Cloudera Runtime 7.3.2

Apache Ranger User Management

Date published: 2020-07-28
Date modified: 2026-03-31

https://docs.cloudera.com/

Legal Notice

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

Administering Ranger Users, Groups, Roles, and Permissions

Ranger Admin users can manage users, group, roles and permissions using Ranger Admin Web UI.

Overview: Ranger User/Groups/Roles

Ranger Admin Web UI allows users with Administrator role (permissions) to create new users, groups and roles that define fine-grained access control in Cloudera. This topic presents an overview of the user, group, role, permission management options you can find under Settings.

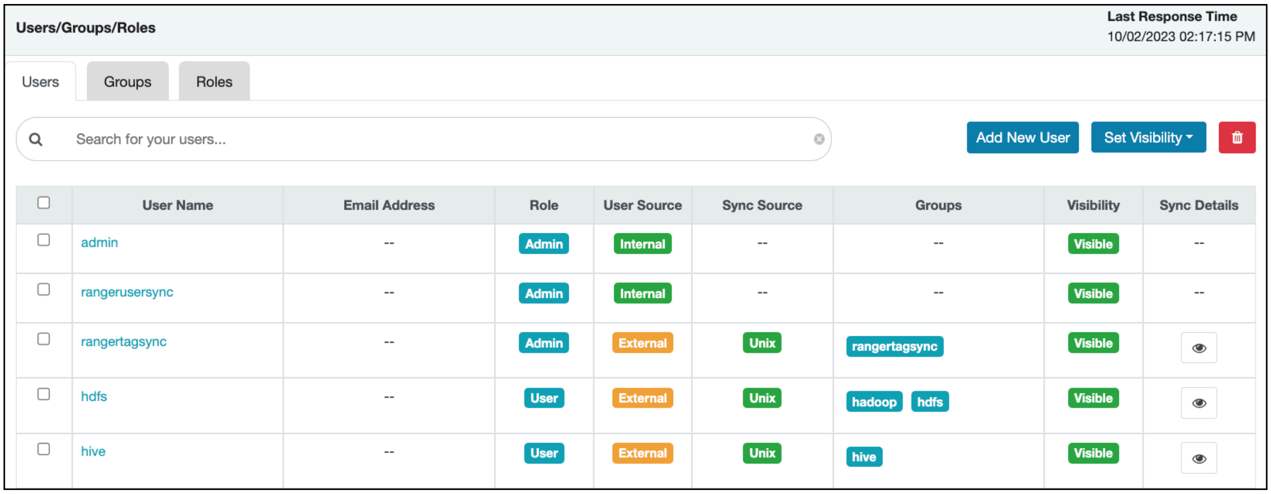
To list the users, groups, and roles for which Ranger manages access to Cloudera services, select Ranger Admin Web UI Settings Users/Groups/Roles .

Users lists:

- Internal users - created by a Ranger admin user who can log in to the Ranger Admin Web UI.
- External users - created at other systems such as Active Directory, LDAP, or UNIX.
- Admin users - who are the only users with permission to create users and services, run reports, and perform other administrative tasks using Ranger Admin Web UI.
- Visible users - those users created in Ranger Admin Web UI, or in other systems who are "active", in other words, not marked for deletion.
- Hidden users - those users that have been marked for deletion for any reason (for example invalid characters, duplicates, or obsolescence).

Users also shows the Groups to which each user belongs.

The following example shows internal, external, and Admin users listed on Service Manager Users :



Users/Groups/Roles

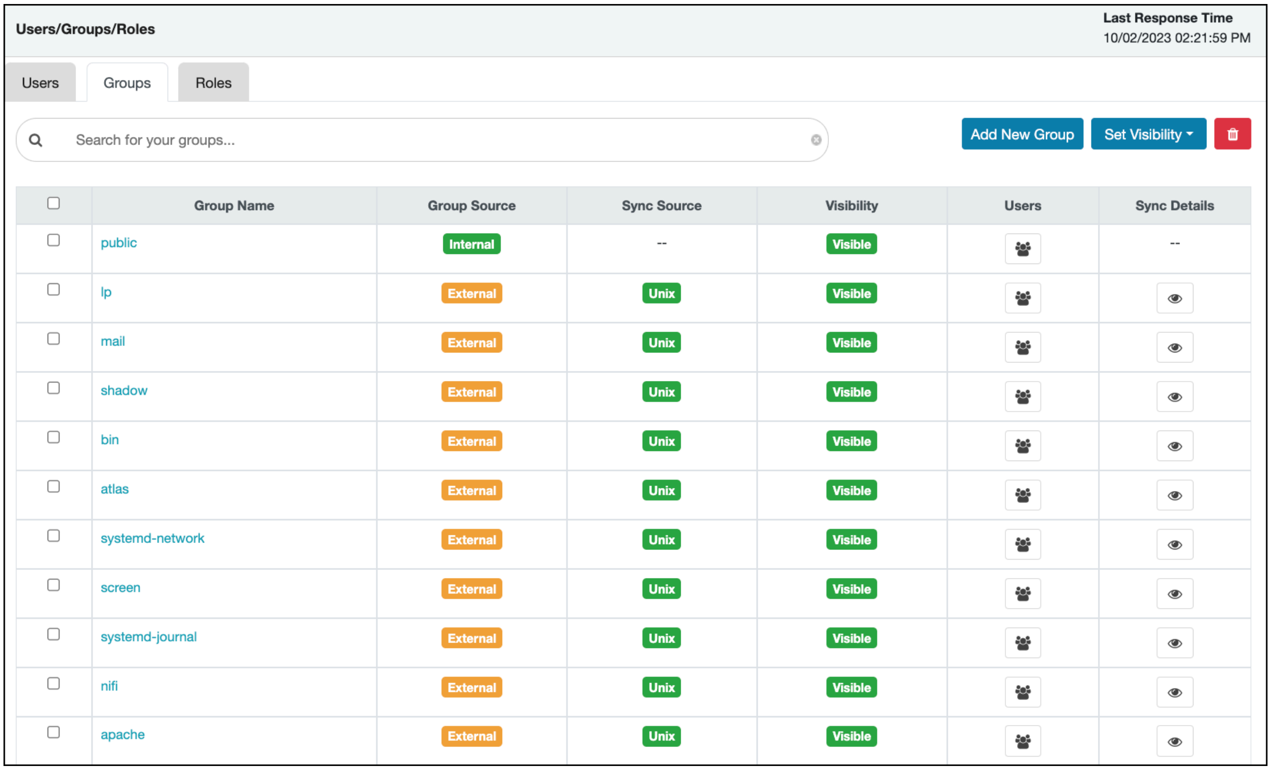
Last Response Time
10/02/2023 02:17:15 PM

□	User Name	Email Address	Role	User Source	Sync Source	Groups	Visibility	Sync Details
□	admin	--	Admin	Internal	--	--	Visible	--
□	rangerusersync	--	Admin	Internal	--	--	Visible	--
□	rangertagsync	--	Admin	External	Unix	rangertagsync	Visible	👁
□	hdfs	--	User	External	Unix	hadoop hdfs	Visible	👁
□	hive	--	User	External	Unix	hive	Visible	👁

Groups lists:

- Internal groups - created by a Ranger admin.
- External groups - created by other systems.
- On Groups, you can click Users to view the members of a specific group.

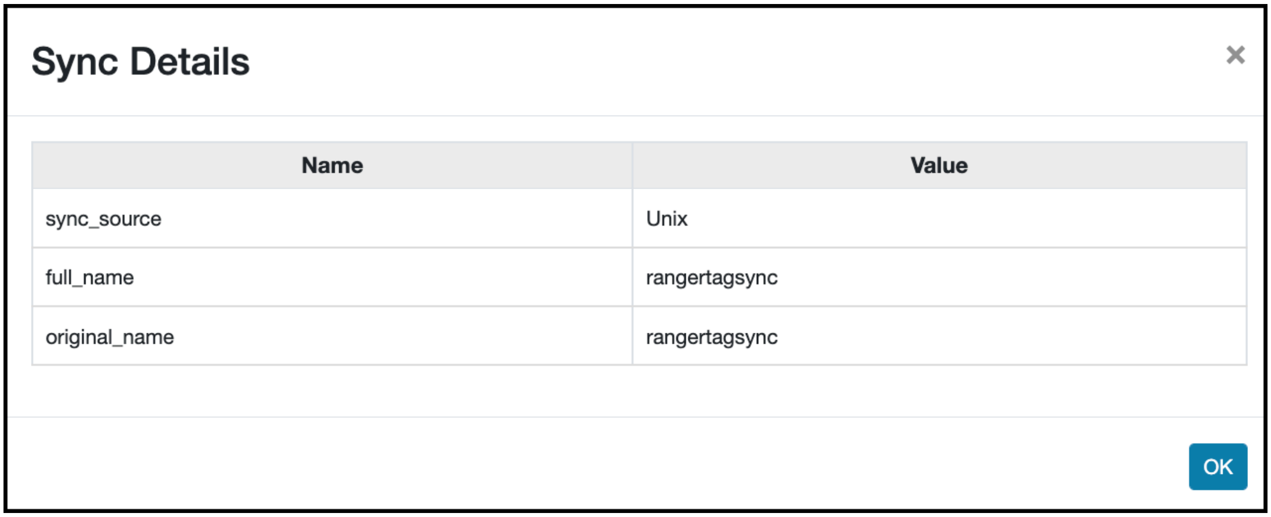
The following figure shows internal and external groups listed on Groups:



□	Group Name	Group Source	Sync Source	Visibility	Users	Sync Details
□	public	Internal	--	Visible	👥	--
□	lp	External	Unix	Visible	👥	👁
□	mail	External	Unix	Visible	👥	👁
□	shadow	External	Unix	Visible	👥	👁
□	bin	External	Unix	Visible	👥	👁
□	atlas	External	Unix	Visible	👥	👁
□	systemd-network	External	Unix	Visible	👥	👁
□	screen	External	Unix	Visible	👥	👁
□	systemd-journal	External	Unix	Visible	👥	👁
□	nifi	External	Unix	Visible	👥	👁
□	apache	External	Unix	Visible	👥	👁

The Users and Groups pages also lists a Sync Source for each user and group. To filter Users and Groups by sync source type, select Sync Source as a search filter, then enter a sync source type, such as Unix or LDAP/AD. To view more information about the sync source, click Sync Details for a user or group.

The following example shows the sync details for the rangertagsync user.



Name	Value
sync_source	Unix
full_name	rangertagsync
original_name	rangertagsync

Roles lists:

- Role names, and related mappings to:
- User names
- Group names
- Other role names

Adding a user

How a Ranger Admin user can add new Ranger users.

About this task

Only a Ranger Admin user can create other Admin users, service users, or Auditor users, based on the full permissions to configure Ranger Admin Web UI.

Admin users can create the following user types:

admin

auditor

keyadmin

user

A Ranger Admin user can also import/ export policies for services (for example hdfs, hive, atlas, etc.) other than kms.

KeyAdmin users cannot create users, but can:

Import kms policies

Export kms policies

access Key Manager module functionality with full permissions.

Auditor users cannot create users, but can:

access Audit Manager module functionality with full permissions.

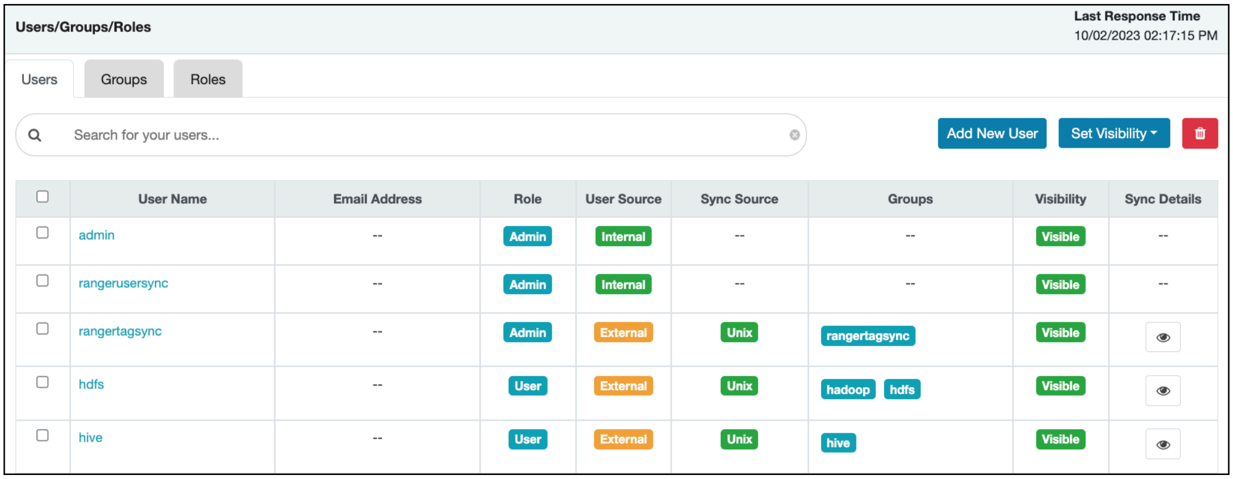
This topic presents the example of logging in to Ranger Admin Web UI, using admin credentials and then creating a new user with Auditor role.

Procedure

1. Log in to Ranger Admin Web UI, using administrator credentials.
2. In the Ranger Admin Web UI, select Settings Users .

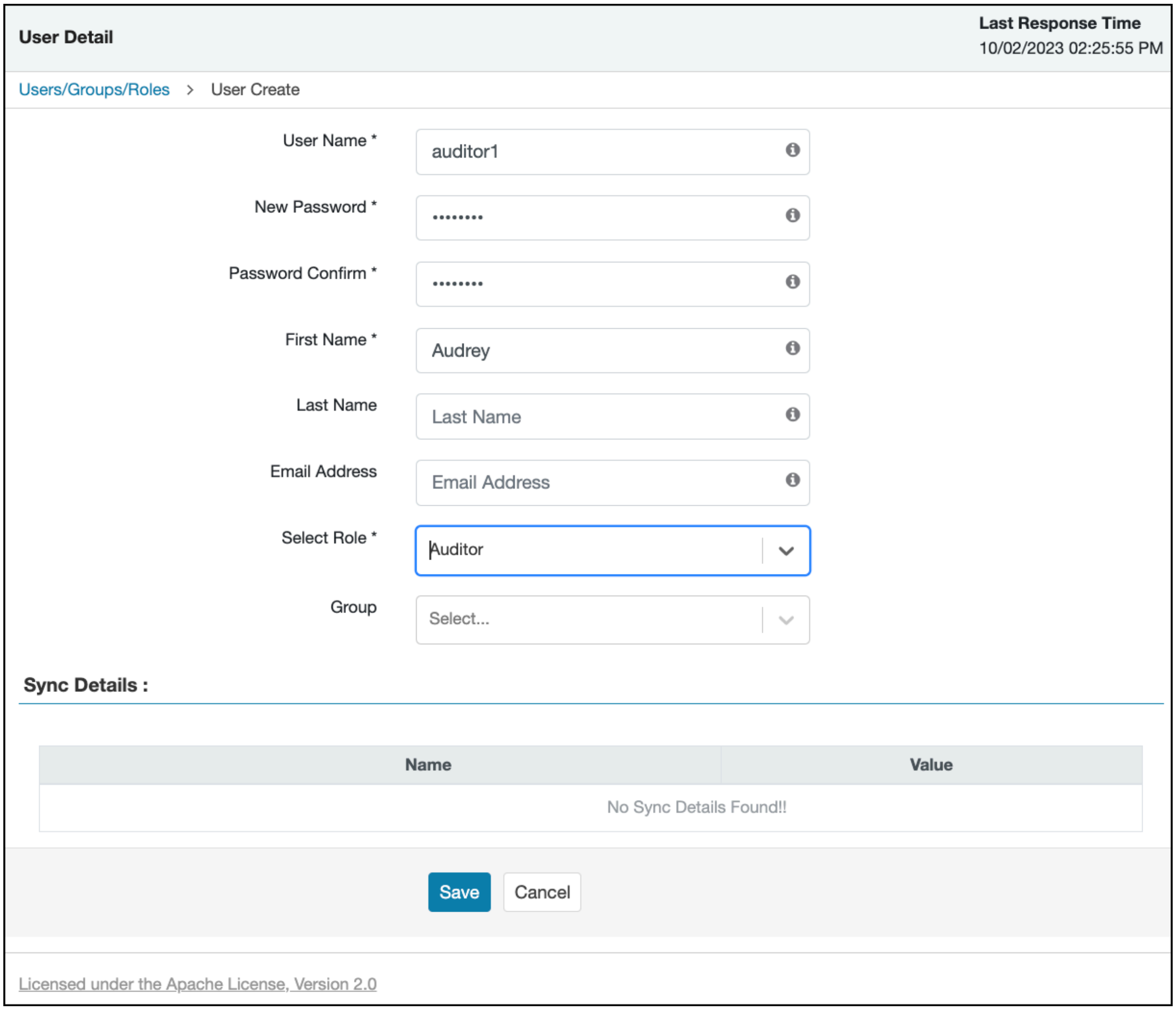
   Users/Groups/Roles displays the Users tab active.

   Note: Users displays the Add New Users option only to users logged in with admin permissions.



□	User Name	Email Address	Role	User Source	Sync Source	Groups	Visibility	Sync Details
□	admin	--	Admin	Internal	--	--	Visible	--
□	rangerusersync	--	Admin	Internal	--	--	Visible	--
□	rangertagsync	--	Admin	External	Unix	rangertagsync	Visible	👁
□	hdfs	--	User	External	Unix	hadoop hdfs	Visible	👁
□	hive	--	User	External	Unix	hive	Visible	👁

3. Click Add New User .
   User Detail displays User Create options.



User Detail

Last Response Time
10/02/2023 02:25:55 PM

Users/Groups/Roles > User Create

User Name *	auditor1
New Password *	••••••••
Password Confirm *	••••••••
First Name *	Audrey
Last Name	Last Name
Email Address	Email Address
Select Role *	Auditor
Group	Select...

Sync Details :

Name	Value
No Sync Details Found!!	

4. On User Create, enter information in ALL REQUIRED (*) fields, then click Save.

   In this example, Auditor has been selected in Select Role. No group has been selected, nor has the password been confirmed.

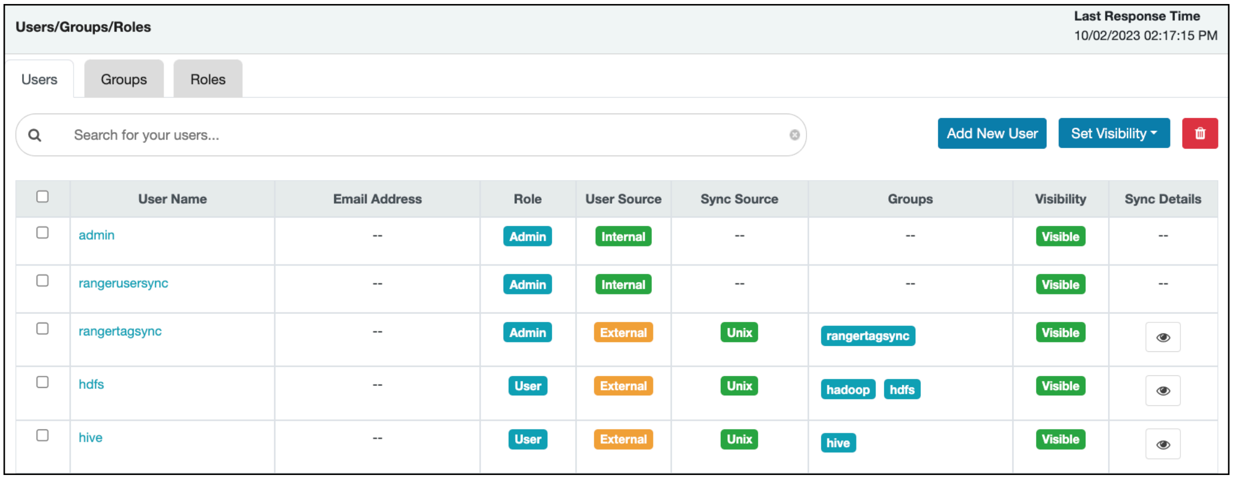
   After completing all required fields, clicking Save immediately creates the user and adds the user to any selected groups.

Editing a user

How to edit a user in Ranger.

Procedure

1. In the Ranger Admin Web UI, select Settings Users .
   Users/Groups/Roles displays the Users tab active.

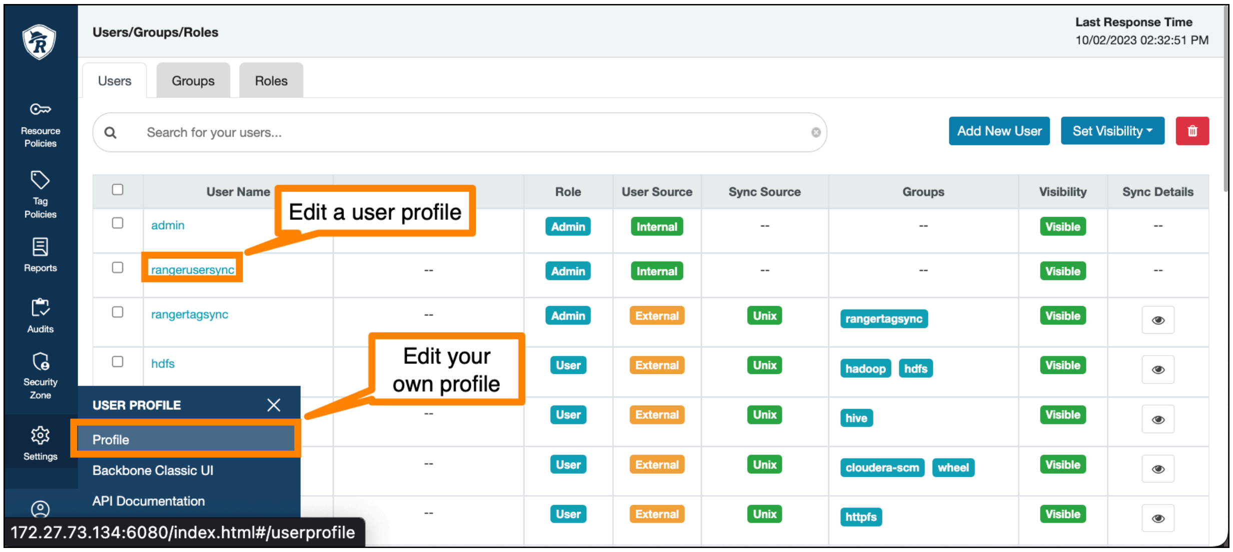


Users/Groups/Roles

Last Response Time
10/02/2023 02:17:15 PM

□	User Name	Email Address	Role	User Source	Sync Source	Groups	Visibility	Sync Details
□	admin	--	Admin	Internal	--	--	Visible	--
□	rangerusersync	--	Admin	Internal	--	--	Visible	--
□	rangertagsync	--	Admin	External	Unix	rangertagsync	Visible	👁
□	hdfs	--	User	External	Unix	hadoop hdfs	Visible	👁
□	hive	--	User	External	Unix	hive	Visible	👁

2. Select a user profile to edit. To edit your own profile, select your user name, then click Profile.



Users/Groups/Roles

Last Response Time
10/02/2023 02:32:51 PM

Search for your users...

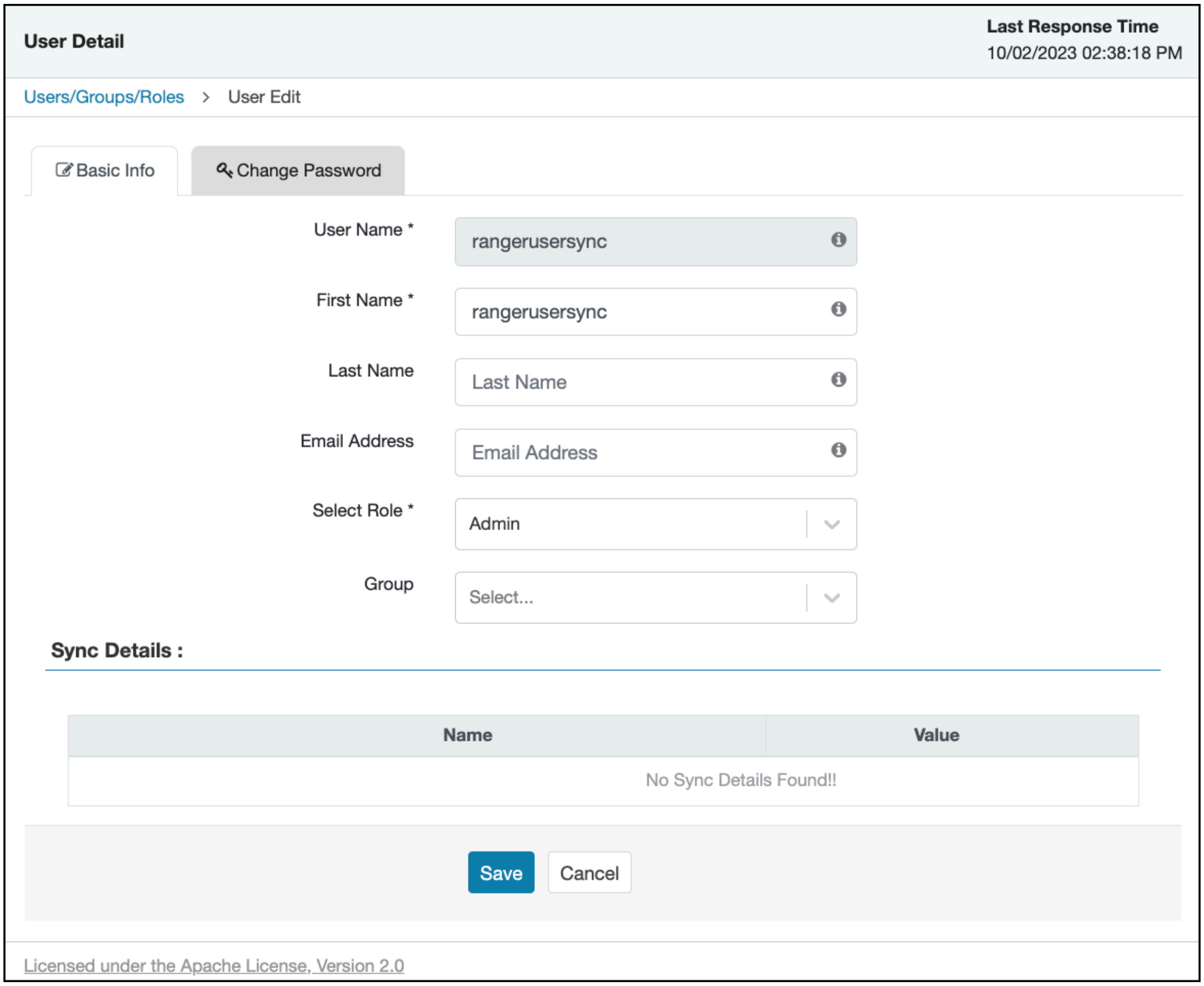
☐	User Name		Role	User Source	Sync Source	Groups	Visibility	Sync Details
☐	admin		Admin	Internal	--	--	Visible	--
☐	rangerusersync	--	Admin	Internal	--	--	Visible	--
☐	rangertagsync	--	Admin	External	Unix	rangertagsync	Visible	👁
☐	hdfs		User	External	Unix	hadoop hdfs	Visible	👁
		--	User	External	Unix	hive	Visible	👁
		--	User	External	Unix	cloudera-scm wheel	Visible	👁
		--	User	External	Unix	httpfs	Visible	👁

Edit a user profile

Edit your own profile

172.27.73.134:6080/index.html#/userprofile

The User Detail page appears.



User Detail

Last Response Time
10/02/2023 02:38:18 PM

Users/Groups/Roles > User Edit

User Name *	rangerusersync
First Name *	rangerusersync
Last Name	Last Name
Email Address	Email Address
Select Role *	Admin
Group	Select...

Sync Details :

Name	Value
No Sync Details Found!!	

Licensed under the Apache License, Version 2.0

Note:

You can only fully edit internal users. For external users, you can only edit the user role.

3. Edit the user details, then click Save.

Related Tasks

Deleting a group

Deleting a user

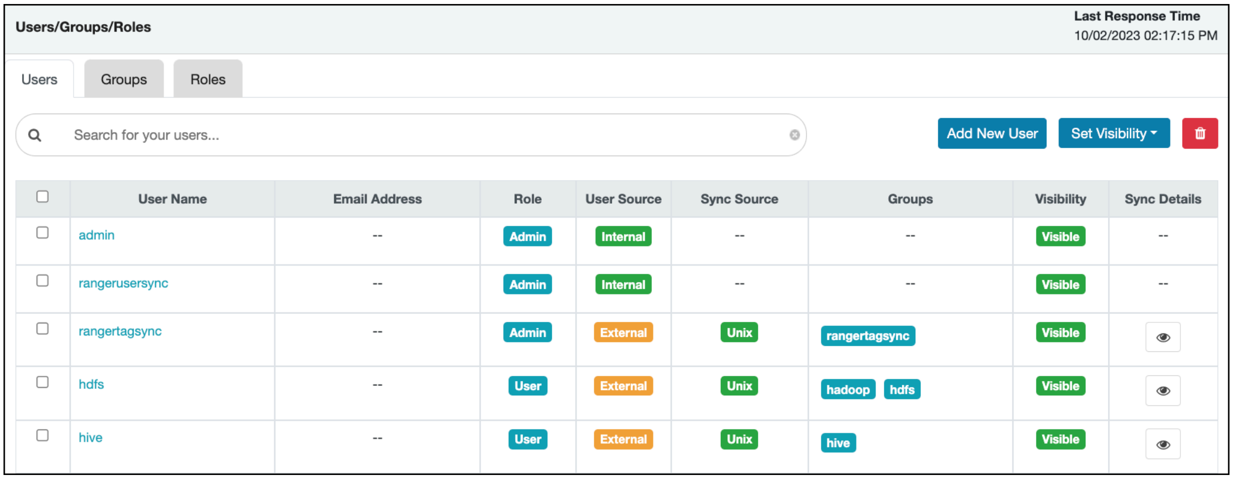
How to delete a user in Ranger.

Before you begin

Only users with the "admin" role can delete a user.

Procedure

1. In the Ranger Admin Web UI, select Settings Users .
   Users/Groups/Roles displays the Users tab active.

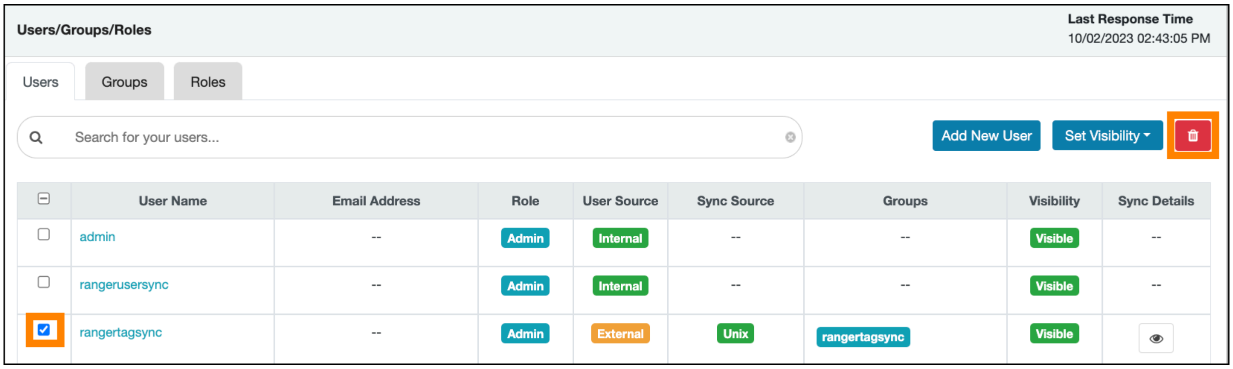


Users/Groups/Roles

Last Response Time
10/02/2023 02:17:15 PM

□	User Name	Email Address	Role	User Source	Sync Source	Groups	Visibility	Sync Details
□	admin	--	Admin	Internal	--	--	Visible	--
□	rangerusersync	--	Admin	Internal	--	--	Visible	--
□	rangertagsync	--	Admin	External	Unix	rangertagsync	Visible	👁
□	hdfs	--	User	External	Unix	hadoop hdfs	Visible	👁
□	hive	--	User	External	Unix	hive	Visible	👁

2. Select the check box of the user you want to delete, then click Delete.



Users/Groups/Roles

Last Response Time
10/02/2023 02:43:05 PM

⊟	User Name	Email Address	Role	User Source	Sync Source	Groups	Visibility	Sync Details
□	admin	--	Admin	Internal	--	--	Visible	--
□	rangerusersync	--	Admin	Internal	--	--	Visible	--
☑	rangertagsync	--	Admin	External	Unix	rangertagsync	Visible	👁

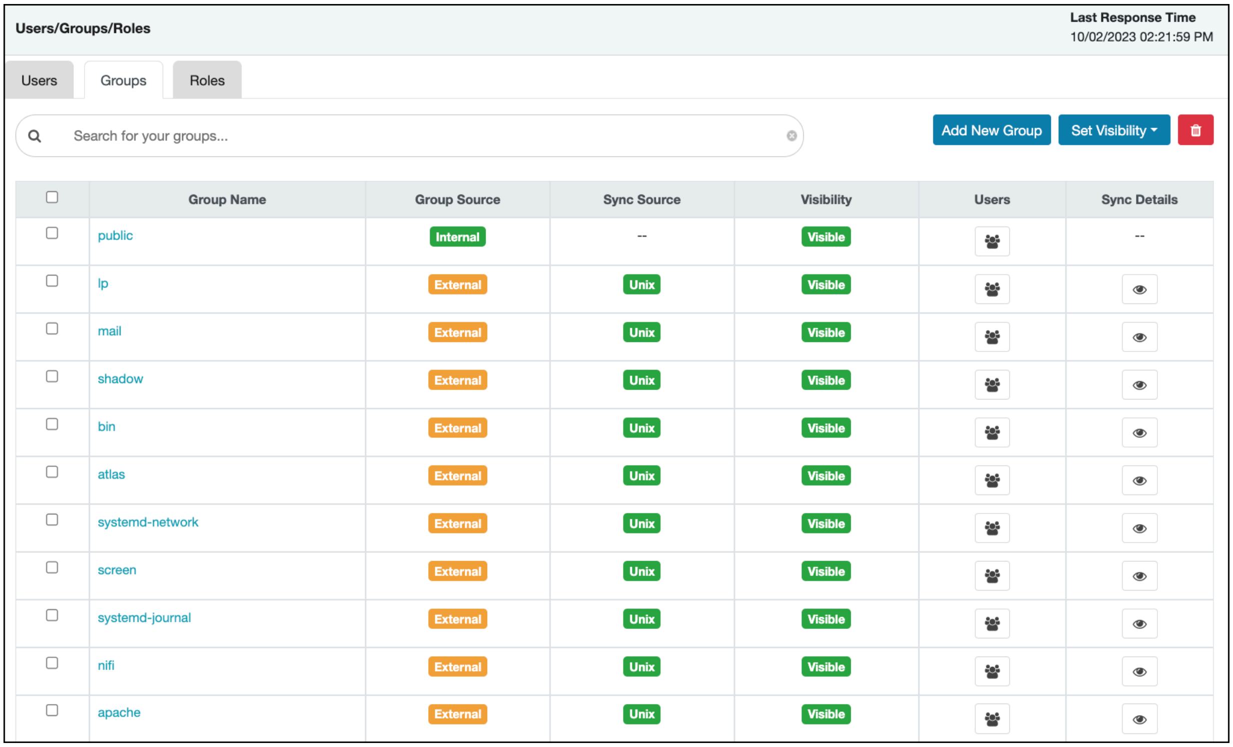
3. Click OK on the confirmation pop-up.

Adding a group

How to add a group in Ranger.

Procedure

1. In the Ranger Admin Web UI, select Settings Groups .
   Users/Groups/Roles displays the Groups tab active.



Users/Groups/Roles

Last Response Time
10/02/2023 02:21:59 PM

Users | Groups | Roles

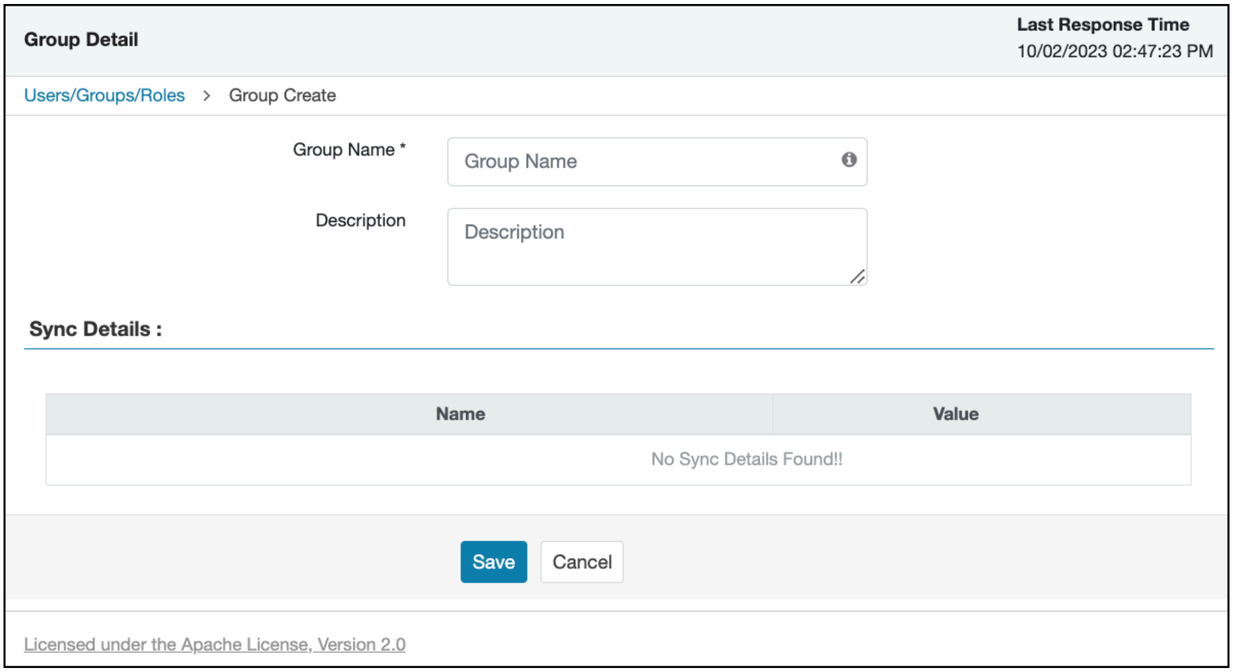
Search for your groups...

Add New Group | Set Visibility | 🗑

□	Group Name	Group Source	Sync Source	Visibility	Users	Sync Details
□	public	Internal	--	Visible	👥	--
□	lp	External	Unix	Visible	👥	👁
□	mail	External	Unix	Visible	👥	👁
□	shadow	External	Unix	Visible	👥	👁
□	bin	External	Unix	Visible	👥	👁
□	atlas	External	Unix	Visible	👥	👁
□	systemd-network	External	Unix	Visible	👥	👁
□	screen	External	Unix	Visible	👥	👁
□	systemd-journal	External	Unix	Visible	👥	👁
□	nifi	External	Unix	Visible	👥	👁
□	apache	External	Unix	Visible	👥	👁

2. Click Add New Group.

   The Group Create page appears.



Group Detail

Last Response Time
10/02/2023 02:47:23 PM

Users/Groups/Roles > Group Create

Group Name * : Group Name

Description : Description

Sync Details :

Name	Value
No Sync Details Found!!	

Save | Cancel

Licensed under the Apache License, Version 2.0

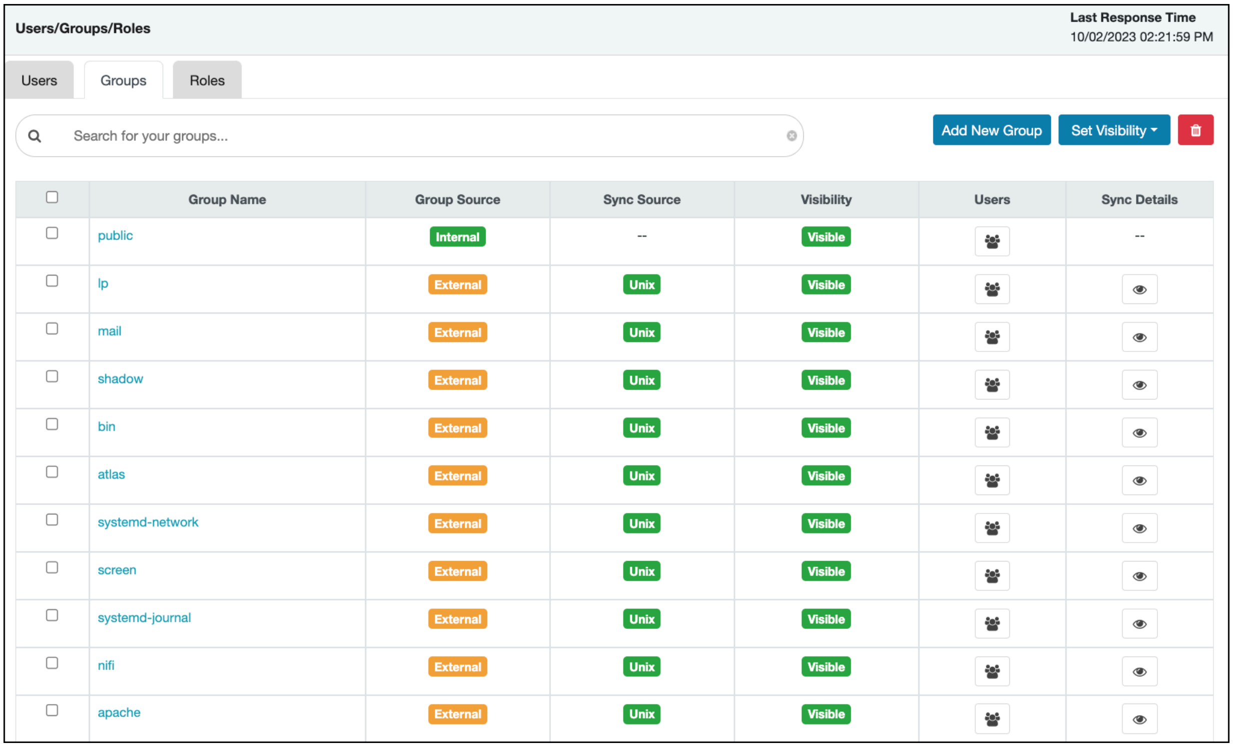
3. Enter a unique name for the group (required) and description (optional), then click Save.

Editing a group

How to edit a group in Ranger.

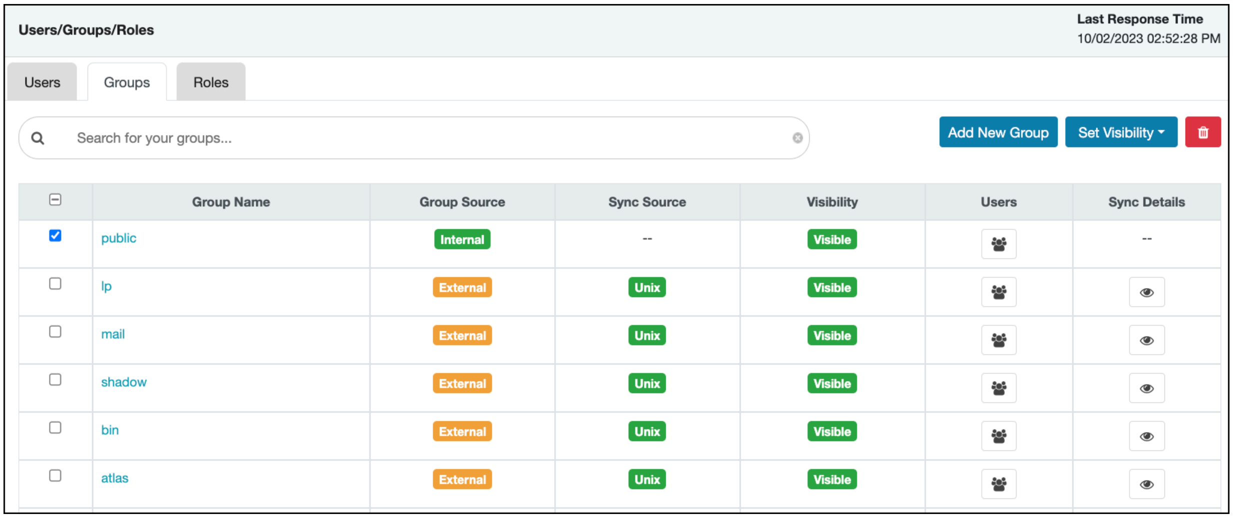
Procedure

1. In the Ranger Admin Web UI, select Settings Groups .
   Users/Groups/Roles displays the Groups tab active.



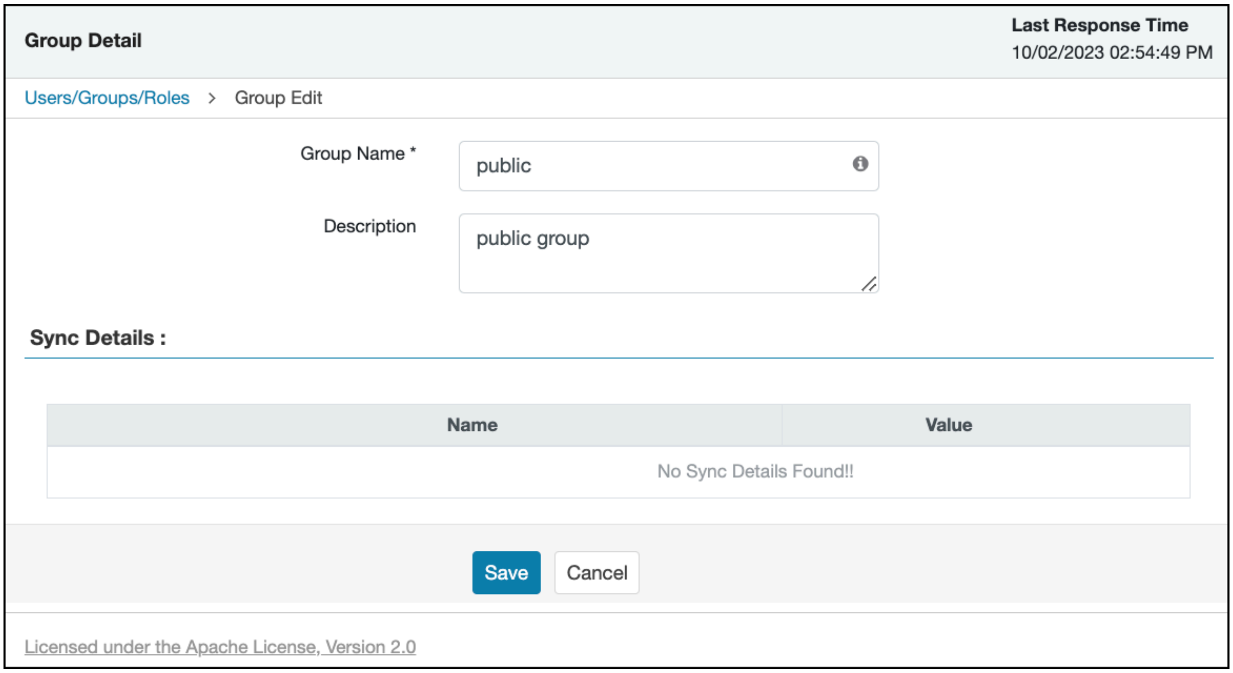
□	Group Name	Group Source	Sync Source	Visibility	Users	Sync Details
□	public	Internal	--	Visible	👥	--
□	lp	External	Unix	Visible	👥	👁
□	mail	External	Unix	Visible	👥	👁
□	shadow	External	Unix	Visible	👥	👁
□	bin	External	Unix	Visible	👥	👁
□	atlas	External	Unix	Visible	👥	👁
□	systemd-network	External	Unix	Visible	👥	👁
□	screen	External	Unix	Visible	👥	👁
□	systemd-journal	External	Unix	Visible	👥	👁
□	nifi	External	Unix	Visible	👥	👁
□	apache	External	Unix	Visible	👥	👁

2. Select a group name to edit, then double-click the group name.



□	Group Name	Group Source	Sync Source	Visibility	Users	Sync Details
☑	public	Internal	--	Visible	👥	--
□	lp	External	Unix	Visible	👥	👁
□	mail	External	Unix	Visible	👥	👁
□	shadow	External	Unix	Visible	👥	👁
□	bin	External	Unix	Visible	👥	👁
□	atlas	External	Unix	Visible	👥	👁

3. The Group Edit page appears.



Group Detail
Last Response Time
10/02/2023 02:54:49 PM
Users/Groups/Roles > Group Edit

Group Name * public

Description public group

Sync Details :

Name	Value
No Sync Details Found!!	

Save
Cancel
Licensed under the Apache License, Version 2.0

4. Edit the group details, then click Save.

Deleting a group

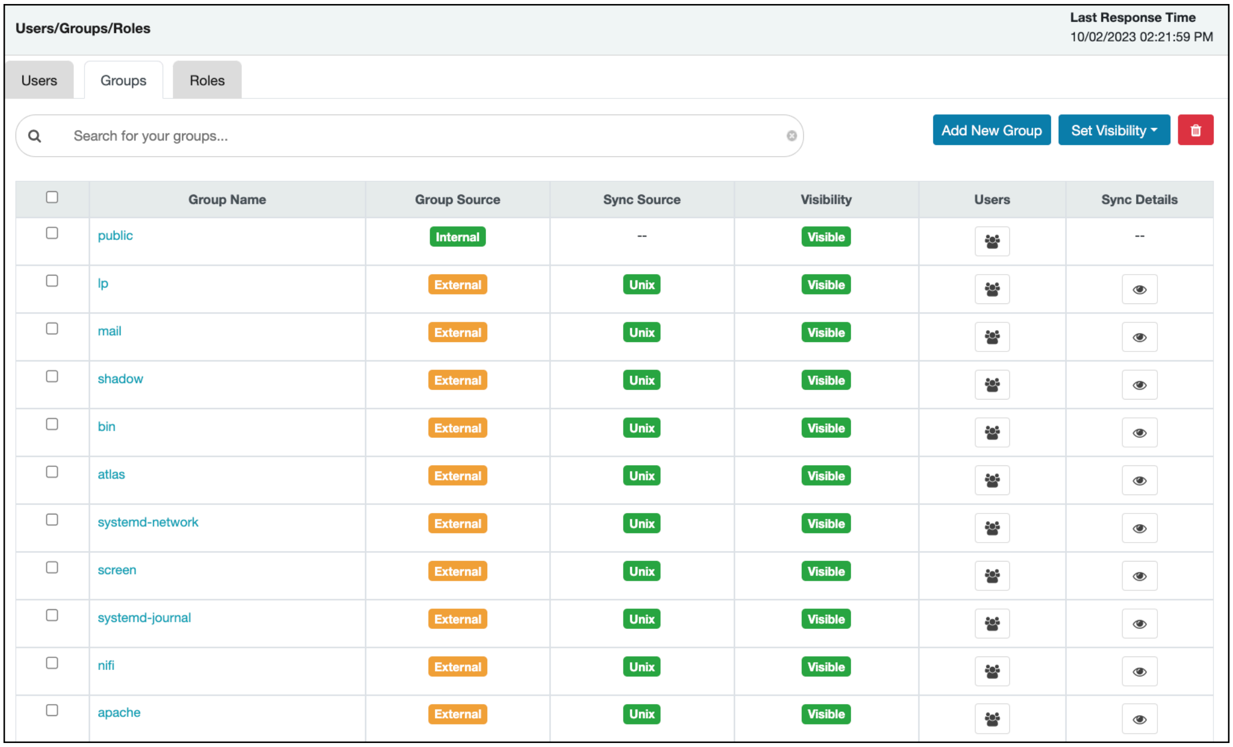
How to delete a group in Ranger.

Before you begin

Only users with the "admin" role can delete a group.

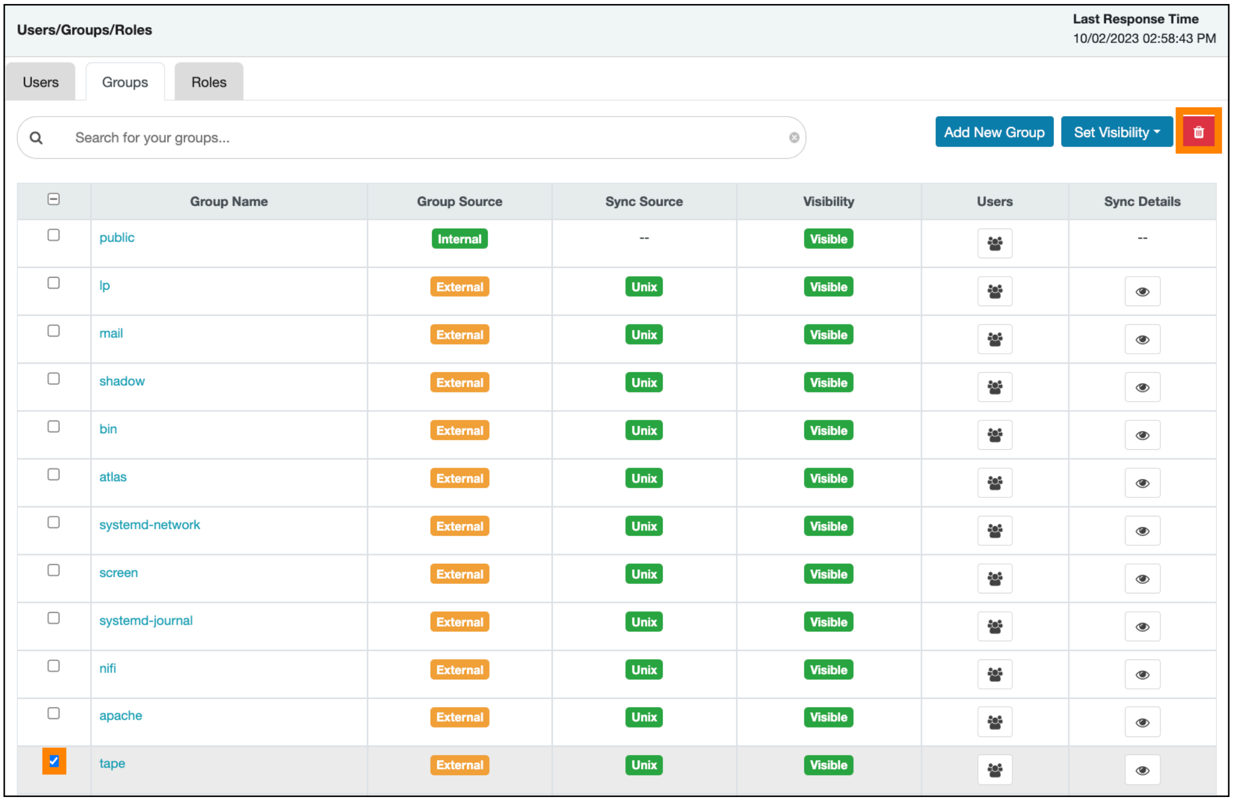
Procedure

1. In the Ranger Admin Web UI, select Settings Groups .
   Users/Groups/Roles displays the Groups tab active.



□	Group Name	Group Source	Sync Source	Visibility	Users	Sync Details
□	public	Internal	--	Visible	👥	--
□	lp	External	Unix	Visible	👥	👁
□	mail	External	Unix	Visible	👥	👁
□	shadow	External	Unix	Visible	👥	👁
□	bin	External	Unix	Visible	👥	👁
□	atlas	External	Unix	Visible	👥	👁
□	systemd-network	External	Unix	Visible	👥	👁
□	screen	External	Unix	Visible	👥	👁
□	systemd-journal	External	Unix	Visible	👥	👁
□	nifi	External	Unix	Visible	👥	👁
□	apache	External	Unix	Visible	👥	👁

2. Select the check box of the group you want to delete, then click Delete.



⊟	Group Name	Group Source	Sync Source	Visibility	Users	Sync Details
□	public	Internal	--	Visible	👥	--
□	lp	External	Unix	Visible	👥	👁
□	mail	External	Unix	Visible	👥	👁
□	shadow	External	Unix	Visible	👥	👁
□	bin	External	Unix	Visible	👥	👁
□	atlas	External	Unix	Visible	👥	👁
□	systemd-network	External	Unix	Visible	👥	👁
□	screen	External	Unix	Visible	👥	👁
□	systemd-journal	External	Unix	Visible	👥	👁
□	nifi	External	Unix	Visible	👥	👁
□	apache	External	Unix	Visible	👥	👁
☑	tape	External	Unix	Visible	👥	👁

3. Click OK on the confirmation pop-up.

What to do next

Users in a deleted group will be reassigned to no group. You can edit these users and reassign them to other groups.

Related Tasks

Editing a user

Adding a role through Ranger

How to add a role in Ranger.

About this task

A role contains a set of users, groups, or other roles. You assign a role by adding a user, group or role to it. By adding multiple roles, you create a role hierarchy in which you manage permission sets at the role level.

Benefits that roles provide in a large environment:

- A role may include many users or groups, all of which may be updated using a single command.
- Adding or revoking a single permission to or from a role requires a single command, which also applies to all users and groups with that role.
- Roles allow for some documentation about why a permission is granted or revoked.

Conceptually, a role functions as a collection. A group is a collection of users. You create a role and add users to it. Then, you grant that role to a group. Roles present an easier way to manage users and groups, based on specific access criteria.

A simple example of a role hierarchy follows:

- FinReadOnly role, which gives read permission on all tables in the Finance database and is defined by a Ranger policy that grants read on database:Finance, table:* to the FinReadOnly role.
- FinWrite role, which gives write permission on all tables in the Finance database and is defined by a Ranger policy that grants write on database:Finance, table:* to the FinWrite role.
- FinReadWrite role, which role is granted both the FinRead and FinWrite roles and thereby inherits read and write permission to all tables in the Finance database.
- FinReporting group whose users require only read permission to the Finance tables. FinReporting group is added to FinReadOnly role in Ranger.
- FinDataPrep group whose users require only write permission to the Finance tables. FinDataPrep group is added to the FinWrite role in Ranger.
- FinPowerUser group whose users require read and write permission to all Finance tables. FinPowerUsers group is added to the FinReadWrite role in Ranger.

You can create a role either through Ranger, or through Hive.

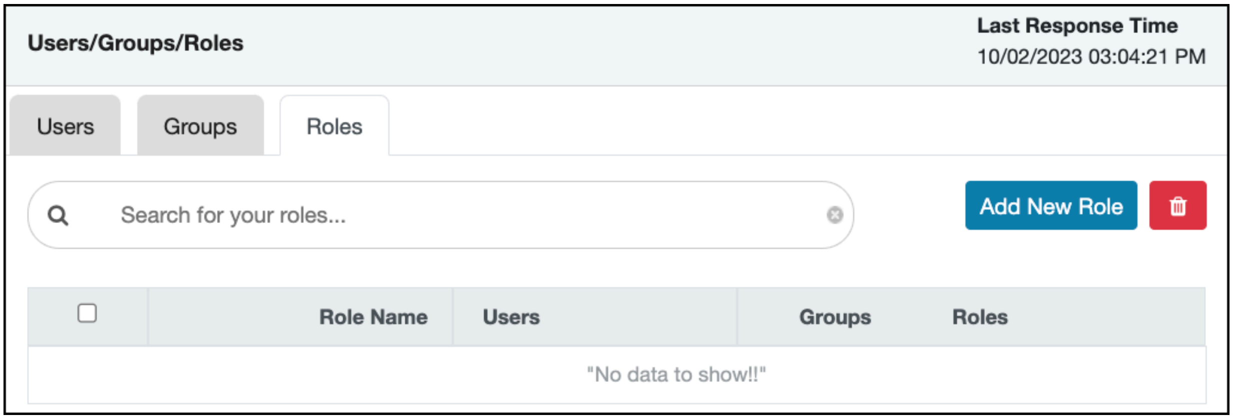
Before you begin

To add a role, the user must have Admin_Role privilege in Ranger.

Procedure

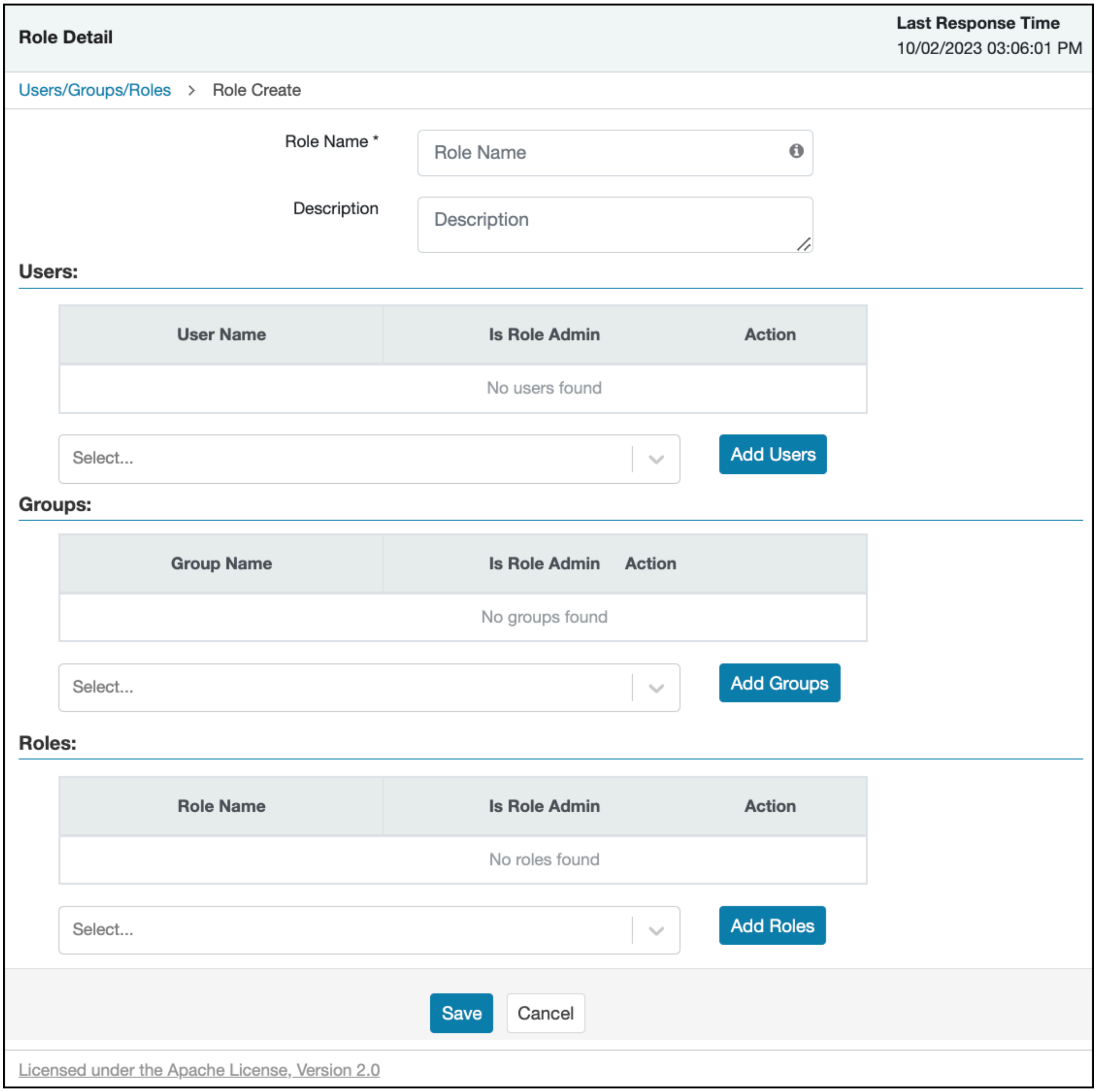
To create a role through Ranger, using Ranger Admin Web UI:

1. Select Settings Roles .
   Users/Groups/Roles displays the Roles tab active.



☐	Role Name	Users	Groups	Roles
"No data to show!!"				

2. Click Add New Role.
   The Role Detail page displays Role Create options.



Users:

User Name	Is Role Admin	Action
No users found		

Groups:

Group Name	Is Role Admin	Action
No groups found		

Roles:

Role Name	Is Role Admin	Action
No roles found		

3. Enter a unique name for the role. Optionally, add users, groups and/or roles to be associated with the role, then click Save.

Adding a role through Hive

How to add a role in Hive.

About this task

You can create a role either through Ranger, or through Hive.

Before you begin

To add a role through Hive, the user must have Admin_Role privilege in Ranger.

Procedure

In Hive:

1. Log in as a user with Admin_Role privilege.

2. Type the following command:

```
CREATE ROLE external_hr_role_01;
```

Any user with Is_Role_Admin privilege has the ability to assign the role to other users in Hive.

For example, to grant this new role to the user hr_user01, type:

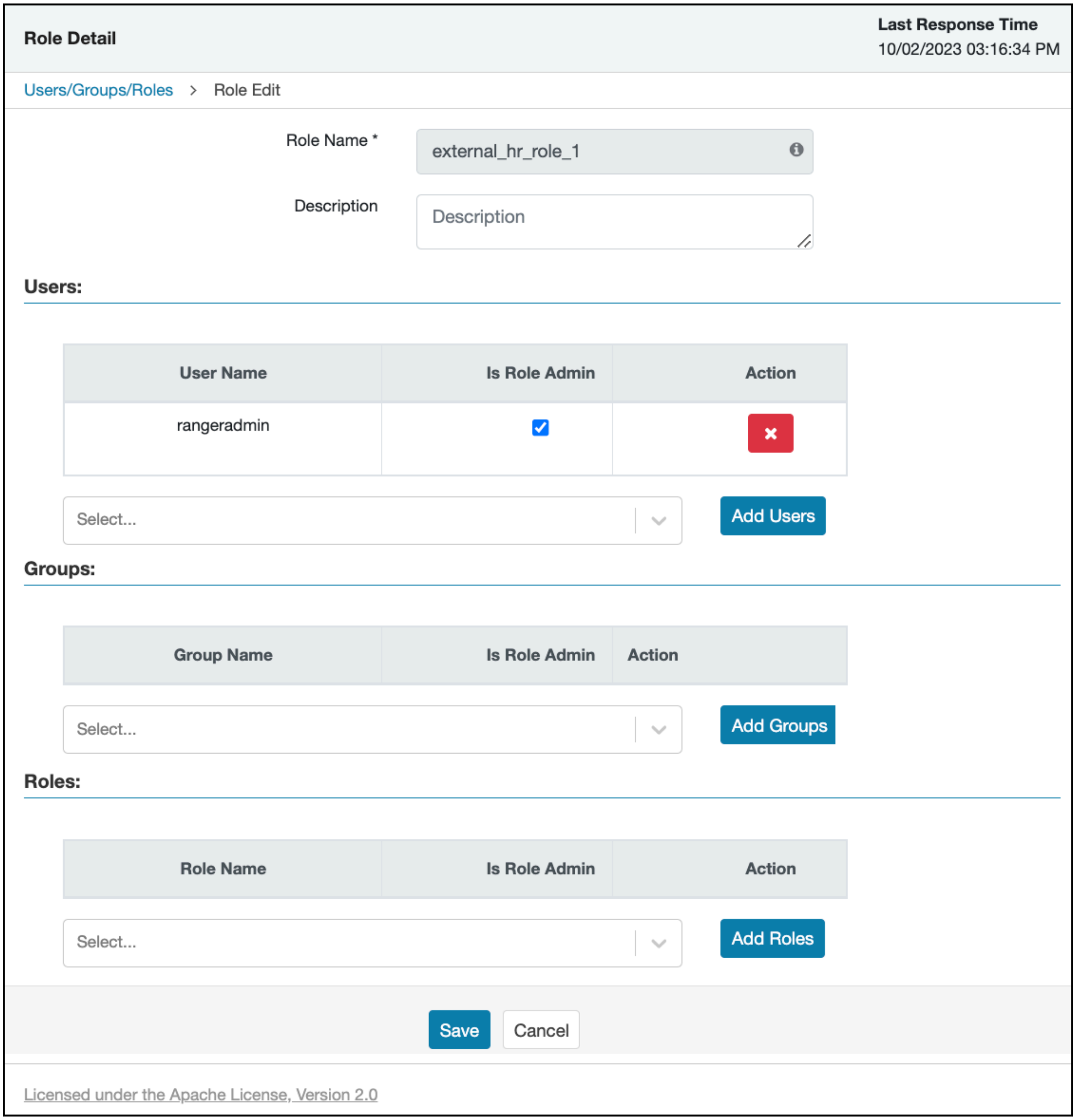
```
GRANT ROLE external_hr_role_01 TO USER hr_user01;
```

hr_user01 appears in Ranger having the external_hr_role_01 role.

You can also grant Is_Role_Admin privilege to a specific user by typing:

```
GRANT ROLE external_hr_role_01 TO USER hr_user02 WITH ADMIN OPTION;
```

The role you create appears in Ranger and is recognized by Hive. The user that creates the role adds automatically to the list of users having that role. The added user has the Is_Role_Admin privilege, as shown in Ranger:



Role Detail

Last Response Time
10/02/2023 03:16:34 PM

Users/Groups/Roles > Role Edit

Role Name *: external_hr_role_1

Description: Description

Users:

User Name	Is Role Admin	Action
rangeradmin	☑	✖

Groups:

Group Name	Is Role Admin	Action

Roles:

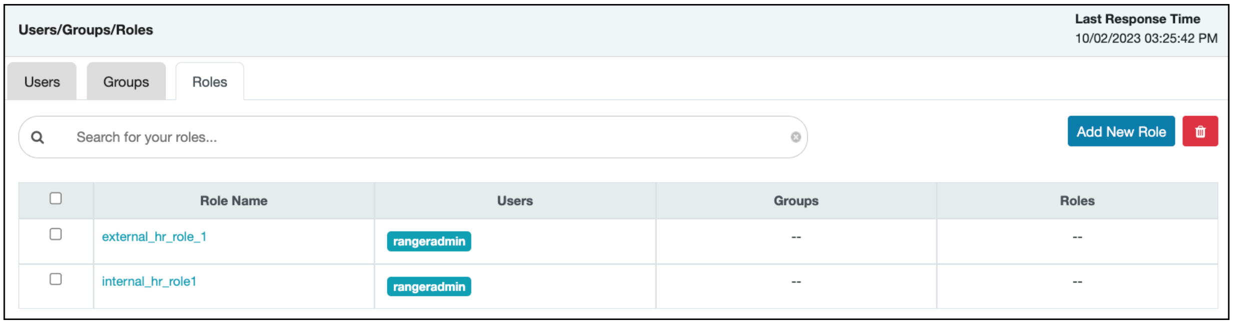
Role Name	Is Role Admin	Action

Editing a role

How to edit a role in Ranger.

Procedure

1. Select Settings Roles .
2. Click the Roles tab.
   Users/Groups/Roles displays the Roles tab active.

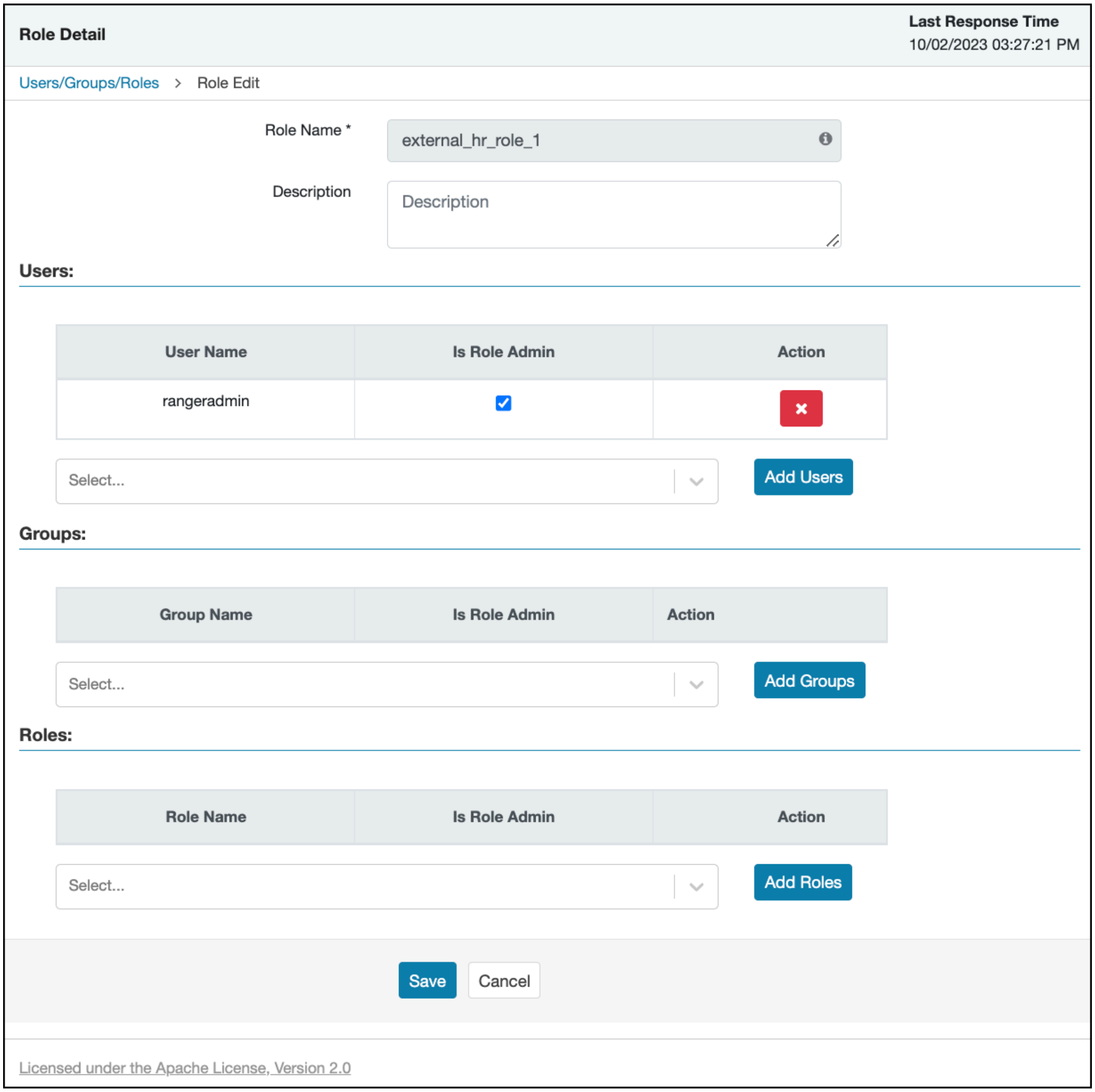


Users/Groups/Roles

Last Response Time
10/02/2023 03:25:42 PM

□	Role Name	Users	Groups	Roles
□	external_hr_role_1	rangeradmin	--	--
□	internal_hr_role1	rangeradmin	--	--

3. Click the role name to edit.
   The selected role opens for editing in Role Detail.



Role Detail

Last Response Time
10/02/2023 03:27:21 PM

Users/Groups/Roles > Role Edit

Role Name * external_hr_role_1

Description

Users:

User Name	Is Role Admin	Action
rangeradmin	☑	✖

Groups:

Group Name	Is Role Admin	Action

Roles:

Role Name	Is Role Admin	Action

Licensed under the Apache License, Version 2.0

4. Add users, groups and roles to the existing role, then click Save.
   If the role was created in Hive, you can add other users in Hive using the GRANT command:

```
GRANT ROLE external_hr_role_01 TO USER hr_user02;
```

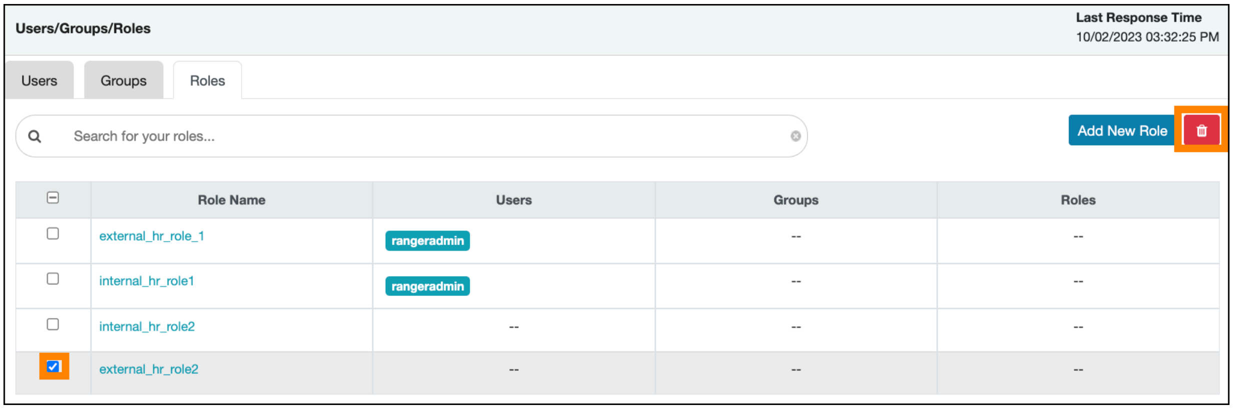
Deleting a role

How to delete a role in Ranger.

Procedure

1. Select Settings Roles .

2. Click the Roles tab.
   Users/Groups/Roles displays the Roles tab active.



⊟	Role Name	Users	Groups	Roles
□	external_hr_role_1	rangeradmin	--	--
□	internal_hr_role1	rangeradmin	--	--
□	internal_hr_role2	--	--	--
☑	external_hr_role2	--	--	--

3. Select the role you want to delete, then click Delete.
4. After deleting any roles, click Save .

   If the role was created in Hive, you can delete the role through Hive using the Drop command:

```
DROP ROLE internal_hr_role_02;
```

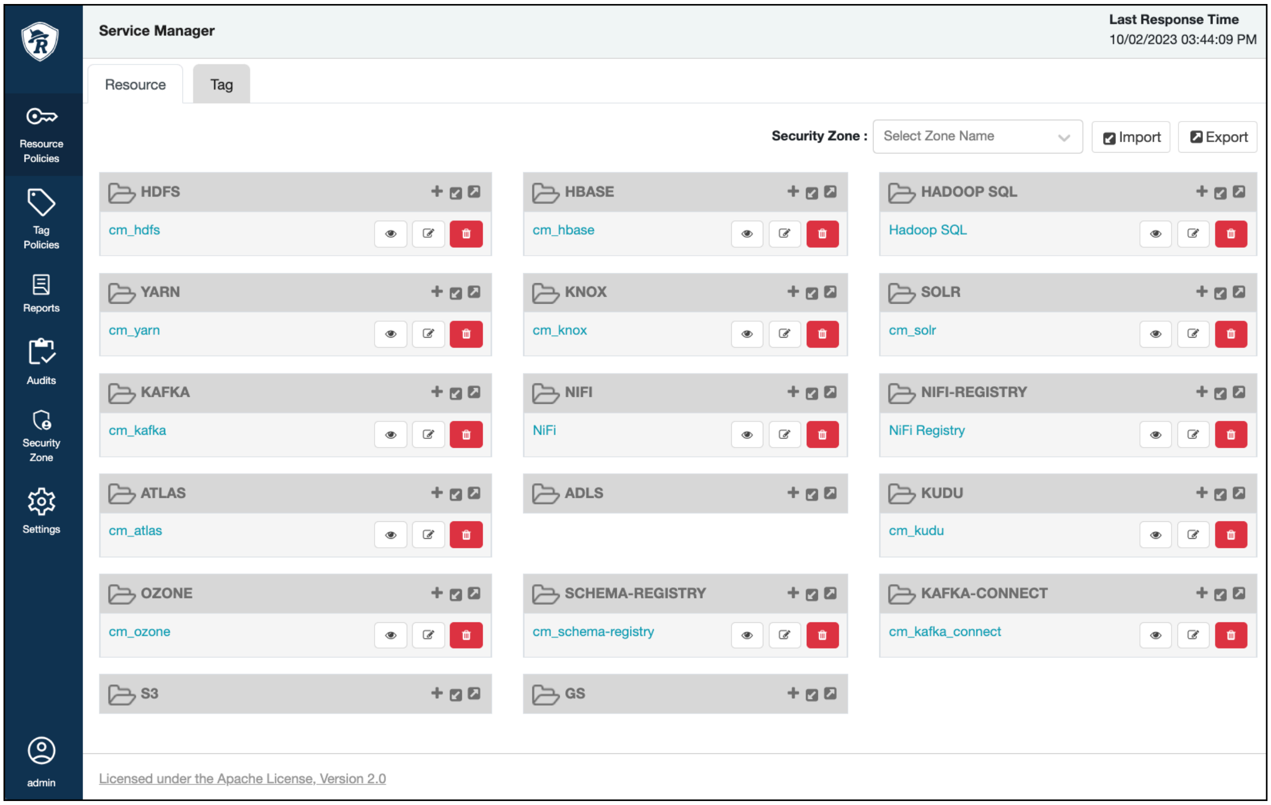
Adding or editing module permissions

How to add or edit users and groups permission to access Ranger modules.

About this task

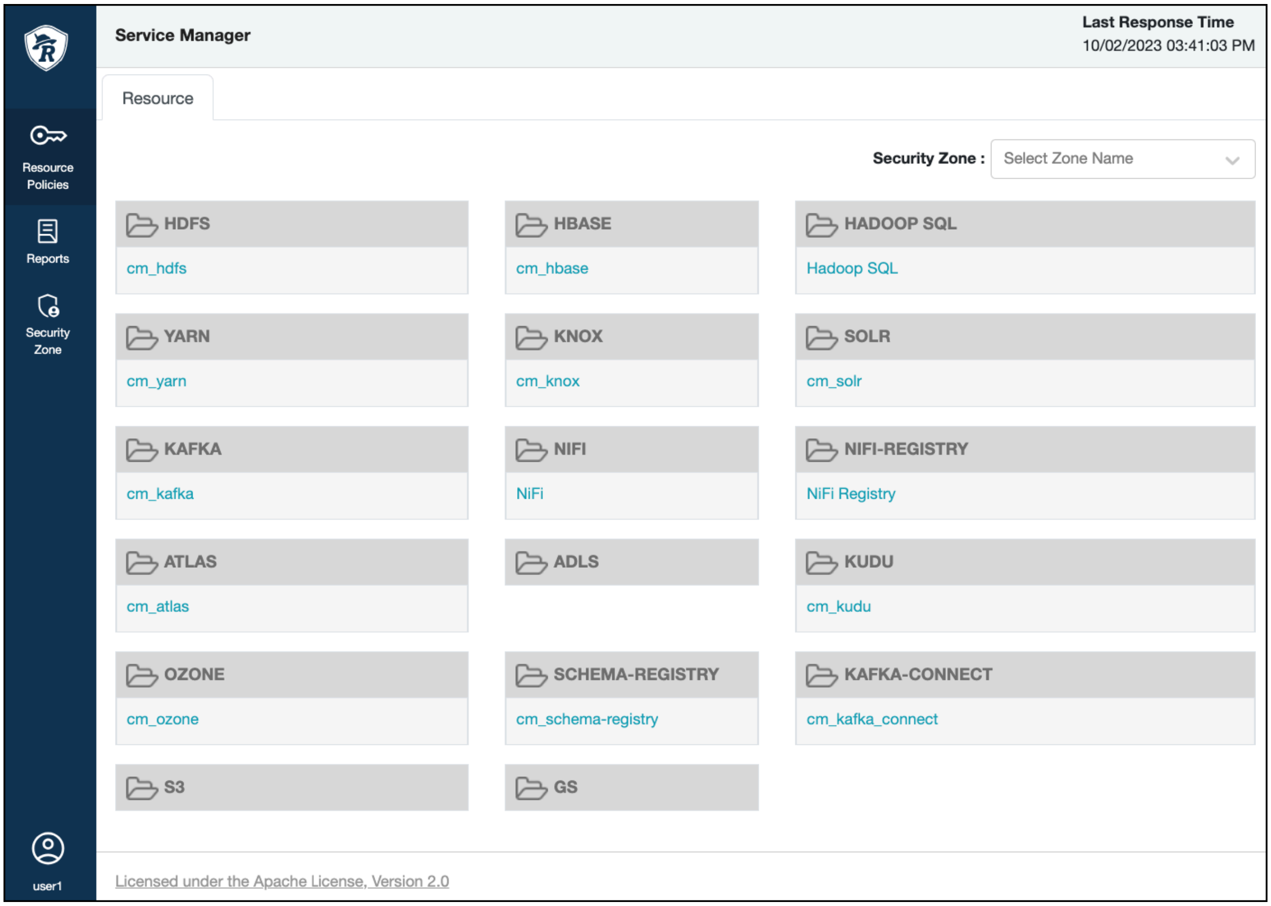
Permissions defines the users and groups that can access Ranger modules using Ranger Admin Web UI. Access to a module determines whether the user can see and use options within that module using Ranger Admin Web UI. For example, a user with access to Resource Based Polices module can see Resource Policies option under Service Manager in Ranger Admin Web UI. An admin user can add, import, and export services policies, as well as view, edit and delete permissions for policies, as shown in the following example:

Figure 1: Ranger Admin User View of Resource-based Policies



A Ranger user (without admin permissions) can only view the resource-based policies, as shown in the following example:

Figure 2: Ranger User View of Resource-based Policies



Service Manager

Last Response Time
10/02/2023 03:41:03 PM

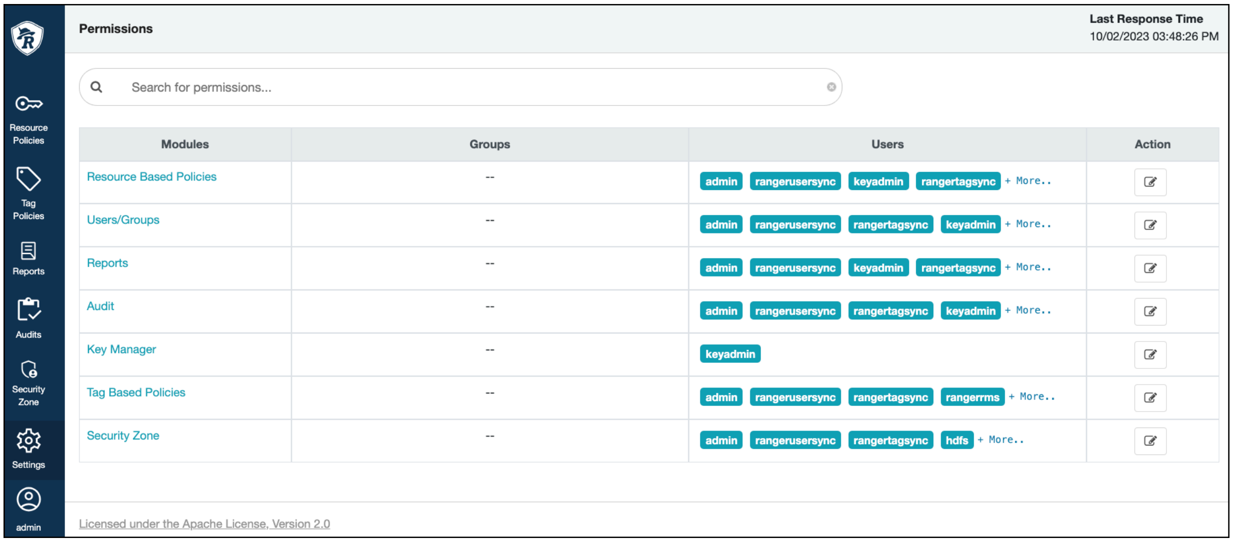
Security Zone : Select Zone Name

HDFS cm_hdfs	HBASE cm_hbase	HADOOP SQL Hadoop SQL
YARN cm_yarn	KNOX cm_knox	SOLR cm_solr
KAFKA cm_kafka	NIFI NiFi	NIFI-REGISTRY NiFi Registry
ATLAS cm_atlas	ADLS	KUDU cm_kudu
OZONE cm_ozone	SCHEMA-REGISTRY cm_schema-registry	KAFKA-CONNECT cm_kafka_connect
S3	GS	

Licensed under the Apache License, Version 2.0

Procedure

1. Select Settings Permissions .
   Permissions displays the Ranger Admin Web UI modules and users that have access permissions to each module.



Permissions

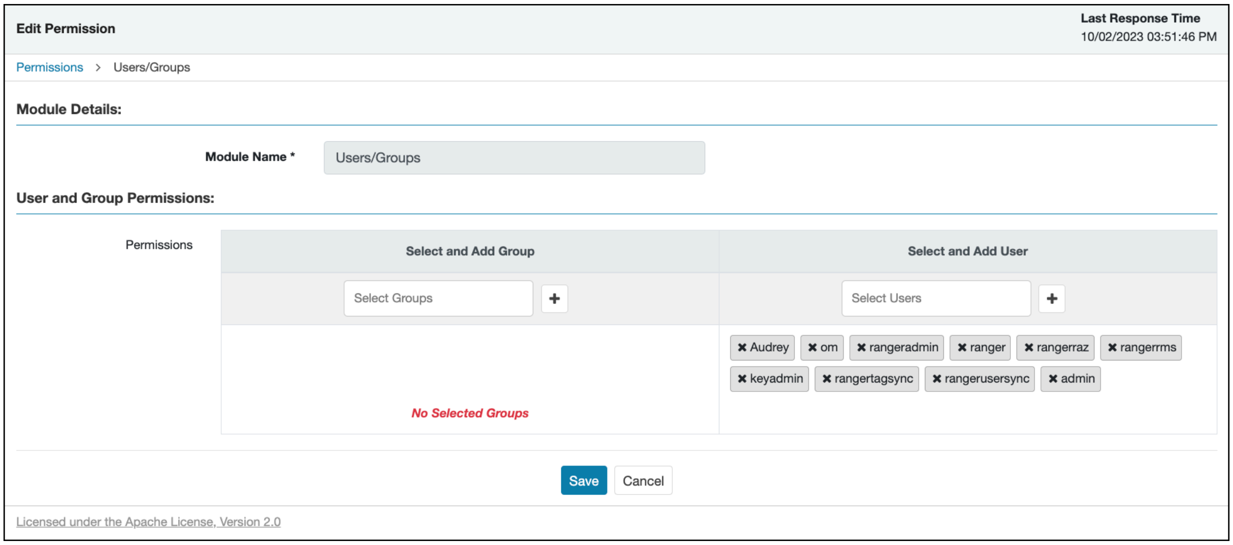
Last Response Time
10/02/2023 03:48:26 PM

Search for permissions...

Modules	Groups	Users	Action
Resource Based Policies	--	admin rangerusersync keyadmin rangertagsync + More..	✎
Users/Groups	--	admin rangerusersync rangertagsync keyadmin + More..	✎
Reports	--	admin rangerusersync keyadmin rangertagsync + More..	✎
Audit	--	admin rangerusersync rangertagsync keyadmin + More..	✎
Key Manager	--	keyadmin	✎
Tag Based Policies	--	admin rangerusersync rangertagsync rangerrms + More..	✎
Security Zone	--	admin rangerusersync rangertagsync hdfs + More..	✎

Licensed under the Apache License, Version 2.0

2. Click the Edit icon (✎) for the module you would like to edit.
   Edit Permission page displays options to select and add user and groups to modules.



Edit Permission

Last Response Time
10/02/2023 03:51:46 PM

Permissions > Users/Groups

Module Details:

Module Name * Users/Groups

User and Group Permissions:

Permissions

Select and Add Group	Select and Add User
Select Groups +	Select Users +
No Selected Groups	✖ Audrey ✖ om ✖ rangeradmin ✖ ranger ✖ rangerraz ✖ rangerrms ✖ keyadmin ✖ rangertagsync ✖ rangerusersync ✖ admin

Save Cancel

Licensed under the Apache License, Version 2.0

3. Edit the users and groups that get permission to access a module, then click Save.

   You can select multiple users and groups using + .

Deleting users or groups in bulk

How to delete users or groups in bulk in Ranger.

About this task

You can delete users and groups from Ranger by using the deleteUserGroupUtil.py script in /opt/cloudera/parcels/CDP/lib/ranger-admin/deleteUserGroupUtil.py.

Before you begin

Only users with the "admin" role can delete users or groups in bulk.

Procedure

1. Get all the users through API or from database directly:

   ```
   # GET http://<ip>:6080/service/xusers/users/
   # GET http://<ip>:6080/service/xusers/groups/
   ```

   Or

   ```
   # select user_name from x_user
   ```

2. Save all the users you want to delete in a file, except service users (keep only AD/LDAP users).
3. On the Ranger host, go to the following location:

   ```
   cd /opt/cloudera/parcels/CDP-<version>/lib/ranger-admin/
   ```

4. Run the following Python script:

   ```
   Example - # python /opt/cloudera/parcels/CDH/lib/ranger-admin/deleteUser
   GroupUtil.py -users <user file path>
   ```

```
-admin <ranger admin user> -force -url <Ranger Admin URL> -sslCertPath: <F
ilepath to ssl certificate to use when Ranger Admin uses HTTPS>
```

Note: The Python script should contain one user or group in each line.

5. Verify on Ranger UI.

Ranger Usersync

How to configure Ranger Usersync to sync users and groups from AD/LDAP.

Overview

The Ranger usersync service syncs users, groups, and group memberships from various sources, such as Unix, File, or AD/LDAP into Ranger. Ranger usersync provides a set of rich and flexible configuration properties to sync users, groups, and group memberships from AD/LDAP supporting a wide variety of use cases.

As a Ranger administrator, you will work with users and groups to configure policies in Ranger and administer access to the Ranger UI. You will use group memberships only to administer access to the Ranger UI.

Note: Group memberships stored in Ranger are not used during authorization. Instead, individual components compute the group memberships for a given user on-demand, using utilities like id or group mappings, during authorization. The authority on this is the output of the id or groups command on the Linux OS, which is populated by SSSD from AD (or whichever LDAP provider is used).

For example:

```
# idsp_test1
uid=40002(sp_test1) gid=40006(sp_test1)
groups=40006(sp_test1),40003(cdf_puas),40005(cdf_policy_admins)
# id sp_auditor
uid=40003(sp_auditor) gid=40007(sp_auditor)groups=40007(sp_auditor),40003(cdf_puas)
```

uses id to show that users sp_test 1 and user sp_auditor each belong to three groups, also

```
#groups sp_test1
sp_test1 : sp_test1 cdf_puas cdf_policy_admins
# groups sp_auditor
sp_auditor : sp_auditor cdf_puas
```

uses groups to show the groups that users sp_test1 and sp_auditor belong to.

You must first understand the specific use-case before syncing users, groups, and group memberships from AD/LDAP. For example, if you want to configure only group-level policies, then you must sync groups to Ranger, but syncing users and group memberships to Ranger is not required.

Determining the users and groups to sync to Ranger:

Typically, you must complete a three-step process to define the complete set of users and groups that you will sync to Ranger:

1. **Define the customer use-case.**

   3 common use cases:

   - A customer Admin or Data Admin wants to configure only group-level policies and restrict access to the Ranger UI to only a few users.
   - A customer's Admin or Data Admin wants to configure only group-level policies and restrict access to the Ranger UI to only members of a group.
   - A customer's Admin or Data Admin wants to configure mostly group-level policies and a few user- level policies.
2. **Define all relevant sync source details.** For every use-case, at least four key questions must by answered:

   - What groups will sync to Ranger?
   - Which organizational units (OUs) in AD/LDAP contain these groups?
   - What users will sync to Ranger?
   - Which organizational units (OUs) in AD/LDAP contain these users?
3. **Configure Usersync properties.**

   This topic describes an example set of Usersync configuration properties and values, based on a simple use-case and example AD source repository.

Example Use Case:

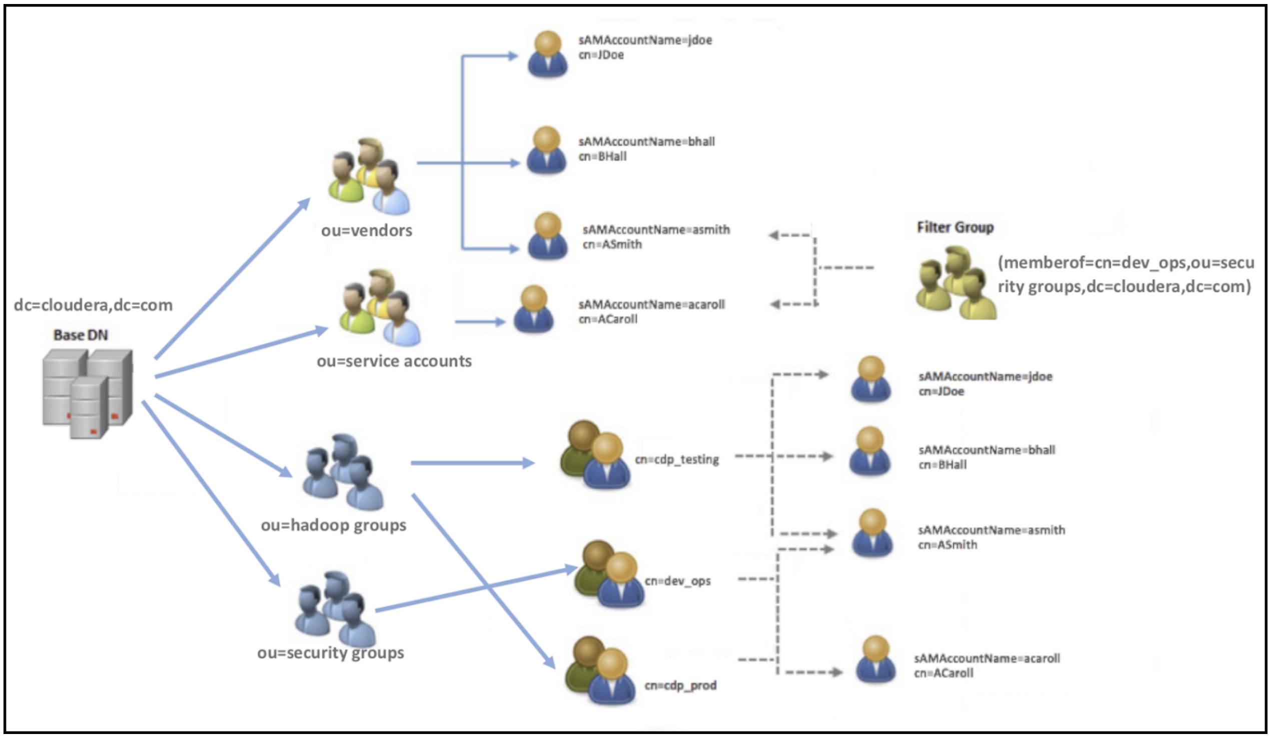
First, consider the following use-case, in order to better understand how to configure Usersync properties:

A customers Admin or Data Admin wants to configure only group-level policies and restrict access to the Ranger UI to only members of a group.

Example AD environment:

Configuring Ranger Usersync with AD/LDAP depends highly on the customer environment. You must understand the organization of users and groups in the customer environment. This illustration shows users and groups organized in an Active Directory environment.

Figure 3: Example Active Directory Structure



Answering the key user and group questions, based on the example AD structure:

In this example, the customer wants to configure group-level policies for groups cdp_testing and cdp_prod and wants to provide admin access to the Ranger UI only for users in the dev_ops group.

Based on the example Active Directory structure, answers to the four key user/group questions are:

Q1: What groups will be synced to Ranger?

A1: cdp_testing, cdp_prod, and dev_ops

Q2: What OUs contain these groups in AD?

A2: hadoop groups and security groups

Q3: What users will be synced to Ranger?

A3: asmith and acaroll (these users are dev_ops group members)

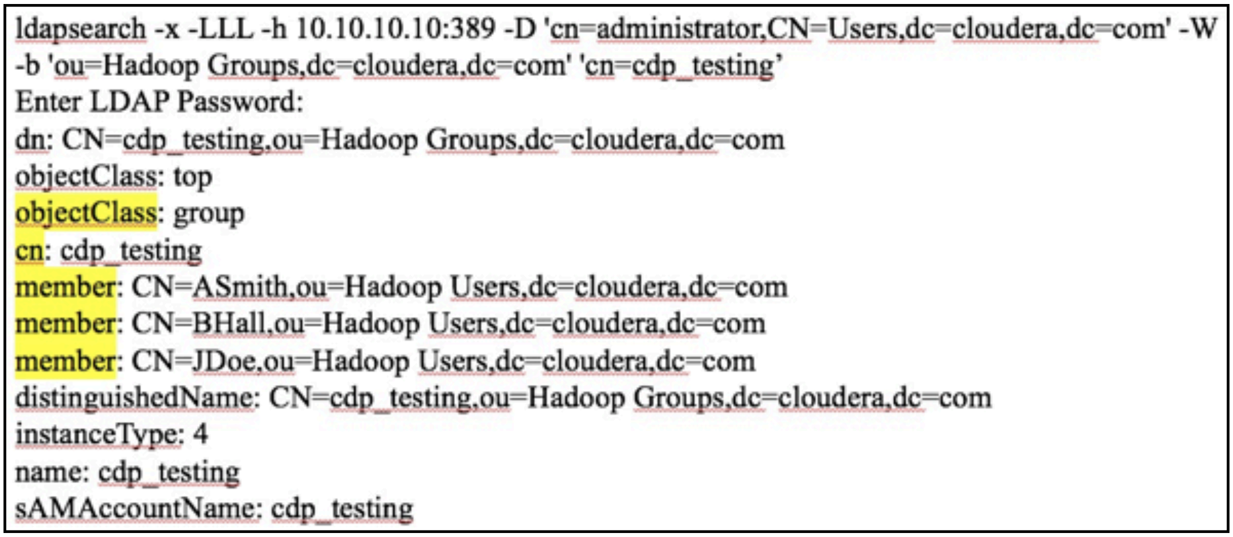
Q4: What OUs contain these users in AD?

A4: vendors and service accounts

To find the specific answers to these questions in a particular environment, use a tool such as Ldapsearch, as shown in the following examples.

- Example: Ldapsearch command to search a particular group cdp_testing and determine what attributes are available for the group.

  Figure 4: Using Ldapsearch to find a specific group

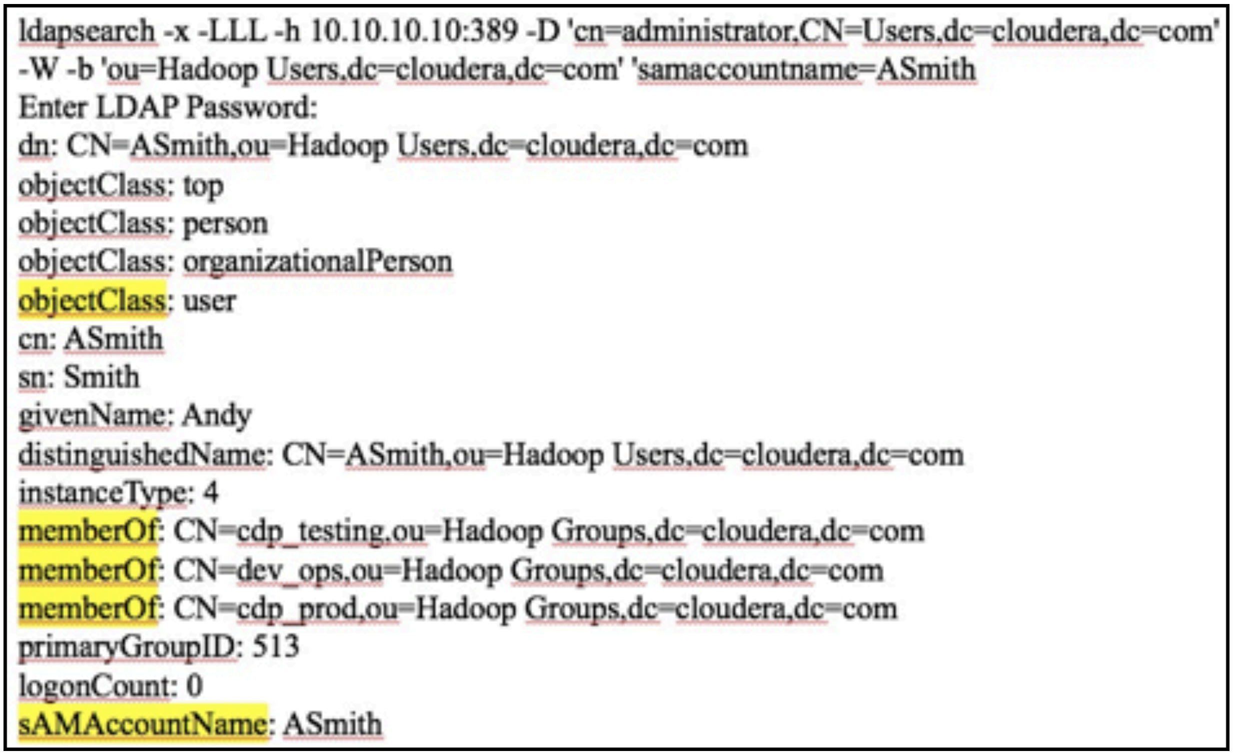


```
ldapsearch -x -LLL -h 10.10.10.10:389 -D 'cn=administrator,CN=Users,dc=cloudera,dc=com' -W
-b 'ou=Hadoop Groups,dc=cloudera,dc=com' 'cn=cdp_testing’
Enter LDAP Password:
dn: CN=cdp_testing,ou=Hadoop Groups,dc=cloudera,dc=com
objectClass: top
objectClass: group
cn: cdp_testing
member: CN=ASmith,ou=Hadoop Users,dc=cloudera,dc=com
member: CN=BHall,ou=Hadoop Users,dc=cloudera,dc=com
member: CN=JDoe,ou=Hadoop Users,dc=cloudera,dc=com
distinguishedName: CN=cdp_testing,ou=Hadoop Groups,dc=cloudera,dc=com
instanceType: 4
name: cdp_testing
sAMAccountName: cdp_testing
```

  Above output shows all the available attributes for cn=cdp_testing. The highlighted attributes are those of interest for usersync configuration. In this case, cdp_testing has three “member” attributes: ASmith, BHall, and JDoe.

- Example: Ldapsearch command to search a particular user ASmith and determine what attributes are available for the user.

  Figure 5: Using Ldapsearch to find a specific user



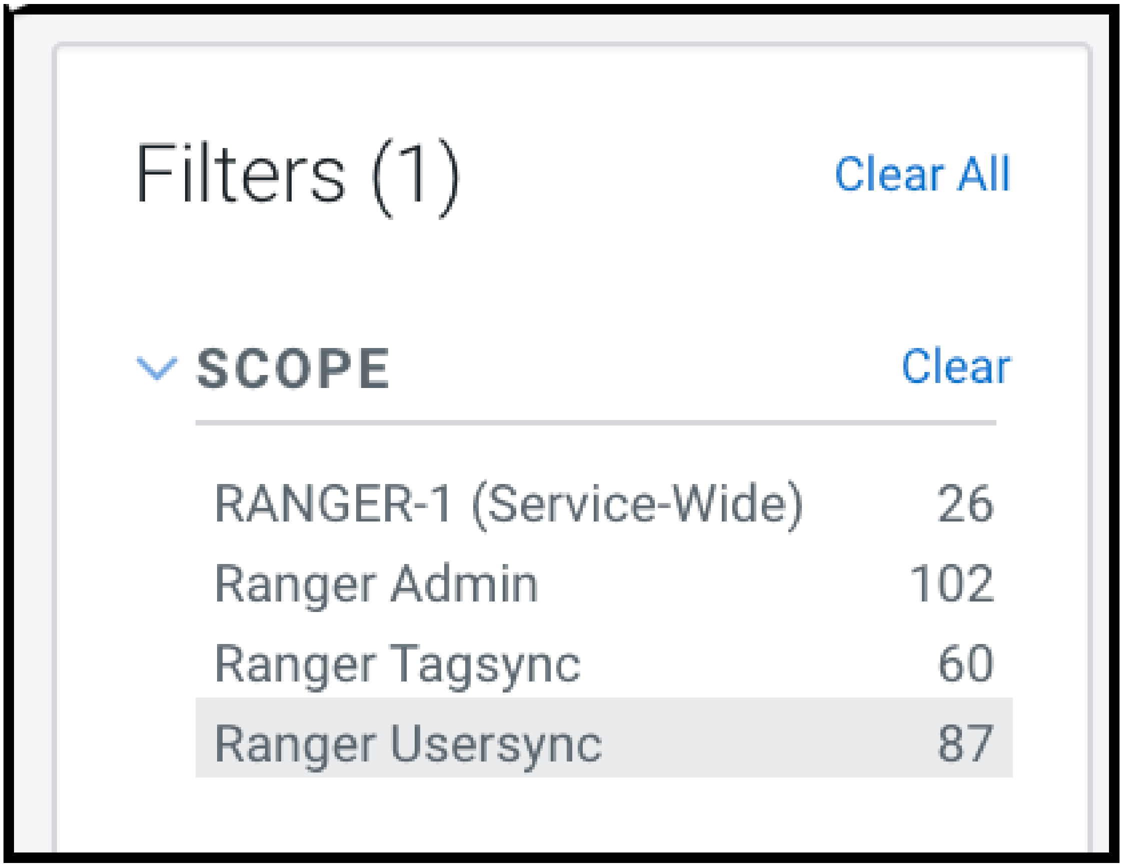
```
ldapsearch -x -LLL -h 10.10.10.10:389 -D 'cn=administrator,CN=Users,dc=cloudera,dc=com'
-W -b 'ou=Hadoop Users,dc=cloudera,dc=com' 'samaccountname=ASmith
Enter LDAP Password:
dn: CN=ASmith,ou=Hadoop Users,dc=cloudera,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: ASmith
sn: Smith
givenName: Andy
distinguishedName: CN=ASmith,ou=Hadoop Users,dc=cloudera,dc=com
instanceType: 4
memberOf: CN=cdp_testing,ou=Hadoop Groups,dc=cloudera,dc=com
memberOf: CN=dev_ops,ou=Hadoop Groups,dc=cloudera,dc=com
memberOf: CN=cdp_prod,ou=Hadoop Groups,dc=cloudera,dc=com
primaryGroupID: 513
logonCount: 0
sAMAccountName: ASmith
```

  Above output shows all the available attributes for a user. The highlighted attributes are those of interest for usersync configuration. In this case, ASmith is a “memberof” 3 groups - cdp_testing, dev_ops, and cdp_prod.

How to configure Usersync, based on the illustrated AD environment example:

In Cloudera Manager Ranger Configuration select the Ranger Usersync filter scope.

Figure 6: Filtering the Ranger Configuration Properties for Usersync

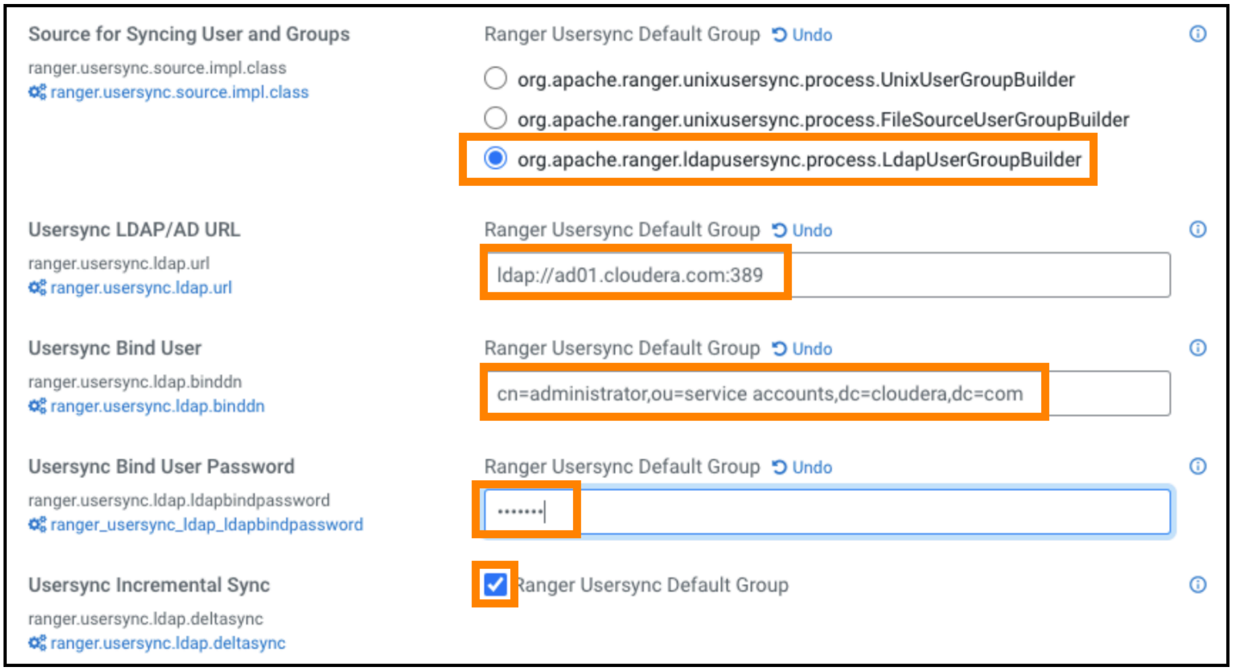


RANGER-1 (Service-Wide)	26
Ranger Admin	102
Ranger Tagsync	60
Ranger Usersync	87

Filtering narrows the list to 87 configuration properties specific to Usersync.

1. To define the common configuration properties that control LDAP URL and bind credentials, scroll to Source for Syncing Users and Groups, then define the configurations properties appropriate for the environment. Configurations shown here match the Example AD environment.

   Figure 7: Ranger Usersync common configuration settings

   Source for Syncing User and Groups
   ranger.usersync.source.impl.class
   ranger.usersync.source.impl.class
   Ranger Usersync Default Group Undo
   ○ org.apache.ranger.unixusersync.process.UnixUserGroupBuilder
   ○ org.apache.ranger.unixusersync.process.FileSourceUserGroupBuilder
   ● org.apache.ranger.ldapusersync.process.LdapUserGroupBuilder

   Usersync LDAP/AD URL
   ranger.usersync.ldap.url
   ranger.usersync.ldap.url
   Ranger Usersync Default Group Undo
   ldap://ad01.cloudera.com:389

   Usersync Bind User
   ranger.usersync.ldap.binddn
   ranger.usersync.ldap.binddn
   Ranger Usersync Default Group Undo
   cn=administrator,ou=service accounts,dc=cloudera,dc=com

   Usersync Bind User Password
   ranger.usersync.ldap.ldapbindpassword
   ranger_usersync_ldap_ldapbindpassword
   Ranger Usersync Default Group Undo
   •••••••

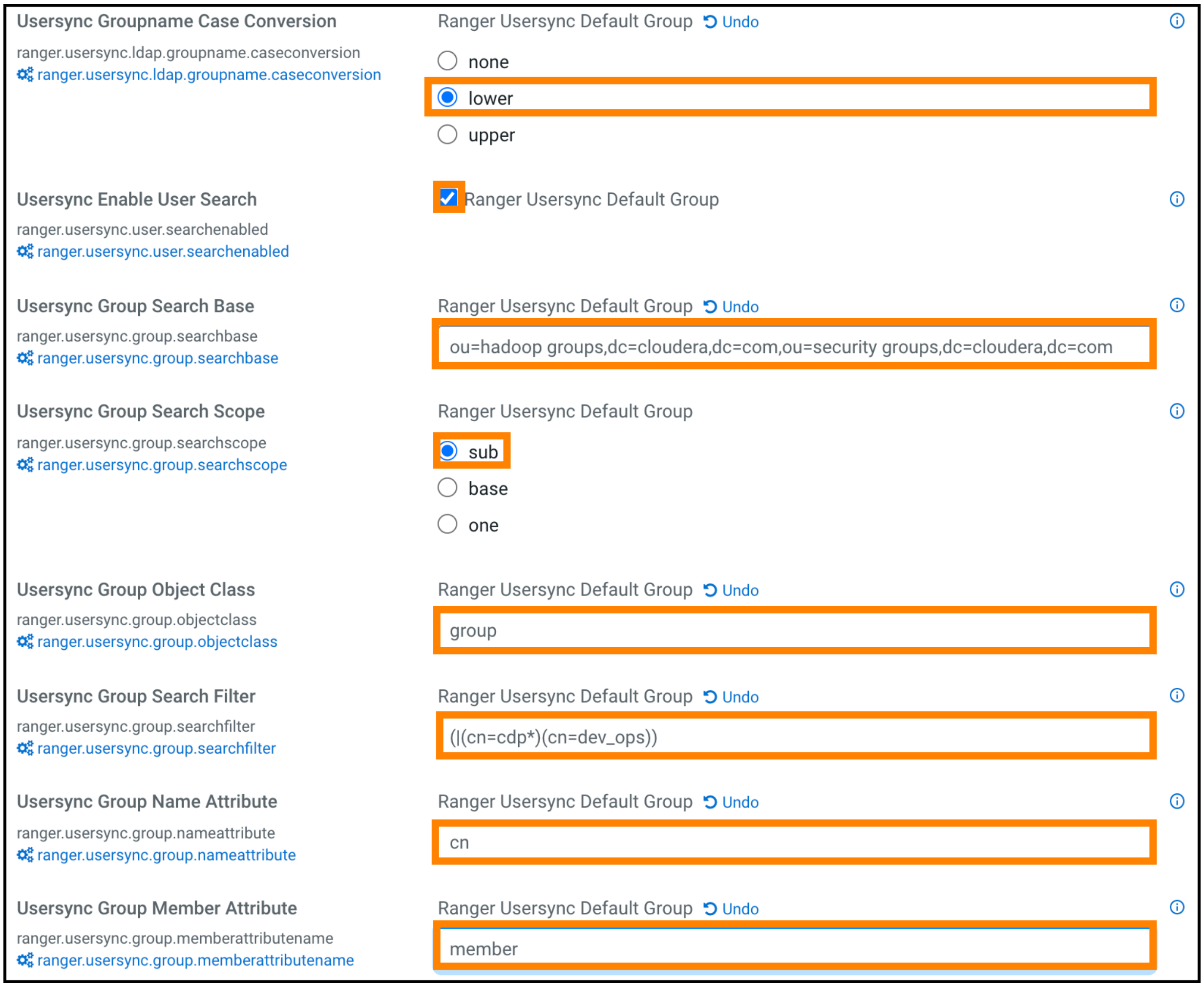
   Usersync Incremental Sync
   ranger.usersync.ldap.deltasync
   ranger.usersync.ldap.deltasync
   ☑ Ranger Usersync Default Group

   Bind credentials are for the user to query Ldap service for users and groups. Bind credentials contain two configuration properties:

   - Usersync Bind User (or bind dn) - specify the username as complete DN (Distinguished Name)
   - Usersync Bind User Password

2. To define the required configuration properties that control group synchronization from AD, scroll to Usersync Enable User Search, then define the configurations properties appropriate for the environment. Configurations shown here match the Example AD environment.

Figure 8: Ranger Usersync group configuration settings



Usersync Groupname Case Conversion
ranger.usersync.ldap.groupname.caseconversion
ranger.usersync.ldap.groupname.caseconversion
Ranger Usersync Default Group Undo
○ none
● lower
○ upper

Usersync Enable User Search
ranger.usersync.user.searchenabled
ranger.usersync.user.searchenabled
☑ Ranger Usersync Default Group

Usersync Group Search Base
ranger.usersync.group.searchbase
ranger.usersync.group.searchbase
Ranger Usersync Default Group Undo
ou=hadoop groups,dc=cloudera,dc=com,ou=security groups,dc=cloudera,dc=com

Usersync Group Search Scope
ranger.usersync.group.searchscope
ranger.usersync.group.searchscope
Ranger Usersync Default Group
● sub
○ base
○ one

Usersync Group Object Class
ranger.usersync.group.objectclass
ranger.usersync.group.objectclass
Ranger Usersync Default Group Undo
group

Usersync Group Search Filter
ranger.usersync.group.searchfilter
ranger.usersync.group.searchfilter
Ranger Usersync Default Group Undo
(|(cn=cdp*)(cn=dev_ops))

Usersync Group Name Attribute
ranger.usersync.group.nameattribute
ranger.usersync.group.nameattribute
Ranger Usersync Default Group Undo
cn

Usersync Group Member Attribute
ranger.usersync.group.memberattributename
ranger.usersync.group.memberattributename
Ranger Usersync Default Group Undo
member

A few specific points to consider about group config settings:

- ranger.usersync.ldap.groupname.caseconversion - Used for converting the case of the groupname. Three possible options are:
  - None - Group names are synced to ranger as is from AD/LDAP. This is the default setting.
  - Lower - All the group names are converted to lowercase while syncing to ranger. This is the recommended setting.
  - Upper - All the group names are converted to uppercase while syncing to ranger

  Note: Policy authorization is case sensitive. Therefore, usernames and groups names synced to ranger must match the exact case of the users and groups resolved by the services such as hdfs, hive, hbase, etc. For example, consider dev_ops (all in lower case). Ranger does not treat this as the same value as Dev_Ops which may have been synced from AD and applied to some policies.

  ranger.usersync.group.searchbase - Used to search a particular OU in AD for groups. Multiple OUs can be specified with ; separated. For example, the example AD shows two OUs that must be searched for groups:

  - ou=hadoop groups,dc=cloudera,dc=com (complete DN for ou=hadoop groups)
  - ou=security groups,dc=cloudera,dc=com (complete DN for ou=security groups)

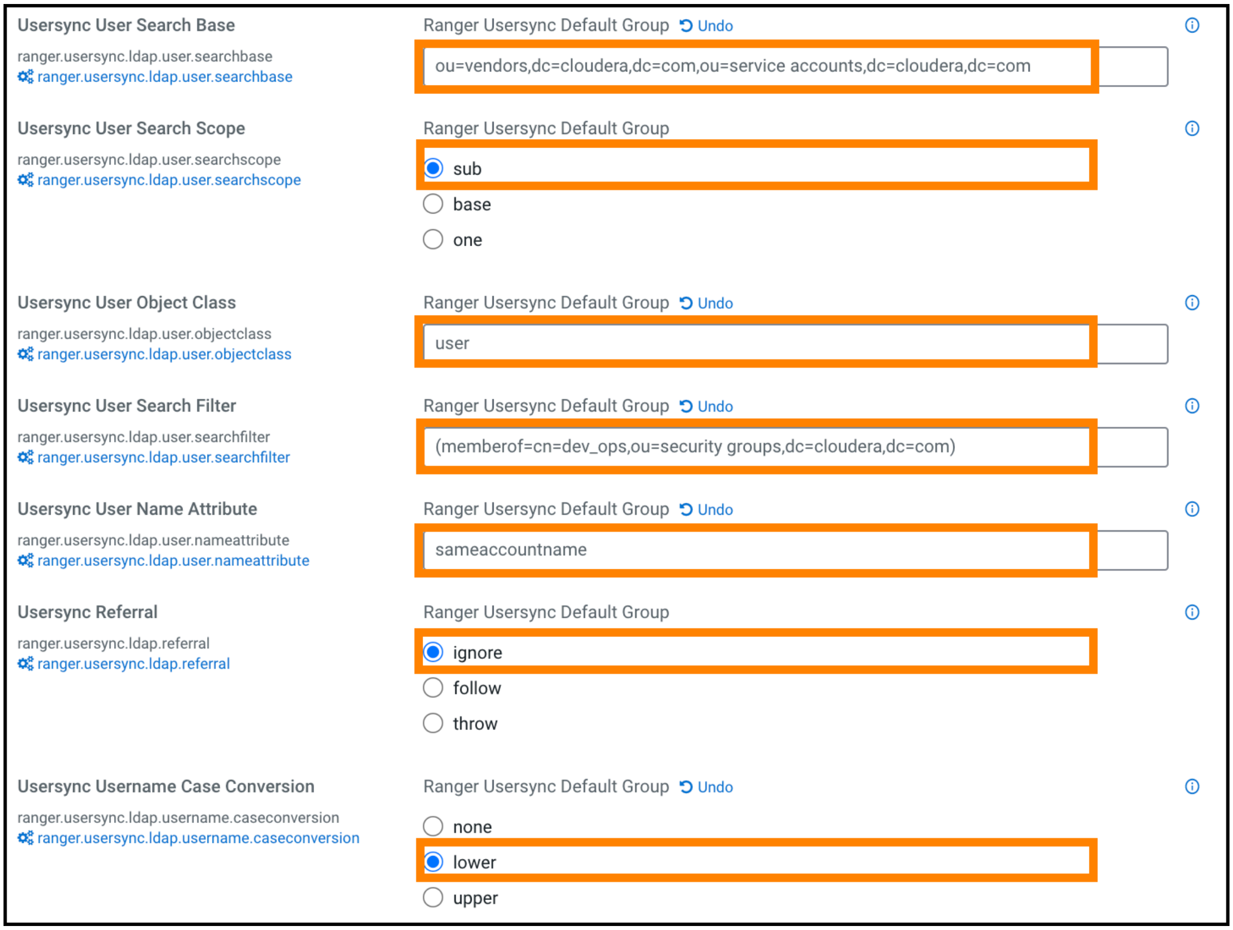
- ranger.usersync.group.searchfilter - In this example, since only 3 groups exist in hadoop groups OU and security groups OU and since all 3 require sync to Ranger, you can specify the filter as cn=* . The value for this property follows standard ldap search query filter format.

  Note: Later, if a new group is added in AD under these OUs and if the customer wants those groups to be sync'd to ranger, no configuration change to usersync is required.

- ranger.usersync.user.searchenabled - In this example, since the customer wants to sync users from dev_ops groups to provide admin access to Ranger UI, this property is set to true .

3. To define the required configuration properties that control user synchronization from AD, scroll to Usersync User Search Base, then define the configurations properties appropriate for the environment. Configurations shown here match the Example AD environment.

Figure 9: Ranger Usersync user configuration settings



Usersync User Search Base
ranger.usersync.ldap.user.searchbase
ranger.usersync.ldap.user.searchbase
Ranger Usersync Default Group ⟲ Undo
ou=vendors,dc=cloudera,dc=com,ou=service accounts,dc=cloudera,dc=com

Usersync User Search Scope
ranger.usersync.ldap.user.searchscope
ranger.usersync.ldap.user.searchscope
Ranger Usersync Default Group
● sub
○ base
○ one

Usersync User Object Class
ranger.usersync.ldap.user.objectclass
ranger.usersync.ldap.user.objectclass
Ranger Usersync Default Group ⟲ Undo
user

Usersync User Search Filter
ranger.usersync.ldap.user.searchfilter
ranger.usersync.ldap.user.searchfilter
Ranger Usersync Default Group ⟲ Undo
(memberof=cn=dev_ops,ou=security groups,dc=cloudera,dc=com)

Usersync User Name Attribute
ranger.usersync.ldap.user.nameattribute
ranger.usersync.ldap.user.nameattribute
Ranger Usersync Default Group ⟲ Undo
sameaccountname

Usersync Referral
ranger.usersync.ldap.referral
ranger.usersync.ldap.referral
Ranger Usersync Default Group
● ignore
○ follow
○ throw

Usersync Username Case Conversion
ranger.usersync.ldap.username.caseconversion
ranger.usersync.ldap.username.caseconversion
Ranger Usersync Default Group ⟲ Undo
○ none
● lower
○ upper

A few specific points to consider about user config settings:

- ranger.usersync.ldap.user.searchbase - This configuration is used to search a particular location in AD for users. Specify multiple OUs with ; separated.

  Note: If users are distributed across several OUs, specifying a base directory, for example, dc=cloudera,dc=com might be convenient and is highly recommended to restrict the search with proper filters.

- ranger.usersync.ldap.user.searchfilter - In this example, since the customer wants to sync only the users that belong to dev_ops, the value for this property is (memberof=cn=dev_ops,ou=security groups,dc=cloudera,dc=com) .

  Note: Wildcards are not supported only when the memberof attribute is used for searching. If you use attributes such as cn or samaccountname for filtering, you can specify wildcards. For example, (|(cn=asm*)(samaccountname=acar*))

- ranger.usersync.ldap.username.caseconversion - Used for converting the case of the username. Three possible options are:

  - None - Usernames are synced to ranger as is from AD/LDAP. This is the default setting.
  - Lower - All the usernames are converted to lowercase while syncing to ranger. This is the recommended setting.
  - Upper - All the usernames are converted to uppercase while syncing to ranger

Note: Policy authorization is case sensitive. Therefore, usernames and groups names synced to ranger must match the exact case of the users and groups resolved by the services such as hdfs, hive, hbase, etc. For example, consider asmith (all in lower case). Ranger does not treat this as the same value as ASmith which may have been synced from AD and applied to some policies.

Adding default service users and roles for Ranger

Cloudera Manager adds a property and default values that define roles for the minimum set of service users by default.

Name

ranger.usersync.whitelist.users.role.assignment.rules

Default Value

&ROLE_SYS_ADMIN:u:admin,rangerusersync,rangertagsync,ranger,rangeradmin,rangerraz,rangerrms&ROLE_KEY_A

Go to Cloudera Manager Ranger Configuration , then type whitelist in Search to see the property and assigned values. Ranger Usersync creates roles for each service user during syncronization.

ranger.usersync.whitelist.users.role.assignment.rules uses same format as ranger.usersync.group.based.role.assignment.rules.

If you add any custom principals, you must update the list of values for ranger.usersync.whitelist.users.role.assignment.rules accordingly so that Ranger usersync applies role assignments rules appropriately. Any change to these configuration values requires a restart of Ranger usersync. Ranger usersync applies these rules during restart and every sync cycle, if changed.

If the same service user exists in:

- ranger.usersync.whitelist.users.role.assignment.rules, and
- ranger.usersync.group.based.role.assignment.rules

with different role assignments, then the role assignment from ranger.usersync.whitelist.users.role.assignment.rules takes priority. This is true even if ranger.usersync.group.based.role.assignment.rules has role assignment rules for a group that has service users as members. Any changes to the role assignments made to these service users from Ranger UI or rest API are temporary. The next Ranger usersync sync cycle resets them.

Configuring Usersync assignment of Admin users

How to automatically assign Admin and Key Admin roles for external users

About this task

Ranger provides configuration for defining roles for external users.

Usersync pulls in users/groups from your external user repository, such as LDAP/AD, and populates the Ranger database with these users/groups. Use this procedure to automatically assign roles to specific users/groups. The example properties shown in this topic automatically assign the ADMIN/KEYADMIN role to external users.

Currently, Ranger supports various roles (or privileges) to be assigned to a user:

ROLE_SYS_ADMIN

Has permission to create users, group, roles, services, and policies, run reports, and perform other administrative tasks. Admin users can also create child policies based on the original policy.

ROLE_KEY_ADMIN

Has permission to manage (create, update, or delete) access policies and keys for Ranger KMS.

ROLE_USER

Has least privilege (and default role) assigned to a user. All users are assigned this default role.

ROLE_ADMIN_AUDITOR

An Admin user with read-only permission.

ROLE_KEY_ADMIN_AUDITOR

An Admin user with read-only permission for Ranger KMS.

Auditor and KMS Auditor roles have been introduced in Ranger Admin. Users with these roles have read-only access to all the services, policies, user/groups,audits and reports.

- The Auditor role allows a user with Auditor role to view all information that a user with Admin role can see. A user with Auditor role will have a read-only view of a user with Admin role. In other words, a user with Auditor role user will be blocked from the create/update/delete/import/exportJson of all API in the Ranger UI and curl command.
- The KMS Auditor role allows a user with KMS Auditor role to view all information that a user with Keyadmin role can see on the Ranger UI. A user with KMS Auditor role will have a read-only view of a user with Keyadmin role. In other words, a user with KMS Auditor role will be blocked from create/update/delete/import/exportJson of all API in the Ranger UI and curl command.
- Users with the Auditor or KMSAuditor role, even if delegated as admin in any policies of any services, will be restricted from create/update/delete/import/exportJson. In other words, users with Auditor or KMS Auditor role have view-only access based on their role.
- A user with KMS Auditor role cannot get keys, even if that user is added in policy.
- Users with Auditor or KMS Auditor role can change their password.
- No user has Auditor or KMS Auditor role by default.
- Users with Auditor or KMS Auditor role can export policies to excel and csv file formats.

A user can have only one role, and that role is determined by the last role assigned, depending in part on group membership.

For example, if the role assignment rules are configured as follows:

ROLE_SYS_ADMIN:u:User1, User2&ROLE_SYS_ADMIN:g:Group1, Group2&ROLE_AUDITOR:g:Group3, Group4&ROLE_USER:g:Group5

and if a user belongs to Group1 & Group5, then the role assigned to that user is ROLE_USER.

Similarly, if a user belongs to Group2 & Group3, then the role assigned to that user is ROLE_AUDITOR.

If the user does not belong to any of these groups (Group1, Group2, Group3, Group4, or Group5), then the default role assigned to the user is ROLE_USER.

If the user belongs to only Group1, then the role assigned to the user is ROLE_SYS_ADMIN.

To automatically assign the ADMIN/KEYADMIN role to external users:

Procedure

1. In Ranger Configuration Search , type role.assignment.
2. In Ranger Usersync Default Group: verify that the following default delimiter values appear for each property:

Property Name	Delimiter Value
ranger.usersync.role.assignment.list.delimiter	&
ranger.usersync.users.groups.assignment.list.delimiter	:
ranger.usersync.username.groupname.assignment.list.delimiter	,
ranger.usersync.group.based.role.assignment.rules	

3. In Ranger UserSync Group Based Role Assignment Rules, type the following value as one string:

   ROLE_SYS_ADMIN:u:User1,User2&ROLE_SYS_ADMIN:g:Group1,Group2&

   ROLE_KEY_ADMIN:u:kmsUser&ROLE_KEY_ADMIN:g:kmsGroup&

   ROLE_USER:u:User3,User4&ROLE_USER:g:Group3,Group4&

   ROLE_ADMIN_AUDITOR:u:auditorUsers,auditors&
   ROLE_ADMIN_AUDITOR:g:adminAuditorGroup,rangerAuditors&

   ROLE_KEY_ADMIN_AUDITOR:u:kmsAuditors&ROLE_KEY_ADMIN_AUDITOR:g:kmsAuditorGroup

   where "u" indicates user and "g" indicates group
4. Click Save Changes (CTRL+S).
5. If Usersync requires no other changes, choose Actions Restart Usersync .

Configuring nested group hierarchies

You must enable non-zero group hierarchy levels in Ranger before adding a group as a member to another group.

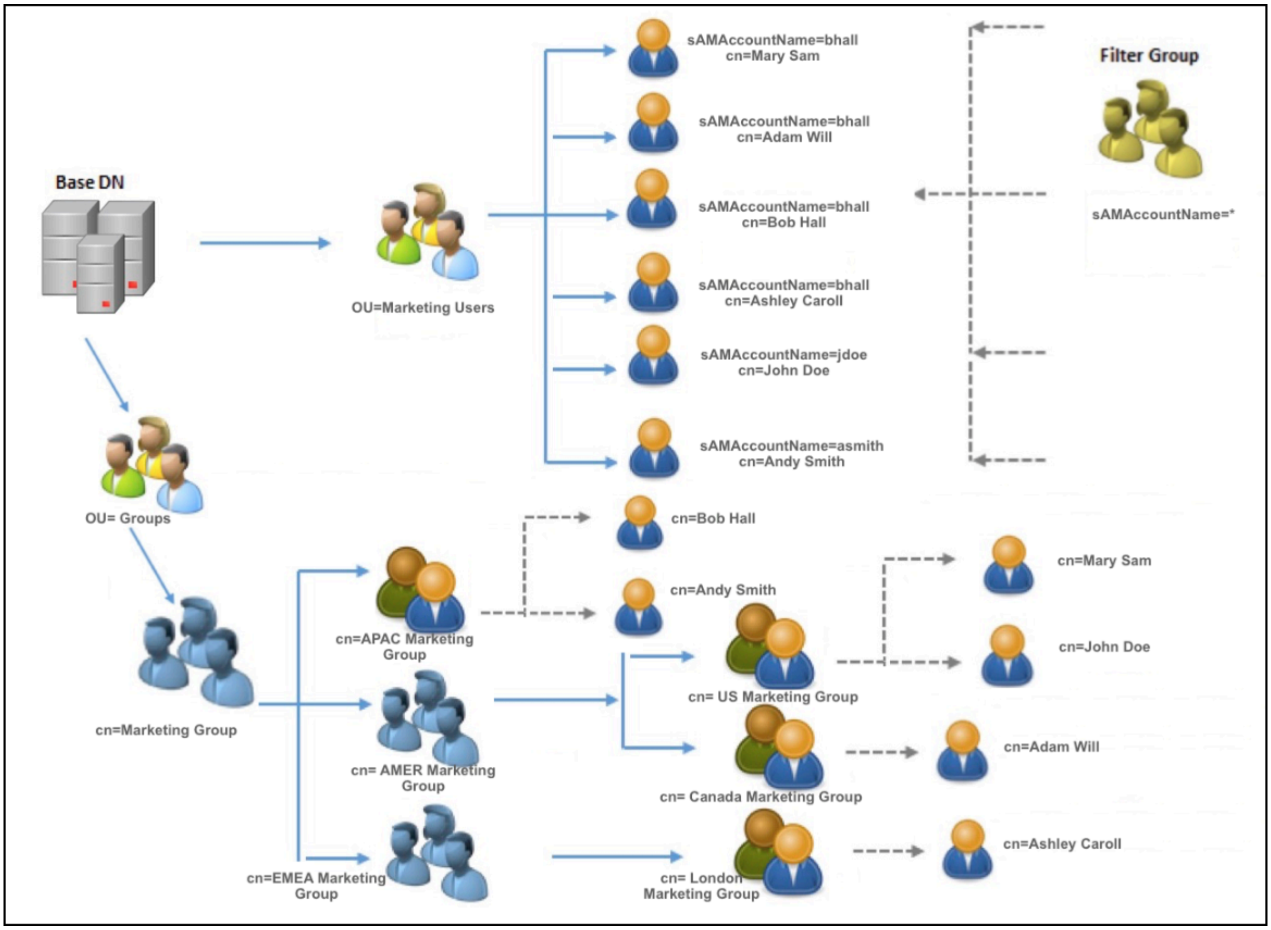
About this task

"Nesting" groups means adding a group as a member to another group. If a group (groupA) is a member of another group (groupB), then the users belonging to the member group (groupA) are part of the parent group (groupB) as well. Nesting can be very useful when delegating access through inheritance. Many large enterprises have their groups in LDAP/AD nested within other groups. Security admins want the users in those nested groups to be associated in Ranger so that they are available for policy authoring in Ranger Admin. Cloudera Ranger Usersync supports nested group membership representation for policy authoring.

Note: In order to utilize nested group mapping, a Ranger Admin user must set the Usersync Group Hierarchy Levels to a non-zero value.

In the following example directory structure, all the marketing users are under one OU "Marketing Users". All these users are members of different groups based on the location like US, Canada, London, etc… For example, user "Adam Will" from "Marketing Users" OU is a member of "Canada Marketing Group". The example directory structure also contains multiple, nested group levels; for example, "US Marketing Group" is a member of "AMER Marketing Group" which again is a member of "Marketing Group".

Figure 10: Example Active Directory structure with Nested Groups


Base DN
OU=Marketing Users
sAMAccountName=bhall cn=Mary Sam
sAMAccountName=bhall cn=Adam Will
sAMAccountName=bhall cn=Bob Hall
sAMAccountName=bhall cn=Ashley Caroll
sAMAccountName=jdoe cn=John Doe
sAMAccountName=asmith cn=Andy Smith
Filter Group
sAMAccountName=*
OU= Groups
cn=Marketing Group
cn=APAC Marketing Group
cn= AMER Marketing Group
cn=EMEA Marketing Group
cn=Bob Hall
cn=Andy Smith
cn= US Marketing Group
cn= Canada Marketing Group
cn= London Marketing Group
cn=Mary Sam
cn=John Doe
cn=Adam Will
cn=Ashley Caroll

Ranger Usersync, by default, computes only the immediate groups for the users. For example, user "Adam Will" is part of "Canada Marketing Group" and only this information is available in ranger without nested group sync configuration. With this information, if an admin wants to provide access to all the users under "AMER Marketing Group", then all the sub groups - "US Marketing Group" and "Canada Marketing Group" must be added to the policy, using Ranger.

In order to simplify the policy configuration at parent-level groups, Ranger supports evaluating nested group memberships by configuring "ranger.usersync.ldap.grouphierarchylevels". If ranger.usersync.ldap.grouphierarchylevels is set to "3", Ranger Usersync computes the group memberships for user "Adam Will" as "Canada Marketing Group", "AMER Marketing Group", "Marketing Group". This way, admin can configure ranger policy at the parent group level ("AMER Marketing Group") which will be applied for all the users (Mary Sam, John Doe, and Adam Will) under each sub group (US Marketing Group and Canada Marketing Group).

Procedure

1. In Cloudera Manager Ranger Configuration, type hierarchy in Search.
2. In Usersync Group Hierarchy Levels (ranger.usersync.ldap.grouphierarchylevels) configuration, set:
   - In Ranger Usersync Default Group, type: 3
3. Click Save Changes.
4. Restart Ranger.

Note: ranger.usersync.ldap.grouphierarchylevels property is set to 0 by default.

Configuring Ranger Usersync for Deleted Users and Groups

How to configure Ranger Usersync for users and groups that have been deleted from the sync source.

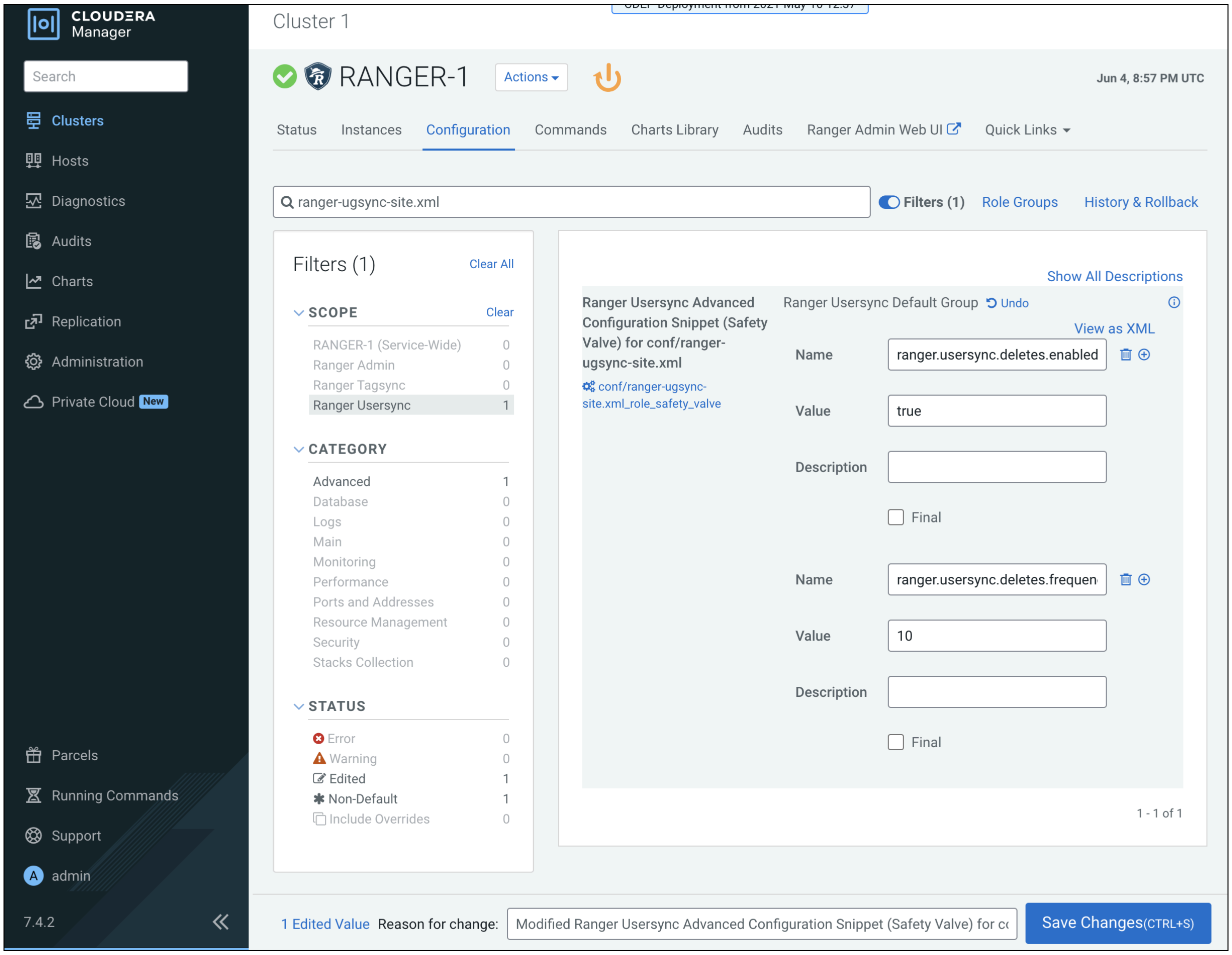
About this task

You can configure Ranger Usersync to update Ranger when users and groups have been deleted from the sync source (UNIX, LDAP, AD or PAM). This ensures that users and groups – and their associated access permissions – do not remain in Ranger when they are deleted from sync source.

Procedure

1. In Cloudera Manager, select Ranger > Configuration, then use the Search box to search for Ranger Usersync Advanced Configuration Snippet (Safety Valve) for conf/ranger-ugsync-site.xml. Use the Add (+) icons to add the following properties, then click Save Changes.

Name	Value	Description
ranger.usersync.deletes.enabled	true	Enables deleted users and groups synchronization. The default setting is false (disabled).
ranger.usersync.deletes.frequency	10	Sets the frequency of delete synchronization. The default setting is 10, or once every 10 Usersync cycles. Delete synchronization consumes cluster resources, so a lower (more frequent) setting may affect performance.



ranger-ugsync-site.xml

Ranger Usersync Advanced Configuration Snippet (Safety Valve) for conf/ranger-ugsync-site.xml

conf/ranger-ugsync-site.xml_role_safety_valve

Ranger Usersync Default Group

Name: ranger.usersync.deletes.enabled

Value: true

Description:

□ Final

Name: ranger.usersync.deletes.frequen

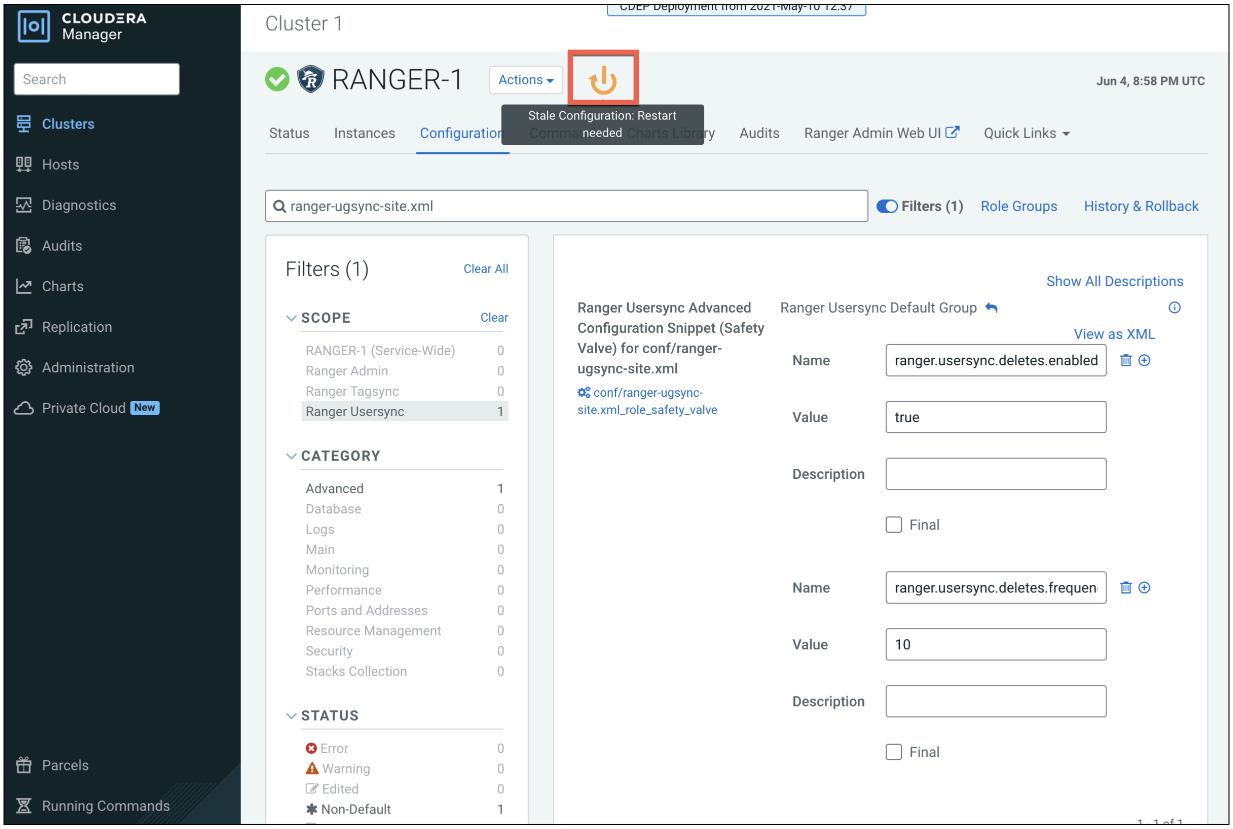
Value: 10

Description:

□ Final

Reason for change: Modified Ranger Usersync Advanced Configuration Snippet (Safety Valve) for co

2. Click the Ranger Restart icon.



ranger-ugsync-site.xml

Filters (1)

SCOPE

RANGER-1 (Service-Wide)	0
Ranger Admin	0
Ranger Tagsync	0
Ranger Usersync	1

CATEGORY

Advanced	1
Database	0
Logs	0
Main	0
Monitoring	0
Performance	0
Ports and Addresses	0
Resource Management	0
Security	0
Stacks Collection	0

STATUS

Error	0
Warning	0
Edited	0
Non-Default	1

Ranger Usersync Advanced Configuration Snippet (Safety Valve) for conf/ranger-ugsync-site.xml

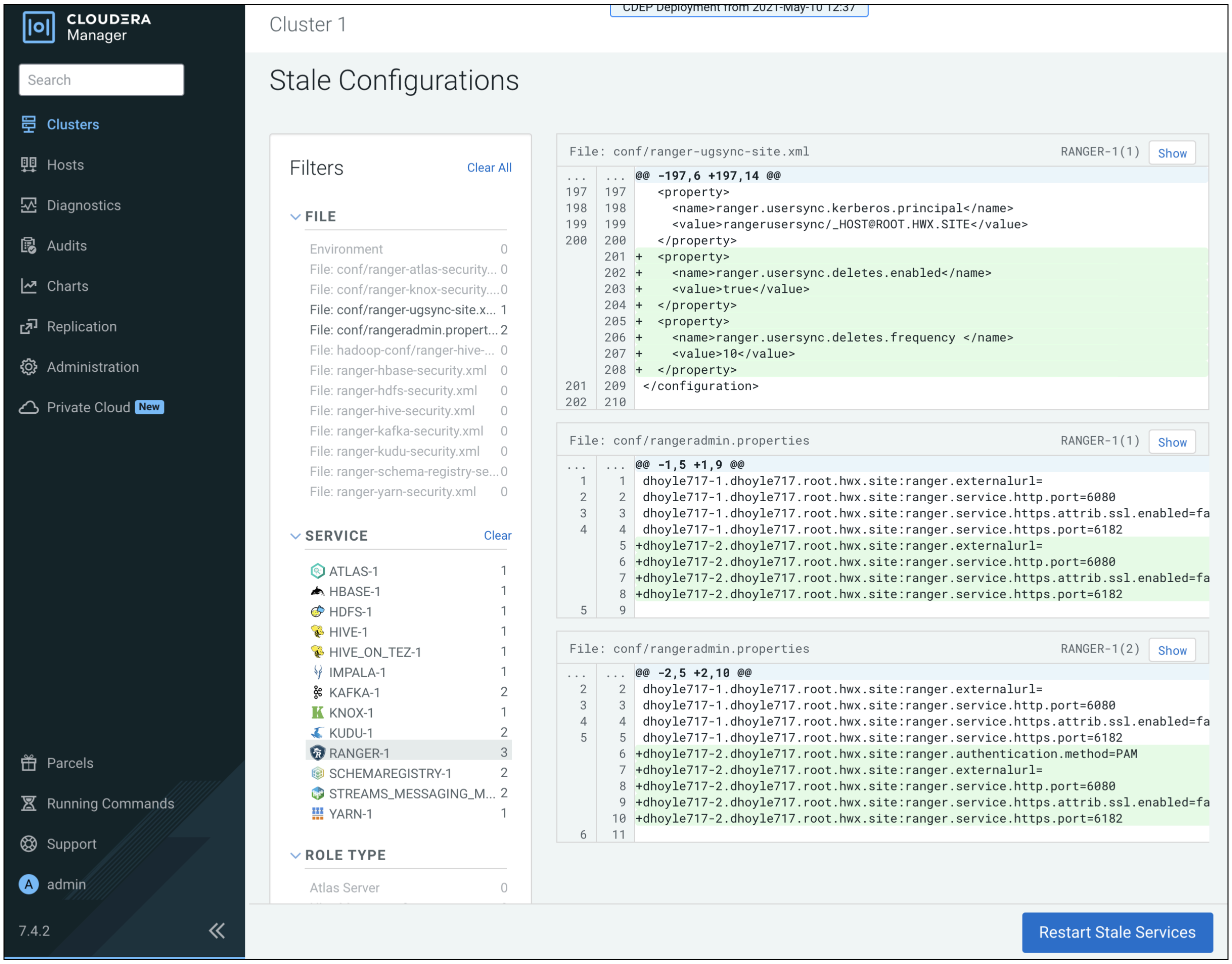
conf/ranger-ugsync-site.xml_role_safety_valve

Ranger Usersync Default Group

Name: ranger.usersync.deletes.enabled
Value: true
Description:
□ Final

Name: ranger.usersync.deletes.frequen
Value: 10
Description:
□ Final

3. On the Stale Configurations page, click Restart Stale Services.



Stale Configurations

Filters Clear All

FILE

Environment	0
File: conf/ranger-atlas-security...	0
File: conf/ranger-knox-security....	0
File: conf/ranger-ugsync-site.x...	1
File: conf/rangeradmin.propert...	2
File: hadoop-conf/ranger-hive-...	0
File: ranger-hbase-security.xml	0
File: ranger-hdfs-security.xml	0
File: ranger-hive-security.xml	0
File: ranger-kafka-security.xml	0
File: ranger-kudu-security.xml	0
File: ranger-schema-registry-se...	0
File: ranger-yarn-security.xml	0

SERVICE Clear

ATLAS-1	1
HBASE-1	1
HDFS-1	1
HIVE-1	1
HIVE_ON_TEZ-1	1
IMPALA-1	1
KAFKA-1	2
KNOX-1	1
KUDU-1	2
RANGER-1	3
SCHEMAREGISTRY-1	2
STREAMS_MESSAGING_M...	2
YARN-1	1

ROLE TYPE

Atlas Server	0

File: conf/ranger-ugsync-site.xml RANGER-1(1)

```
@@ -197,6 +197,14 @@
   <property>
     <name>ranger.usersync.kerberos.principal</name>
     <value>rangerusersync/_HOST@ROOT.HWX.SITE</value>
   </property>
+  <property>
+    <name>ranger.usersync.deletes.enabled</name>
+    <value>true</value>
+  </property>
+  <property>
+    <name>ranger.usersync.deletes.frequency </name>
+    <value>10</value>
+  </property>
 </configuration>
```

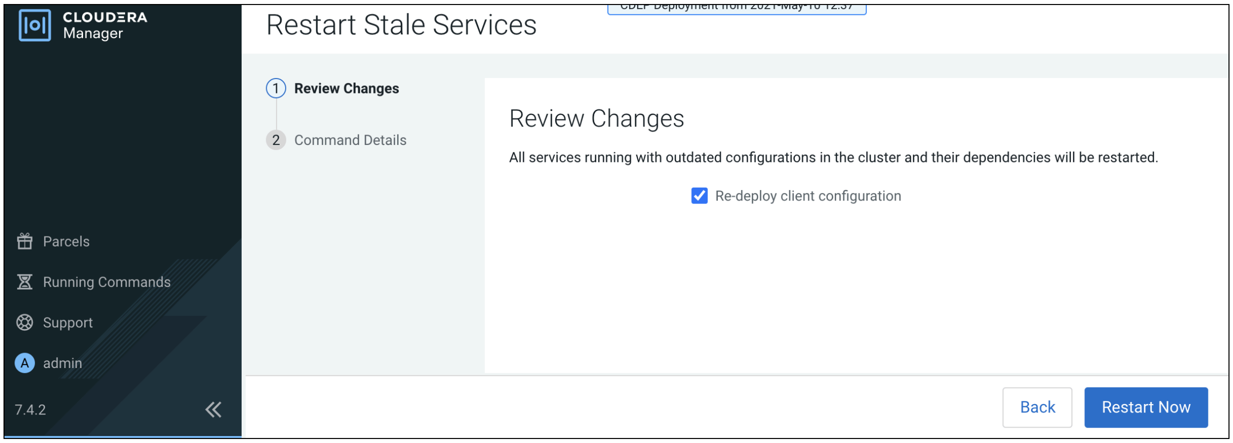
File: conf/rangeradmin.properties RANGER-1(1)

```
@@ -1,5 +1,9 @@
 dhoyle717-1.dhoyle717.root.hwx.site:ranger.externalurl=
 dhoyle717-1.dhoyle717.root.hwx.site:ranger.service.http.port=6080
 dhoyle717-1.dhoyle717.root.hwx.site:ranger.service.https.attrib.ssl.enabled=fa
 dhoyle717-1.dhoyle717.root.hwx.site:ranger.service.https.port=6182
+dhoyle717-2.dhoyle717.root.hwx.site:ranger.externalurl=
+dhoyle717-2.dhoyle717.root.hwx.site:ranger.service.http.port=6080
+dhoyle717-2.dhoyle717.root.hwx.site:ranger.service.https.attrib.ssl.enabled=fa
+dhoyle717-2.dhoyle717.root.hwx.site:ranger.service.https.port=6182
```

File: conf/rangeradmin.properties RANGER-1(2)

```
@@ -2,5 +2,10 @@
 dhoyle717-1.dhoyle717.root.hwx.site:ranger.externalurl=
 dhoyle717-1.dhoyle717.root.hwx.site:ranger.service.http.port=6080
 dhoyle717-1.dhoyle717.root.hwx.site:ranger.service.https.attrib.ssl.enabled=fa
 dhoyle717-1.dhoyle717.root.hwx.site:ranger.service.https.port=6182
+dhoyle717-2.dhoyle717.root.hwx.site:ranger.authentication.method=PAM
+dhoyle717-2.dhoyle717.root.hwx.site:ranger.externalurl=
+dhoyle717-2.dhoyle717.root.hwx.site:ranger.service.http.port=6080
+dhoyle717-2.dhoyle717.root.hwx.site:ranger.service.https.attrib.ssl.enabled=fa
+dhoyle717-2.dhoyle717.root.hwx.site:ranger.service.https.port=6182
```

4. On the Restart Stale Services page, select the Re-deploy client configuration check box, then click Restart Now.



Restart Stale Services

Review Changes

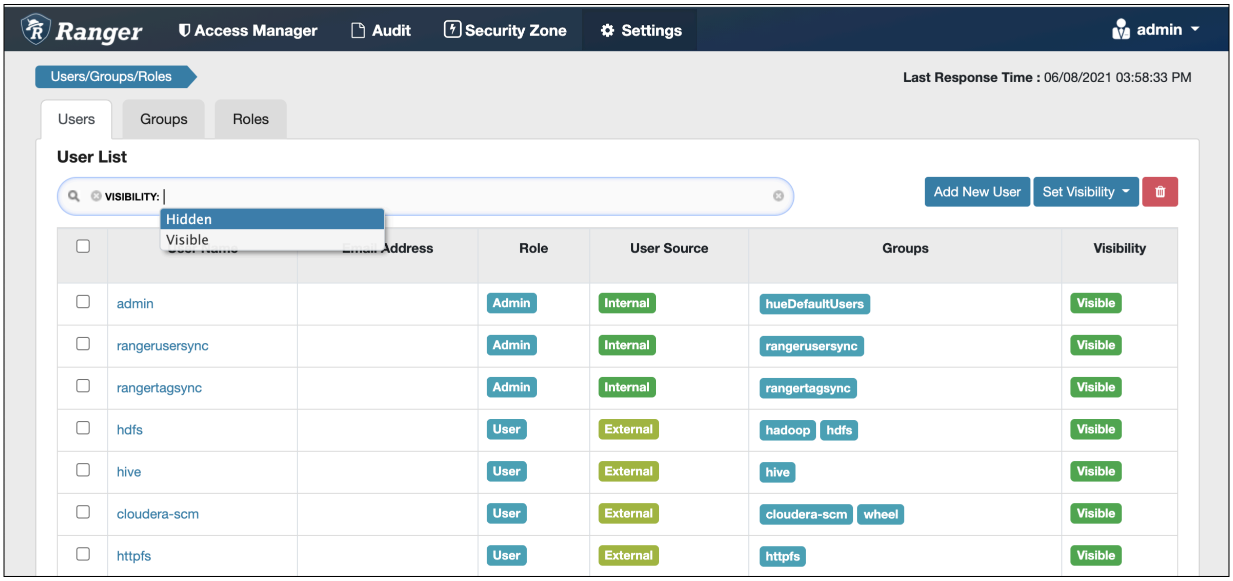
All services running with outdated configurations in the cluster and their dependencies will be restarted.

☑ Re-deploy client configuration

5. A progress indicator page appears while the services are being restarted. When the services have restarted, click Continue.

6. Users that have been deleted in sync source are not automatically deleted in Ranger – they are marked as Hidden and must be manually deleted by the Ranger Admin user, and then Ranger Usersync must be restarted.

   In the Ranger Admin Web UI, select Settings > Users/Groups/Roles. Click in the User List text box, then select Visibility > Hidden.



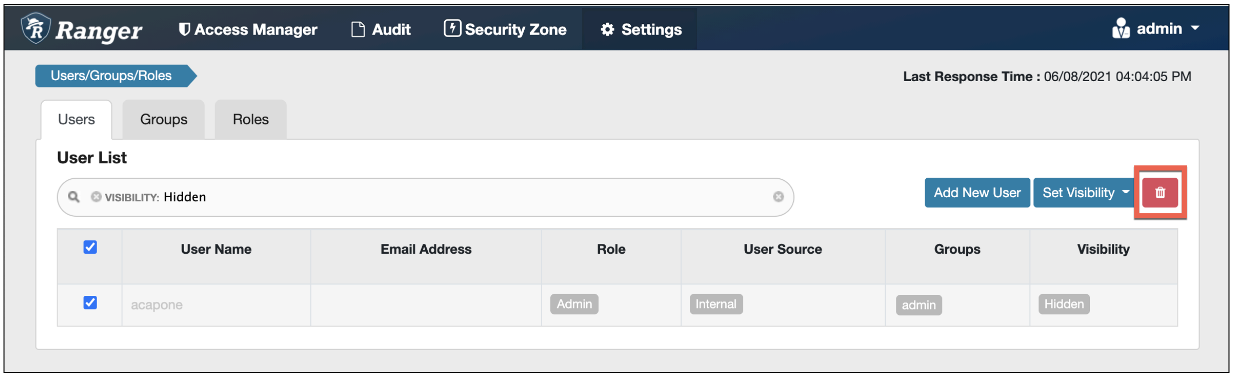
Last Response Time : 06/08/2021 03:58:33 PM

User List

VISIBILITY: |

☐	User Name	Email Address	Role	User Source	Groups	Visibility
☐	admin		Admin	Internal	hueDefaultUsers	Visible
☐	rangerusersync		Admin	Internal	rangerusersync	Visible
☐	rangertagsync		Admin	Internal	rangertagsync	Visible
☐	hdfs		User	External	hadoop hdfs	Visible
☐	hive		User	External	hive	Visible
☐	cloudera-scm		User	External	cloudera-scm wheel	Visible
☐	httpfs		User	External	httpfs	Visible

7. To delete a hidden user or group manually, select the applicable check boxes, then click the red Delete icon, as shown in the following example.



Last Response Time : 06/08/2021 04:04:05 PM

User List

VISIBILITY: Hidden

☑	User Name	Email Address	Role	User Source	Groups	Visibility
☑	acapone		Admin	Internal	admin	Hidden

You can delete multiple users or groups by running a "delete" script on the command line interface.

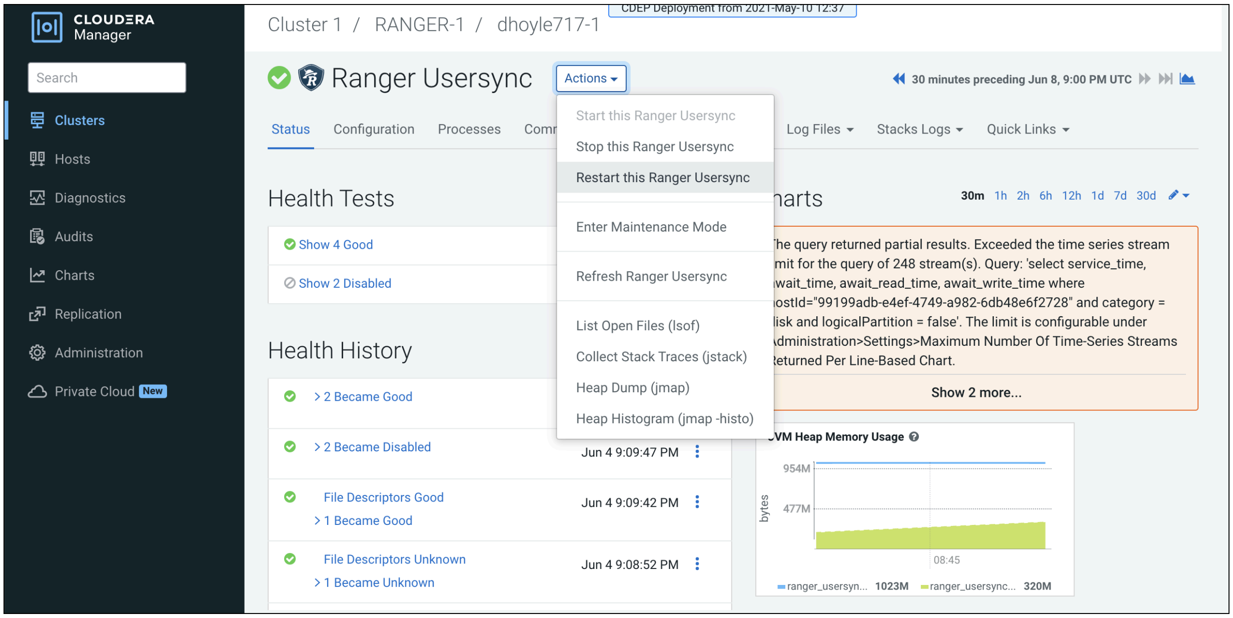
For example:

```
Sample command to delete users:
python deleteUserGroupUtil.py -users <user file path> -admin <ranger admin
 user> -url <rangerhosturl> [-force] [-sslCertPath <cert path>] [-debug]

Sample command to delete groups:
python deleteUserGroupUtil.py -groups <group file path> -admin <ranger
admin user> -url <rangerhosturl> [-force] [-sslCertPath <cert path>] [-d
ebug]
```

Note: The deleteUserGroupUtil.py script installs as part of the Ranger installation on the node where Ranger Admin runs, in the following location: /opt/cloudera/parcels/CDH/lib/ranger-admin/ .

8. In Cloudera Manager, select Ranger > Ranger Usersync, then select Actions > Restart this Ranger Usersync.



The query returned partial results. Exceeded the time series stream
mit for the query of 248 stream(s). Query: 'select service_time,
wait_time, await_read_time, await_write_time where
ostId="99199adb-e4ef-4749-a982-6db48e6f2728" and category =
lisk and logicalPartition = false'. The limit is configurable under
Administration>Settings>Maximum Number Of Time-Series Streams
Returned Per Line-Based Chart.

Note:

- Sync source is tracked when processing Ranger users and groups for deletion. If the same user name for a separate sync source already exists in Ranger DB, that user will not be updated or marked as hidden.
- For AD/LDAP sync:
  - After marking a user or group as deleted/hidden in Ranger, the user or group status does not change automatically. The user or group must be manually deleted (or deleted using the cli "delete" script). Usersync must be restarted to reflect any changes to the same user name in the source.
  - For example, a user (Bob) from one OU (say Engineering) is deleted from the source and is marked as deleted in Ranger admin. If the same user name (Bob) is subsequently added back to the same OU, the user status will not be automatically enabled. The user must be manually deleted and Usersync must be restarted to implement the changes.
  - If an identical user name (say Bob) is deleted from one OU (say Engineering) and added to a different OU (say Finance) between the sync cycles, user Bob is marked as hidden/deleted only when the delete cycle is triggered. Until then there is a security risk that user Bob from Finance will be granted the permissions for Bob from Engineering.

Configuring Ranger Usersync for invalid usernames

How to configure Ranger Usersync to manage usernames containing invalid characters.

About this task

Ranger Usersync pulls in users/groups from your external user repository, such as LDAP/AD, and populates the Ranger database with these users/groups.

An invalid username contains at least one invalid character. Ranger fails to create a set of users if an invalid username exists within that set of users. Usersync perpetually tries to recreate this user set without creating Ranger or Cloudera Manager alerts. This error appears in both Usersync and admin logs, but the log output lacks necessary details such as the invalid username. By adding the following configuration, you cause Usersync to recognize invalid characters in a user/group name and then skip synchronization for any names that contain invalid characters.

Procedure

1. In Cloudera Manager Ranger Configuration, type Ranger Usersync Advanced Configuration Snippet (Safety Valve) in Search.
2. In the Ranger Usersync Advanced Configuration Snippet (Safety Valve) for conf/ranger-ugsync-site.xml configuration, select + to include the following property:
   - In Name, type: ranger.usersync.name.validation.enabled
   - In Value, type: true
3. Click Save Changes.
4. Restart Ranger.

   Note: This configuration property is set to false by default.

Results

Ranger Usersync now successfully synchronizes all valid usernames from the external user repository and skips any usernames containing invalid characters.

Setting credentials for Ranger Usersync custom keystore

How to set the custom keystore file location and password for a Ranger Usersync custom keystore.

About this task

Ranger Usersync role creates a default keystore file, ranger.usersync.keystore.file during restart. UNIX authentication in Ranger Admin requires this keystore file. The keystore file takes a password from the ranger.usersync.keystore.password configuration, exposed in Cloudera Manager supporting Cloudera Runtime 7.1.6 and higher.

Setting custom keystore credentials for Ranger Usersync overrides the default credentials.

Note: Setting custom keystore credentials addresses the issue of using the default, self-signed certificate created for usersync for port 5151. After performing this procedure, you can use your custom, CA-signed certificate.

To set Ranger Usersync custom keystore credentials:

Procedure

1. In Cloudera Manager Ranger Configuration , type Ranger Usersync Advanced Configuration Snippet in the search field.
2. In Ranger Usersync Advanced Configuration Snippet (Safety Valve) for conf/ranger-ugsync-site.xml , enter the following:
   a) In Name, type: ranger.usersync.keystore.file
   b) In Value, type: <keystore_file_path>
3. In Cloudera Manager Ranger Configuration , type Usersync Keystore Password in the search field.
4. In ranger.usersync.keystore.password, type a new password.
5. Click Save Changes.
6. Restart Ranger Usersync.

Results

Ranger uses the custom keystore file location and password values instead of the default values.

Enabling Ranger Usersync search to generate internally

You can configure Ranger Usersync to generate a search filter internally when Search includes a list of group names or group names with a wildcard character.

About this task

When you want to filter users who are members of “cdp_prod”, “cdp_testing”, or “dev_ops” groups, you can add a configuration, ranger.usersync.ldap.groupnames, that accepts each group name, as a domain name, a short name, or as a group name that contains a wildcard character. Usersync only reads ranger.usersync.ldap.groupnames when the sync source is AD/LDAP and ranger.usersync.ldap.user.searchfilter is empty. This also requires that ranger.usersync.group.searchbase is not empty and the configured value for ranger.usersync.group.searchbase must be part of the group searchbase in AD/LDAP. When ranger.usersync.ldap.user.searchfilter is not empty, Usersync ignores the value of ranger.usersync.ldap.groupnames. Values can be either DN of the groups, short name of the groups, or the group names with wildcard character. For example:

- Domain names of the groups
  - memberof=CN=dev_ops,ou=Hadoop Groups,dc=cloudera,dc=com
  - memberof=CN=cdp_prod,ou=Hadoop Groups,dc=cloudera,dc=com
  - memberof=CN=cdp_testing,ou=Hadoop Groups,dc=cloudera,dc=com
- Short names of the groups
  - CN=dev_ops
  - CN=cdp_prod
  - CN=cdp_testing
- Group names with wildcard character
  - CN=cdp*
  - CN=dev_ops

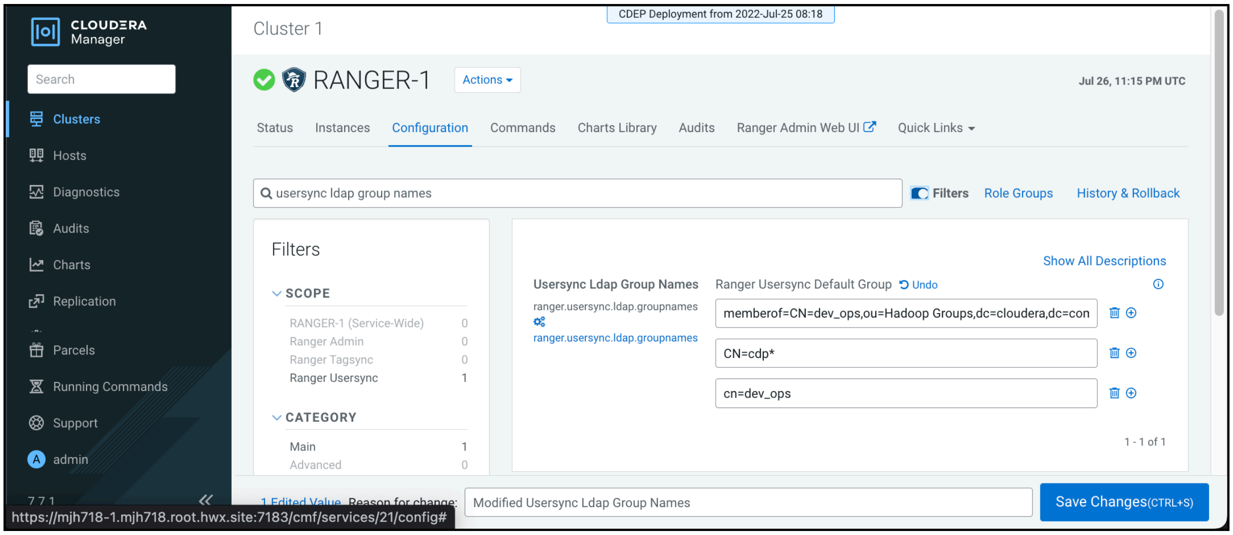
To enable Usersync search to generate an internal search filter for multiple groups names that include wildcard characters:

Procedure

1. In Cloudera Manager Ranger Configuration Search , type ranger.usersync.ldap.groupnames.
2. In Ranger Usersync Default Group, click +1.
3. Type <group_name>.

4. Repeat steps 2 and 3 for each group name.

Figure 11: Example supported group name formats for Usersync LDAP



usersync ldap group names

Usersync Ldap Group Names
ranger.usersync.ldap.groupnames
ranger.usersync.ldap.groupnames

Ranger Usersync Default Group Undo

memberof=CN=dev_ops,ou=Hadoop Groups,dc=cloudera,dc=con

CN=cdp*

cn=dev_ops

Reason for change: Modified Usersync Ldap Group Names

5. Click Save Changes.
6. In Actions, choose Restart Usersync.

Results

The search filter now includes all group names that you saved.

What to do next

To confirm, log in to Ranger Admin Web UI. In Settings Users/Group/Roles Groups , in Groups List, select Group Name. You should see group names that you configured available as search filter values.

Configuring Usersync to sync directly with LDAP/AD

Ranger Usersync can be manually configured to sync directly with LDAP/AD .

About this task

By default, Ranger Usersync uses sssd to sync users and groups from a Unix source. This can affect performance and limit scale. This runtime release supports LDAP/AD as a default sync source. Additionally, Ranger Usersync can be manually configured to:

- update users and groups from multiple (LDAP/AD, Unix and file) sync sources
- customize the default sync interval

Procedure

1. Go to Cloudera Manager Ranger Configuration Filters Ranger Usersync .
2. In Search, type safety valve.
   This filters all Ranger configs to expose only the Usersync safety valves.
3. In Ranger Usersync Advanced Configuration Snippet (Safety Valve) for conf/ranger-ugsync-site.xml, press +.
   a) In Name, type ranger.usersync.syncsource.validation.enabled
   b) In Value, type false
   c) Click Save Changes(CTRL+S)

   Allows sync from multiple source types.

4. In Ranger Usersync Advanced Configuration Snippet (Safety Valve) for conf/ranger-ugsync-extra-attrs.xml, press +.
   a) In Name, type <cloud user.cloud.id.mapping>
   b) In Value, type clouduser1,clouduser2,….
   c) Click Save Changes(CTRL+S)

   Cloud ids for these users and groups are synced to ranger admin.

   Note: This will be populated by default. The above can also be manually overridden.

Configure SASL Bind in Ranger Usersync

Learn how to configure GSSAPI/Kerberos SASL Bind in Ranger Usersync.

About this task

Note: GSSAPI SASL Bind in Ranger Usersync is applicable for 7.3.1.100 and above releases.

Usersync of Ranger supports Simple Bind to authenticate to a directory service and GSSAPI SASL Bind using Kerberos mechanism to authenticate to a directory service from Usersync. This topic describes how to configure GSSAPI SASL Bind in Ranger Usersync.

Procedure

1. In Cloudera Manager, go to Ranger Configuration .
2. Configure the following properties of admin and usersync:

Table 1: LDAP configurations

Property	Values
Admin Configurations	
ranger.authentication.method	ACTIVE_DIRECTORY
ranger.ldap.ad.domain	<ldap-ad-domain>
ranger.ldap.ad.url	ldap://<ldaphost>:389 For SSL: ldaps://<ldaphost>:636
ranger.ldap.ad.base.dn	DC=qe-infra-ad,DC=example,DC=com
ranger.ldap.ad.bind.dn	rangerusersync@<ldap-ad-domain>
ranger.ldap.ad.bind.password	<Ldap-password>
ranger.ldap.ad.referral	ignore
ranger.ldap.ad.user.searchfilter	(sAMAccountName={0})
Usersync Configurations	
ranger.usersync.ldap.user.searchbase	CN=Users,DC=qe-infra-ad,DC=example,DC=com
ranger.usersync.source.impl.class	org.apache.ranger.ldapusersync.process.LdapUserGroupBuilder
ranger.usersync.ldap.url	ldap://<ldaphost>:389 For SSL, ldaps://<ldaphost>:636
ranger.usersync.ldap.binddn	rangerusersync@<ldap-ad-domain>
ranger_usersync_ldap_ldapbindpassword	<Ldap-password>

Property	Values
ranger.usersync.ldap.starttls	false For SSL, true
ranger.usersync.ldap.user.nameattribute	sAMAccountName
ranger.usersync.ldap.user.objectclass	person
ranger.usersync.ldap.user.searchscope	sub
ranger.usersync.ldap.user.searchfilter	cn=*
ranger.usersync.group.nameattribute	cn
ranger.usersync.group.memberattributename	member
ranger.usersync.group.searchfilter	cn=*
ranger.usersync.group.searchbase	OU=groups,DC=qe-infra-ad,DC=example,DC=com
ranger.usersync.group.objectclass	group
ranger.usersync.ldap.authentication.mechanism	GSSAPI **Note:** Add the ranger.usersync.ldap.authentication.mechanism configuration in Safety Valve for conf/ranger-ugsync-site.xml.

3. Click Save Changes.
4. Add truststore details in usersync truststore:

   a) To get the AD cert to pem file:

   ```
   openssl s_client -showcerts -connect <ldap-host>:636 -servername <server
   _name> </dev/null 2>/dev/null > my_ldaps_cert.pem
   ```

   b) To view the content of the pem file:

   ```
   <path_to_keytool>/keytool -printcert -file my_ldaps_cert.pem
   ```

   c) To add the pem file to usersync truststore:

   ```
   <path_to_keytool>/keytool -import -alias ALIAS -file my_ldaps_cert.pem -
   storetype JKS -keystore <path to truststore.jks> -storepass <truststoe_p
   assword>
   ```

5. Restart the service.

Force deletion of external users and groups from the Ranger database

You can force delete external users and groups from a Ranger database using either REST endpoints or python client APIs.

If Ranger Usersync operates without proper configuration, a Ranger database may be (over)-populated with user and group records. Huge user/group and mapping tables may cause Ranger administration issues. To aid in removal of unnecessary users/groups, customers may use this feature to delete specific external user/groups or even all external users/groups if required.

Note: Deleting users and groups manually is not an optimal solution for all customers.

The following 2 REST endpoints are available:

- service/xusers/delete/external/groups
- service/xusers/delete/external/users

The following 2 python client APIs are available:

- force_delete_external_users
- force_delete_external_groups

Invoking REST endpoints (via cURL):

```
# to delete a group named 'group_1'
service/xusers/delete/external/groups?name=group_1
# to delete all hidden groups
service/xusers/delete/external/groups?isVisible=0
# to delete all visible groups
service/xusers/delete/external/groups?isVisible=1
# to delete all groups with sync source as Unix
service/xusers/delete/external/groups?syncSource=Unix
# to delete a user named 'user_1'
service/xusers/delete/external/users?name=user_1
# to delete all hidden users
service/xusers/delete/external/users?isVisible=0
# to delete a user with a particular email id
service/xusers/delete/external/users?emailAddress=xyz5@gmail.com
```

To use the python client APIs:

Note: apache_ranger python package version should be >= 0.0.11

```
$ pip3 install apache_ranger
$ python3
>>> from apache_ranger.client.ranger_client import *
>>> ranger = RangerClientPrivate('<ranger_url>', ('<ranger_user>', '<ranger_
pass>'))
```

Specific use-case examples using the python client APIs:

```
# to delete user_1
>>> ranger.force_delete_external_users("name=user_1")
# to delete group_1
>>> ranger.force_delete_external_groups("name=group_1")
# to delete all users with sync source as Unix
>>> ranger.force_delete_external_users("syncSource=Unix")
# to delete all users with Auditor Role
>>> ranger.force_delete_external_users("userRole=ROLE_ADMIN_AUDITOR")
# to delete all external users
>>> ranger.force_delete_external_users()
# to delete all external groups
>>> ranger.force_delete_external_groups()
```

Transforming username and/or groupname

You can configure Ranger Usersync to perform username and/or groupname transformations in order to make them POSIX compliant.

About this task

You need to configure multiple properties to enable this feature. The values for these properties should be in the following format (similar to Sed format):

```
s/regex/replacement/g
```

Where,

- "s" stands for substitution and is mandatory.
- "/ " is the delimiter and is mandatory.
- Regex is the regular expression to match.
- Replacement is the value to replace and is optional. If not specified, the found pattern is removed from the resulting string.
- "g" stands for replacing all the occurrences. It is optional, and if not specified, it defaults to "g".

To enable this feature, perform the following steps:

Procedure

1. Go to Cloudera Manager Ranger Configuration .
2. Under Ranger Usersync, search and update the following properties:
   - ranger.usersync.mapping.username.handler (default value is org.apache.ranger.ugsyncutil.transform.RegEx)

     This property defines the specific class or handler responsible for mapping usernames from the external source.
   - ranger.usersync.mapping.groupname.handler (default value is org.apache.ranger.ugsyncutil.transform.RegEx)

     This property defines the specific class or handler responsible for mapping groupnames from the external source.
   - ranger.usersync.mapping.username.regex

     This property defines the regular expression which will be used for username conversion.
   - ranger.usersync.mapping.groupname.regex

     This property defines the regular expression which will be used for groupname conversion.
3. Restart Ranger.

   Note: To configure multiple transformations/mappings, you can add new regex properties in Ranger Usersync Advanced Configuration Snippet (Safety Valve) for conf/ranger-ugsync-site.xml as: ranger.usersync.mapping.<attribute name>.<tranformation>.1, ranger.usersync.mapping.<attribute name>.<tranformation>.2, and so on...

   Where, <attribute name> is “username” or “groupname” or "email ID", which comes from AD/LDAP, and <transformation> is “regex”, which is the conversion method certified by Cloudera.

   All the transformations will be applied in the order you configure them.