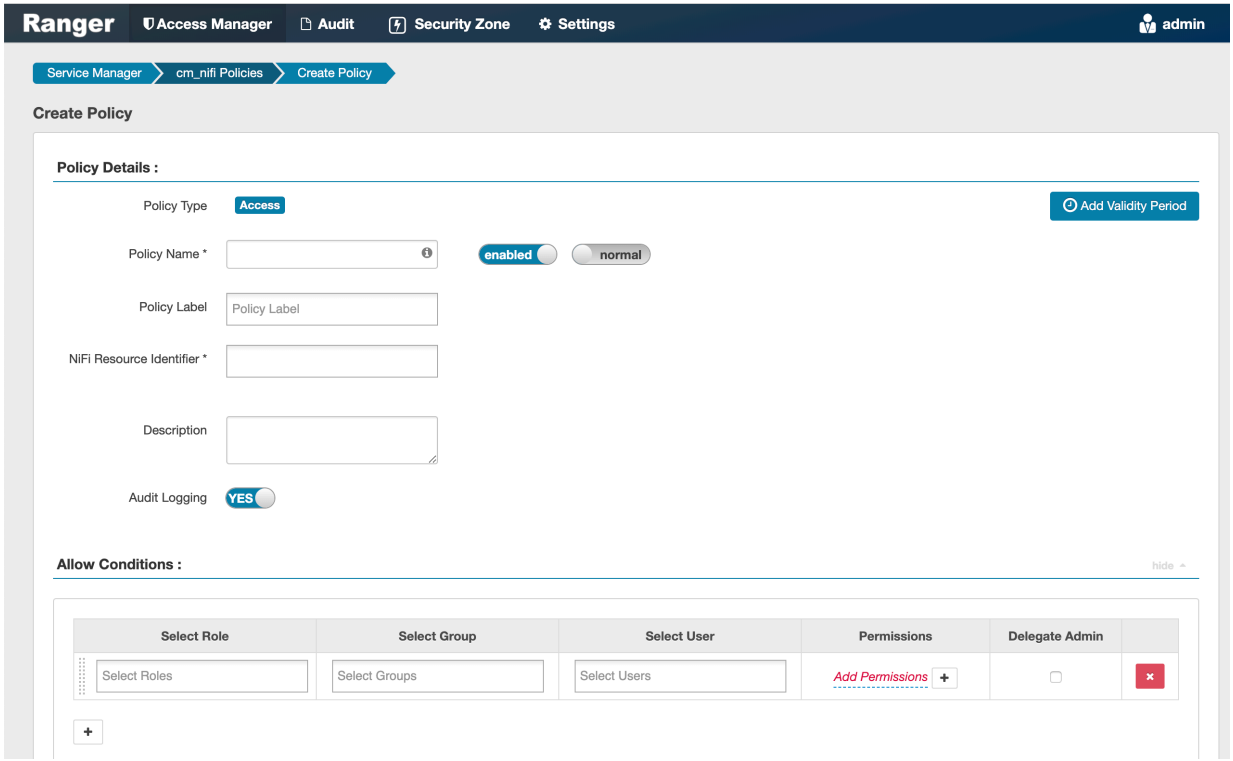
Configure a resource-based policy: NiFi

How to add a new policy to an existing Atlas service.

Procedure

1. On the Service Manager page, select an existing NiFi service.
   The List of Policies page appears.

2. Click Add New Policy.
   The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 37: Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
NiFi Resource Identifier	In a NiFi cluster, all nodes must be granted the ability to view and modify component data in order for user to list or empty queues in processor component outbound connections. With Ranger this can be accomplished by using a wildcard to grant all of the NiFi nodes read and write access to the /data/* NiFi resource.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.

Field	Description
Add Validity Period	Specify a start and end time for the policy.

Table 38: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Permissions	Add or edit permissions: Read, Write, Select/Deselect All.
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

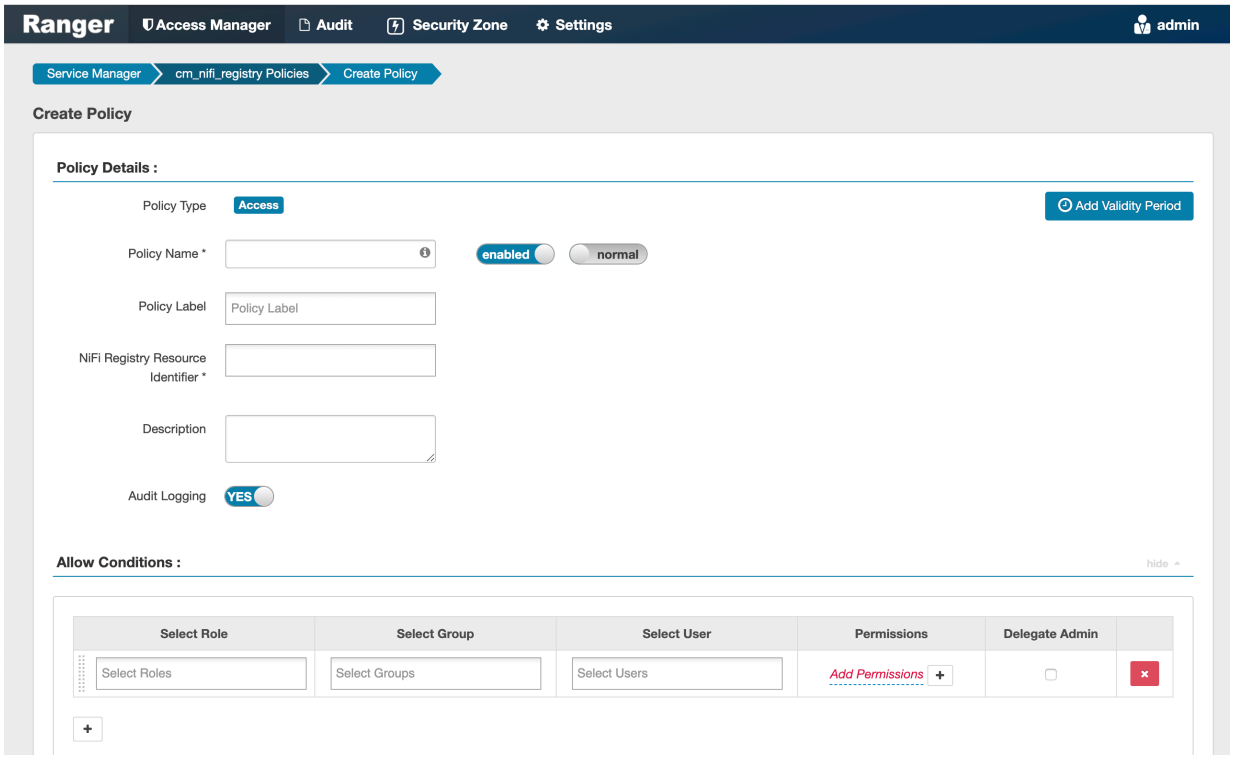
Configure a resource-based policy: NiFi Registry

How to add a new policy to an existing Atlas service.

Procedure

1. On the Service Manager page, select an existing NiFi Registry service.
   The List of Policies page appears.

2. Click Add New Policy.
The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 39: Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
NiFi Registry Resource Identifier	In a NiFi cluster, all nodes must be granted the ability to view and modify component data in order for user to list or empty queues in processor component outbound connections. With Ranger this can be accomplished by using a wildcard to grant all of the NiFi nodes read and write access to the /data/* NiFi resource.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.

Field	Description
Add Validity Period	Specify a start and end time for the policy.

Table 40: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Permissions	Add or edit permissions: Read, Write, Delete, Select/Deselect All.
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

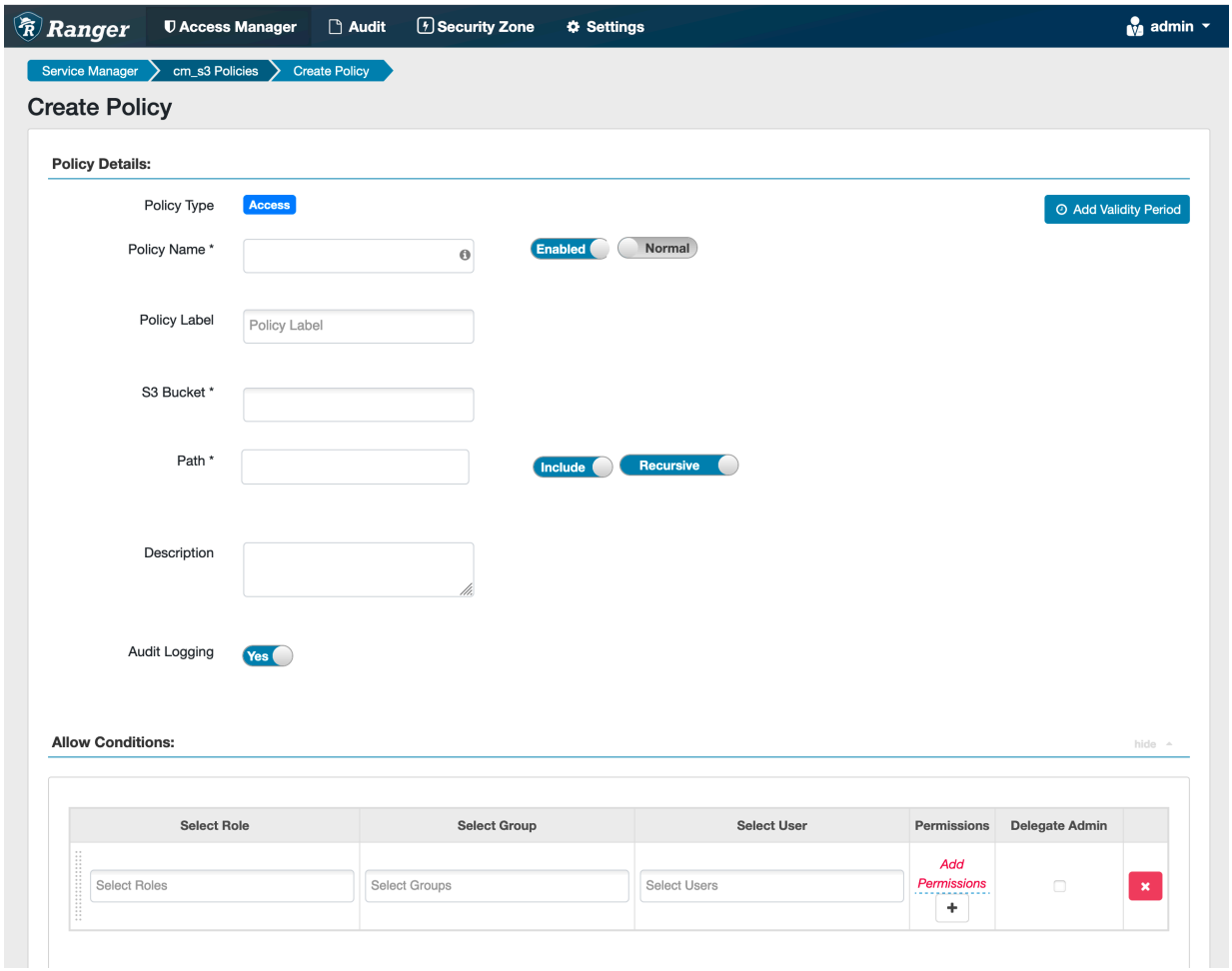
Related Information
SQL Standard Based Hive Authorization

Configure a resource-based policy: S3
How to add a new policy to an existing S3 service.

Procedure
1. On the Service Manager page, select an existing S3 service.

   The List of Policies page appears.

2. Click Add New Policy.
The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 41: Policy Details

Field	Description
Policy Name	Enter a unique name for this policy. The name cannot be duplicated anywhere in the system.
Policy Label	An optional label for the policy. You can search reports and filter policies based on these labels.
Enabled/Disabled	Enables or disables the policy.
Normal/Override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
S3 Bucket	The S3 bucket.
Path	Specify the path for the policy. The default Recursive setting specifies that the path is recursive; you can also specify a non-recursive path. The default Include setting specifies that the path is included; you can also exclude the path.
Description	(Optional) Describe the purpose of the policy.

Field	Description
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Add Validity Period	Specify a start and end time for the policy.

Table 42: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Permissions	Add or edit permissions: Read, Write, Select/Deselect All.
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

Configure a resource-based policy: Solr

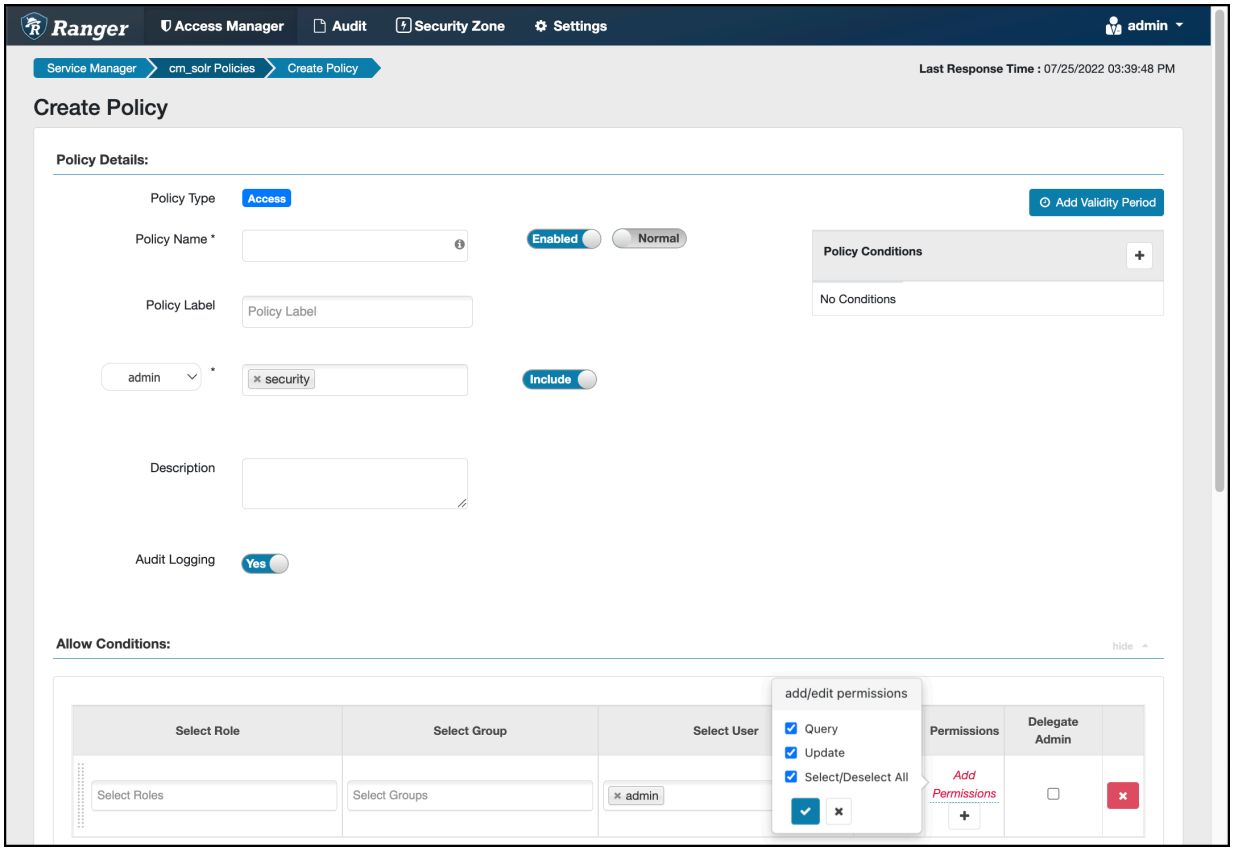
How to add a new policy to an existing Solr service.

Procedure

1. On the Service Manager page, select an existing Solr service.

   The List of Policies page appears.

2. Click Add New Policy.
The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 43: Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
Resource Type	collection - click to select from a list of dynamic values config - click to select from a list of dynamic values schema - click to select from a list of dynamic values admin - click to select COLLECTIONS, CORES, METRICS, SECURITY or AUTOSCALING
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.
Add Validity Period	Specify a start and end time for the policy.

Field	Description
Policy Conditions (applied at the policy level)	Click the + icon, then specify an IP address range.

Table 44: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Policy Conditions (applied at the item level)	Specify an IP address range.
Permissions	Add or edit permissions: Query, Update
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

Related Information

Wildcards and variables in resource-based policies

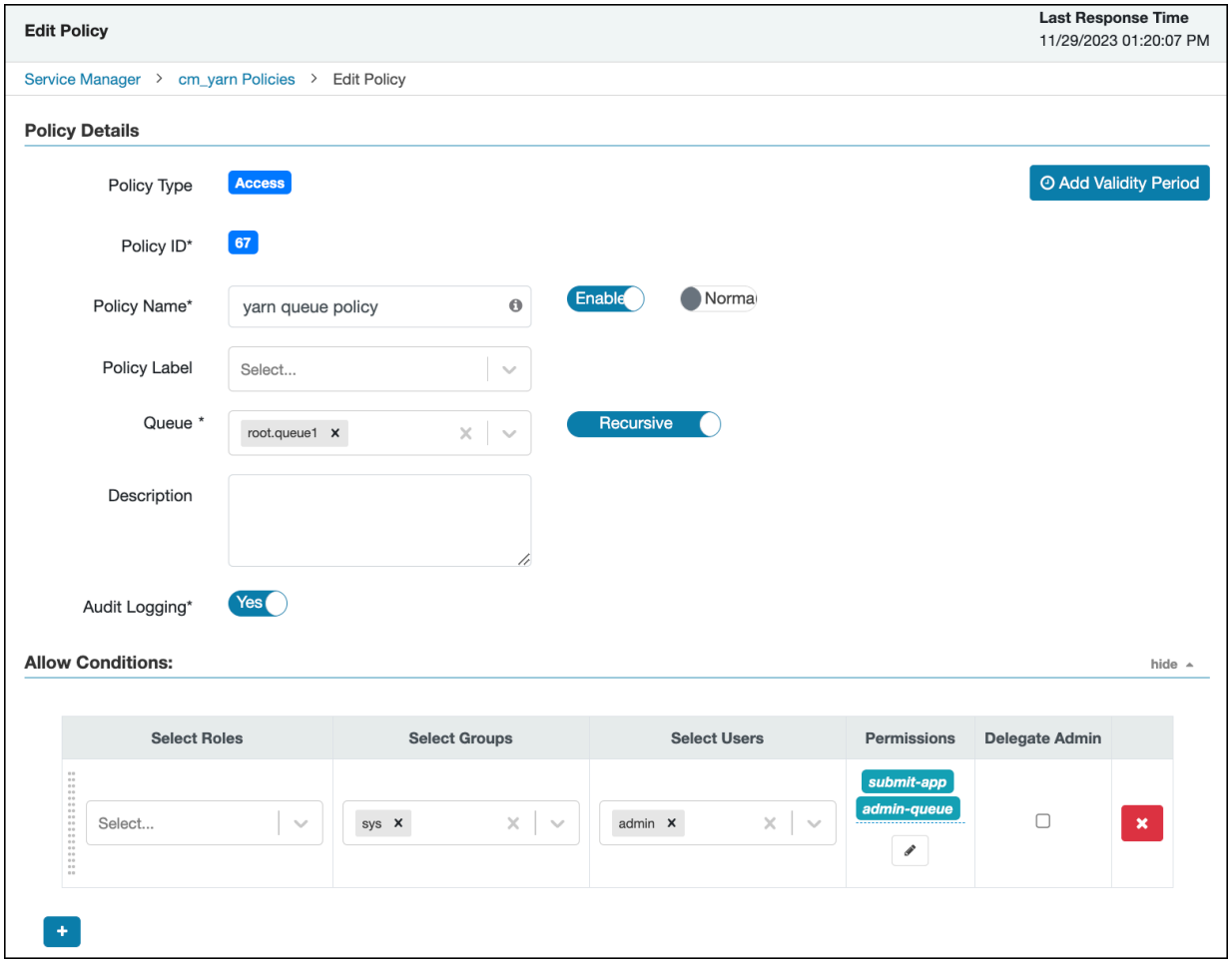
Configure a resource-based policy: YARN

How to add a new policy to an existing YARN service.

Procedure

1. On the Service Manager page, select an existing YARN service.
   The List of Policies page appears.

2. Click Add New Policy.
The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 45: Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
Queue	The YARN queue to which the policy applies. For example, enter root.xyz for the xyz queue.
Recursive	The default recursive setting specifies that the policy will also be applied to all sub-queues; you can also specify a non-recursive path.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.

Field	Description
Add Validity Period	Specify a start and end time for the policy.

Table 46: Allow Conditions

Label	Description
Select Role	Specify the roles to which this policy applies. To designate a role as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Select Group	Specify the groups to which this policy applies. To designate a group as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify the users to which this policy applies. To designate a user as an Administrator, select the Delegate Admin check box. Administrators can edit or delete the policy, and can also create child policies based on the original policy.
Permissions	Add or edit permissions: submit-app, admin-queue, Select/Deselect All.
Delegate Admin	You can use Delegate Admin to assign administrator privileges to the roles, groups, or users specified in the policy. Administrators can edit or delete the policy, and can also create child policies based on the original policy.

4. You can use the Plus (+) symbol to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add.

Related Information

Wildcards and variables in resource-based policies

Wildcards and variables in resource-based policies

Reference for wildcards and variables in resource-based policies.

Ranger Authorization Resource Policy Wildcard Characters

Wildcard characters can be included in the resource path, the database name, the table name, or the column name:

- * indicates zero or more occurrences of characters
- ? indicates a single character

Ranger Authorization Resource Policy {USER} Variable

The variable {USER} can be used to autofill the accessing user, for example:

In Select User, choose {USER}.

In Resource Path, enter data_{USER}.

Ranger Authorization Resource Policy {USER} Variable Recommended Practices and Customizability

Ranger requires that string '{USER}' is used to represent accessing user as the user in the policy-item in a Ranger policy. However, Ranger provides flexible way of customizing the string that is used as shorthand to represent the

accessing user's name in the policy resource specification. By default, Ranger policy resource specification expects characters '{' and '}' as delimiters for string 'USER', however, ranger supports customizable way of specifying delimiter characters, escaping those delimiters, and the string 'USER' itself by prefixing it with another, user-specified string on a per resource-level basis in the service definition of each component supported by Ranger.

For example, if for a certain HDFS installation, if the path names may contain '{' or '}' as valid characters, but not '%' character, then the service-definition for HDFS can be specified as:

```
"resources": [
{
      "itemId": 1,
      "name": "path",
      "type": "path",
      "level": 10,
      "parent": "",
      "mandatory": true,
      "lookupSupported": true,
      "recursiveSupported": true,
      "excludesSupported": false,
      "matcher": "org.apache.ranger.plugin.resourcematcher.RangerPathReso
urceMatcher",
      "matcherOptions": {"wildcard": true, "ignoreCase": false}, "replaceT
okens":true, "tokenDelimiterStart":"%", "tokenDelimiterEnd":"%", "tokenDelim
iterPrefix":"rangerToken:"}
      "validationRegEx":"",
      "validationMessage": "",
      "uiHint":"",
      "label": "Resource Path",
      "description": "HDFS file or directory
path"
}
]
```

Corresponding ranger policy for the use case for HDFS will be written as follow:

```
resource: path=/home/%rangerToken:USER%
user: {USER}
permissions: all, delegateAdmin=true
```

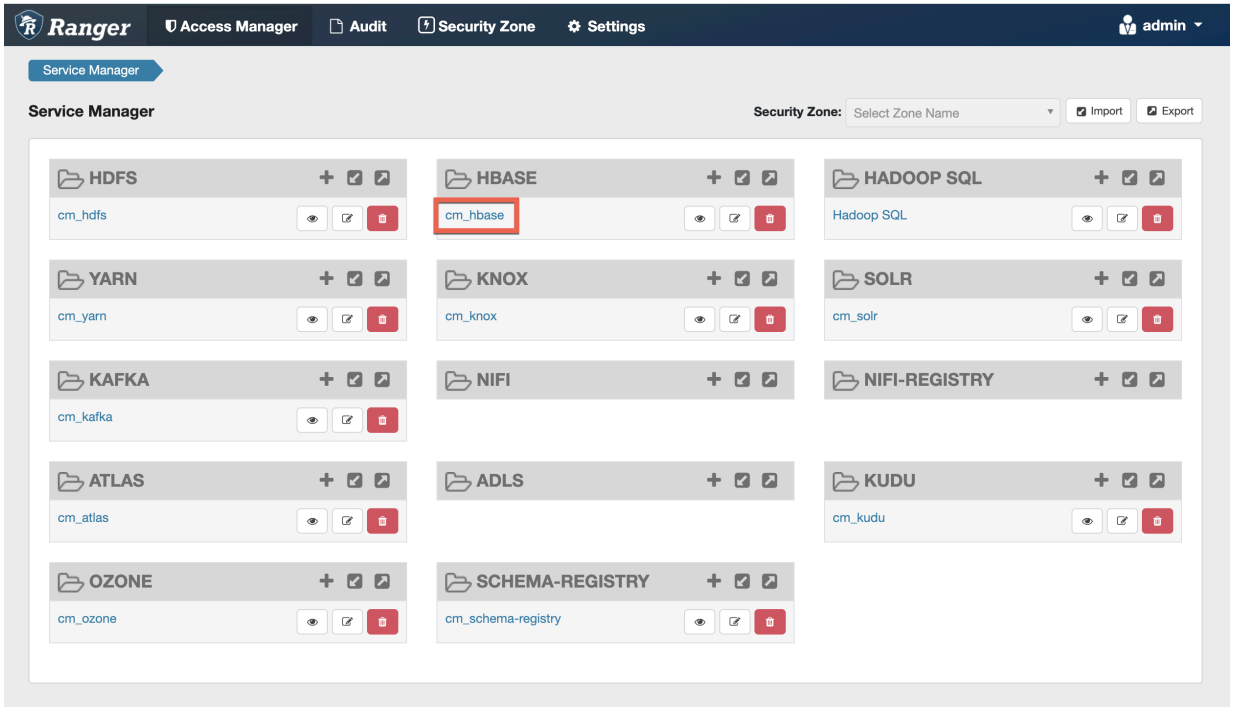
The following customizable matcherOptions are available for this feature:

- replaceTokens: true if short-hand for user in resource-spec needs to be replaced at run-time with current-user's name; false if the resource-spec needs to be interpreted as it is. Default value: true.
- tokenDelimiterStart: Identifies start character of short-hand for current-user in resource specification. Default value: {.
- tokenDelimiterEnd: Identifies end character of short-hand for current-user in resource specification. Default value:}.
- tokenDelimiterEscape: Identifies escape character for escaping tokenDelimiterStart or tokenDelimiterEnd values in resource specification. Default value: \.
- tokenDelimiterPrefix: Identifies special prefix which together with string 'USER' makes up short-hand for current-user's name in the resource specification. Default value: .

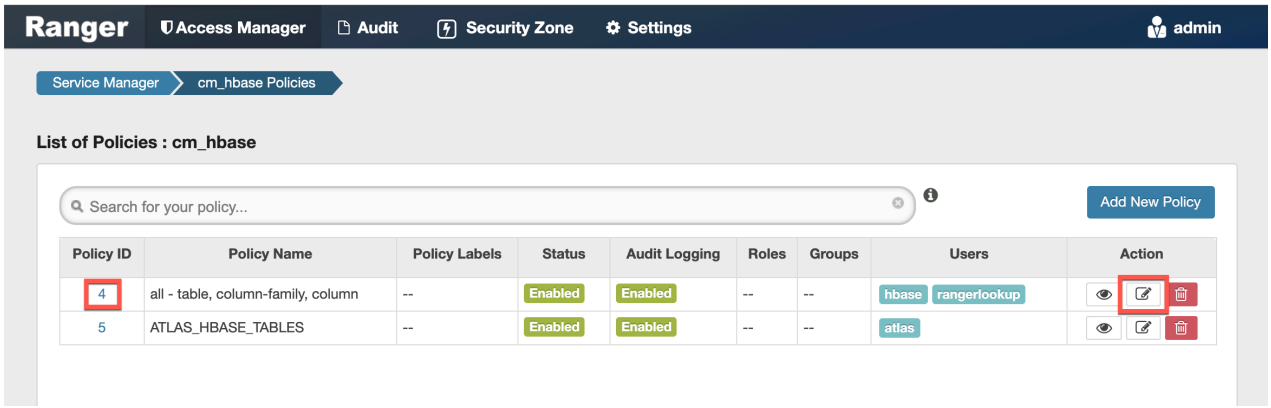
Preloaded resource-based services and policies

Apache Ranger includes preloaded resource-based services and policies.

- The preloaded resource-based services appear on the Service Manager page for resource-based policies, and are prefixed with "cm_", with the exception of Hadoop SQL, which applies to multiple SQL stack components (Hive, Impala, and Hue).



- To view the policies for each preloaded service, click the service name. To view policy details, click the applicable edit icon or policy ID number.



Index

cm_atlas

cm_hbase

cm_hdfs

cm_kafka

cm_knox

cm_nifi

cm_solr

cm_yarn

Hadoop SQL

cm_atlas

all - entity-type, entity-classification, entity, entity-business-metadata

This is a default policy of type "entity" that gives access to all entities and their business metadata attributes for the following users and groups, with the specified permissions:

- admin, dpprofiler, beacon – Update Business Metadata
- rangertagsync, rangerlookup – Read entity
- public group – Read entity

all - entity-type, entity-classification, entity

This is a default policy of type "entity" that gives access to all entities and their classifications for the following users and groups, with the specified permissions:

- admin, dpprofiler, beacon – Read, Create, Update, Delete entity & Add, Update, Remove classification
- rangertagsync, rangerlookup – Read entity
- public group – Read entity

all - entity-type, entity-classification, entity, entity-label

This is a default policy of type "entity" that gives access to all entities and classifications and their labels for the following users and groups, with the specified permissions:

- admin, dpprofiler, beacon – Add, Remove label
- rangertagsync, rangerlookup – Read entity
- public group – Read entity

all - relationship-type, end-one-entity-type, end-one-entity-classification, end-one-entity, end-two-entity-type, end-two-entity-classification, end-two-entity

This is a default policy of type "relationship" that gives access to all to all Entity-Relationships between End1-Entity-Type, End1-Entity-Classification, End1-Entity-ID and End2-Entity-Type, End2-Entity-Classification, End2-Entity-ID for the following users and groups, with the specified permissions:

- admin, dpprofiler, beacon – Add, Update, and Remove relationship
- public group – Add, Update, and Remove relationship

all - atlas-service

This is a default policy of type "atlas-service" that gives access to all atlas-services [export, import, purge, server] for the following users, with the specified permissions:

- admin, dpprofiler, beacon – Admin Export and Admin Import

all - type-category, type

This is a default policy of type "type-category" that gives access to all type categories [ENUM, ENTITY, CLASSIFICATION, RELATIOSHIP, STRUCT] and type names for the following users, with the specified permissions:

- admin, dpprofiler, beacon – Create, Update, and Delete type

Allow users to manage favorite searches

This is a default policy of type "entity-type" that gives access to __AtlasUserProfile and __AtlasUserSavedSearch resources which are internal types for favorite search. This policy provides Read, Create, Update, and Delete Entity permissions to validated users who create a favorite search.

cm_hbase

all - table, column-family, column

Provides access to all HBase tables, column-families, and columns to the following users, with the specified permissions:

- hbase, rangerlookup – Read, Write, Create, Admin

ATLAS_HBASE_TABLES

Provides access to all HBase column-families and columns in the atlas_janus and ATLAS_ENTITY_AUDIT_EVENTS HBase tables, to the following user, with the specified permissions:

- atlas – Read, Write, Create, Admin

cm_hdfs

all - path

Provides access to all HDFS resource paths to the following users, with the specified permissions:

- hdfs, rangerlookup – Read, Write, Execute

kms-audit-path

Provides access to the /ranger/audit/kms resource path to the following user, with the specified permissions:

- keyadmin – Read, Write, Execute

cm_kafka

all - topic

Provides access to all topics to the following users, with the specified permissions:

- kafka, rangerlookup, streamsmsgmgr, streamsrepmgr – Publish, Consume, Configure, Describe, Create, Delete, Describe Configs, Alter Configs

all - cluster

Provides access to all clusters to the following users, with the specified permissions:

- kafka, rangerlookup, streamsmsgmgr, streamsrepmgr – Configure, Describe, Create, Kafka Admin, Idempotent Write, Describe Configs, Alter Configs

all - transactionalid

Provides transactionalid access to the following users, with the specified permissions:

- kafka, rangerlookup, streamsmsgmgr, streamsrepmgr – Publish, Describe

all - delegationtoken

Provides delegationtoken access to the following users, with the specified permissions:

- kafka, rangerlookup, streamsmsgmgr, streamsrepmgr – Describe

ATLAS_HOOK

Provides ATLAS_HOOK topic access to the following users, with the specified permissions:

- hbase, hive, impala, mlgov – Publish
- atlas – Create, Configure, and Consume

ATLAS_ENTITIES

Provides ATLAS_ENTITIES topic access to the following users, with the specified permissions:

- atlas – Create, Configure, and Publish
- rangertagsync – Consume

ATLAS_SPARK_HOOK

Provides ATLAS_SPARK_HOOK topic access to the following user, with the specified permissions:

- atlas – Create, Configure, and Consume

Also provides ATLAS_SPARK_HOOK topic access to the following group, with the specified permissions:

- public – Publish

cm_knox

all - topology, service

Provides access to all Knox topologies and services to the following users, with the specified permissions:

- admin, rangerlookup – Allow

cm_nifi

all - nifi-resource

Provides access to all NiFi resource identifiers to the following user, with the specified permissions:

- rangerlookup – Read, Write

cm_solr

all - collection

Provides access to all Solr collections to the following users, with the specified permissions:

- solr, rangerlookup, ranger, atlas – Query, Update, Others, Solr Admin

RANGER_AUDITS_COLLECTION

Provides access to the RANGER_AUDITS_COLLECTION Solr collection to the following users, with the specified permissions:

- atlas, hbase, hdfs, hive, impala, kafka, knox, nifi, ranger, storm, yarn – Query, Update, Others
- ranger – Query, Update, Others, Solr Admin

cm_yarn

all - queue

Provides access to all YARN queues to the following users, with the specified permissions:

- yarn, rangerlookup – submit-app, admin-queue

Hadoop SQL

all - global

Provides global access to the following users, with the specified permission:

- hive, beacon, dpprofiler, hue, admin, impala, rangerlookup – Temporary UDF Admin

Note: The Ranger web UI may show additional permissions for the all-global policy, but the only valid permission is Temporary UDF Admin.

all - database, table, column

Provides access to all databases, tables, and columns to the following users, with the specified permissions:

- hive, rangerlookup, impala – Select, Update, Create, Drop, Alter, Index, Lock, All, Read, Write, ReplAdmin, Service Admin, Temporary UDF Admin, Refresh
- {OWNER} – All

all - database, table

Provides access to all databases and tables to the following users, with the specified permissions:

- hive, rangerlookup, impala – Select, Update, Create, Drop, Alter, Index, Lock, All, Read, Write, ReplAdmin, Service Admin, Temporary UDF Admin, Refresh
- {OWNER} – All

all - storage-type, storage-url

Ranger introduces new resources “storage-type” and “storage-url” in HadoopSQL Service and supports only one permission “RW Storage ”. When a user creates / alters a table, they will be authorized against this resource policy. Users granted “RW Storage” permission on the resource representing the storage-type + storage-url, can create/alter the table in the respective storage. Provides ccess to all databases to the following users, with the RW Storage permission only:

- hive, rangerlookup, impala, beacon, dpprofiler, hue, admin

Note: {OWNER} macro should NOT be configured for StorageHandler policies.

all - database

Provides access to all databases to the following users, with the specified permissions:

- hive, rangerlookup, impala – Select, Update, Create, Drop, Alter, Index, Lock, All, Read, Write, ReplAdmin, Service Admin, Temporary UDF Admin, Refresh
- {OWNER} – All

Also provides access to all databases to the following group, with the specified permissions:

- public – Create

all - hiveservice

Provides hiveservice access to the following users, with the specified permissions:

- hive, rangerlookup, impala – Select, Update, Create, Drop, Alter, Index, Lock, All, Read, Write, ReplAdmin, Service Admin, Temporary UDF Admin, Refresh

all - database, udf

Provides database and udf access to the following users, with the specified permissions:

- hive, rangerlookup, impala – Select, Update, Create, Drop, Alter, Index, Lock, All, Read, Write, ReplAdmin, Service Admin, Temporary UDF Admin, Refresh
- {OWNER} – All

all - url

Provides url access to the following users, with the specified permissions:

- hive, rangerlookup, impala – Select, Update, Create, Drop, Alter, Index, Lock, All, Read, Write, ReplAdmin, Service Admin, Temporary UDF Admin, Refresh

default database tables columns

Provides access to all tables and columns in the default database to the following user, with the specified permissions:

- impala – Create

Also provides access to all tables and columns in the default database to the following group, with the specified permissions:

- public – Create

information_schema database tables columns

Provides access to all tables and columns in the information_schema database to the following user, with the specified permissions:

- impala – Select

Also provides access to all tables and columns in the information_schema database to the following group, with the specified permissions:

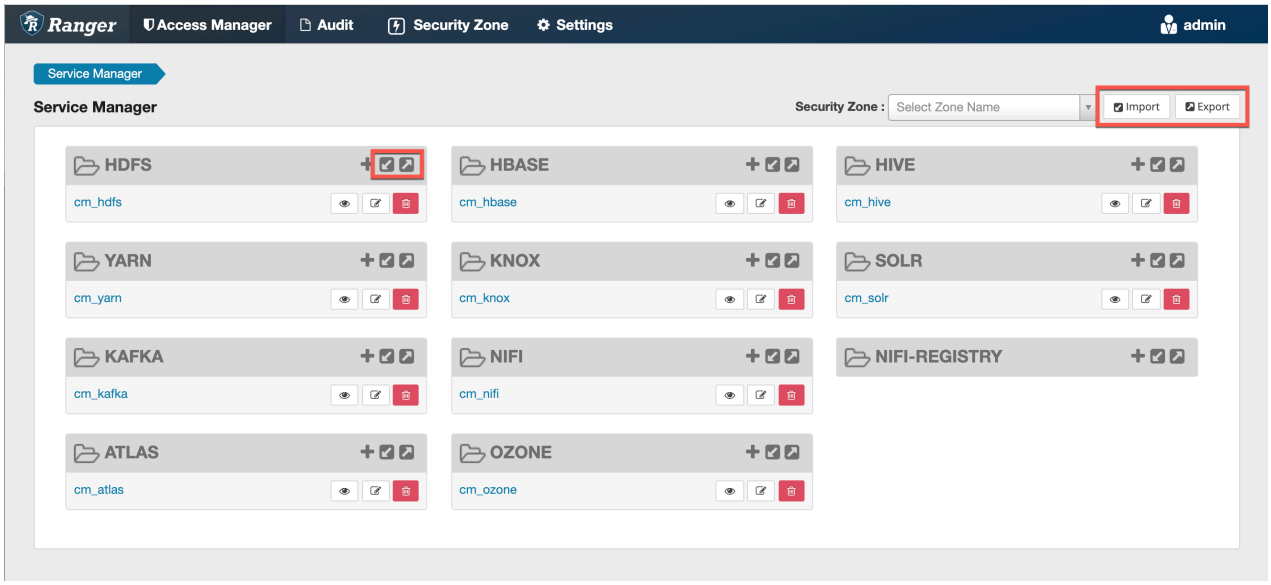
- public – Select

Importing and exporting resource-based policies

You can export and import policies from the Ranger Admin UI for cluster resiliency (backups), during recovery operations, or when moving policies from test clusters to production clusters. You can export/import a specific subset of policies (such as those that pertain to specific resources or user/groups) or clone the entire repository (or multiple repositories) via Ranger Admin UI.

Interfaces

You can import and export policies from the Service Manager page:



You can also export policies from the Reports page:

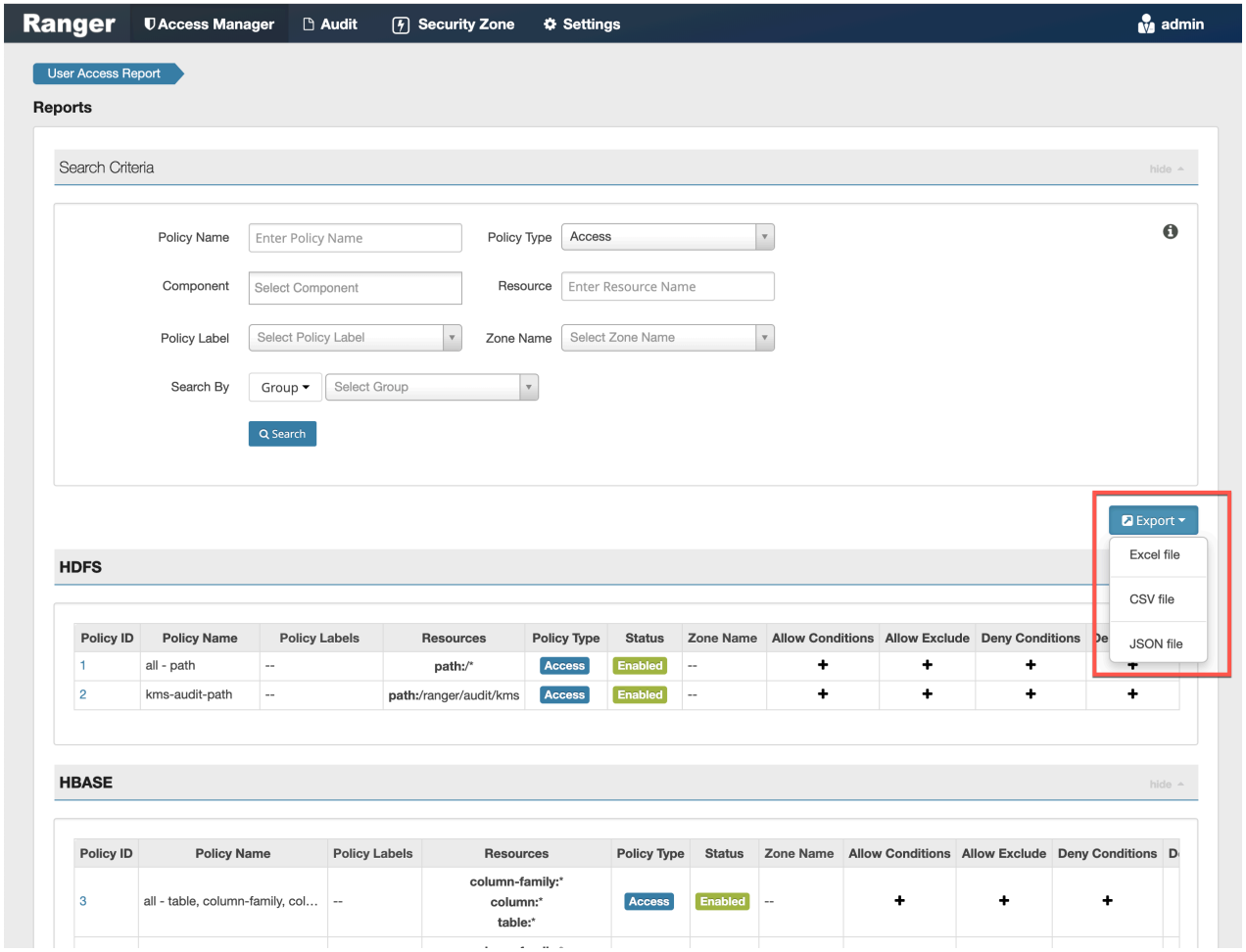


Table 47: Export Policy Options

	Service Manager Page	Reports Page
Formats	JSON	JSON Excel CSV
Filtering Supported	No	Yes
Specific Service Export	Yes	Via filtering

Filtering

When exporting from the Reports page, you can apply filters before saving the file.

Export Formats

You can export policies in the following formats:

- Excel
- JSON
- CSV

  Note: CSV format is not supported for importing policies.

When you export policies from the Service Manager page, the policies are automatically downloaded in JSON format. If you wish to export in Excel or CSV format, export the policies from the Reports page dropdown menu.

Required User Roles

The Ranger admin user can import and export only Resource & Tag based policies. The credentials for this user are set in Ranger Configs > Advanced ranger-env in the fields labeled admin_username (default: admin/admin).

The Ranger KMS keyadmin user can import and export only KMS policies. The default credentials for this user are keyadmin/keyadmin.

Limitations

To successfully import policies, use the following database versions:

- MariaDB: 10.1.16+
- MySQL: 5.6.x+
- Oracle: 11gR2+
- PostgreSQL: 8.4+
- MS SQL: 2008 R2+

Partial import is not supported.

Related Information

Importing and exporting tag-based policies

Import resource-based policies for a specific service

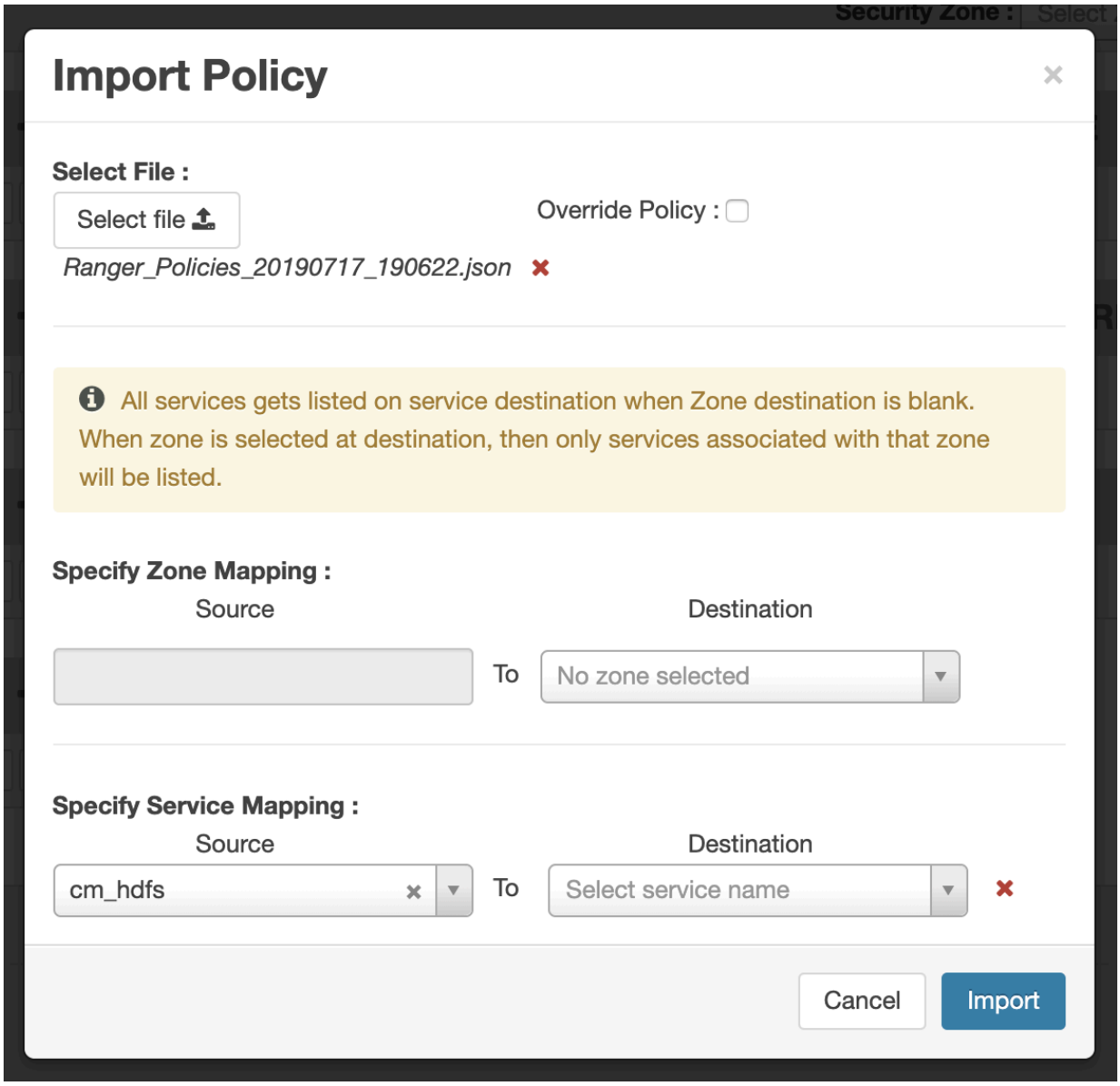
How to import resource-based policies for a specific service (HBase, YARN, etc.).

Procedure

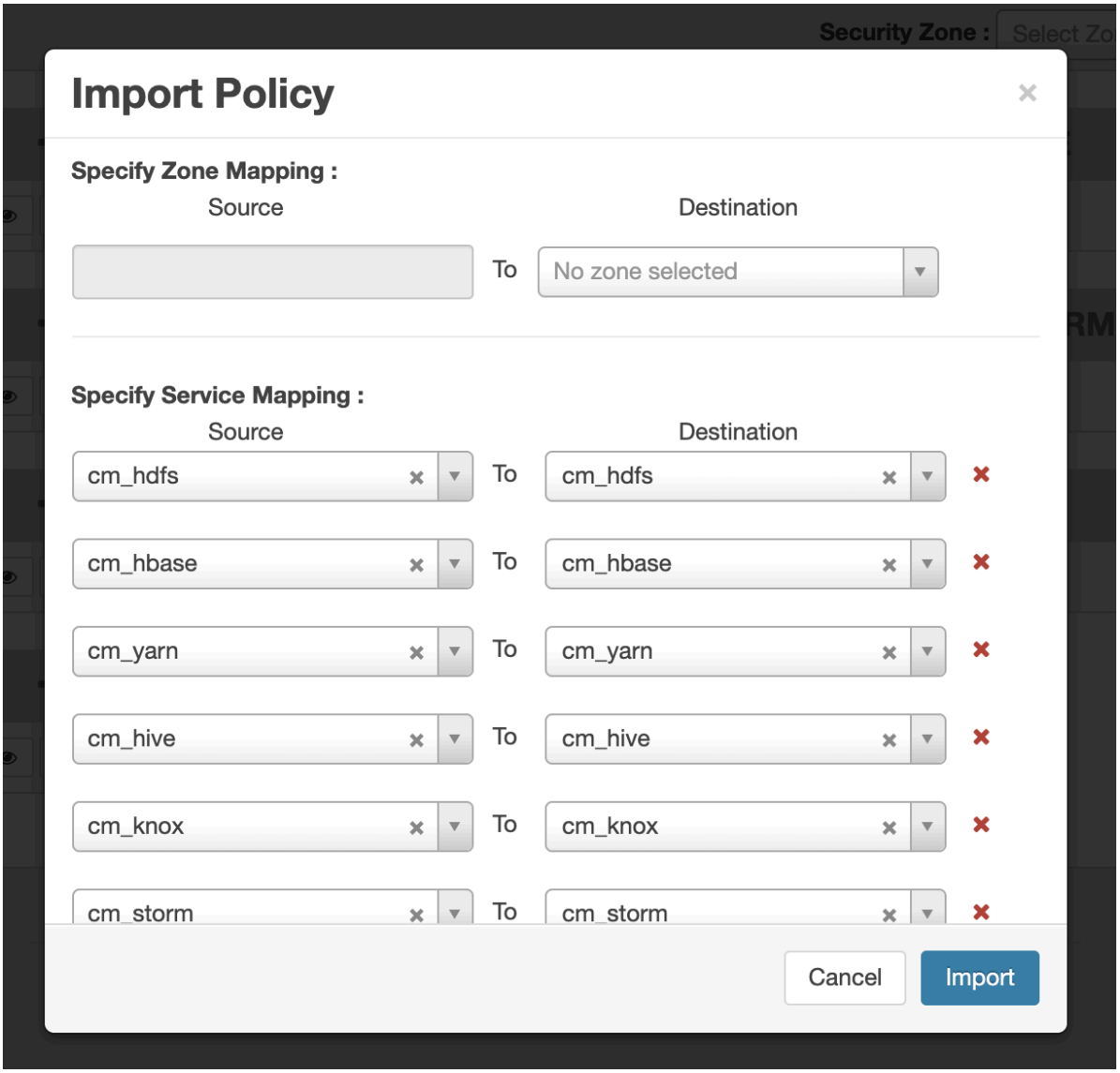
1. On the Service Manager page, click the Import icon for the service:

The Import Policy page appears.

2. Select the file to import.
   You can only import policies in JSON format.



3. (Optional) Configure the import operation:
   a) The Override Policy option deletes all policies of the destination repositories.
   b) Zone Mapping – when no destination is selected, all services are imported. When a destination is selected, only the services associated with that security zone are imported.
   c) Service Mapping maps the downloaded file repository, i.e. source repository to destination repository. You can use the red x symbols to remove services from the import. Scroll down to view all service mappings.



4. Click Import.
   A confirmation message appears after the file is imported.

Related Information

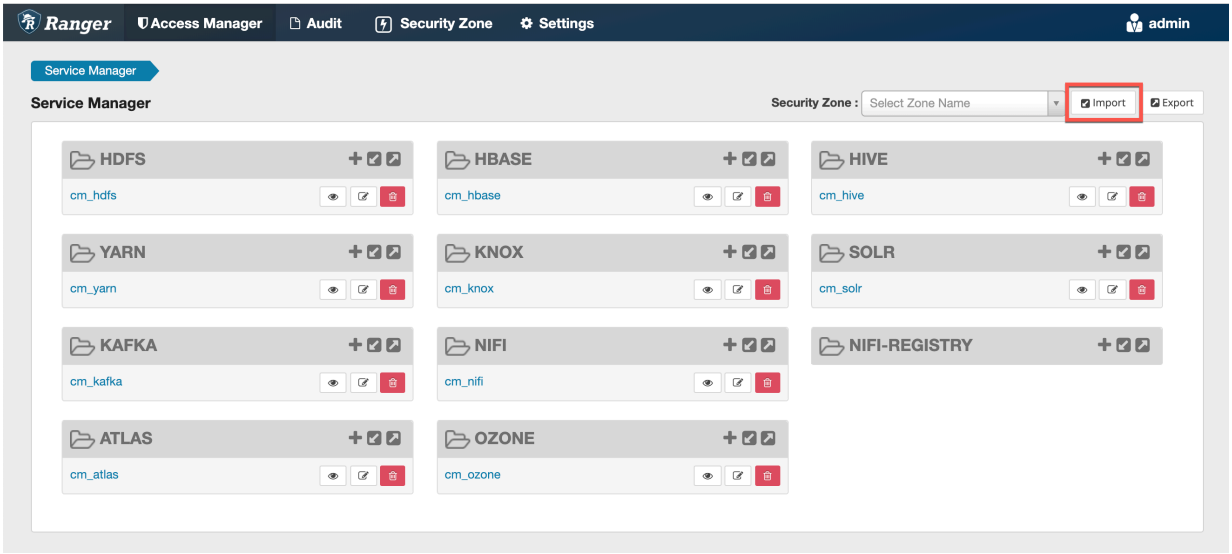
Import resource-based policies for all services

Import resource-based policies for all services

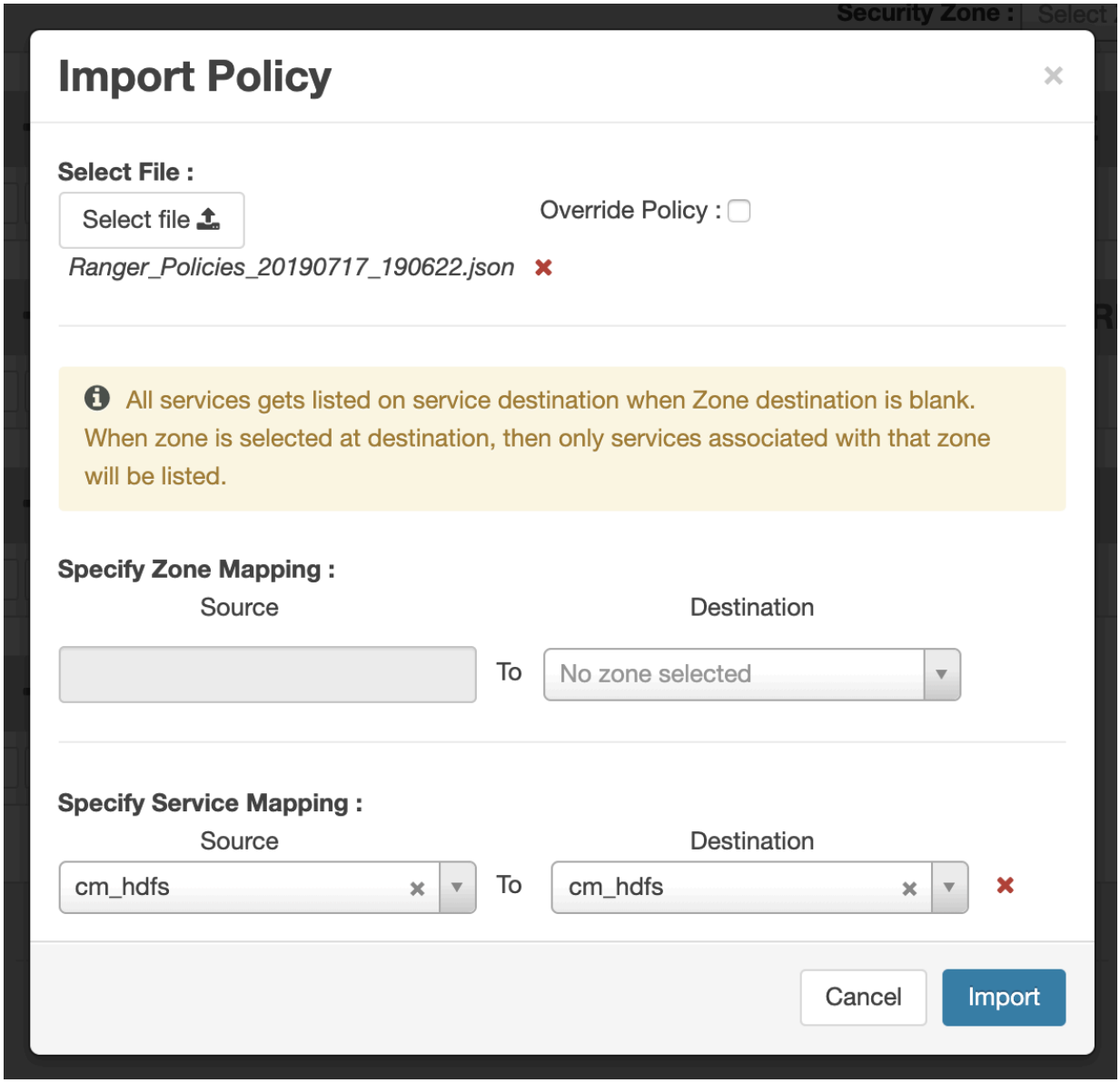
How to import policies for all services.

Procedure

1. On the Service Manager page, click Import.

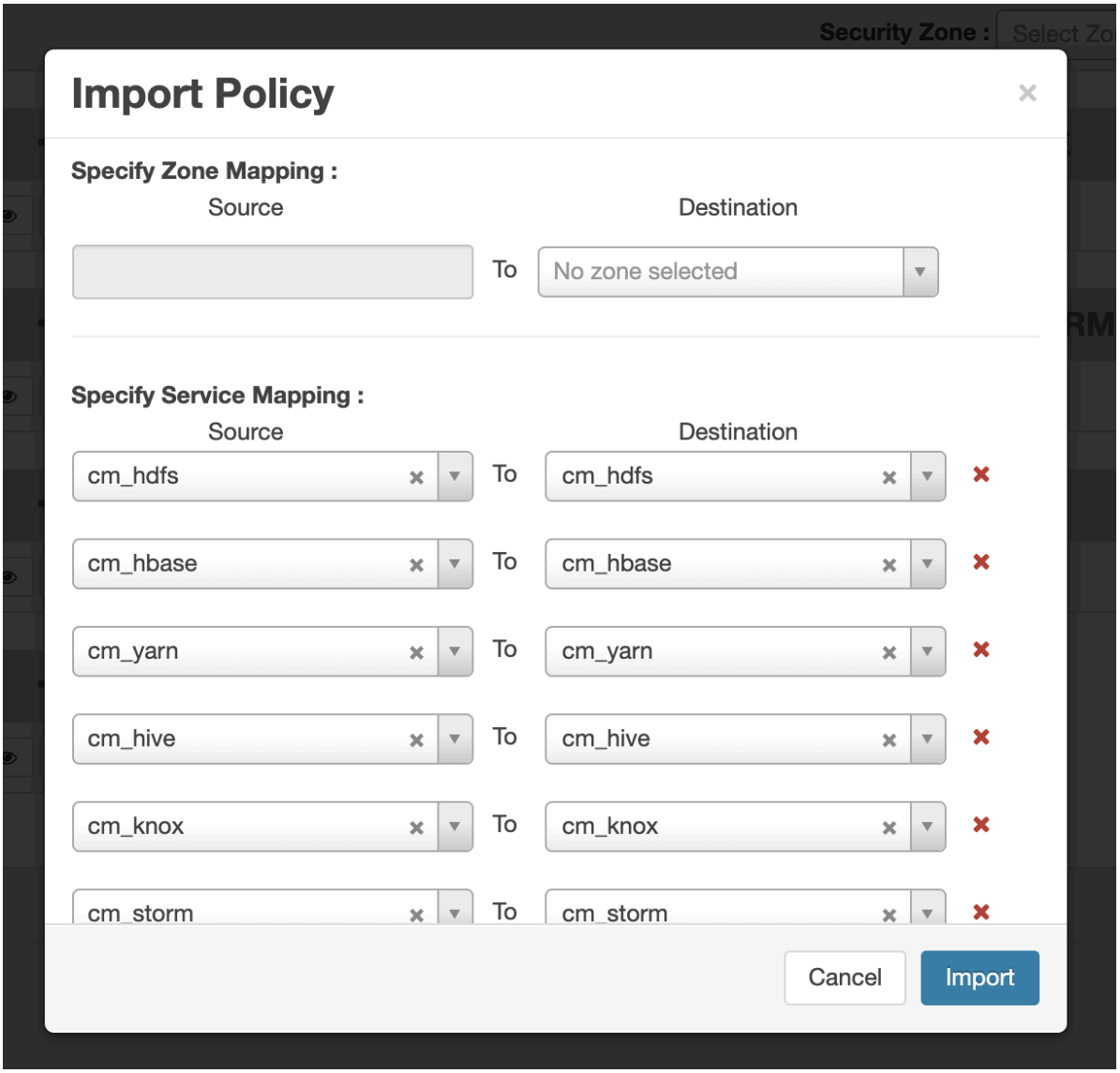


The Import Policy page appears.



2. Select the file to import.
   You can only import policies in JSON format.

3. (Optional) Configure the import operation:
   a) The Override Policy option deletes all policies of the destination repositories.
   b) Zone Mapping – when no destination is selected, all services are imported. When a destination is selected, only the services associated with that security zone are imported.
   c) Service Mapping maps the downloaded file repository, i.e. source repository to destination repository. You can use the red x symbols to remove services from the import. Scroll down to view all service mappings.



4. Click Import.
   A confirmation message appears after the file is imported.

Related Information

Import resource-based policies for a specific service

Export resource-based policies for a specific service

How to export the policies for a specific service (HBase, YARN, etc).

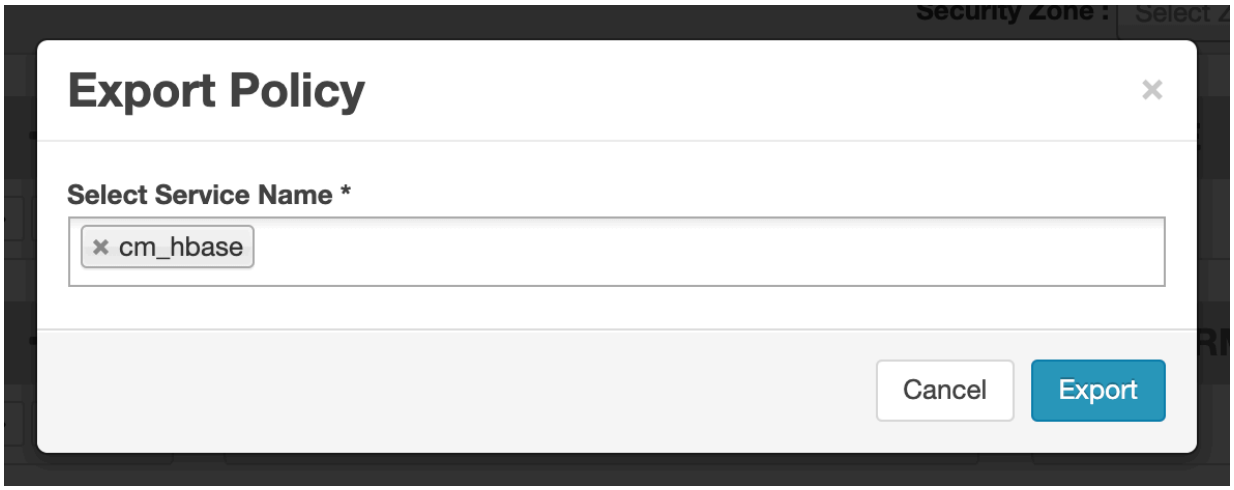
About this task

If you would like to export in Excel or CSV format, export the policies from the Reports page dropdown menu.

Procedure

1. On the Service Manager page, click the Export icon for the service:

   The Export Policy page appears.
2. Click Export.



   The file downloads in your browser as a JSON file.

Related Information

Export all resource-based policies for all services

Export all resource-based policies for all services

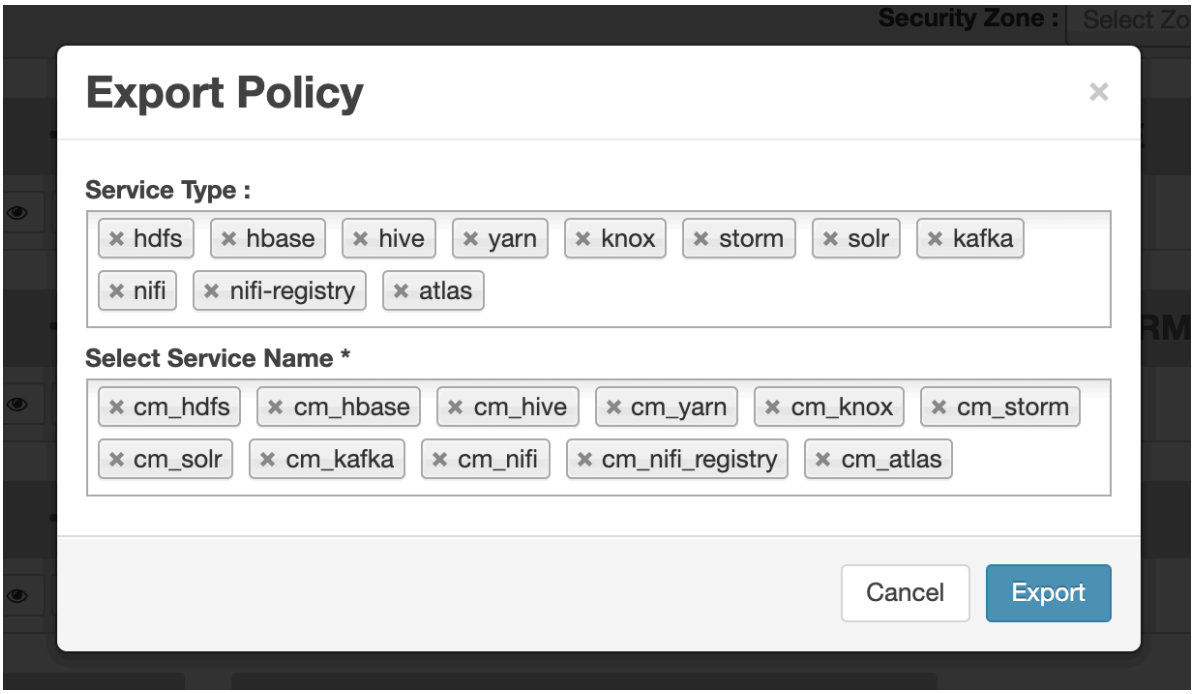
How to export the policies for all service.

About this task

If you would like to export in Excel or CSV format, export the policies from the Reports page drop-down menu.

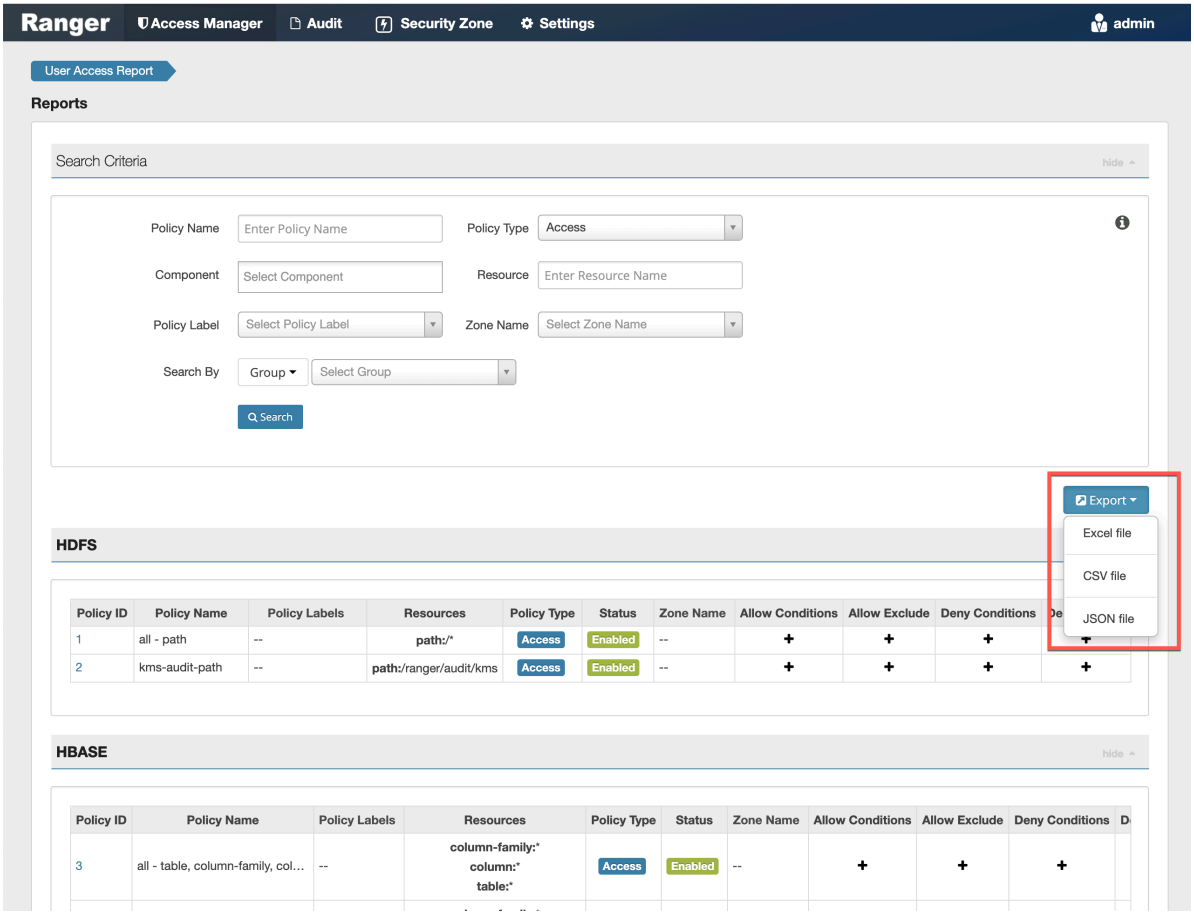
Procedure

- From the Service Manager page:
  a) Click Export.
  The Export Policy page appears.
  b) Remove components or specific services, then click Export.



The file downloads in your browser as a JSON file.

- From the Reports page:
  a) Apply filters before exporting the file.
  b) Open the Export drop-down menu:



  c) Select the file format.
  The file downloads in your browser.

Related Information

Export resource-based policies for a specific service

Row-level filtering and column masking in Hive

You can use Apache Ranger row-level filters to set access policies for rows in Hive tables. You can also use Ranger column masking to set policies that mask data in Hive columns, for example to show only the first or last four characters of column data.

Related Information

Compaction prerequisites

Row-level filtering in Hive with Ranger policies

Row-level filtering helps simplify Hive queries. By moving the access restriction logic down into the Hive layer, Hive applies the access restrictions every time data access is attempted. This helps simplify authoring of the Hive query, and provides seamless behind-the-scenes enforcement of row-level segmentation without having to add this logic to the predicate of the query.

About this task

Row-level filtering also improves the reliability and robustness of Hadoop. By providing row-level security to Hive tables and reducing the security surface area, Hive data access can be restricted to specific rows based on user characteristics (such as group membership) and the runtime context in which this request is issued.

Typical use cases where row-level filtering can be beneficial include:

- A hospital can create a security policy that allows doctors to view data rows only for their own patients, and that allows insurance claims administrators to view only specific rows for their specific site.
- A bank can create a policy to restrict access to rows of financial data based on the employee's business division, locale, or based on the employee's role (for example: only employees in the finance department are allowed to see customer invoices, payments, and accrual data; only European HR employees can see European employee data).
- A multi-tenant application can create logical separation of each tenant's data so that each tenant can see only their own data rows.

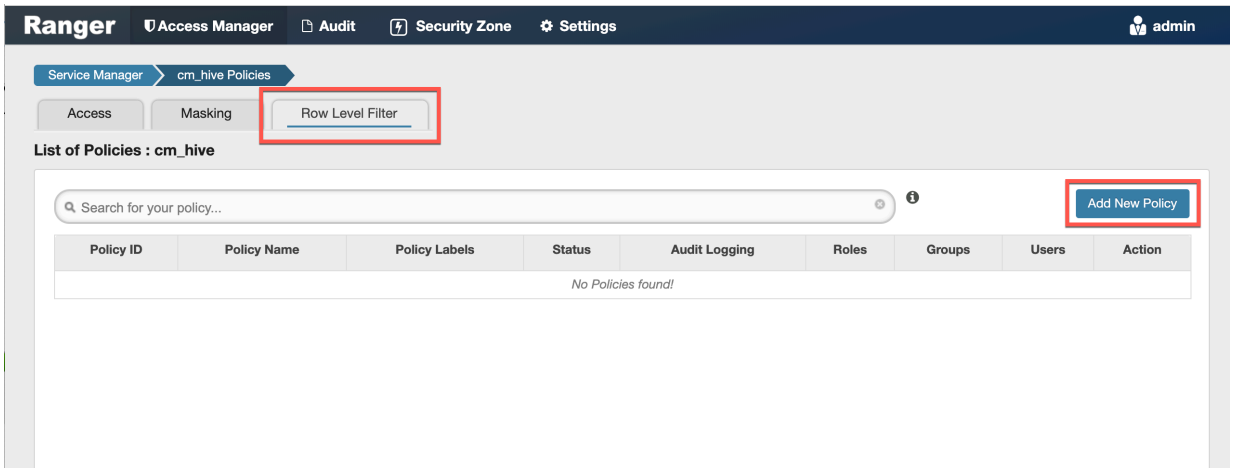
You can use Apache Ranger row-level filters to set access policies for rows in Hive tables. Row-level filter policies are similar to other Ranger access policies. You can set filters for specific users, groups, and conditions.

The following conditions apply when using row-level filters:

- The filter expression must be a valid WHERE clause for the table or view.
- Each table or view should have its own row-level filter policy.
- Wildcard matching is not supported on database or table names.
- Filters are evaluated in the order listed in the policy.
- An audit log entry is generated each time a row-level filter is applied to a table or view.

Procedure

1. On the Service Manager page, select an existing Hive Service.
2. Select the Row Level Filter tab, then click Add New Policy.



3. On the Create Policy page, add the following information for the row-level filter:

Table 48: Policy Details

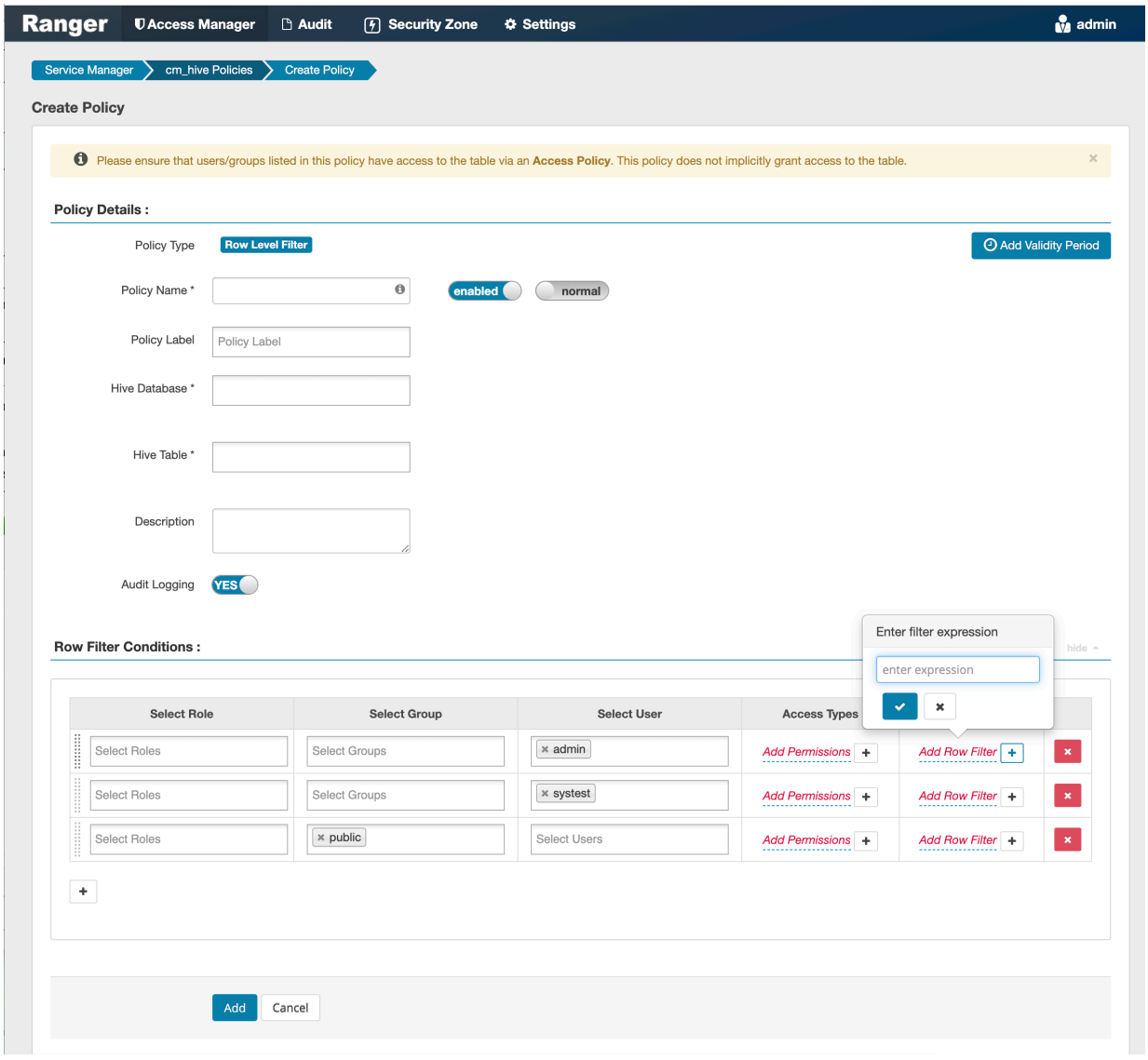
Field	Description
Policy Name (required)	Enter an appropriate policy name. This name cannot be duplicated across the system. The policy is enabled by default.

Field	Description
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
Hive Database (required)	Type in the applicable database name. The auto-complete feature displays available databases based on the entered text.
Hive Table (required)	Type in the applicable table name. The auto-complete feature displays available tables based on the entered text.
Audit Logging	Audit Logging is set to Yes by default. Select No to turn off audit logging.
Description	Enter an optional description for the policy.
Add Validity Period	Specify a start and end time for the policy.

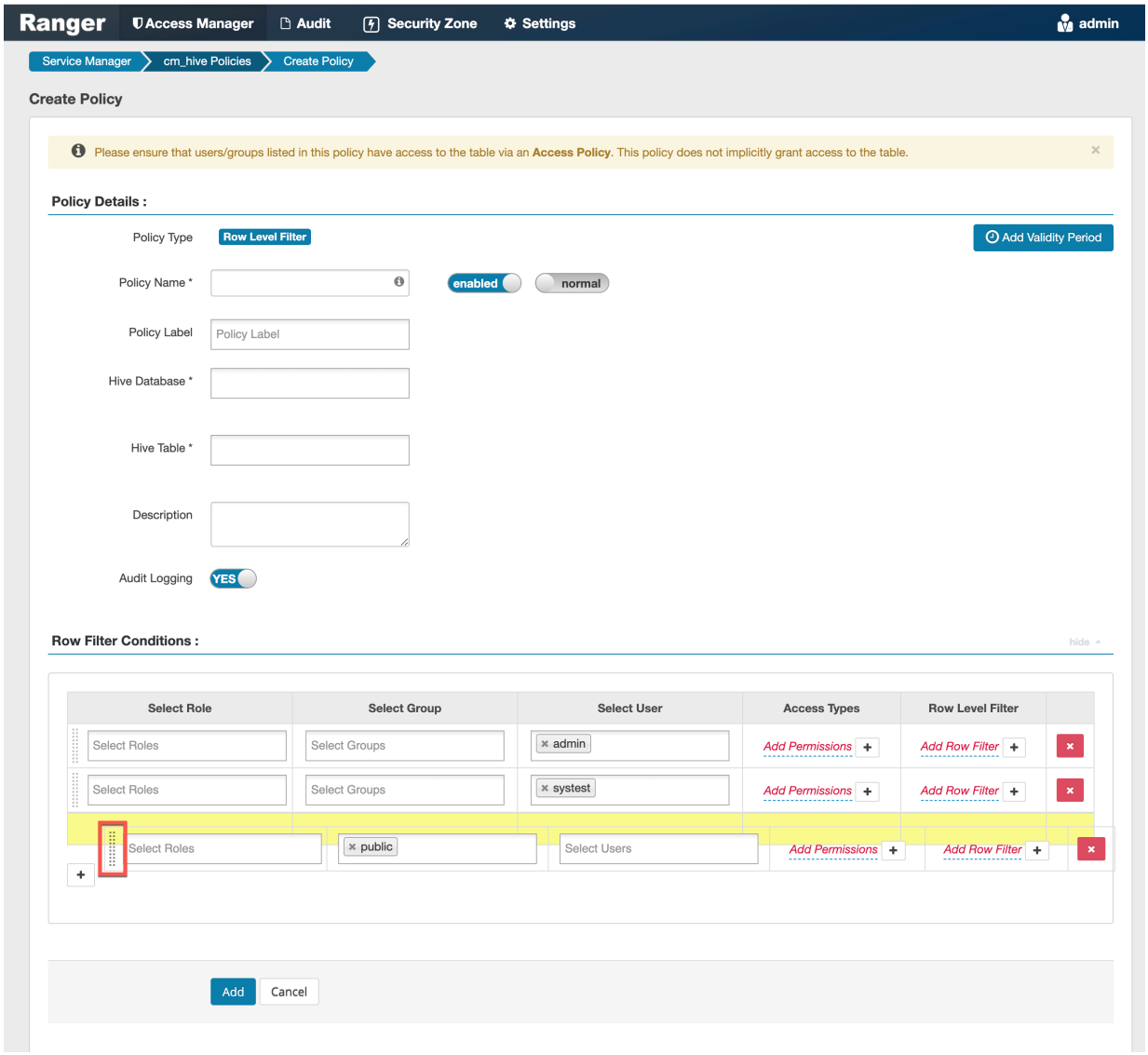
Table 49: Row Filter Conditions

Label	Description
Select Role	Specify the roles to which this policy applies.
Select Group	Specify the groups to which this policy applies. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify one or more users to which this policy applies.
Access Types	Currently select is the only available access type. This will be used in conjunction with the WHERE clause specified in the Row Level Filter field.

Label	Description
Add Row Filter	• To create a row filter for the specified users, groups, and roles, Click Add Row Filter, then type a valid WHERE clause in the Enter filter expression box. • To allow Select access for the specified users and groups without row-level restrictions, do not add a row filter (leave the setting as "Add Row Filter"). • 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.



4. To move a condition in the Row Filter Conditions list (and therefore change the order in which it is evaluated), click the dotted rows icon at the left of the condition row, then drag the condition to a new position in the list.



5. Click Add to add the new row-level filter policy.

Dynamic resource-based column masking in Hive with Ranger policies

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

About this task

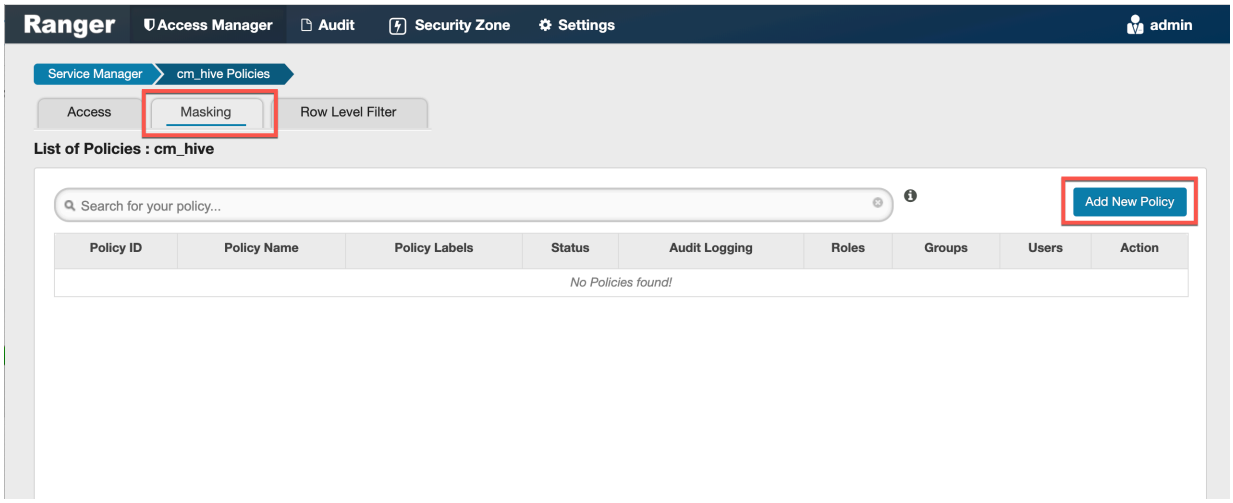
Dynamic column masking policies are similar to other Ranger access policies for Hive. You can set filters for specific users, groups, and conditions. With dynamic column-level masking, sensitive information never leaves Hive, and no changes are required at the consuming application or the Hive layer. There is also no need to produce additional protected duplicate versions of datasets.

The following conditions apply when using Ranger column masking policies to mask data returned in Hive query results:

- A variety of masking types are available, such as show last 4 characters, show first 4 characters, Hash, Nullify, and date masks (show only year).
- You can specify a masking type for specific users, groups, and conditions.
- Wildcard matching is not supported.
- Each column should have its own masking policy.
- Masks are evaluated in the order listed in the policy.
- An audit log entry is generated each time a masking policy is applied to a column.

Procedure

1. On the Service Manager page, select an existing Hive Service.
2. Select the Masking tab, then click Add New Policy.



3. On the Create Policy page, add the following information for the column-masking filter:

Table 50: Policy Details

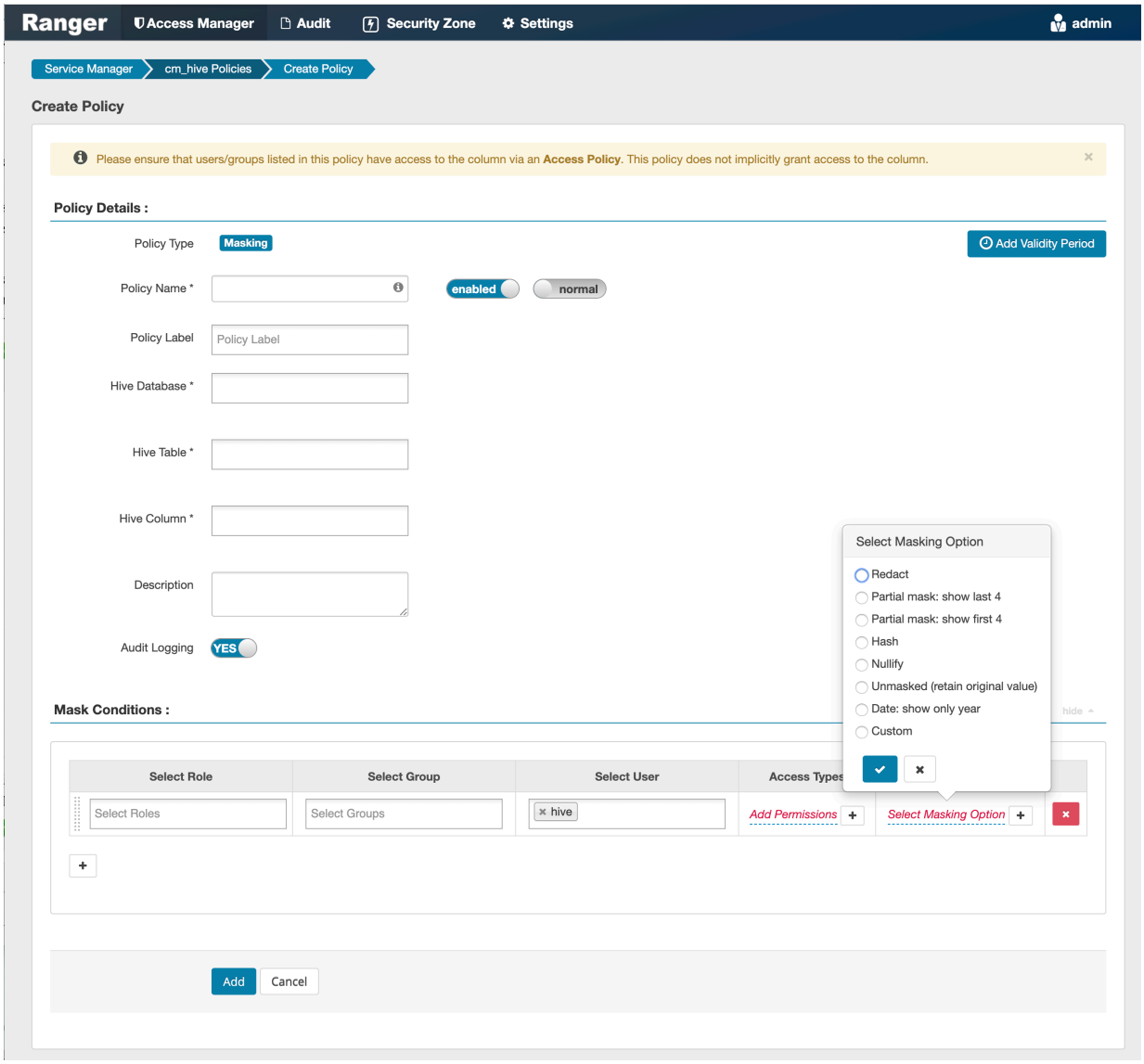
Field	Description
Policy Name (required)	Enter an appropriate policy name. This name cannot be duplicated across the system. The policy is enabled by default.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
Hive Database (required)	Type in the applicable database name. The auto-complete feature displays available databases based on the entered text.
Hive Table (required)	Type in the applicable table name. The auto-complete feature displays available tables based on the entered text.
Hive Column (required)	Type in the applicable column name. The auto-complete feature displays available columns based on the entered text.
Audit Logging	Audit Logging is set to Yes by default. Select No to turn off audit logging.
Description	Enter an optional description for the policy.

Field	Description
Add Validity Period	Specify a start and end time for the policy.

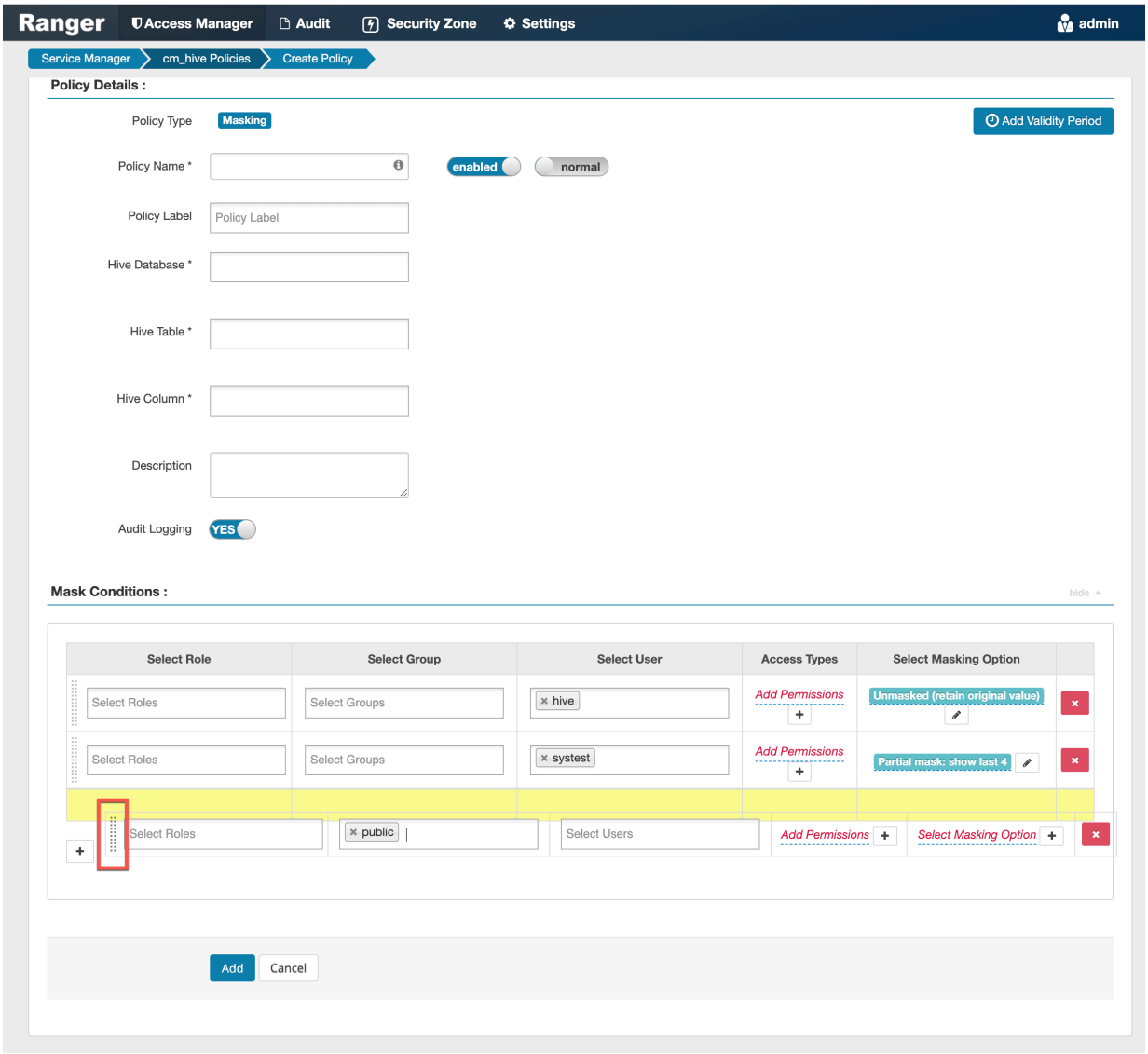
Table 51: Mask Conditions

Label	Description
Select Role	Specify the roles to which this policy applies.
Select Group	Specify the groups to which this policy applies. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify one or more users to which this policy applies.
Access Types	Currently select is the only available access type.

Label	Description
Select Masking Type	To create a row filter for the specified users, groups, and roles, click Select Masking Option, then select a masking type: • Redact – mask all alphabetic characters with "x" and all numeric characters with "n". • Partial mask: show last 4 – Show only the last four characters. • Partial mask: show first 4 – Show only the first four characters. • Hash – Replace all characters with a hash of entire cell value. • Nullify – Replace all characters with a NULL value. • Unmasked (retain original value) – No masking is applied. • Date: show only year – Show only the year portion of a date string and default the month and day to 01/01 • Custom – Specify a custom masked value or expression. Custom masking can use any valid Hive UDF (Hive that returns the same data type as the data type in the column being masked). Masking conditions are evaluated in the order listed in the policy. The condition at the top of the Masking Conditions list is applied first, then the second, then the third, and so on.



4. To move a condition in the Mask Conditions list (and therefore change the order in which it is evaluated), click the dotted rows icon at the left of the condition row, then drag the condition to a new position in the list.



5. Click Add to add the new column masking filter policy.

Dynamic tag-based column masking in Hive with Ranger policies

Where Ranger resource-based masking policy for Hive anonymizes data from a Hive column identified by the database, table, and column, Ranger tag-based masking policy anonymizes Hive column data based on tags and tag attribute values associated with Hive column (usually specified as metadata classification in Atlas).

About this task

The following conditions apply when using Ranger column masking policies to mask data returned in Hive query results:

- A variety of masking types are available, such as show last 4 characters, show first 4 characters, Hash, Nullify, and date masks (show only year).

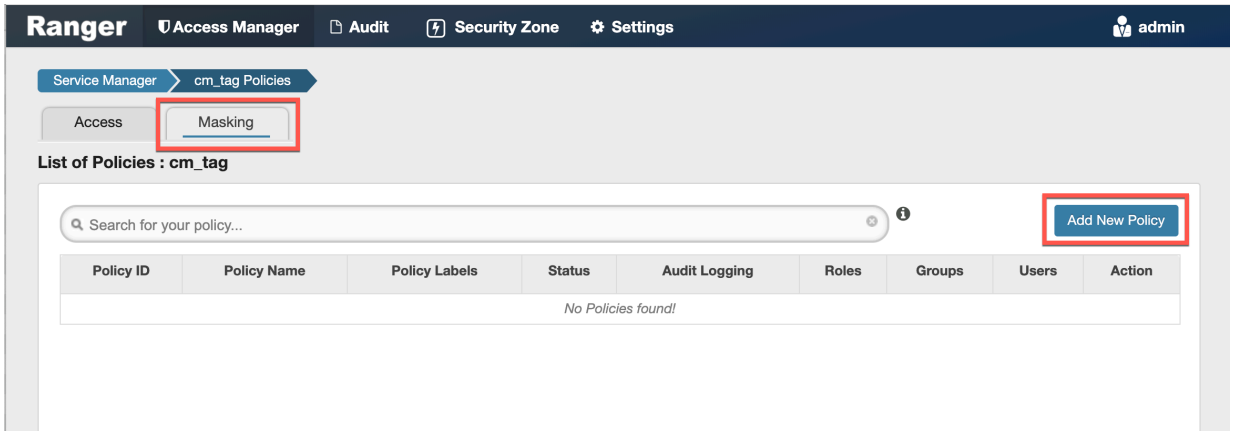
Note: Ranger depends on Hive/Impala's hashing functions for hash masking.

- Impala uses sha512 in FIPS mode, sha256 in non-FIPS mode.
- Hive uses sha256. Plans to update to sha512 in FIPS mode.

- You can specify a masking type for specific users, groups, and conditions.
- Wildcard matching is not supported.
- If there are multiple tag masking policies applied to the same Hive column, the masking policy with the lexicographically smallest policy-name is chosen for enforcement, E.G., policy "a" is enforced before policy "aa".
- Masks are evaluated in the order listed in the policy.
- An audit log entry is generated each time a masking policy is applied to a column.

Procedure

1. Select Access Manager Tag Based Policies , then select a tag-based service.
2. Select the Masking tab, then click Add New Policy.



3. In Create Policy, add the following information for the column-masking filter:

Table 52: Policy Details

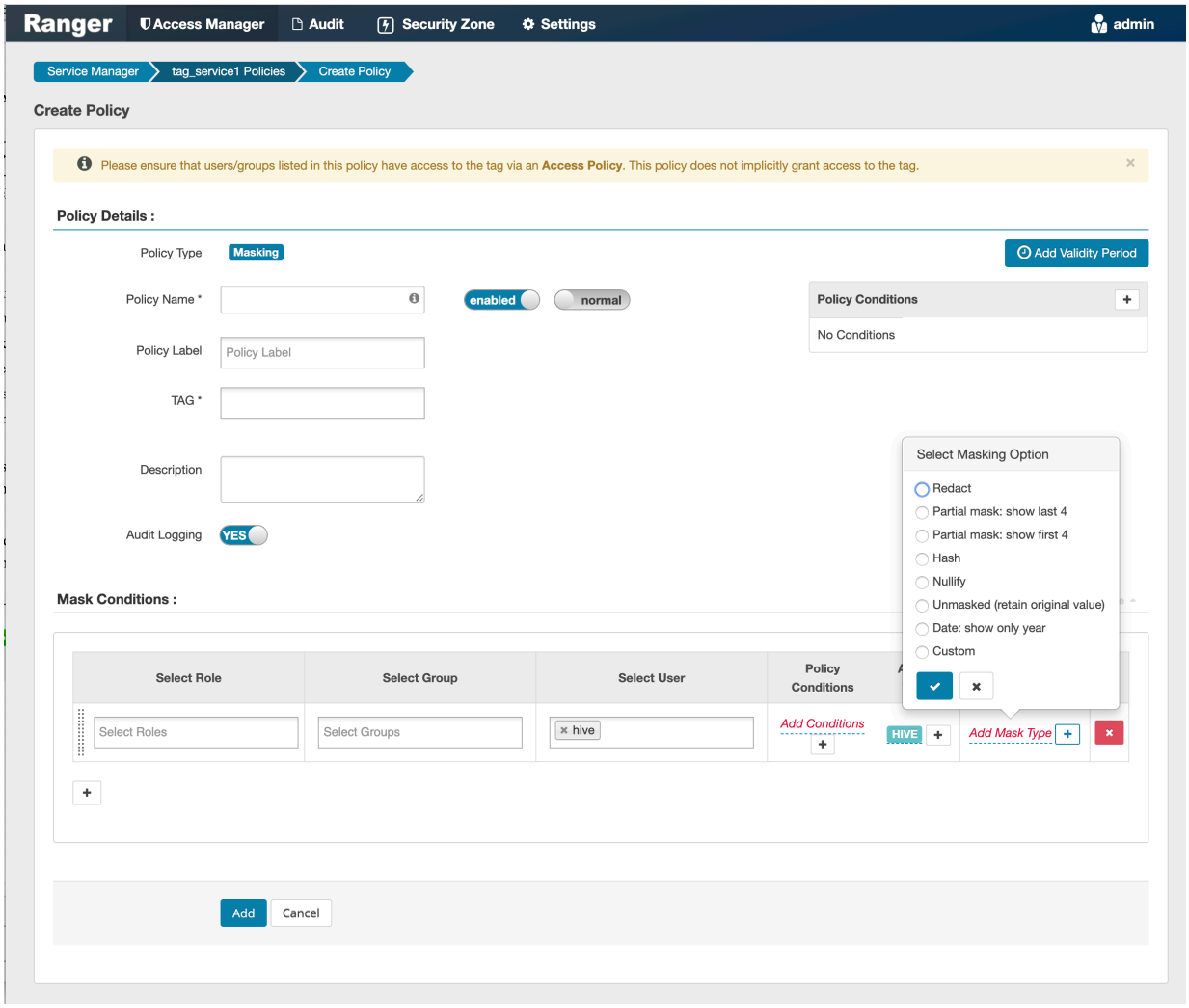
Field	Description
Policy Type (required)	Set to Masking by default.
Policy Name (required)	Enter an appropriate policy name. This name cannot be duplicated across the system. The policy is enabled by default.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
TAG (required)	Enter the applicable tag name, for example MASK.
Audit Logging	Audit Logging is set to Yes by default. Select No to turn off audit logging.
Description	Enter an optional description for the policy.
Add Validity Period	Specify a start and end time for the policy.

Field	Description
Policy Conditions (applied at the policy level)	Click the + icon to add policy conditions. Currently "Accessed after expiry_date? (yes/no)" is the only available policy condition. "Accessed after expiry_date (yes/no)?": To set this condition, type yes in the text box, then click check mark to add the condition. Enter boolean expression: Available for allow or deny conditions on tag-based policies. For examples and details, see “Using Tag Attributes and Values in Ranger Tag-Based Policy Conditions”. Click Save to save the policy condition.

Table 53: Mask Conditions

Label	Description
Select Role	Specify the roles to which this policy applies.
Select Group	Specify the groups to which this policy applies. The public group contains all users, so granting access to the public group grants access to all users.
Select User	Specify one or more users to which this policy applies.
Policy Conditions (applied at the item level)	Click Add Conditions to add policy conditions. Currently "Accessed after expiry_date? (yes/no)" is the only available policy condition. "Accessed after expiry_date (yes/no)?": To set this condition, type yes in the text box, then click check mark to add the condition. Enter boolean expression: Available for allow or deny conditions on tag-based policies. For examples and details, see “Using Tag Attributes and Values in Ranger Tag-Based Policy Conditions”.
Access Types	Currently select is the only available access type for the hive component.

Label	Description
Select Masking Option	To create a row filter for the specified users, groups, and roles, click Select Masking Option, then select a masking type: • Redact – mask all alphabetic characters with "x" and all numeric characters with "n". • Partial mask: show last 4 – Show only the last four characters. • Partial mask: show first 4 – Show only the first four characters. • Hash – Replace all characters with a hash of entire cell value. • Nullify – Replace all characters with a NULL value. • Unmasked (retain original value) – No masking is applied. • Date: show only year – Show only the year portion of a date string and default the month and day to 01/01 • Custom – Specify a custom masked value or expression. Custom masking can use any valid Hive UDF (Hive that returns the same data type as the data type in the column being masked). Masking conditions are evaluated in the order listed in the policy. The condition at the top of the Masking Conditions list is applied first, then the second, then the third, and so on.



4. Click + to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. Click Add to add the new policy.

Related Information

Using tag attributes and values in Ranger tag-based policy conditions

Tag-based Services and Policies

Ranger enables you to create tag-based services and add access policies to those services.

Adding a tag-based service

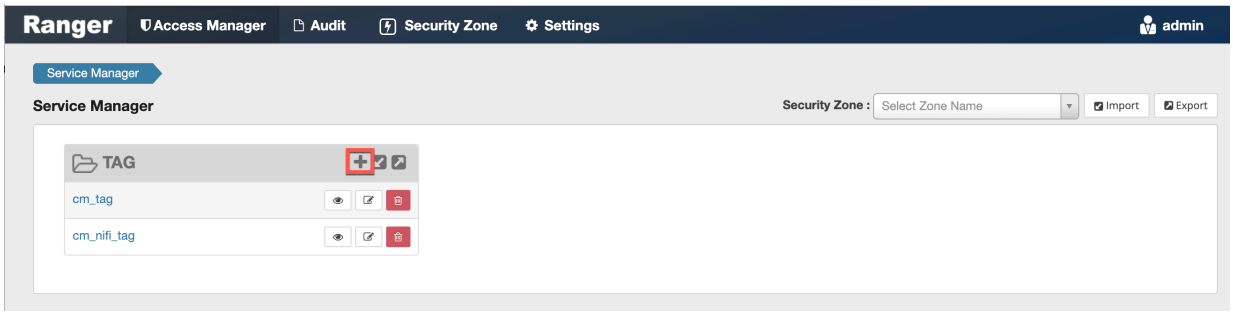
How to add a tag-based service to Ranger.

About this task

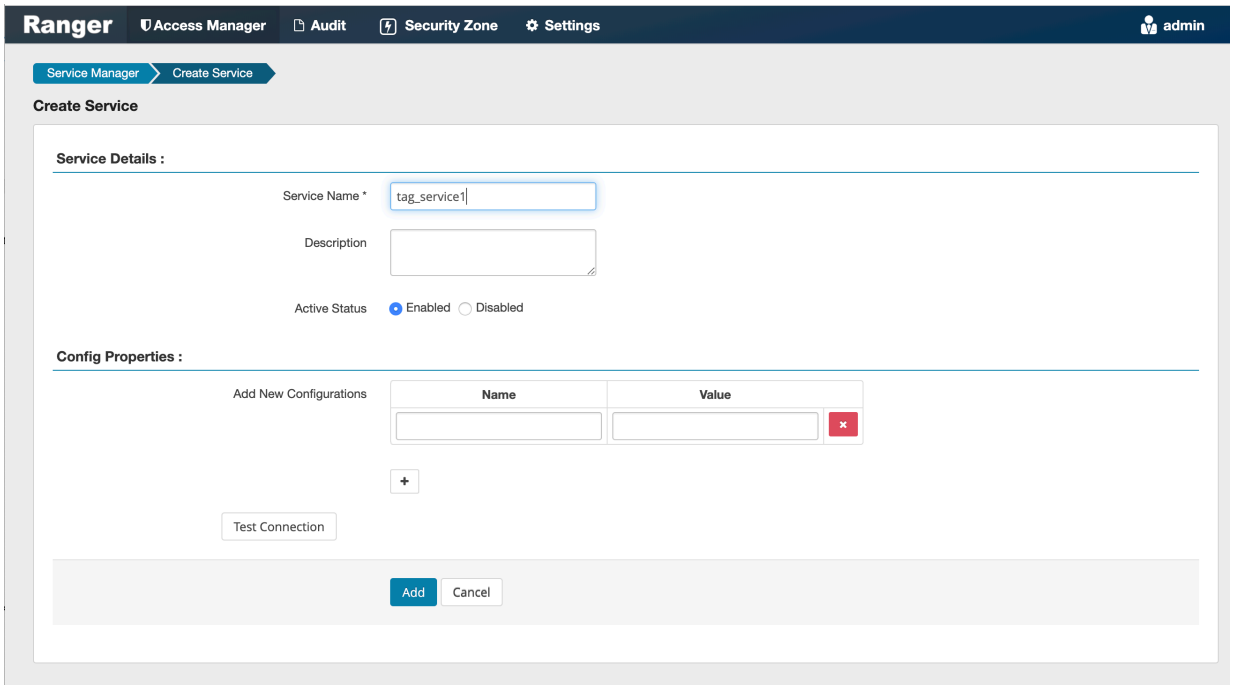
You can use the Service Manager for Tag-Based Policies page to create tag-based services and add tag-based access policies that can be applied to Hadoop resources. Using tag-based policies enables you to control access to resources across multiple Hadoop components without creating separate services and policies in each component. You can also use Ranger TagSync to synchronize the Ranger tag store with an external metadata service such as Apache Atlas.

Procedure

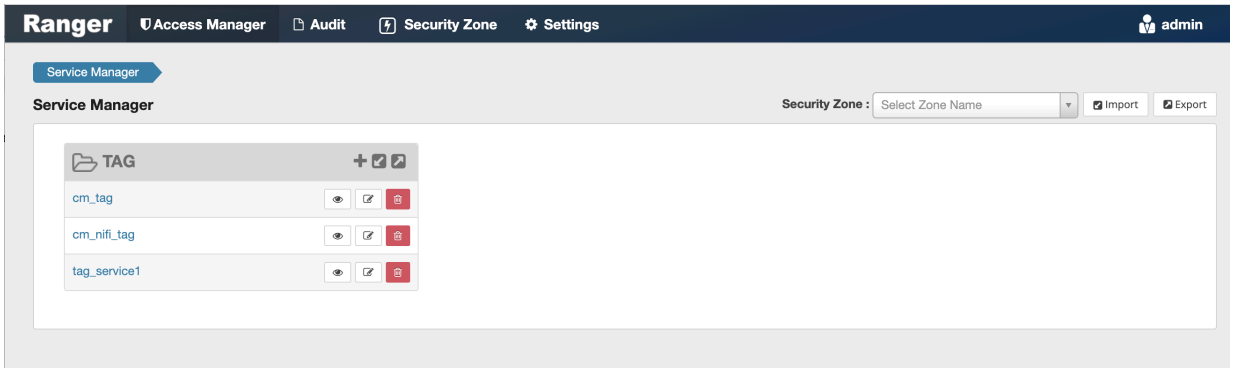
1. Select Access Manager > Tag Based Policies, then click the Add icon ( ) in the TAG box on the Service Manager page.



2. On the Create Service page, type in a service name and an optional description. The service is enabled by default, but you can disable it by selecting Disabled. To add the service, click Add.



3. The new tag service appears on the Service Manager page.



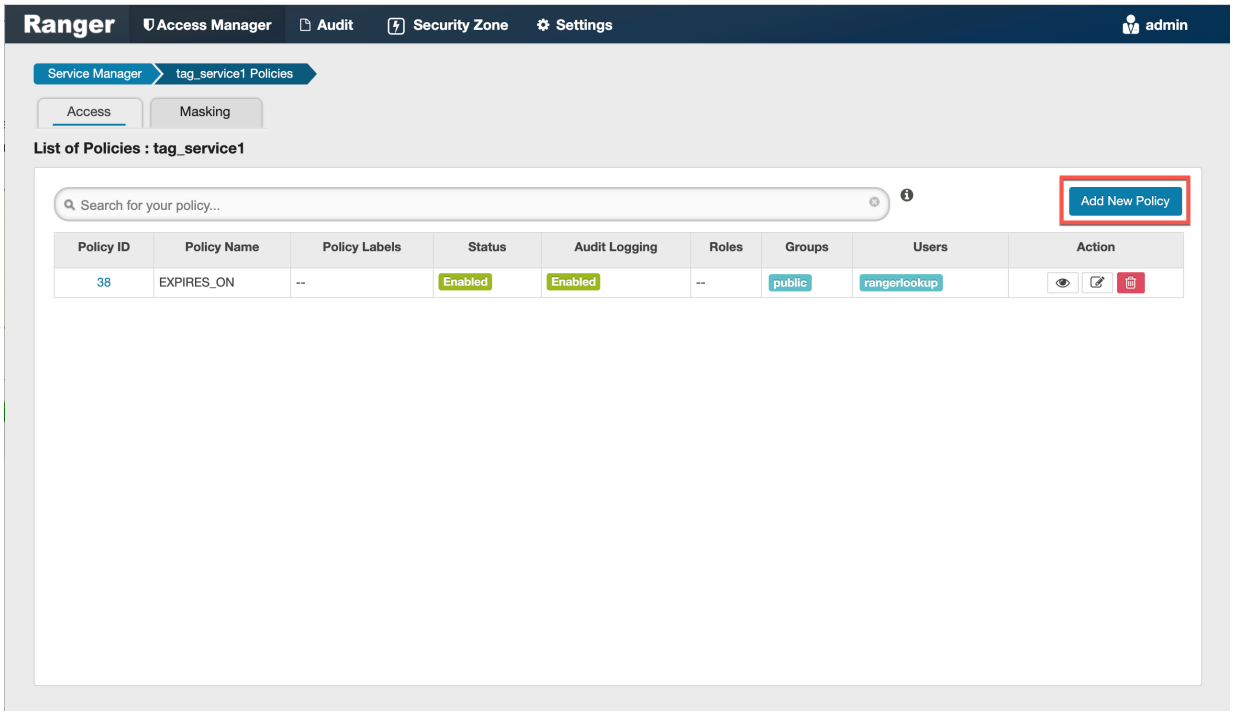
Adding tag-based policies

Tag-based policies enable you to control access to resources across multiple Hadoop components without creating separate services and policies in each component. You can also use Ranger TagSync to synchronize the Ranger tag store with an external metadata service such as Apache Atlas.

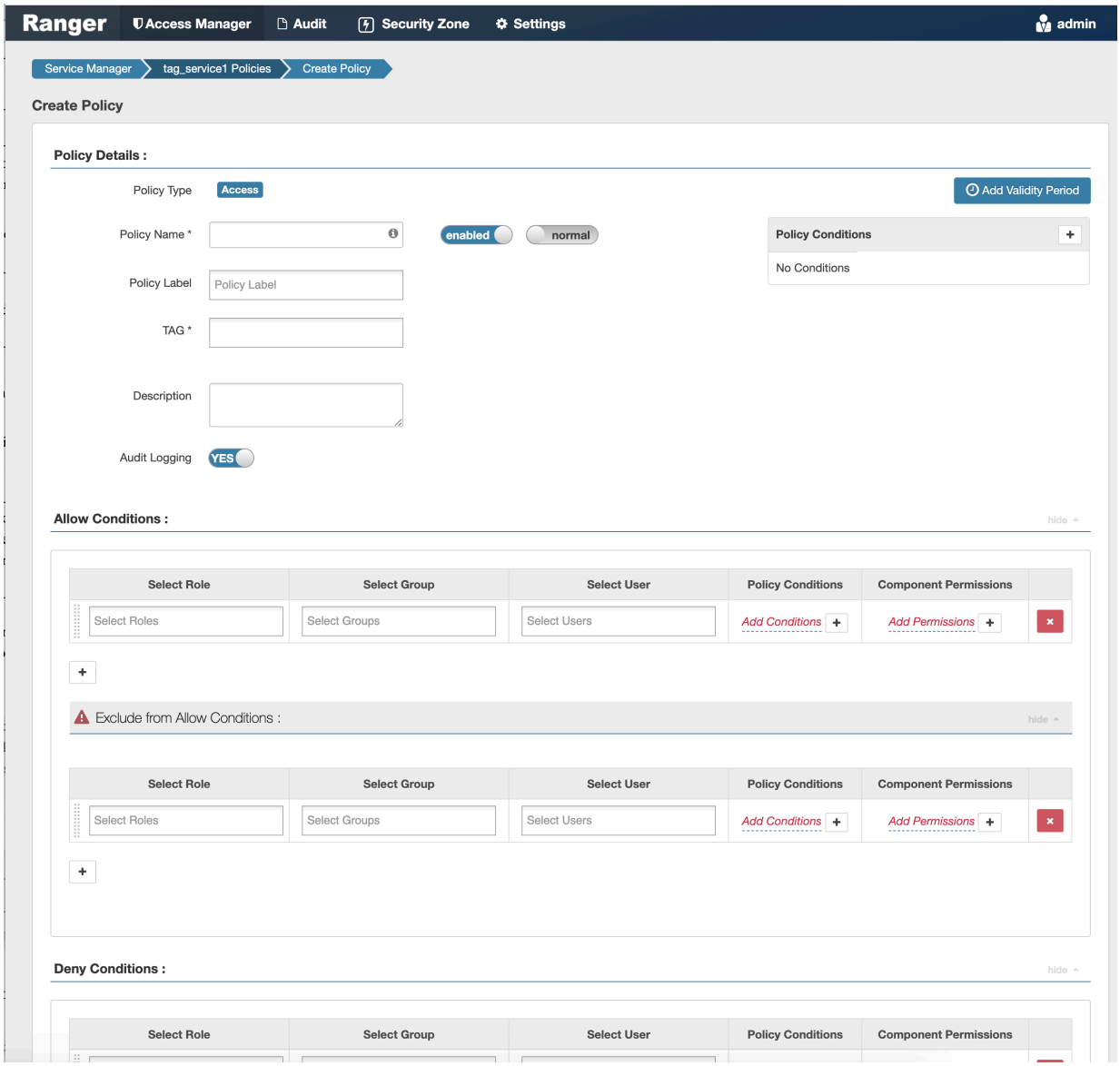
Procedure

1. Select Access Manager> Tag Based Policies, then select a tag-based service.

2. Thee List of Policies page appears with the Access tab selected by default. Click Add New Policy.



The Create Policy page appears:



3. Enter information on the Create Policy page as follows:

Table 54: Policy Details

Field	Description
Policy Type	Set to Access by default.
Policy Name	Enter a unique policy name. This name cannot be duplicated across the system. This field is mandatory.
normal/override	Enables you to specify an override policy. When override is selected, the access permissions in the policy override the access permissions in existing policies. This feature can be used with Add Validity Period to create temporary access policies that override existing policies.
TAG	Enter the applicable tag name.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).

Field	Description
Policy Label	Specify a label for this policy. You can search reports and filter policies based on these labels.
Add Validity Period	Specify a start and end time for the policy.
Policy Conditions (applied at the policy level)	Click the + icon to add policy conditions. Currently "Accessed after expiry_date? (yes/no)" is the only available policy condition. "Accessed after expiry_date (yes/no)?": To set this condition, type yes in the text box, then click the check mark button to add the condition. Enter boolean expression: Available for allow or deny conditions on tag-based policies. For examples and details, see “Using Tag Attributes and Values in Ranger Tag-Based Policy Conditions”. Click Save to save the policy condition.

Table 55: Allow, Exclude from Allow, Deny, and Exclude from Deny Conditions

Label	Description
Select Role	Specify the roles to which this policy applies.
Select Group	Specify the group to which this policy applies. To designate the group as an Administrator for the chosen resource, specify Admin permissions. (Administrators can create child policies based on existing policies). The public group contains all users, so setting a condition for the public group applies to all users.
Select User	Specify a particular user to which this policy applies (outside of an already-specified group) OR designate a particular user as Admin for this policy. (Administrators can create child policies based on existing policies).
Policy Conditions (applied at the item level)	Click Add Conditions to add policy conditions. Currently "Accessed after expiry_date? (yes/no)" is the only available policy condition. "Accessed after expiry_date (yes/no)?": To set this condition, type yes in the text box, then click the check mark button to add the condition. Enter boolean expression: Available for allow or deny conditions on tag-based policies. For examples and details, see “Using Tag Attributes and Values in Ranger Tag-Based Policy Conditions”.
Component Permissions	Click Add Permissions to add or edit component conditions. To add component permissions, enter the component name in the text box, then use the check boxes to specify component permissions. Click the check mark button to add the chosen component conditions to the policy.

4. You can use the Plus (+) symbols to add additional conditions. Conditions are evaluated in the order listed in the policy. The condition at the top of the list is applied first, then the second, then the third, and so on.
5. You can use Deny All Other Accesses to deny access to all other users, groups, and roles other than those specified in the allow conditions for the policy.
6. Click Add to add the new policy.

Related Information

Using tag attributes and values in Ranger tag-based policy conditions

Using tag attributes and values in Ranger tag-based policy conditions

Enter boolean expression allows Ranger to use tag attributes and values when configuring tag-based policy Allow or Deny conditions. It allows admins to provide boolean expression(s) using tag attributes.

The policy condition is introduced in the tag service definition:

```
{
      "itemId":2,
      "name":"expression",
      "evaluator": "org.apache.ranger.plugin.conditionevaluator.RangerScr
iptConditionEvaluator",
      "evaluatorOptions" : {"engineName":"JavaScript", "ui.isMultiline":"tru
e"},
      "label":"Enter boolean expression",
      "description": "Boolean expression"
    }
```

The following variables can be referenced in the boolean expression:

- ctx: Context handler containing APIs to access metadata information from the request.
- tag: Information about the current tag.
- tagAttr: Map containing all the current tag attributes and corresponding values.

The following APIs available from the request:

- getUser(): Returns a string.
- getUserGroups(): Returns a set of strings containing groups.
- getClientIPAddress(): Returns a string containing client IP address.
- getAction(): Returns a string containing information about the action being requested.

For two scenarios:

- User “sam” needs to be denied a policy based on the IP address of the machine from where the resources are accessed.

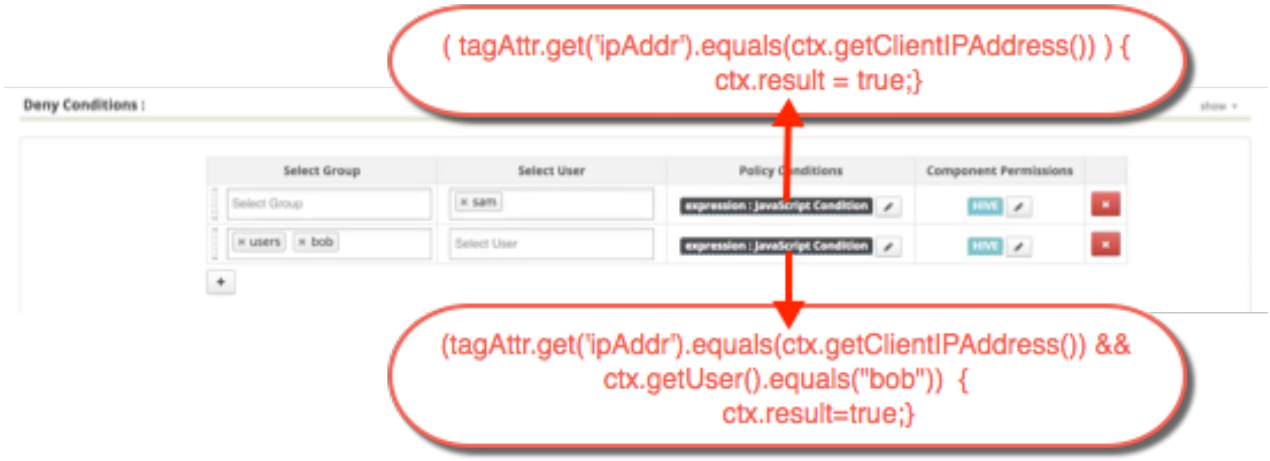
  Set the deny condition for user sam with the following boolean expression:

  ```
  if ( tagAttr.get('ipAddr').equals(ctx.getClientIPAddress()) ) {
   ctx.result = true;
     }
  ```

- Deny one particular user, “bob” from a group, “users”, only when this user is accessing resources from a particular IP defined as an tag attribute in Atlas.

  Set the deny condition for group users with the following boolean expression:

  ```
  if (tagAttr.get('ipAddr').equals(ctx.getClientIPAddress()) && ctx.getUse
  r().equals("bob"))  {
   ctx.result=true;
        }
  ```



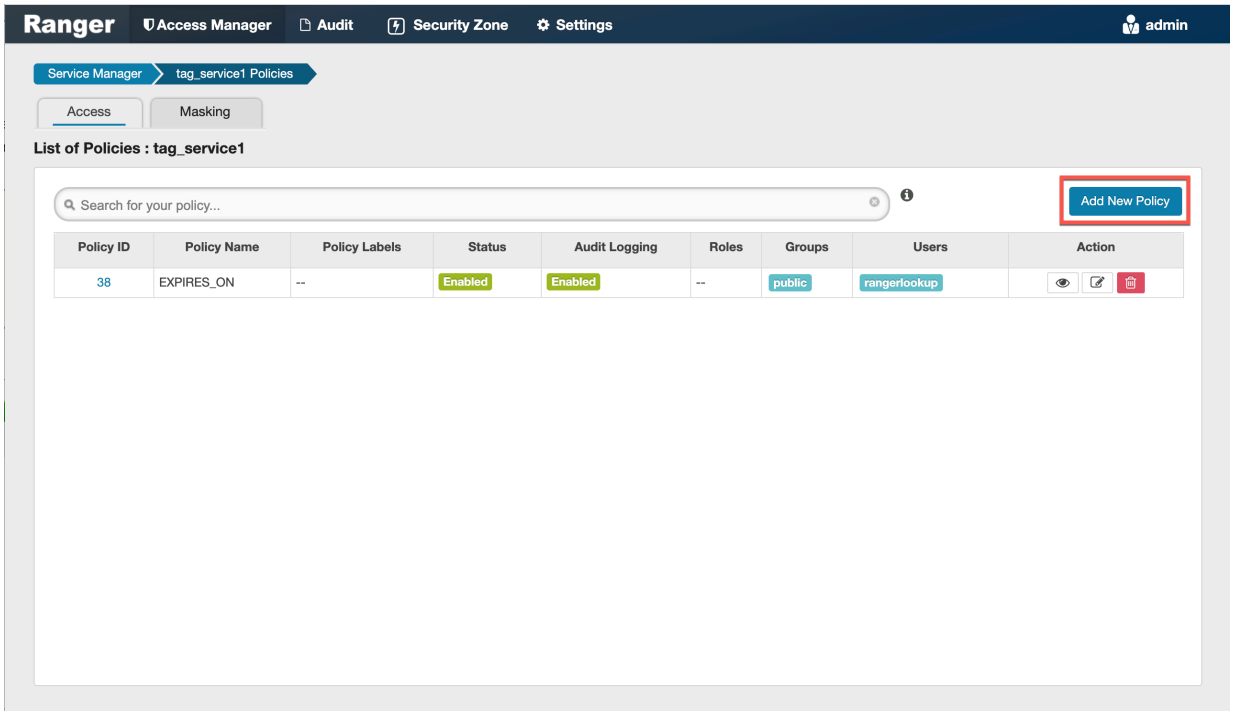
Adding a tag-based PII policy

Example of how to add a PII tag-based policy. In this example we create a tag-based policy for objects tagged "PII" in Atlas. Access to objects tagged "PII" is allowed for members of the "audit" group. All other users (the "public" group) are denied access.

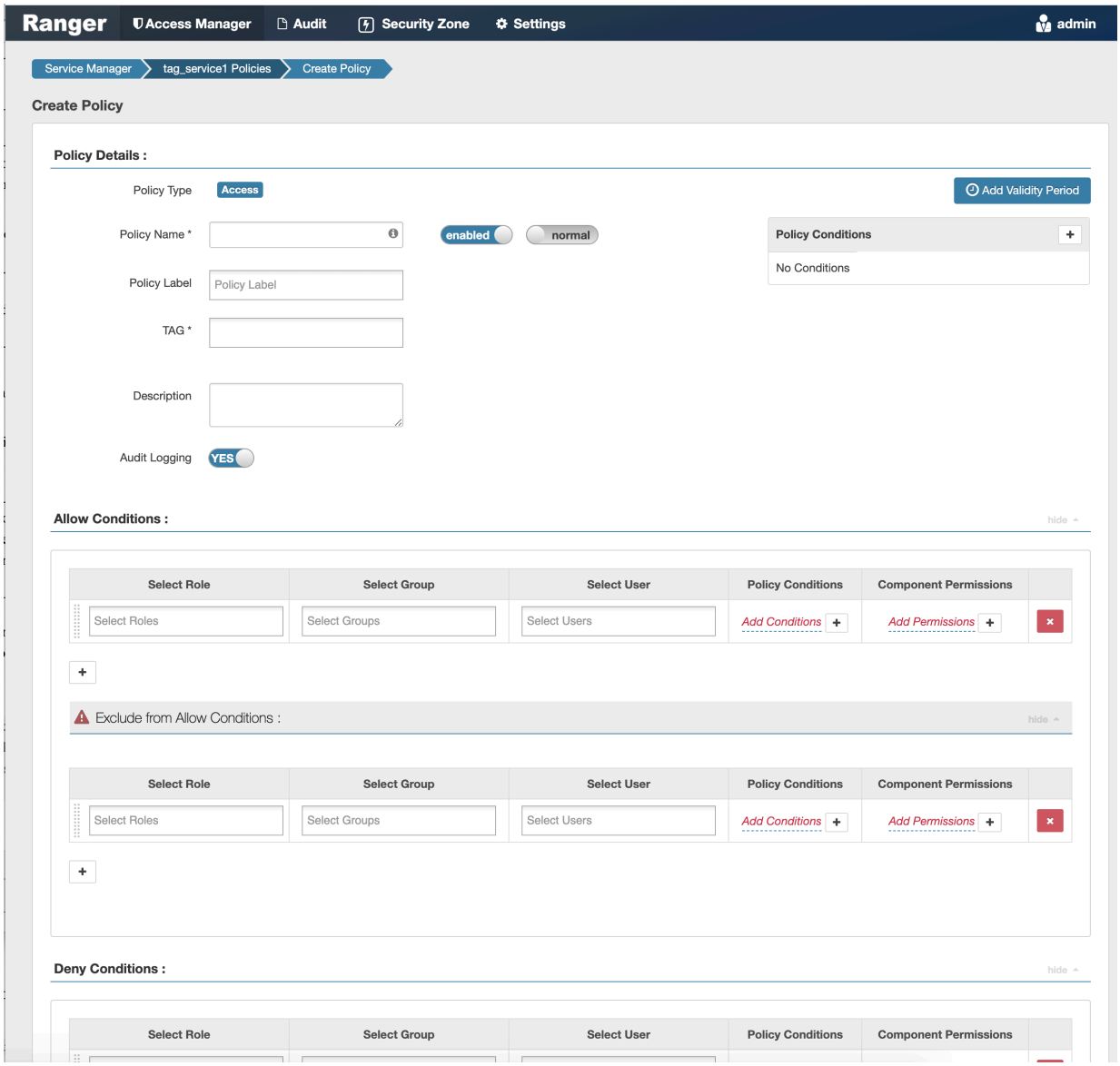
Procedure

1. Select Access Manager > Tag Based Policies, then select a tag-based service.

2. On the List of Policies page, click Add New Policy.



The Create Policy page appears:



3. Enter the following information on the Create Policy page:

Table 56: Policy Details

Field	Description
Policy Type	Set to Access by default.
Policy Name	PII
TAG	PII
Audit Logging	YES
Description	Restrict access to resources with the PII tag.

Table 57: Allow Conditions

Label	Description
Select Group	audit
Select User	<none>

Label	Description
Policy Conditions	<none>
Component Permissions	hive (select all permissions)

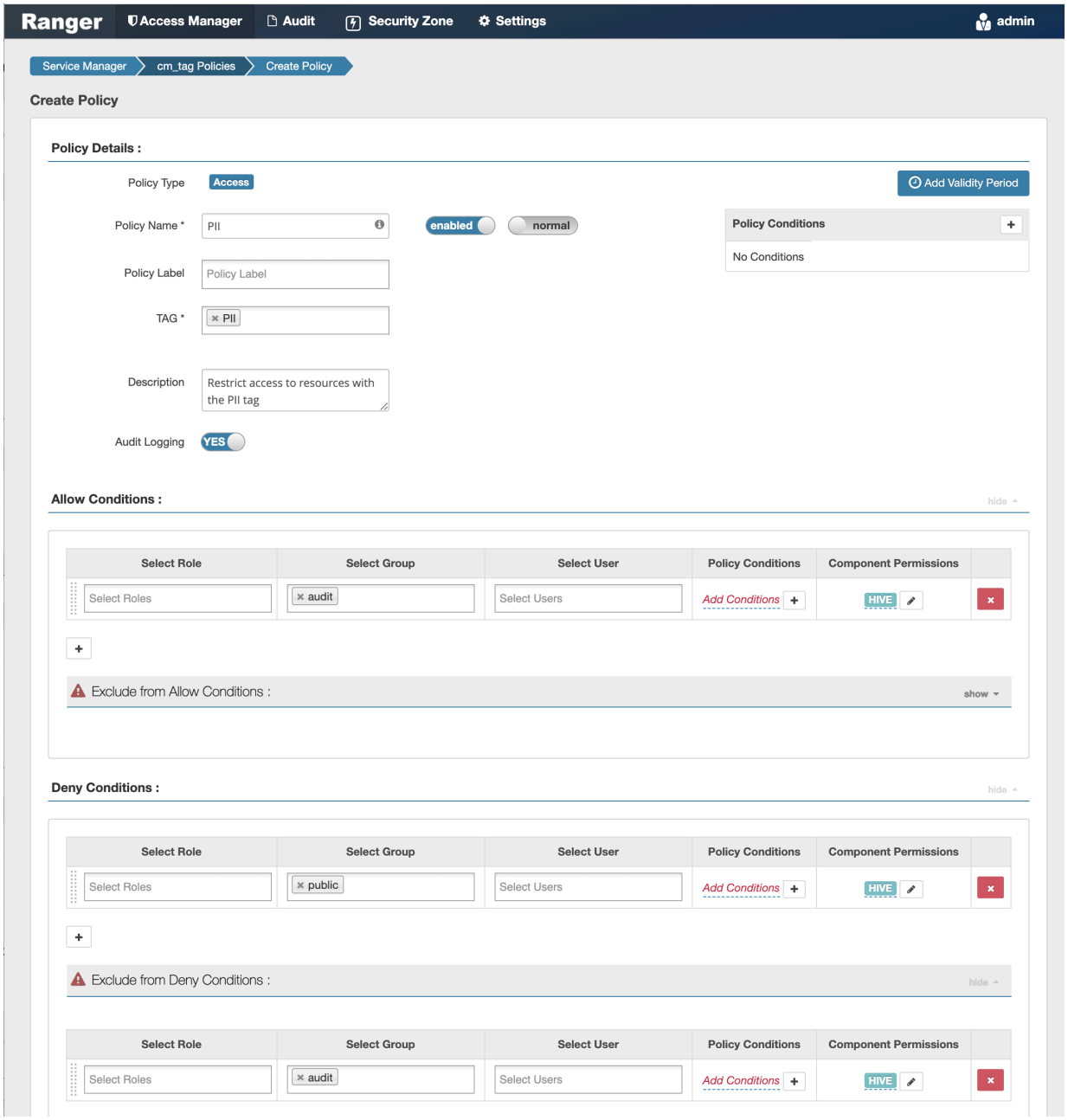
Table 58: Deny Conditions

Label	Description
Select Group	public
Select User	<none>
Policy Conditions	<none>
Component Permissions	hive (select all permissions)

Table 59: Exclude from Deny Conditions

Label	Description
Select Group	audit
Select User	<none>
Policy Conditions	<none>

Label	Description
Component Permissions	hive (select all permissions)



In this example we used Allow Conditions to grant access to the "audit" group, and then used Deny Conditions to deny access to the "public" group. Because the "public" group includes all users, we then used Exclude from Deny Conditions to exclude the "audit" group, in effect reinstating the "audit" group's original Allow access condition.

4. Click Add to add the new policy.

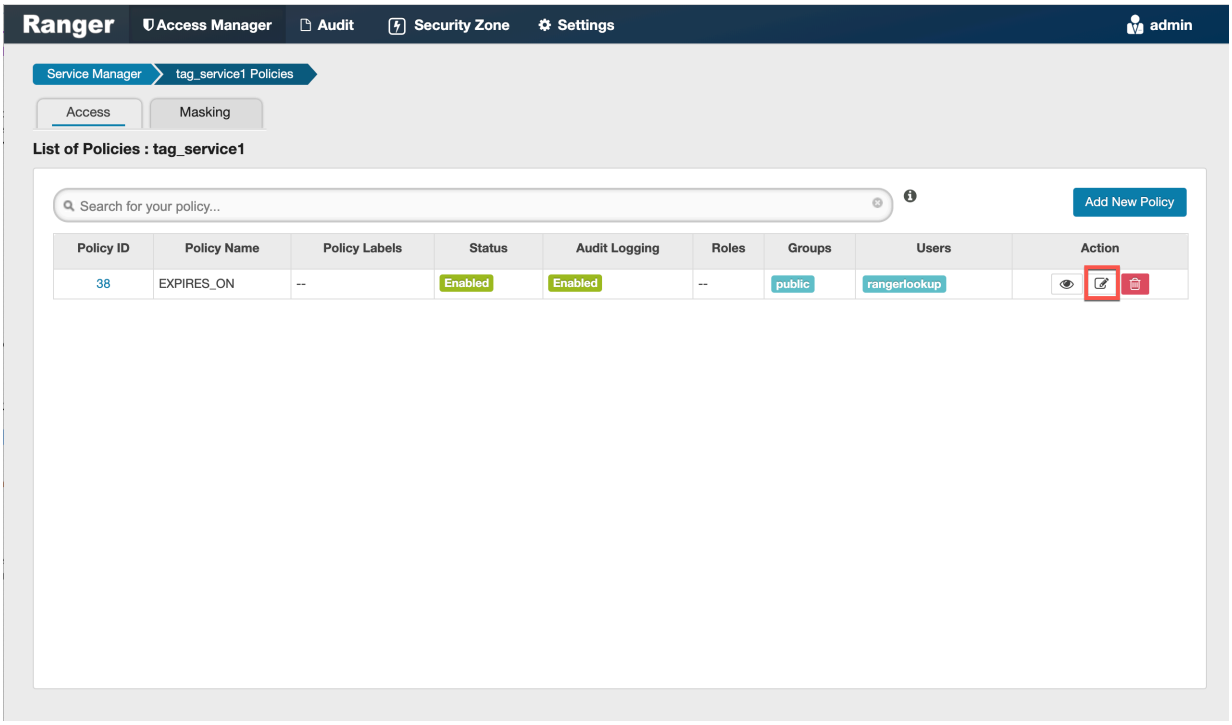
Default EXPIRES ON tag policy

An EXPIRES_ON tag-based policy is created automatically when a tag service instance created. This default policy denies access to objects tagged with EXPIRES_ON after the expiry date specified in the Atlas tag attribute. You can use the following steps to review the default EXPIRES_ON policy.

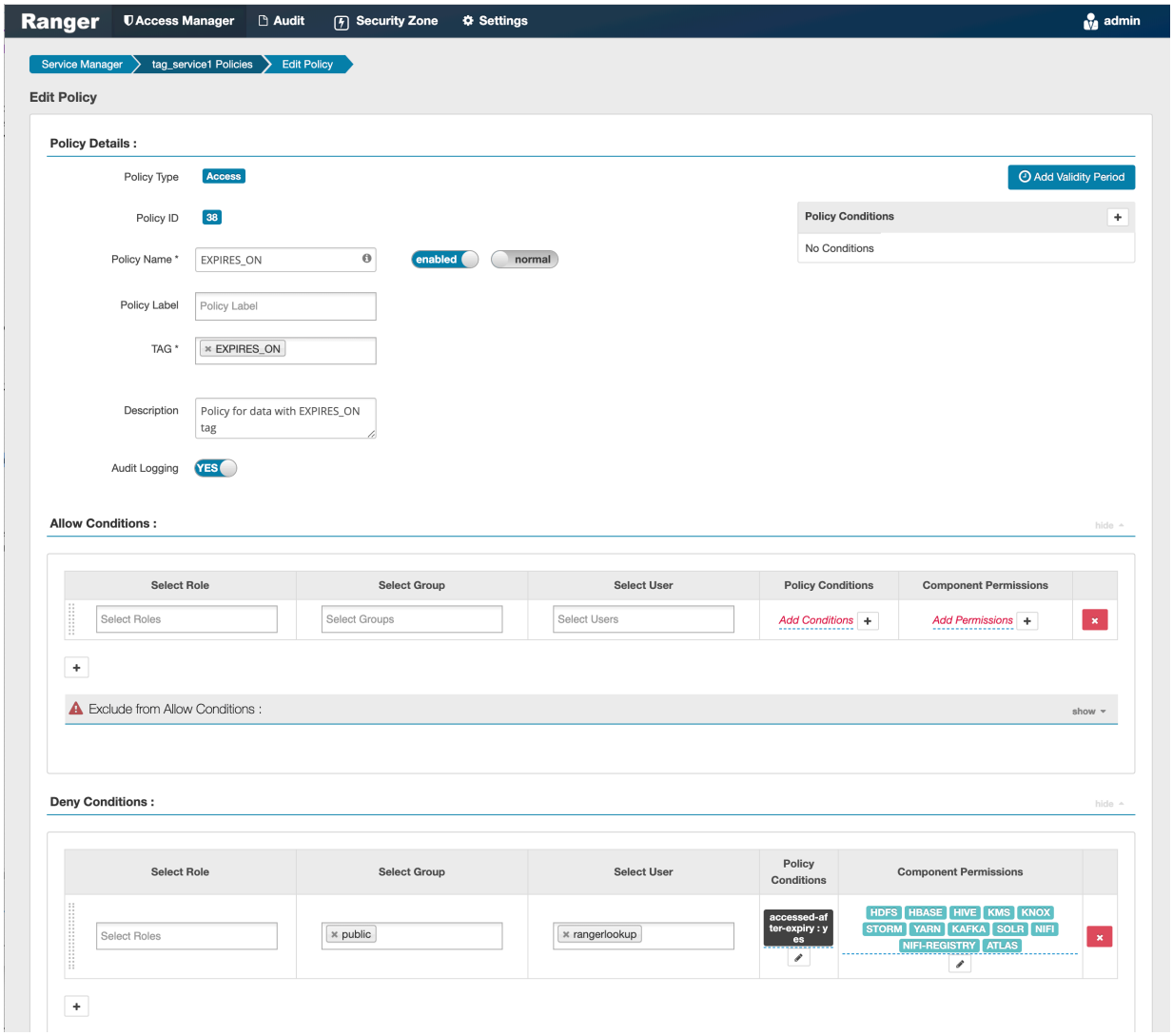
Procedure

1. Select Access Manager > Tag Based Policies, then select a tag-based service.

2. On the List of Policies page, click the Edit icon for the default EXIRES_ON policy.



The Edit Policy page appears:



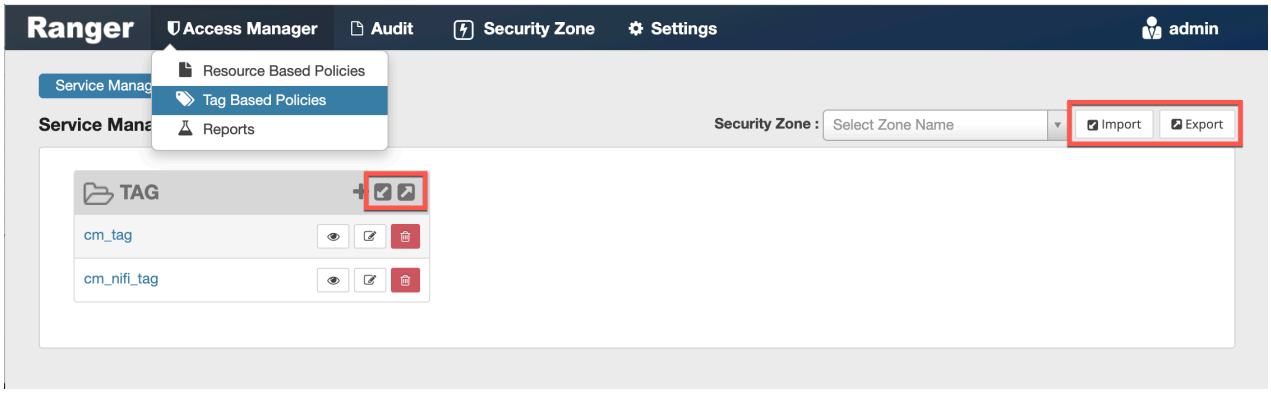
3. We can see that the default EXPIRES_ON policy denies access to all users, and for all components, after the expiry date specified in the Atlas tag attribute.

Importing and exporting tag-based policies

You can export and import policies from the Ranger Admin UI for cluster resiliency (backups), during recovery operations, or when moving policies from test clusters to production clusters. You can import or export a specific subset of policies (such as those that pertain to specific resources or user/groups) or clone the entire repository (or multiple repositories) via the Ranger Admin UI.

Interfaces

You can import and export policies from the Tag Based Policies page:



You can also export policies from the Reports page:

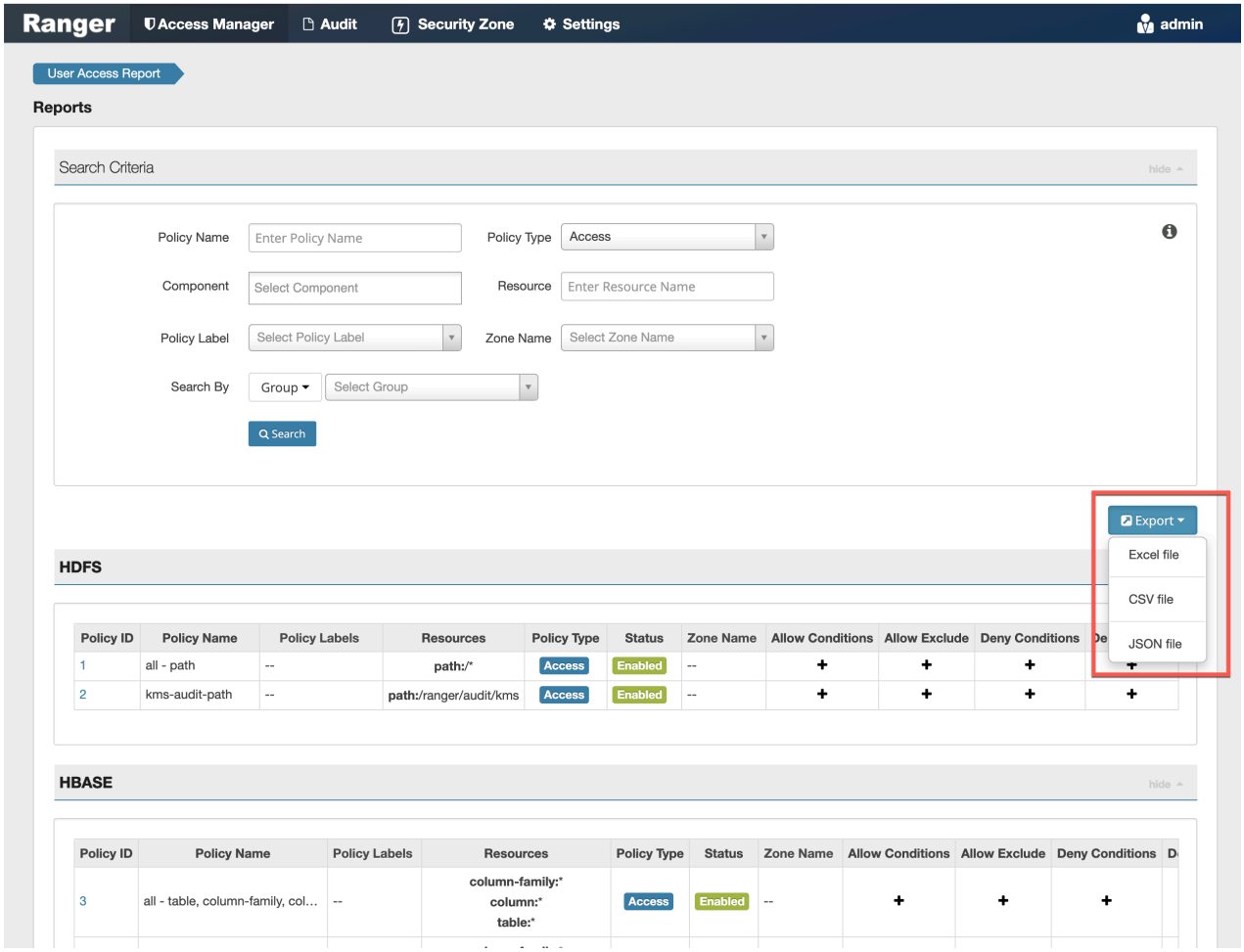


Table 60: Export Policy Options

	Service Manager Page	Reports Page
Formats	JSON	JSON Excel CSV
Filtering Supported	No	Yes
Specific Service Export	Yes	Via filtering

Filtering

When exporting from the Reports page, you can apply filters before saving the file.

Export Formats

You can export policies in the following formats:

- Excel
- JSON
- CSV

  Note: CSV format is not supported for importing policies.

When you export policies from the Service Manager page, the policies are automatically downloaded in JSON format. If you wish to export in Excel or CSV format, export the policies from the Reports page dropdown menu.

Required User Roles

The Ranger admin user can import and export only Resource & Tag based policies. The credentials for this user are set in Ranger Configs > Advanced ranger-env in the fields labeled admin_username (default: admin/admin).

The Ranger KMS keyadmin user can import and export only KMS policies. The default credentials for this user are keyadmin/keyadmin.

Limitations

To successfully import policies, use the following database versions:

- MariaDB: 10.1.16+
- MySQL: 5.6.x+
- Oracle: 11gR2+
- PostgreSQL: 8.4+
- MS SQL: 2008 R2+

Partial policy import is not supported.

Related Information

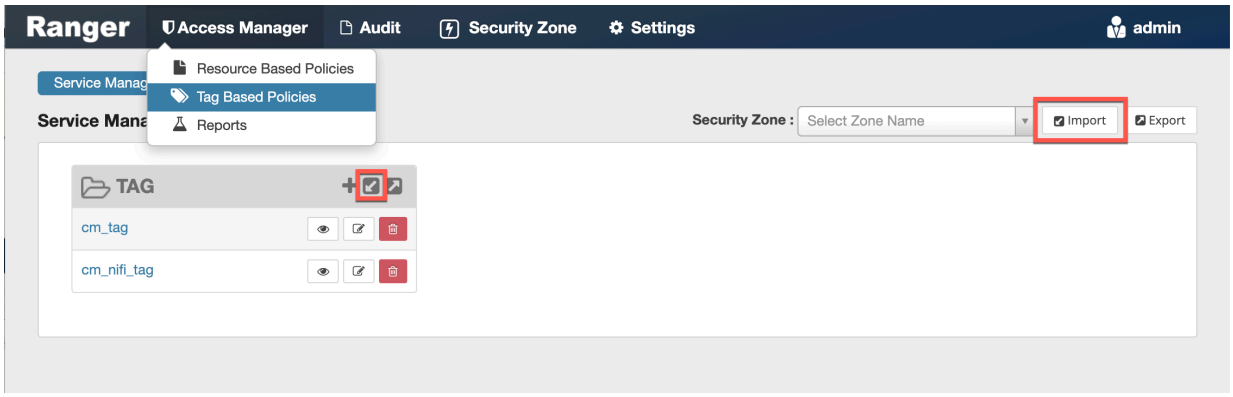
Importing and exporting resource-based policies

Import tag-based policies

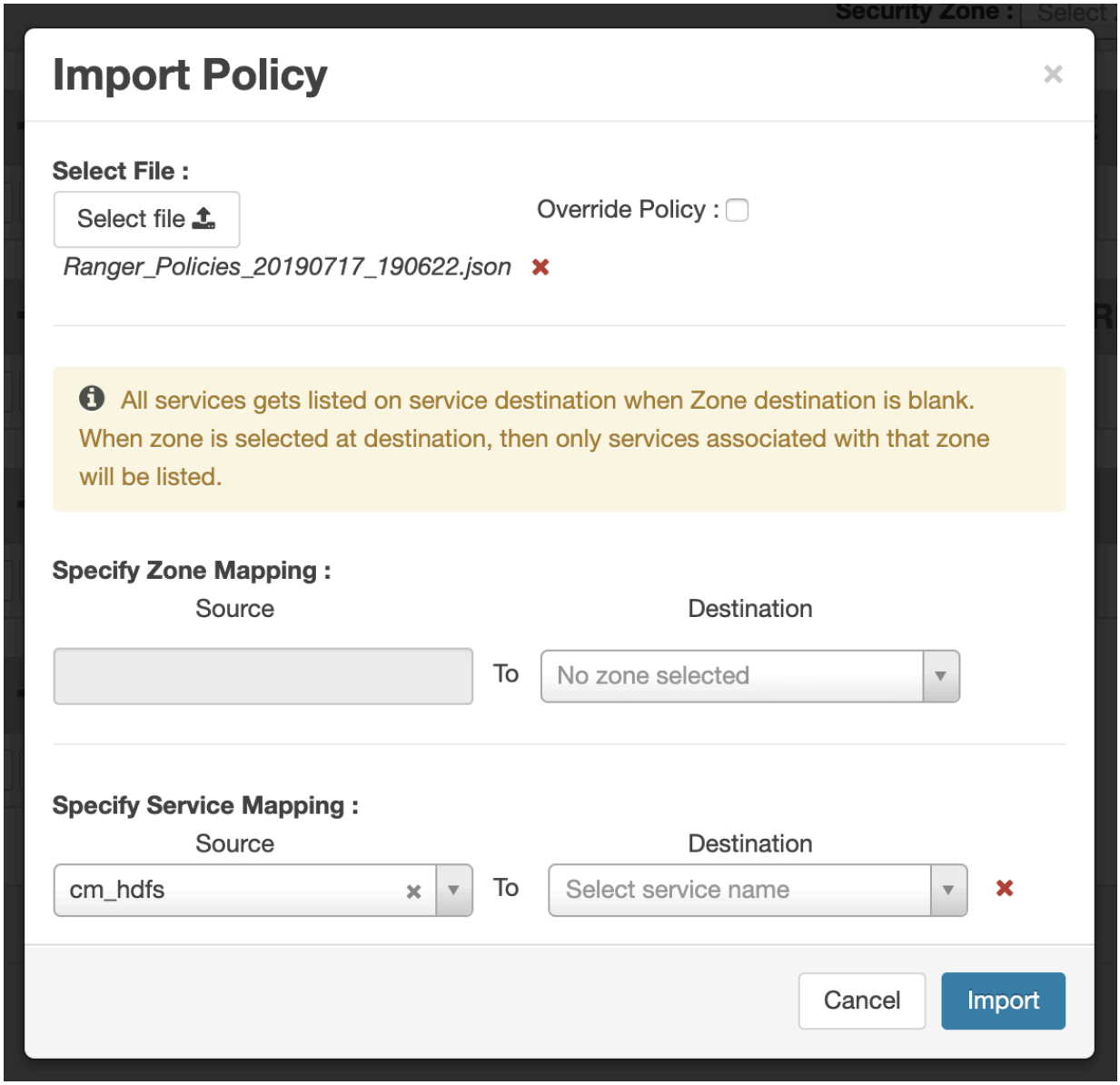
How to import tag-based policies.

Procedure

1. On the Tag Based Policies page, click one of the Import icons:

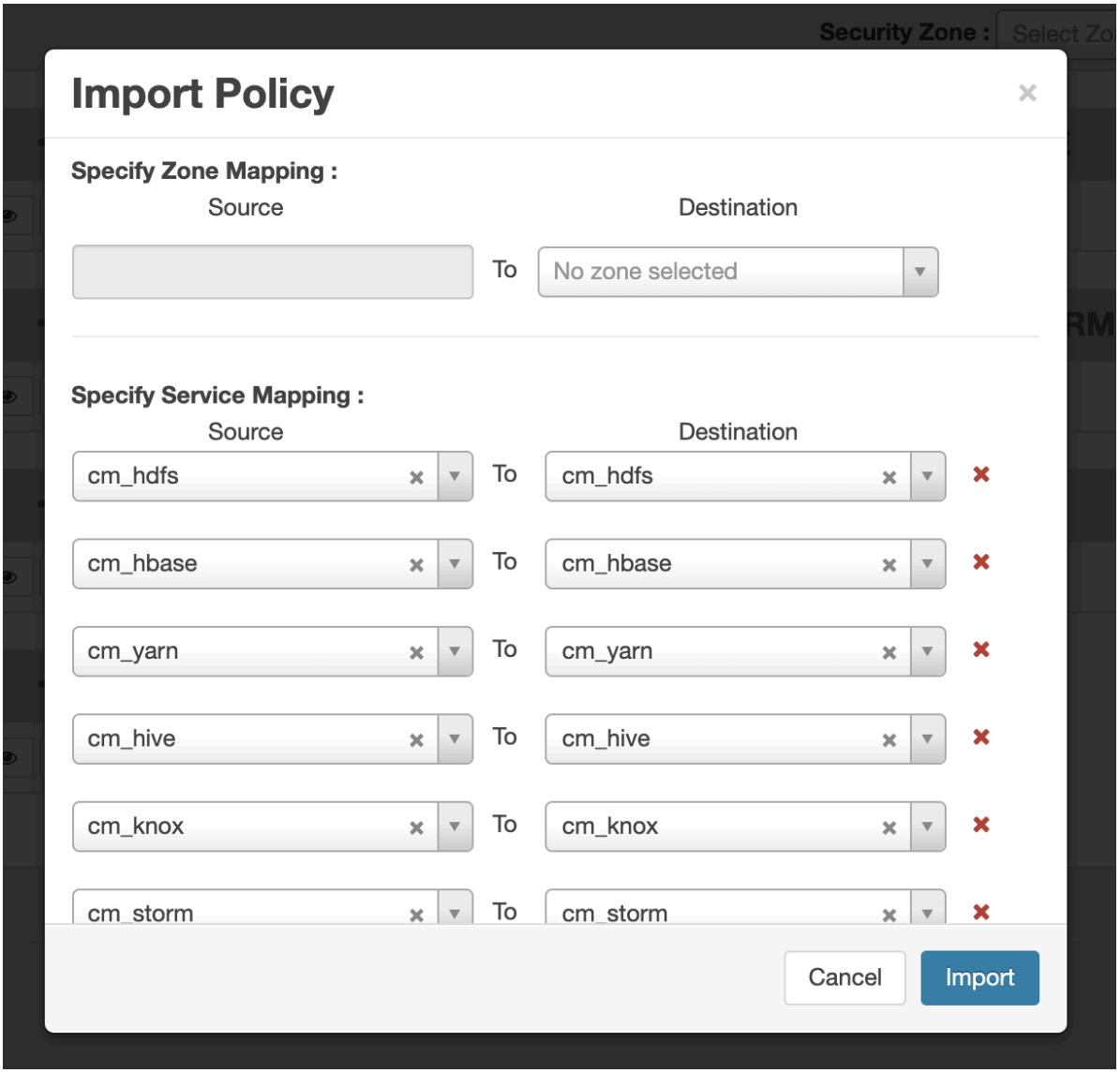


The Import Policy page appears.



2. Select the file to import.
   You can only import policies in JSON format.

3. (Optional) Configure the import operation:
   a) The Override Policy option deletes all policies of the destination repositories.
   b) Zone Mapping – when no destination is selected, all services are imported. When a destination is selected, only the services associated with that security zone are imported.
   c) Service Mapping maps the downloaded file repository, i.e. source repository to destination repository. You can use the red x symbols to remove services from the import. Scroll down to view all service mappings.



4. Click Import.
   A confirmation message appears after the file is imported.

Related Information

Export tag-based policies

Export tag-based policies

How to export all tag-based policies.

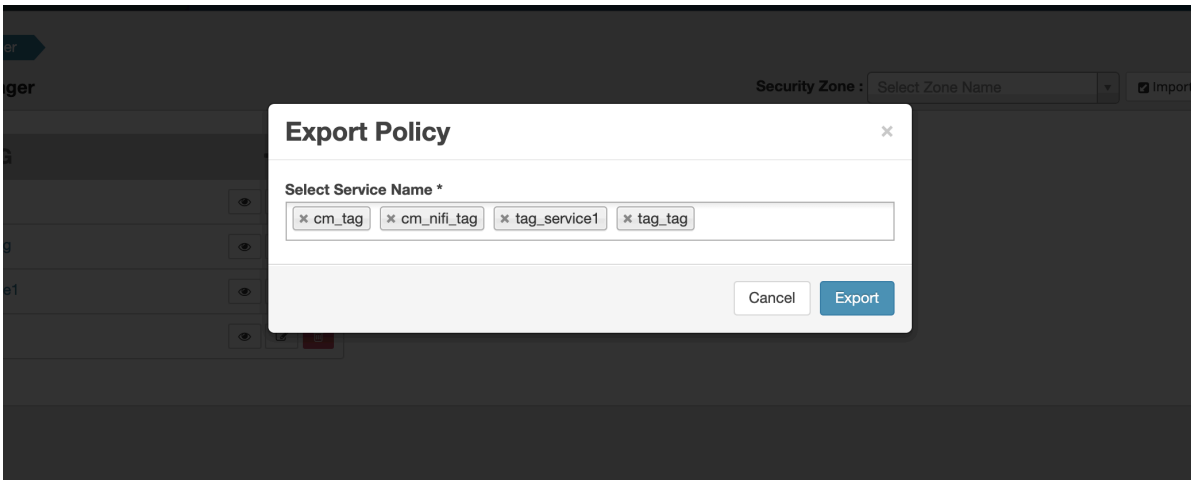
About this task

You can only export policies in JSON format from the Tag-based polices page. If you would like to export in Excel or CSV format, export the policies from the Reports page drop-down menu.

Procedure

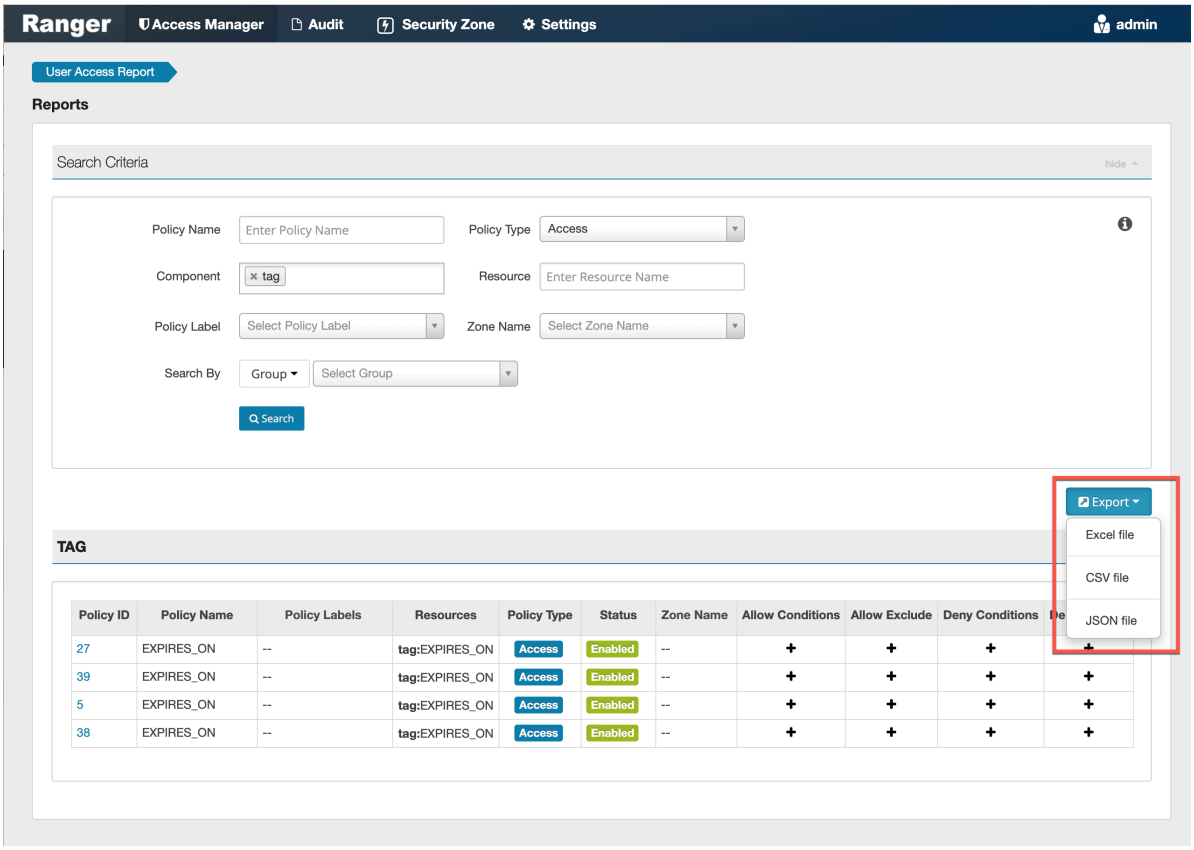
- From the Access Manager >Tag Based Policies page:
  a) Click the Export button or icon:

  The Export Policy page appears.
  b) Remove components or specific services, then click Export.



  c) The file downloads in your browser as a JSON file.

- From the Reports page:
  a) Filter Component to tag and click Search.
  b) (Optional) Apply filters before exporting the file.
  c) Open the Export drop-down menu:



  d) Select the file format.
  The file downloads in your browser.

Create a time-bound policy

Ranger policy validity periods enable you to configure a policy to be effective for a specified time range. You can add a validity period to both resource-based and tag-based policies.

About this task

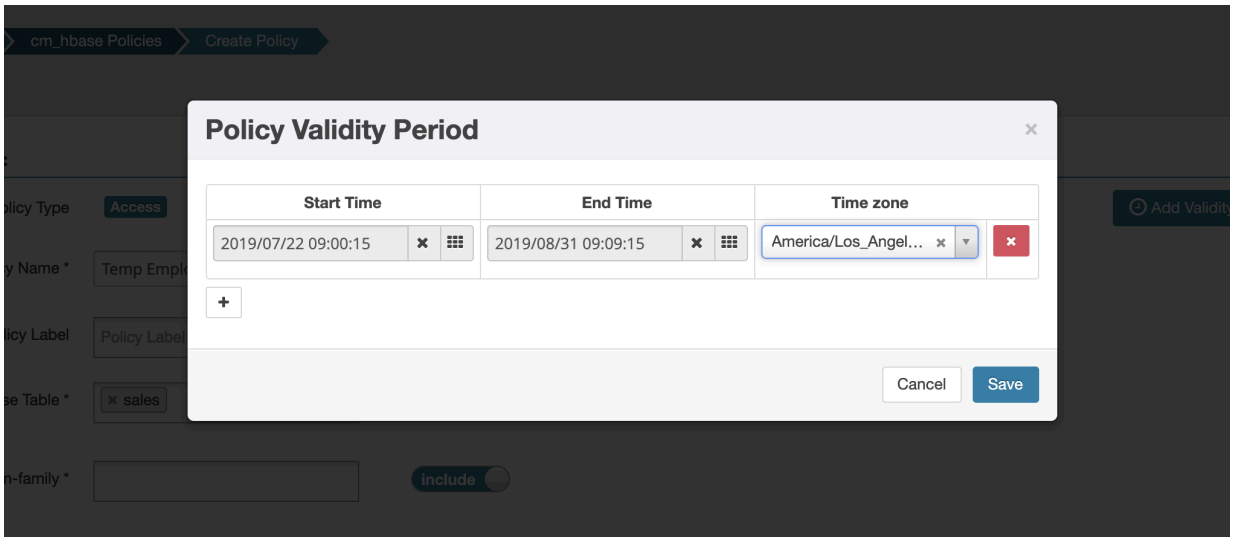
Time-bound policy use-case examples:

- To restrict access to sensitive financial information until the earnings release date.
- To block a certain user for a specific time period (e.g., a compromised user account being investigated needs to be put on "hold" from accessing resources in Hadoop services).
- To block a certain group for a specific time (e.g., excluding temporary employees from writing on resources during the holiday season).

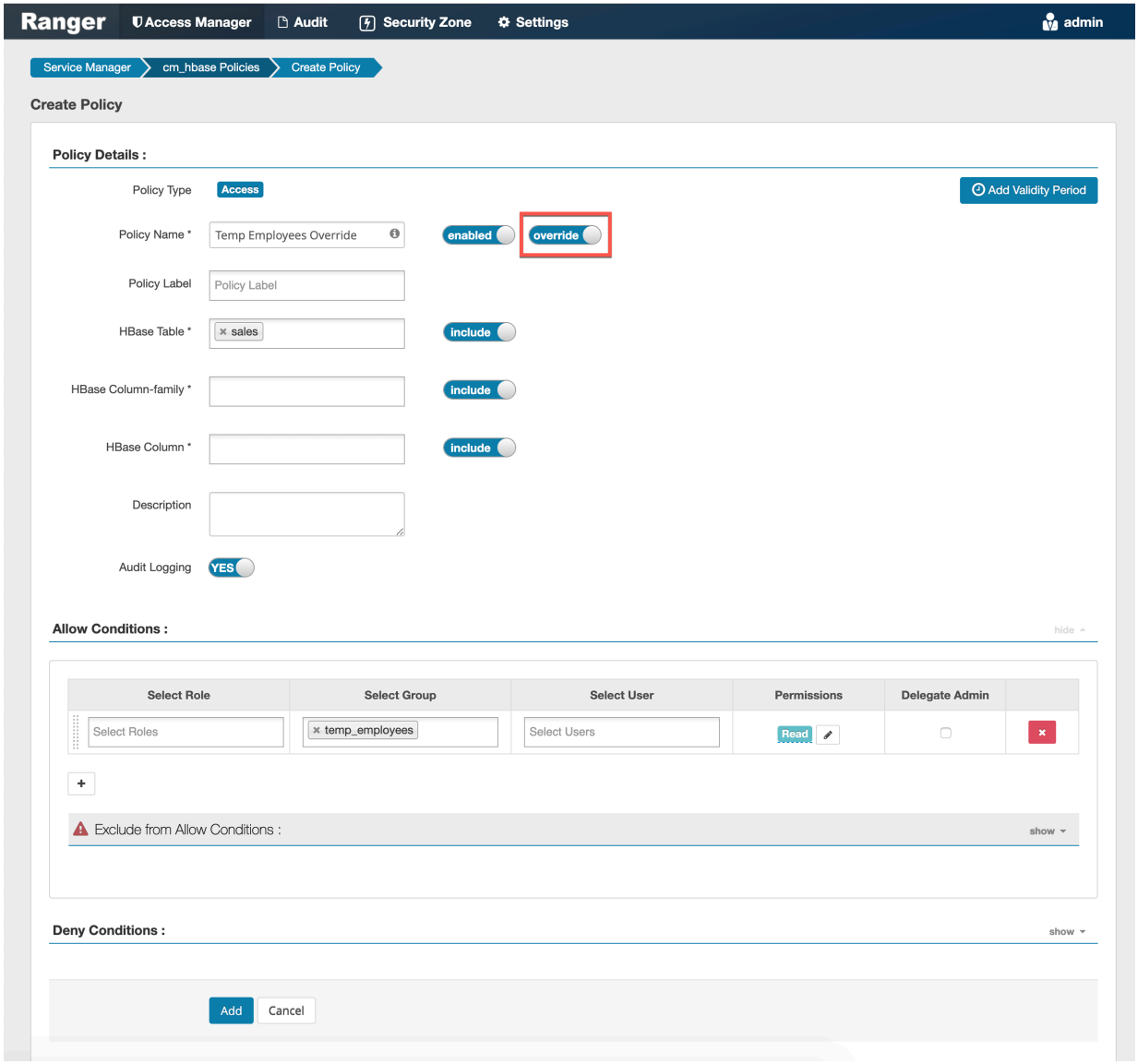
Note: The following procedure shows how to create a time-bound resource-based policy. The procedure is essentially the same for a tag-based resource policy.

Procedure

1. On the Ranger Service Manger page, select a service, then click Add New Policy.
2. Complete the fields on the **Create Policy** page.
3. Click Add Validity Period.
4. On the **Policy Validity Period** pop-up, specify a start time, end time, and time zone. To add additional validity periods, click the + symbol. Click Save to save the specified validity periods.



5. If you would like the policy to override all other policies during its validity period, select override.



6. Click Add.

Create a Hive authorizer URL policy

You can create a Hive Authorizer URL policy in Ranger that maintains Read and Write permissions for a location or folder.

About this task

Hive supports several commands that include URLs which refer to a current or future data location. Such locations must authorize end user access to that location. Currently, you can create a Ranger HDFS policy that grants "All" permissions for a location, recursively. If no such policy exists, HDFS authorization "falls back" to the current ACL that defines access to a location or folder. By default the value of the parameter is “hdfs:,file:”. If you remove “hdfs:”, access requests will be authorized against the HIVE URL policy and won't check for hdfs plugin or Hadoop ACL. This solution requires maintaining many policies or ACLs at the storage level. You can create a Hive Authorizer URL policy in Ranger that maintains Read and Write permissions for a location or folder.

To create a Hive Authorizer policy:

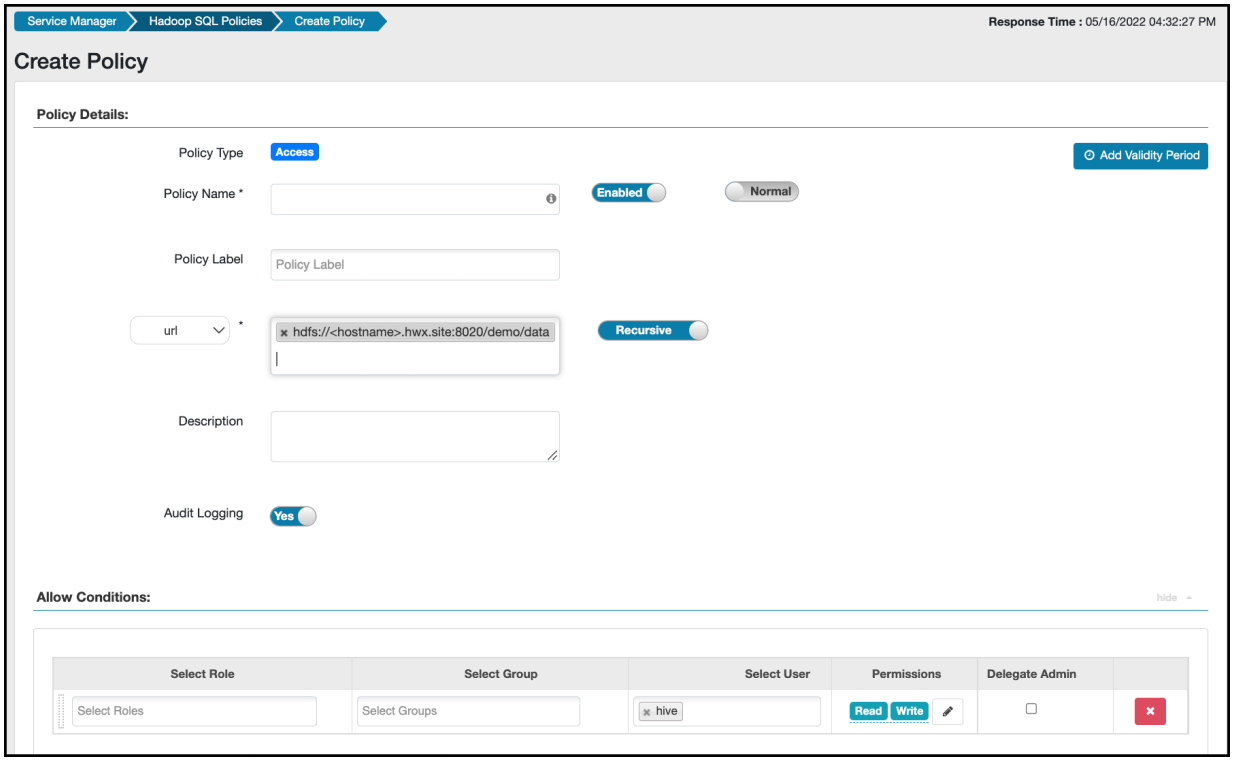
Procedure

1. In Cloudera Manager HIVE-1 Configuration Search . type ranger-hive.
2. In Hive Service Advanced Configuration Snippet (Safety Valve) for ranger-hive-security.xml, click +.
   a) Under HIVE-1, in Name, type: ranger.plugin.hive.urlauth.filesystem.schemes.
   b) In Value, type: file.
   c) Click Save Changes.
3. In Cloudera Manager Hive_On_Tez-1 Configuration Search . type ranger-hive.
4. In Hive Service Advanced Configuration Snippet (Safety Valve) for ranger-hive-security.xml, click +.
   a) Under HIVE_ON_TEZ-1, in Name, type: ranger.plugin.hive.urlauth.filesystem.schemes.
   b) In Value, type: file.
   c) Click Save Changes.
5. In HIVE-1 Actions , click Restart.
6. In HIVE_ON_TEZ-1 Actions , click Restart.

   By default the value of the parameter is “hdfs:,file:”. If you remove “hdfs:”, access requests will be authorized against the HIVE URL policy and won't check for hdfs plugin or Hadoop ACL.
7. In Ranger Resource Policies Hadoop SQL , click Add New Policy.
8. In Policy Details, select URL, then type the URL represents the location or folder to which you want Ranger to authorize access: hdfs://<hostname>.root.hwx.site:8020/demo/data.

9. In Allow Conditions, select user(s), then choose Read and Write permissions, as shown in the following example:

Figure 5: Creating a Hive Authorizer URL Policy



This policy allows the user to READ / WRITE into the location defined by the URL.

CREATE EXTERNAL TABLE IF NOT EXISTS STUDENT (student_ID INT, FirstName STRING, LastName STRING, year STRING, Major STRING) COMMENT 'Student Names' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION '/demo/data';

This will create a table reading data from the location /demo/data provided user will have the necessary READ permission to the location along with CREATE permission for table STUDENT

If the storage system is S3A or ADFS, then URL policy would be maintained for the scheme. For example, s3a://<folder>, abfs://<folder>.

Hive supports URL policies for the following commmands that have URLs defined for the respective location:

CREATE TABLE

external table location

ALTER TABLE LOCATION

new location

ALTER PARTITION LOCATION

new partition location

ALTER TABLE ADD PARTITION

for partition location

LOAD

input location

For additional information about creating Hive commands with URL, see https://cwiki.apache.org/confluence/display/RANGER/Hive+Commands+to+Ranger+Permission+Mapping .

Showing Role|Grant definitions from Ranger HiveAuthorizer

You can use beeline to show the roles granted to users, groups, and roles.

About this task

You can create roles in Ranger or in Hive. You create roles in HIVE using ROLE commands. such as CREATE ROLE, GRANT / REVOKE ROLE. You can create roles in Ranger, using the Ranger Admin Web UI, if you have Admin permissions. See related links for more information about creating roles. The Hive2 command line infterface Beeline returns role grant defnitions for a specific principal, such as a user, group or role.

Before you begin

Roles must be defined before using beeline to show role|grant definitions.

Procedure

1. Run beeline, (the hive2 command line interface) on the Ranger host.

```
beeline -u jdbc:hive2://<ranger_host_name>
```

2. Enter valid syntax to return the role definitions for a specific principal.

   Syntax

   SHOW ROLE GRANT (USER|GROUP|ROLE) principal_name;

   where

   principal_name is USER | GROUP | ROLE name

Results

Beeline outputs query results, as shown in following examples:

Example

SHOW ROLE GRANT USER HDFS -> show roles for user "hdfs"

```
0: jdbc:hive2://rm-ranger-3.rm-ranger.root.hw> show role grant user hdfs;
INFO  : Compiling command(queryId=hive_20211109235258_b9211cfe-0e78-47d7-8a2a-1b6111ddcd18): show role grant user hdfs
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:role, type:string, comment:from deserializer), FieldSchema(name:grant_option,
_time, type:bigint, comment:from deserializer), FieldSchema(name:grantor, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20211109235258_b9211cfe-0e78-47d7-8a2a-1b6111ddcd18); Time taken: 0.02 seconds
INFO  : Executing command(queryId=hive_20211109235258_b9211cfe-0e78-47d7-8a2a-1b6111ddcd18): show role grant user hdfs
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20211109235258_b9211cfe-0e78-47d7-8a2a-1b6111ddcd18); Time taken: 0.008 seconds
INFO  : OK
+-----------+--------------+---------------+----------+
|    role   | grant_option |   grant_time  | grantor  |
+-----------+--------------+---------------+----------+
| ITManager | false        | 1636501912000 |          |
+-----------+--------------+---------------+----------+
```

Example

SHOW ROLE GRANT ROLE -> show roles for role "ITManagers"

```
0: jdbc:hive2://rm-ranger-3.rm-ranger.root.hw> show role grant role ITManager;
INFO  : Compiling command(queryId=hive_20211109235607_2d543c50-7c7b-4d54-bed0-159f67c24079): show role grant role ITManager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:role, type:string, comment:from deserializer), FieldSchema(name:grant_option,
_time, type:bigint, comment:from deserializer), FieldSchema(name:grantor, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20211109235607_2d543c50-7c7b-4d54-bed0-159f67c24079); Time taken: 0.177 seconds
INFO  : Executing command(queryId=hive_20211109235607_2d543c50-7c7b-4d54-bed0-159f67c24079): show role grant role ITManager
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20211109235607_2d543c50-7c7b-4d54-bed0-159f67c24079); Time taken: 0.007 seconds
INFO  : OK
+------------+---------------+---------------+----------+
|    role    | grant_option  |   grant_time  | grantor  |
+------------+---------------+---------------+----------+
| Managers   | false         | 1636502074000 |          |
| TeamLeads  | false         | 1636502122000 |          |
+------------+---------------+---------------+----------+
```

Related Information

SQL Standard Based Hive Authorization
Apache documentation on Role operations
NiFi Ranger based policy descriptions
Adding a role through Hive
Adding a role through Ranger

Ranger Security Zones

Ranger security zones let you organize service resources into multiple security zones.

Security Zones allow carving/bucketing of resources in a service into multiple zones for better administration of security policies. Defining Security Zones can enable multiple administrators to setup security policies for a service – based on the zones to which they have been granted administration rights.

Security Zones Administration

A Security Zone enables a Ranger administrator to separate resource policies into different administrative zones.

What is a Security Zone?

Security Zones help simplify security policy administration, and allow a limited amount of policies to be checked when doing authorization against certain resources. Only policies under a particular zone that contains the requested resource are loaded and checked by Ranger.

For example, let us consider two security zones: finance and sales:

- Security zone finance includes all content in a Hive database named finance.
- Security zone sales includes all content in a sales database.
- Policies defined in a security zone apply only to resources of that zone.
- A zone can be extended to include resources from multiple services such as HDFS, Hive, HBase, Kafka, etc. Extending a zone across multiple services allows zone administrators to set up policies for resources owned by their organization across multiple services.

For example:

```
Zone: finance
    service: prod_hdfs; path=/finance/*, /taxes/*
    service: prod_hive; database=finance
    service: prod_kafka; topic=FIN_*
    service: test_hadoop; path=/finance/*, /taxes/*
Zone: sales
    service: prod_hadoop; path=/sales/*
    service: prod_hive; database=sales
```

```
    service: prod_kafka; topic=SALES_*
```

- As shown above, resources can be specified using wildcards (FIN_*, SALES_*).
- Sets of users and groups are designated as administrators in each security zone.
- Users are allowed to set up policies only in security zones in which they are administrators.
- A resource cannot map to more than one security zone. Ranger does not allow creating security zones that specify resources that match resources in another zone. For example, an attempt to update the finance zone in the above example with the HDFS path /sales/finance/* is not permitted, because this conflicts with the HDFS path /sales/* specified in the sales zone.
- A set of users and groups can be designated as administrators of a security zone. Administrators can create, update, and delete security policies for the resources in that security zone.
- A set of users and groups can be authorized to view audit logs for that security zone's resources. Other users are not allowed to view access-audit logs for that security zone's resources.
- The security zone name appears in the zonename column of the access-audit log.

Security Zone Administration

- Security zones can only be created, updated, or deleted by a user with the ROLE_SYS_ADMIN role in Ranger.
- Users can view, retrieve, and update policies only in security zones in which they have administrator privileges.
- Users can view/retrieve and cannot update zone policies for which they have zone auditor permission.

How are Security Zones Used in Authorization?

When a Ranger authorization plugin authorizes a resource access request, it first determines the zone in which the accessed resource resides. If the resource matches a security zone, only the policies of that security zone are used to authorize the access. If resource does not match any security zone, the policies in the default (unnamed) security zone are used to authorize the access.

Tag-based Policies in Security Zones

In a given service, each security zone can be configured to use tag-based policies from a specific security zone in a tag-service. This enables different tag-based authorization policies to be used, based on the security zone of the resource.

Audit Logs

Audit logs generated by Ranger include the name of the security zone in which the accessed resource resides. Only users who have been assigned as an Admin or Auditor for the security zone are allowed to view the audit logs.

Security Zones Example Use Cases

Four example use cases for admininistering security zones.

Based on the following example:

```
Zone: finance
          service: prod_hdfs; path=/finance/*, /taxes/*
          service: prod_hive; database=finance
          service: prod_kafka; topic=FIN_*
          service: test_hadoop; path=/finance/*, /taxes/*
Zone: sales
          service: prod_hadoop; path=/sales/*
          service: prod_hive; database=sales
          service: prod_kafka; topic=SALES_*
```

Use case 1 : Access HDFS path using zone policy

For example, let us access hdfs path using unixuser1 user from finance zone.

Finance zone resource:

Ranger Service : prod_hdfs

Resource : /finance/*

Finance zone policy:

Resource Path : /finance/*

User : unixuser1

Permission : read, write, execute

Now, when unixuser1 user tries to create dir in /finance dir, Ranger checks for zone with resource /finance and policy for that user in that zone and then allows access for that user. Also, access-audit logs for that operation appear in the Ranger Admin Web UI, Access Audit tab.

Use case 2 : Hive access policy and tag masking policy

For example, we want to manage access policies and masking policy for taxation-related information in multiple finance databases for an organization.

Zone Resource :

Zone Tag service: cm_tag

Ranger Service : prod_hive

Resource :

Database : finance

Zone policy resource

Tag policy

resource:TDS

Hive policy

Resource :

Database : finance

Now, the Admin and security zone admin can create access policies and masking policies for all the resources associated with tag TDS and as and when new tables on Hive / Hbase are created for saving any taxation related data. They can associate a TDS tag with a related Hive / Hbase column. This will enable zone admin to create policies for masking the confidential data of its organization.

Use case 3 : Knox topologies

For example, suppose we want to manage access to a service. We can mange access to a service using topology.

Zone Resource :

Ranger Service : prod_knox

Resource:

Knox Topology:cdp-proxy-api

Knox Service:WEBHDFS

Zone deny policy Resource:

Knox Topology:cdp-proxy-api

Knox Service:WEBHDFS

Without a security zone, access to webhdfs is allowed since the default policy has a 'public' group in it.

Use case 4 : Import and export of zone policy

We can import and export zone policies from stage to prod.

Suppose we want to have the same policy in production that exists on stage. We can export the zone policy from the stage where the exported json has a zone name as a parameter in the json. While importing, we can map the zone name of stage to prod and then import the policies.

Adding a Ranger security zone

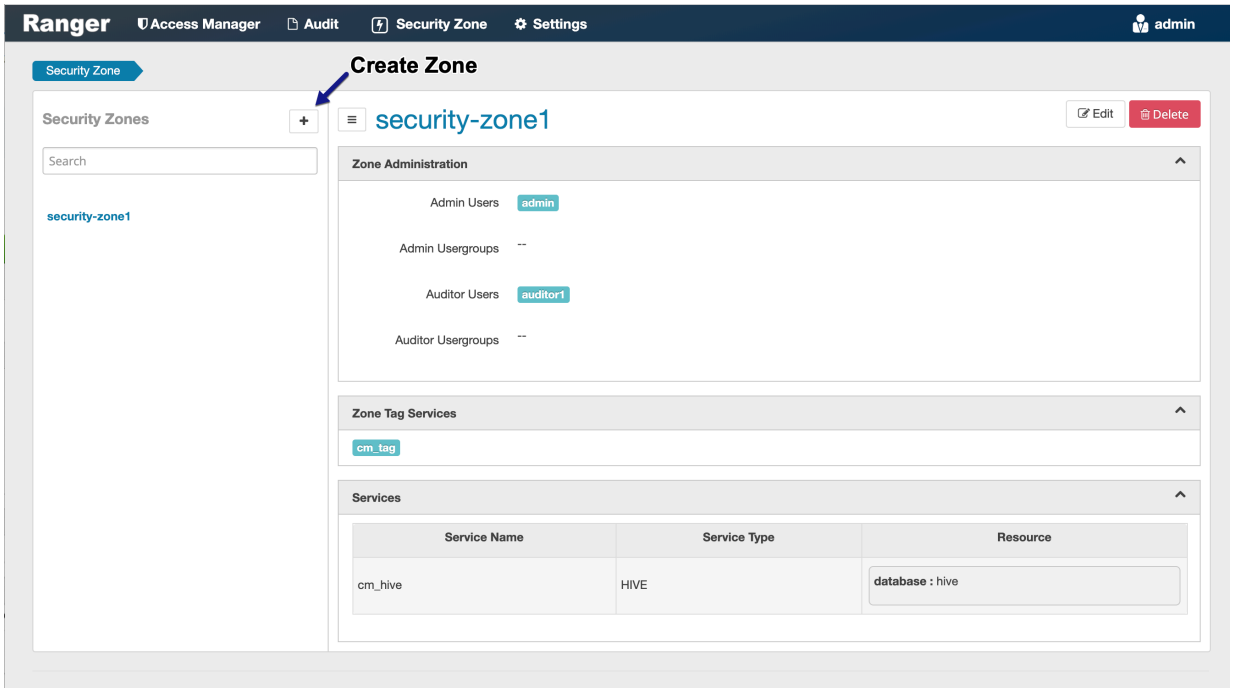
How to add a new Ranger Security Zone.

Procedure

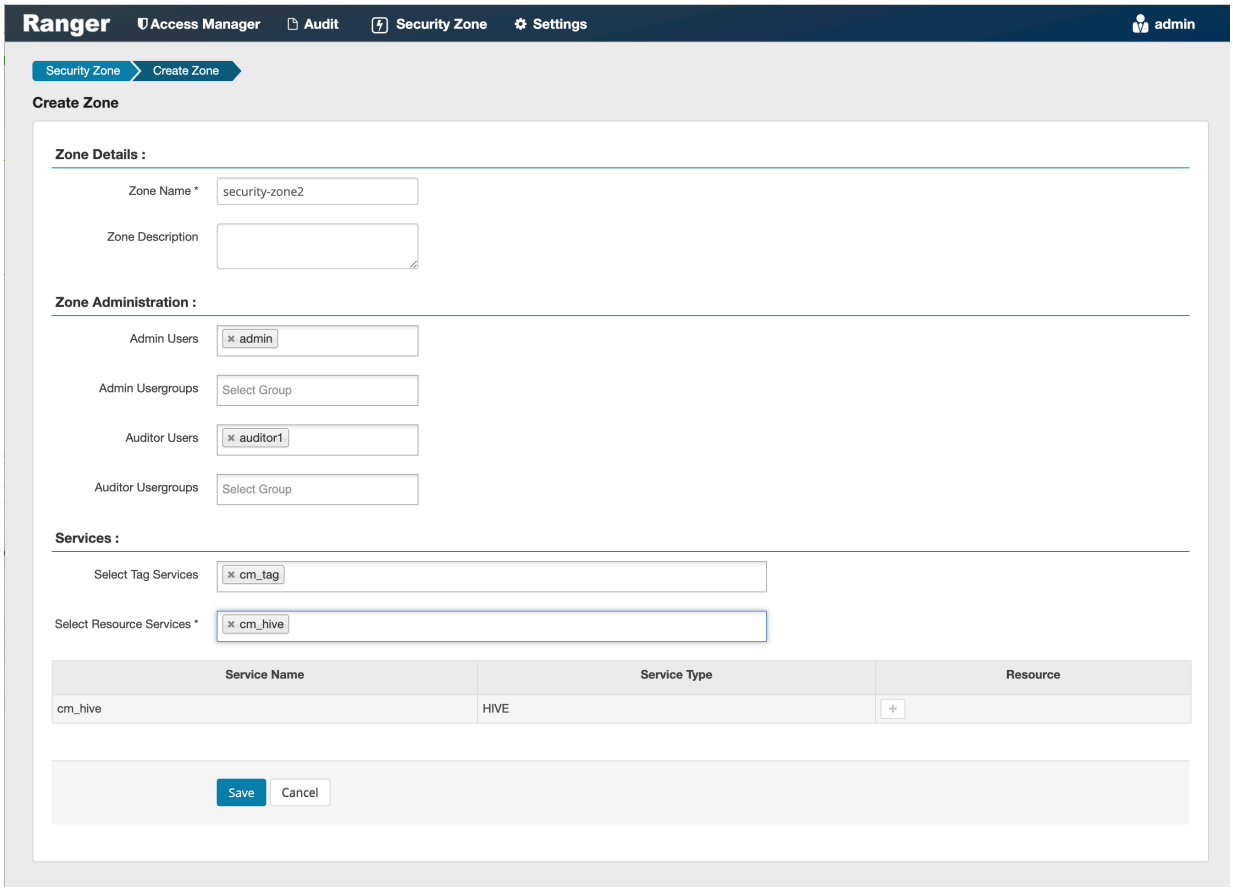
1. Click Security Zone in the top menu.

   The Security Zone page appears.

2. On the Security Zone page, click the + icon.



The Create Zone page appears.



3. Complete the Create Zone page as follows:

Table 61: Zone Details

Field	Description
Zone Name	The security zone name.
Zone Description	An optional description.

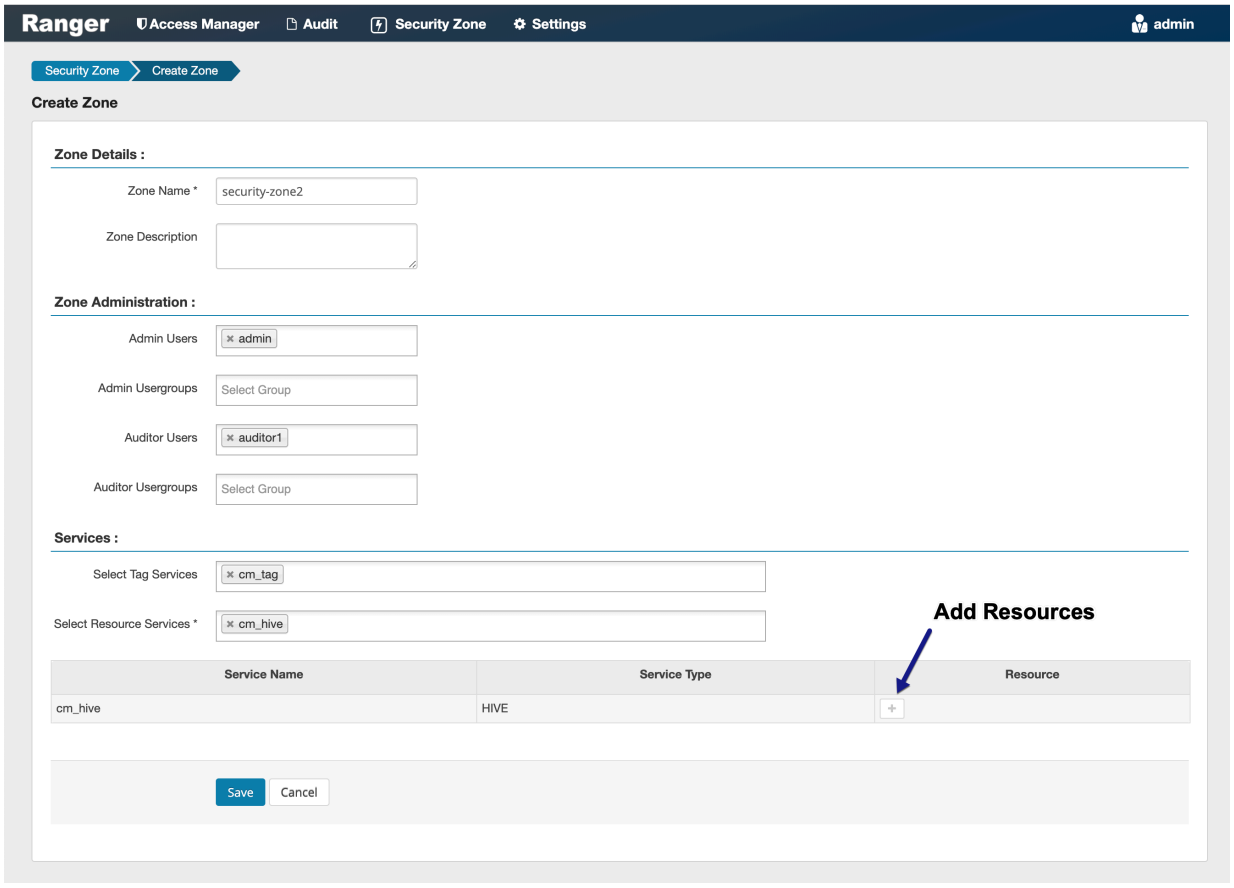
Table 62: Zone Administration

Field	Description
Admin Users	The Admin users for the security zone.
Admin Usergroups	The Admin user groups for the security zone.
Auditor Users	The Auditor users for the security zone.
Auditor Usergroups	The Auditor user groups for the security zone.

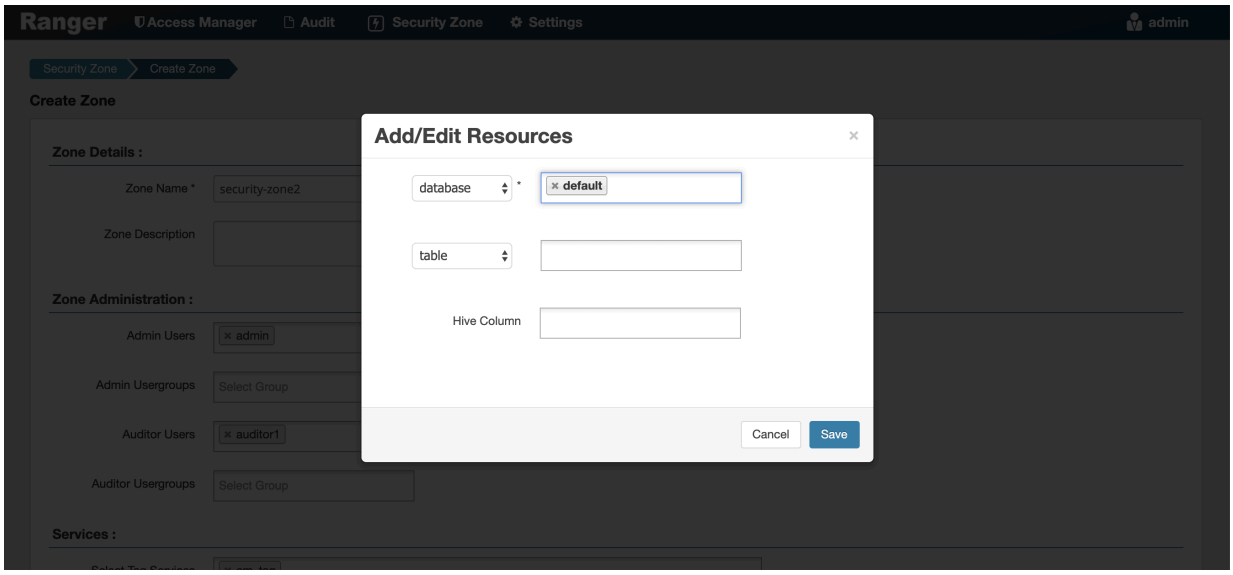
Table 63: Services

Label	Description
Select Tag Services	Select tag-based services for the security zone.
Select Resource Services	Select resource-based services for the security zone.

4. Selected Services are listed in the Services table. To add resources for each selected service, click the + icon in the Resources column for the applicable service.



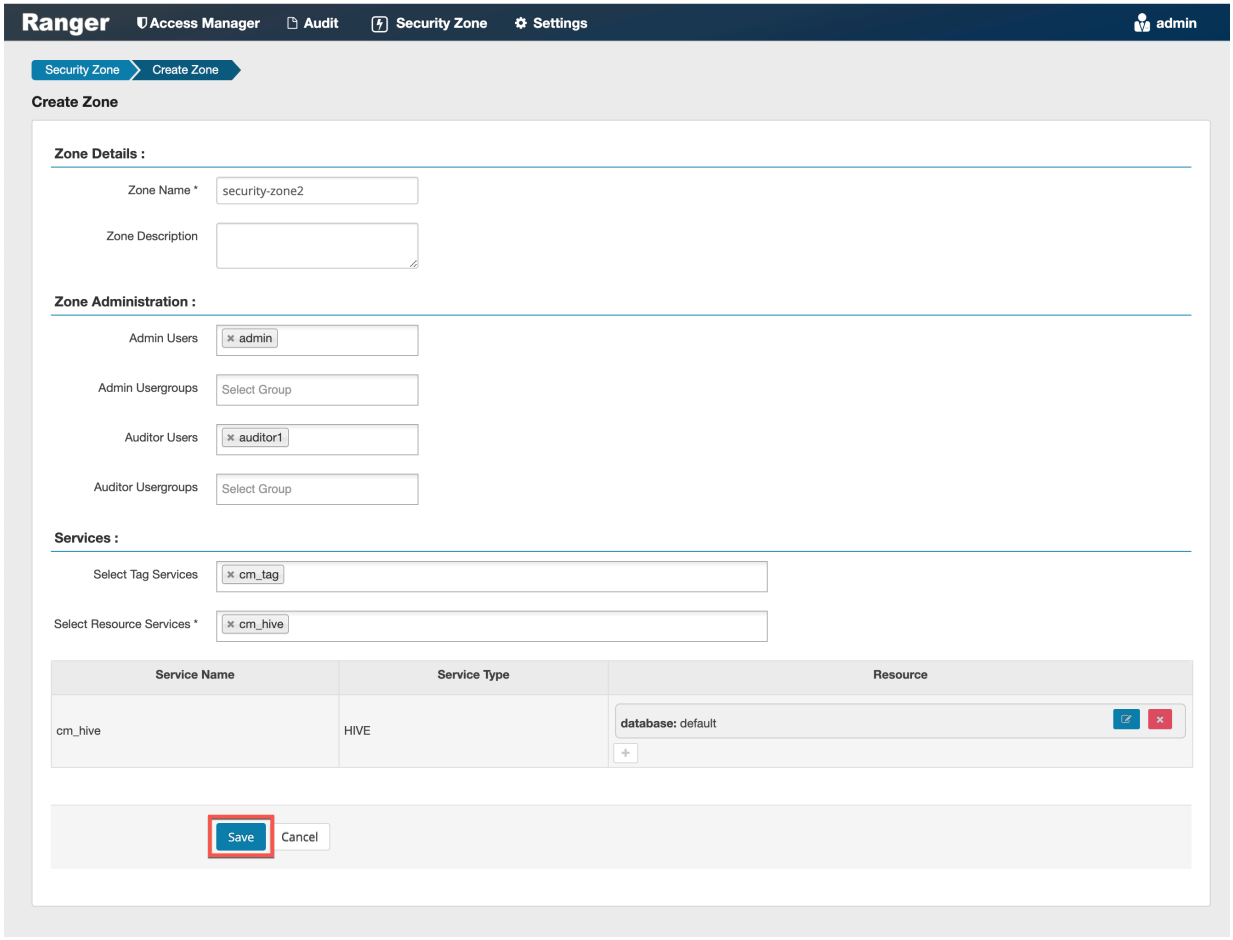
5. Use the Add/Edit Resources pop-up to specify resources for the service, then click Save.



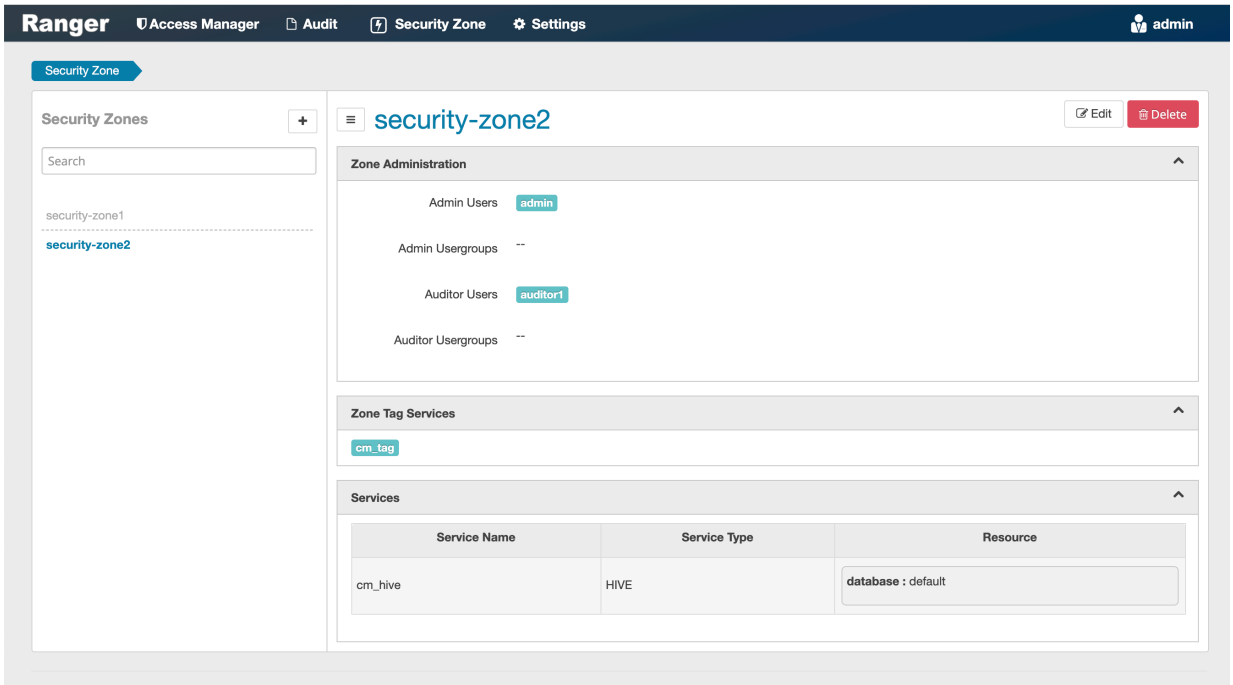
The resources are listed in the Resources column of the Services table.

Note: The solr plugin supports fine-grained authorization similar to legacy Sentry privileges. A part of this support introduces the following new solr resources: collection, config, schema and admin. To perform any operation on a collection, a user also requires admin-level permission. To create a security zone for with the solr service that includes a collection resource, you must also add an admin resource. Currently, if you use one solr service to create a security zone that has a collection resource (and therefore includes an admin resource) you cannot create another solr security zone using another collection. (currently, only one admin resource can be used per solr security zone). This limitation exists for security zones in cr-7.1.8.

6. Click Save at the bottom of the Create Zone page to save the new security zone.

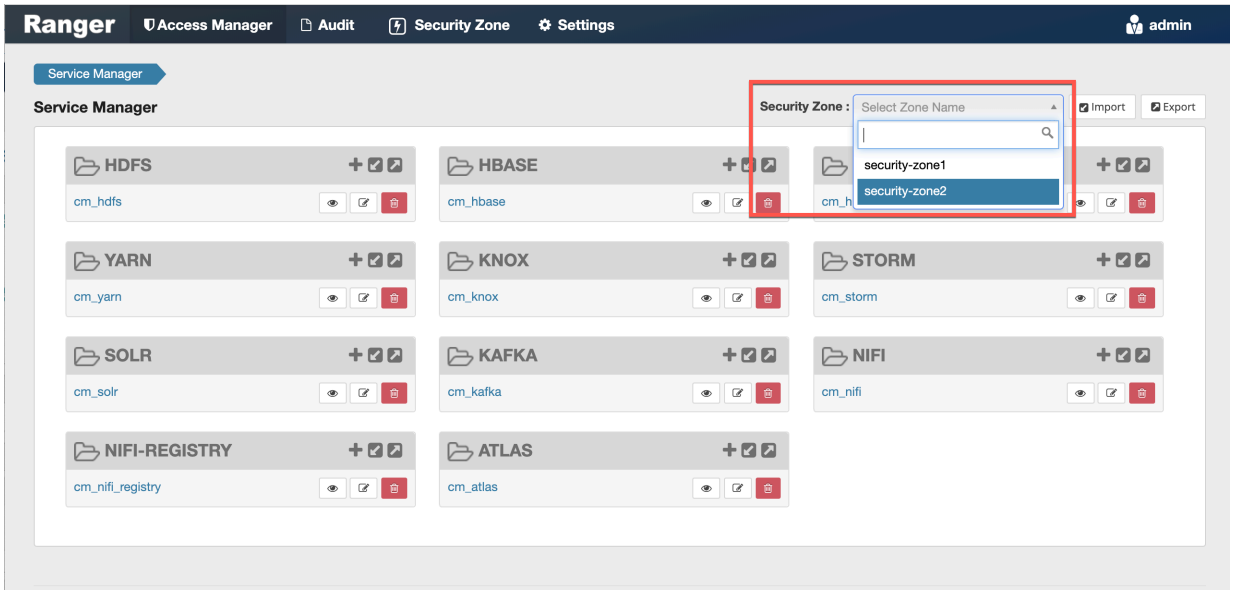


7. The new security zone is listed on the Security Zone page.



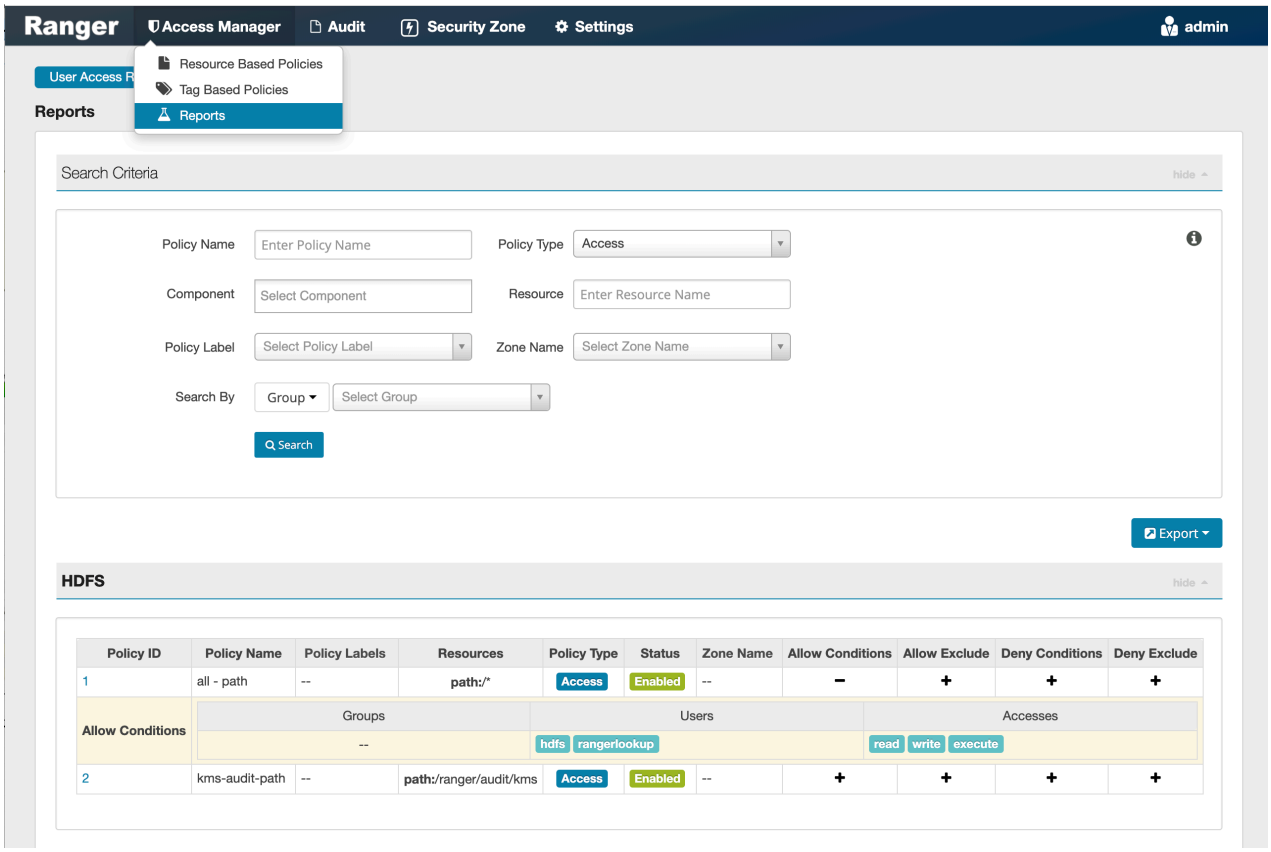
8. To edit a security zone, click the security zone name in the Security Zones list, then click Edit.

9. After security zones have been created, you can use the Security Zone selection box on the Service Manager page to display the services assigned to the selected security zone. A Zone Name column appears in the table on the Audit > Access page, and also in the Access Manager > Reports tables.



Administering Ranger Reports

You can use the Reports page to help manage policies more efficiently as the number of policies increases. This page lists all resource-based and tag-based policies.

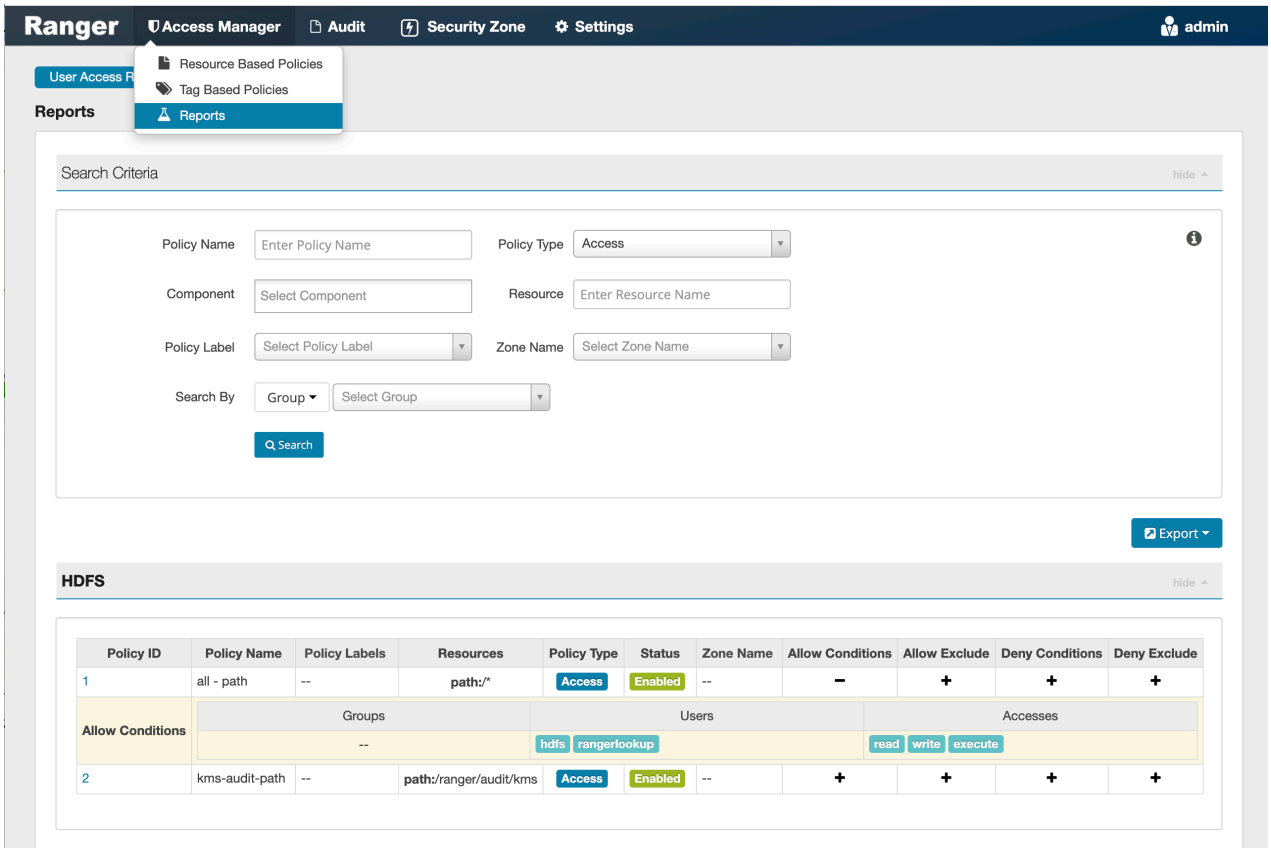


View Ranger reports

How to view reports for Ranger policies.

To view reports for one or more policies, select Access Manager > Reports.

- To view Allow Condition details for each policy, click the icon in the Allow Conditions column. You can use the same method to view details for other policy conditions (Allow Exclude, Deny Conditions, etc.).
- To edit a policy from the Reports page, click the Policy ID.

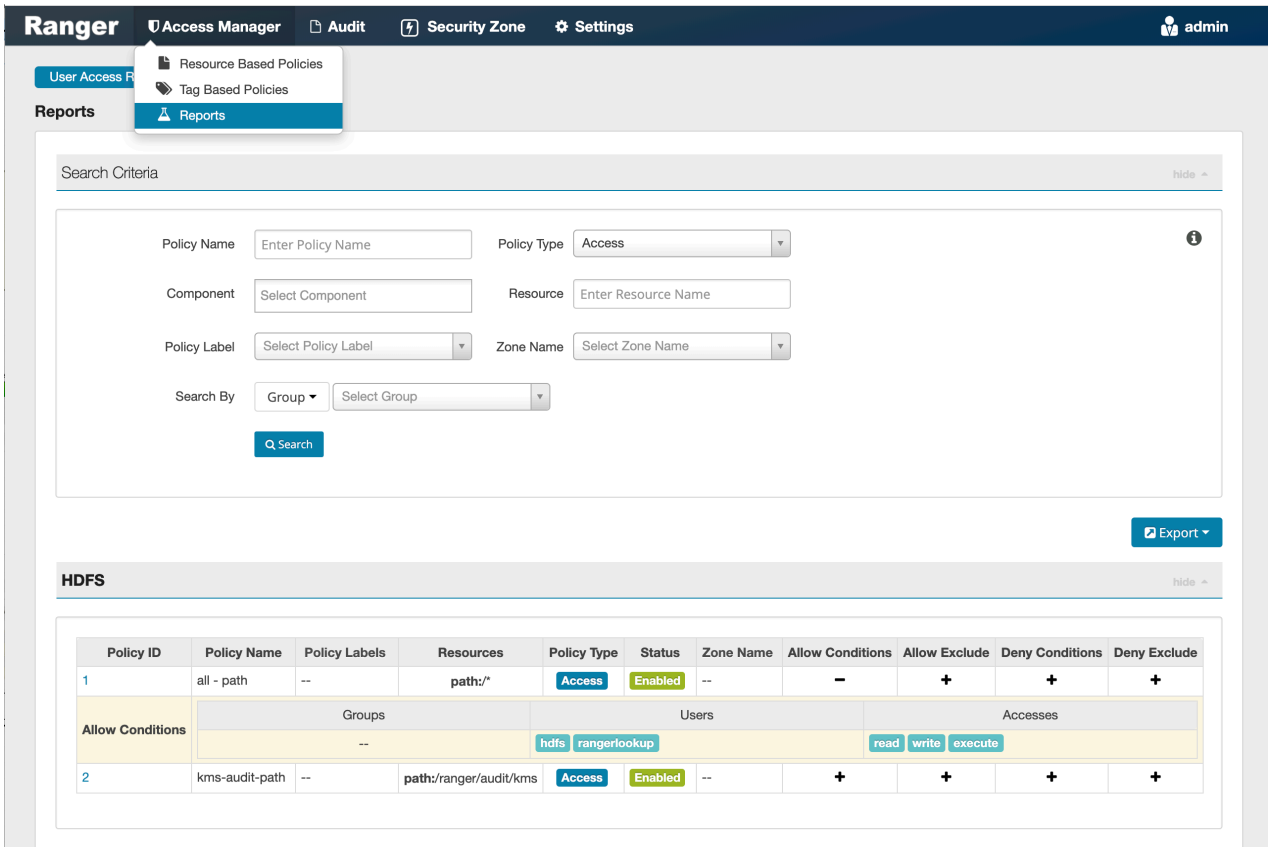


Search Ranger reports

Reference information for searching Ranger reports on one or more policies.

You can search based on:

- Policy Name – The policy name.
- Policy Type – The policy type (Access, Masking, or Row Level Filter).
- Policy Label – The policy label.
- Component – The policy resource or tag component.
- Resource – The resource path used when creating the policy.
- Zone Name – The security zone name.
- Group, Username – The group or user name assigned to the policy.

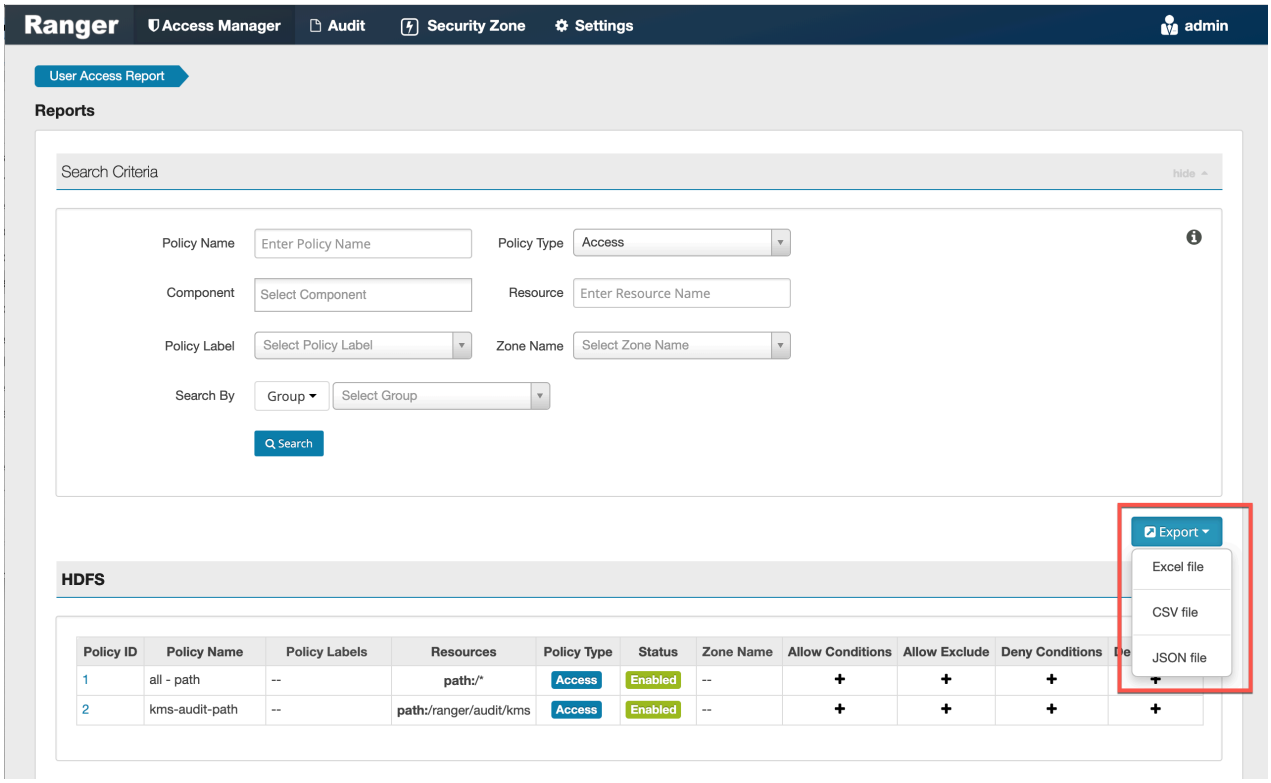


Export Ranger reports

Reference information for exporting Ranger reports on one or more policies.

You can export a list of reports in three file formats:

- CSV file
- Excel file
- JSON



Related Information

Export tag-based policies
Export resource-based policies for a specific service
Export all resource-based policies for all services

Using Ranger client libraries

Ranger now supports clients written in java and python which enable applications to access Ranger REST APIs programmatically. Using client library code simplifies access using java or python, compared with making direct HTTP requests to Ranger REST APIs.

Summary

Ranger client libraries:

- Provide idiomatic, hand-written code in Java and Python, making Ranger REST APIs simple and intuitive to use.
- Handle all low-level details of communication with the server including complexities involved in JSON parsing.
- Support installing the python client using the familiar package management tool pip.

Table 64: Ranger Client Installation Repo and Library Reference Links

Language	Installation	Library Reference
java	github source repository	java library reference
python	github source repository	python library reference

Authentication

The Apache Ranger release 2.2 client supports two authentication types:

- Basic authentication (username/password)
- Kerberos authentication

Java client prompts for the authentication mode to be used at runtime. For Kerberos-based authentications, a principal and keytab file path is required.

SSL

Java and Python clients support SSL/TLS-enabled ranger. To connect to HTTPS ranger using java client, provide the path to the SSL configuration file, as shown in this example:

```
$ ./run-sample-client.sh -n <ranger_admin_url>
SSL Configuration File: /path/to/config.xml
```

Sample SSL configuration file which requires values to be populated:

```
<configuration>
  <property>
    <name>xasecure.policymgr.clientssl.truststore</name>
    <value></value>
  </property>
  <property>
    <name>xasecure.policymgr.clientssl.truststore.credential.file</name>
    <value></value>
  </property>
  <property>
    <name>xasecure.policymgr.clientssl.truststore.type</name>
    <value></value>
  </property>
</configuration>
```

Environment variables

The Java client requires requires that you initialize the following environment variables:

```
$ export JAVA_HOME=/usr/java/<jdk_version>/bin
$ export PATH=$PATH:$JAVA_HOME
$ export HADOOP_CREDSTORE_PASSWORD=<hadoop_credstore_password>
```

Using session cookies to validate Ranger policies

Apache Ranger REST Client uses cookie sessions to download policies, tags and roles from Ranger Admin.

In earlier versions, each Ranger plugin used a kerberos login to request a ticket granting ticket (TGT) from the KDC/AD server in order to download policies, tags and roles. This casued high traffiic levels when mulitple Ranger plugins requested downloads.

Ranger Admin now supports cookie-based sessions. The flag used to enable cookie sessions, ranger.plugin.<service-name>.policy.rest.client.cookie.enabled, where <service-name> is the name of the service for which a Ranger plugin is eanbled, such as hive, solr, or kafka, is set to "enabled" by default.

To check whether the cookie session is used, open the Ranger Admin access.log in the /var/log/ranger/admin folder. Any policy, tag, or role download call to Ranger Admin displays either a 200 or 304 value as response status. A 401 value for response status indicates the call to the KDC server for a TGT for autherntication at service start or when the session cookie expires.

Configure optimized rename and recursive delete operations in Ranger Ozone plugin

You can enable performance optimized authorization approach for rename and recursive delete operations in the Ranger Ozone plugin.

About this task

Ozone introduced support for FSO (FILE_SYSTEM_OPTIMIZED) Bucket layout. FSO Bucket layout is a Hierarchical FileSystem namespace view with directories and files. Similar to HDFS, with FSO bucket layout, Ozone has an efficient directory rename and delete operations. Ranger supports not only authorization for rename and recursive delete operations, but also provides an option to enable performance optimized solution when these operations are performed on directory containing large set of subpaths (directories/files) within it.

Property name - ranger.plugin.ozone.optimized.subaccesspath.enabled

Default is set to false.

To enable authorization for rename and recursive delete operations in the Ranger Ozone plugin:

Procedure

1. In Cloudera Manager Ozone Ozone Manager Configuration Search . type ranger-ozone-security.xml.
2. In Ozone Service Advanced Configuration Snippet (Safety Valve) for ranger-ozone-security.xml, click +.
   a) Under Ozone, in Name, type: ranger.plugin.ozone.optimized.subaccesspath.enabled.
   b) In Value, type: true.
   c) Click Save Changes.
3. In Ozone Actions , click Restart.

Results

Ranger not only authorizes rename and recursive delete operations, but also provides an option to enable performance optimized solution when these operations are performed on a directory containing a large set of subpaths (directories/files) within it.