7.1.8..

Apache Ranger User Management

Date published: 2022-07-21
Date modified:

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

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:

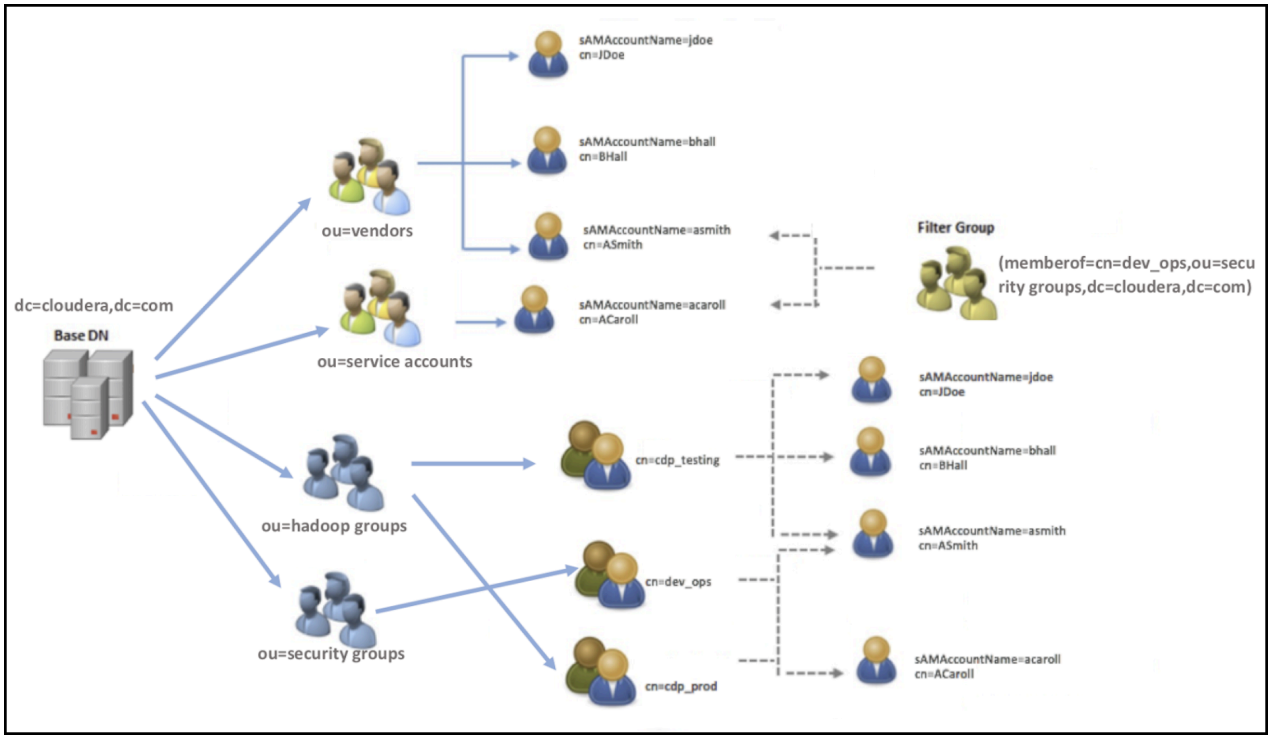
First, consider the following use-case, in order to better understand how to configure Usersync properties:

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 1: 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 2: 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 3: Using Ldapsearch to find a specific user

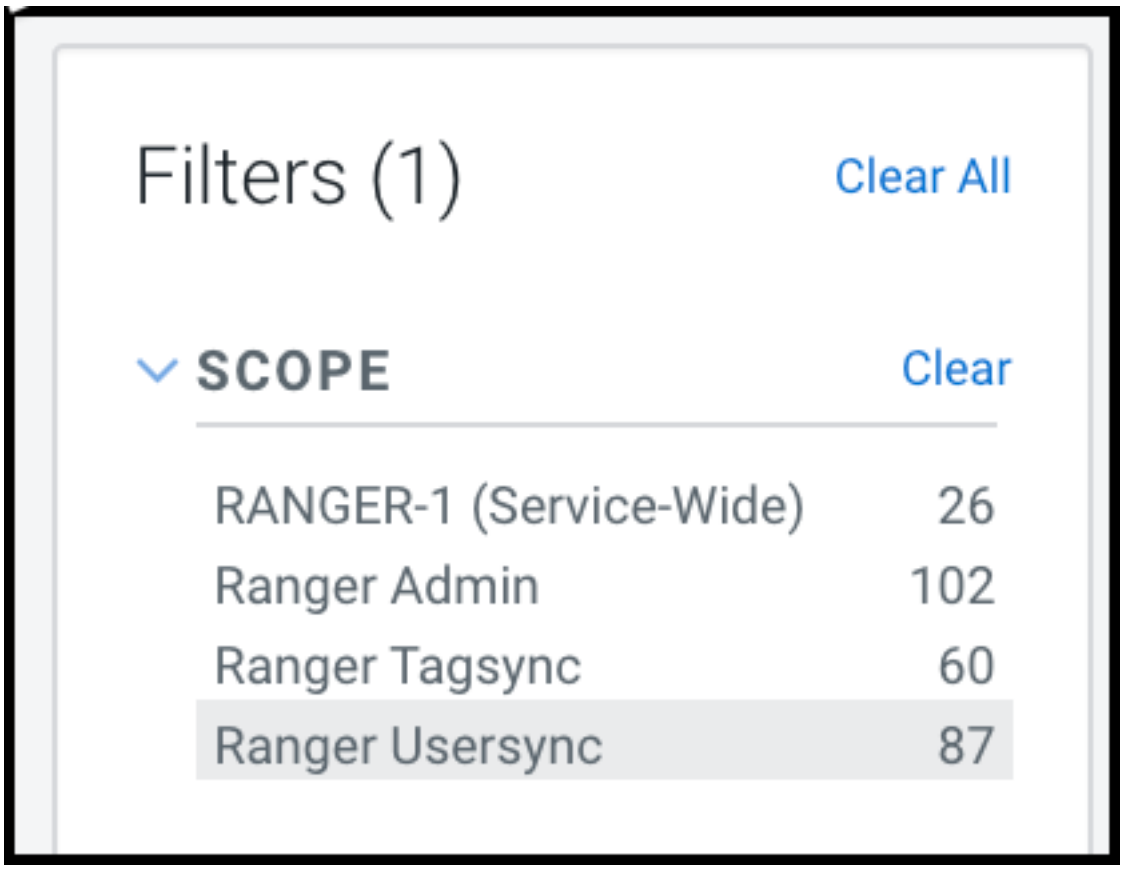
```
ldapsearch -x -LLL -h 10.10.10.10:389 -D 'cn=administrator,CN=Users,dc=cloudera,dc=com'
-W -b 'ou=Hadoop Users,dc=cloudera,dc=com' 'samaccountname=ASmith
Enter LDAP Password:
dn: CN=ASmith,ou=Hadoop Users,dc=cloudera,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: 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.

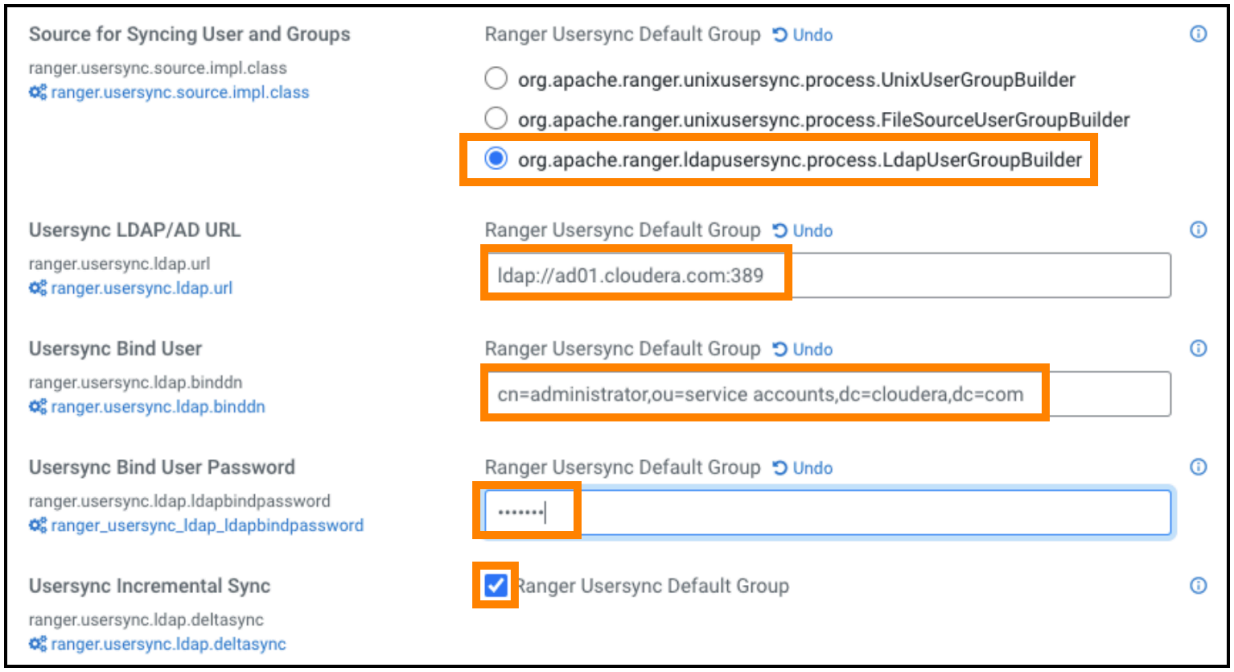
Figure 4: Filtering the Ranger Configuration Properties for Usersync



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 5: Ranger Usersync common configuration settings

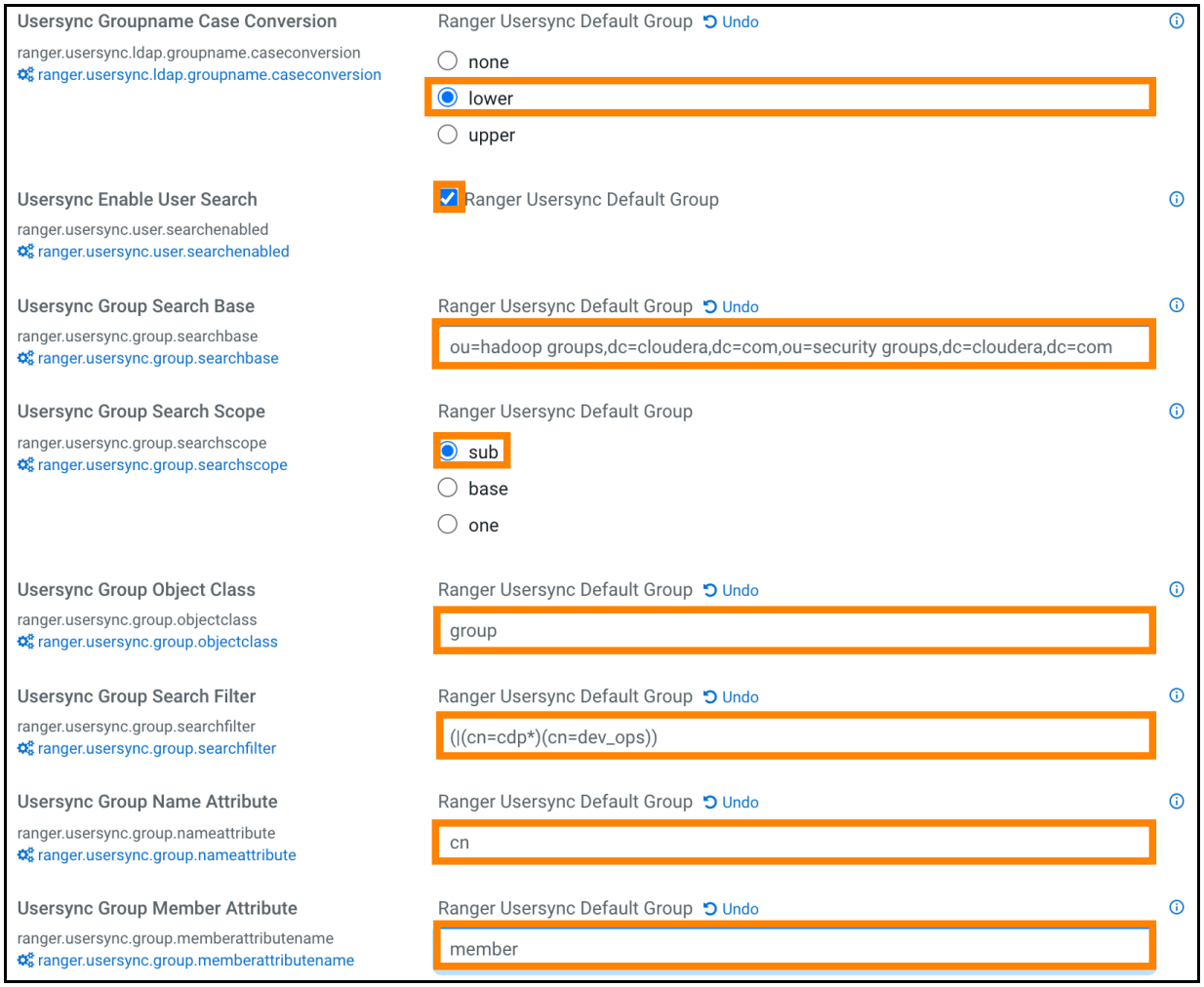


Bind credentials are for the user to query Ldap service for users and groups. Bind credentials contain two configuration properties:

- 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 6: Ranger Usersync group configuration settings



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)

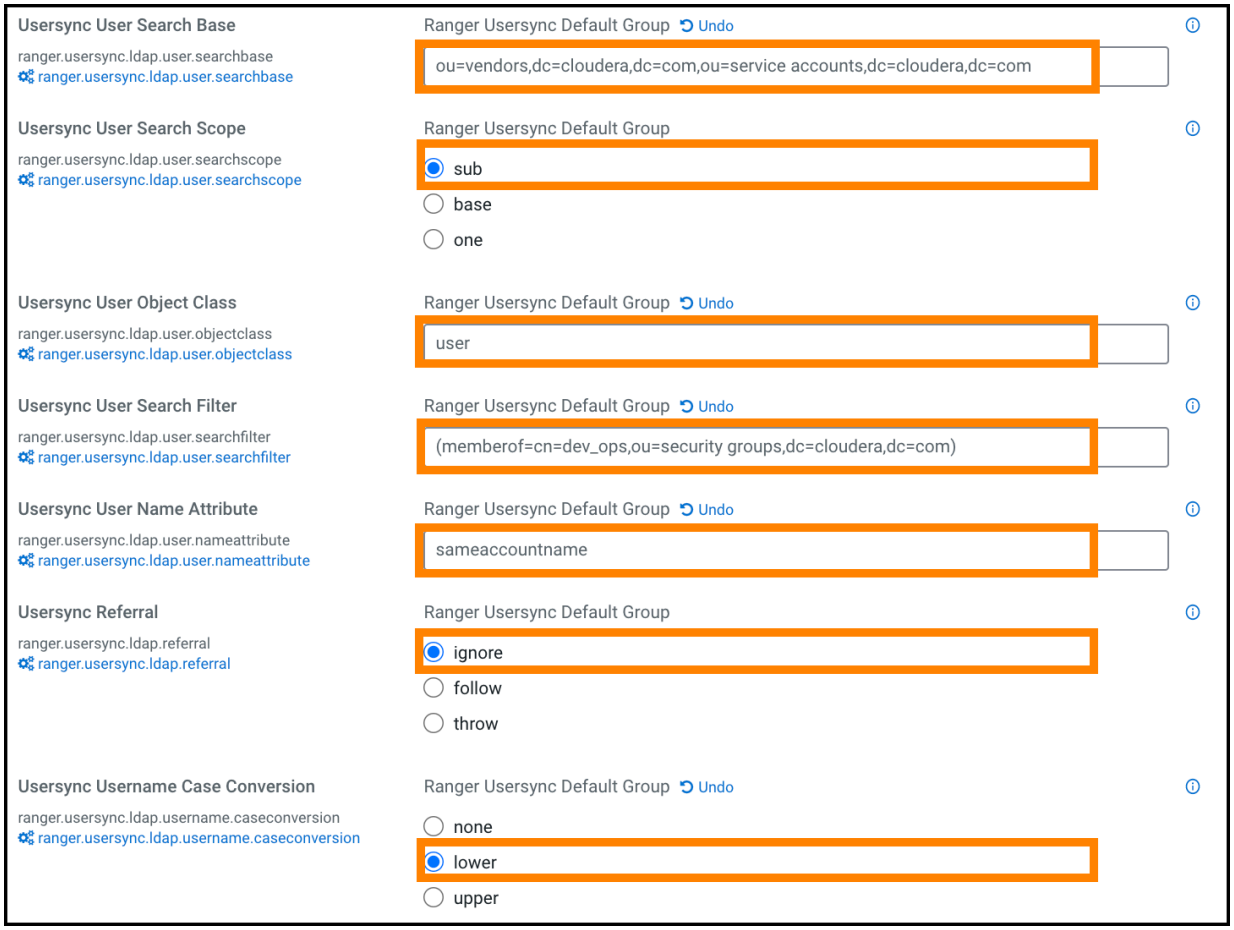
- ranger.usersync.group.searchfilter - In this example, since only 3 groups exist in hadoop groups OU and security groups OU and since all 3 require sync to Ranger, you can specify the filter as cn=* . The value for this property follows standard ldap search query filter format.

  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 7: Ranger Usersync user configuration settings



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.

Configure 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 .

Configure Ranger Usersync for Deleted Users and Groups

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

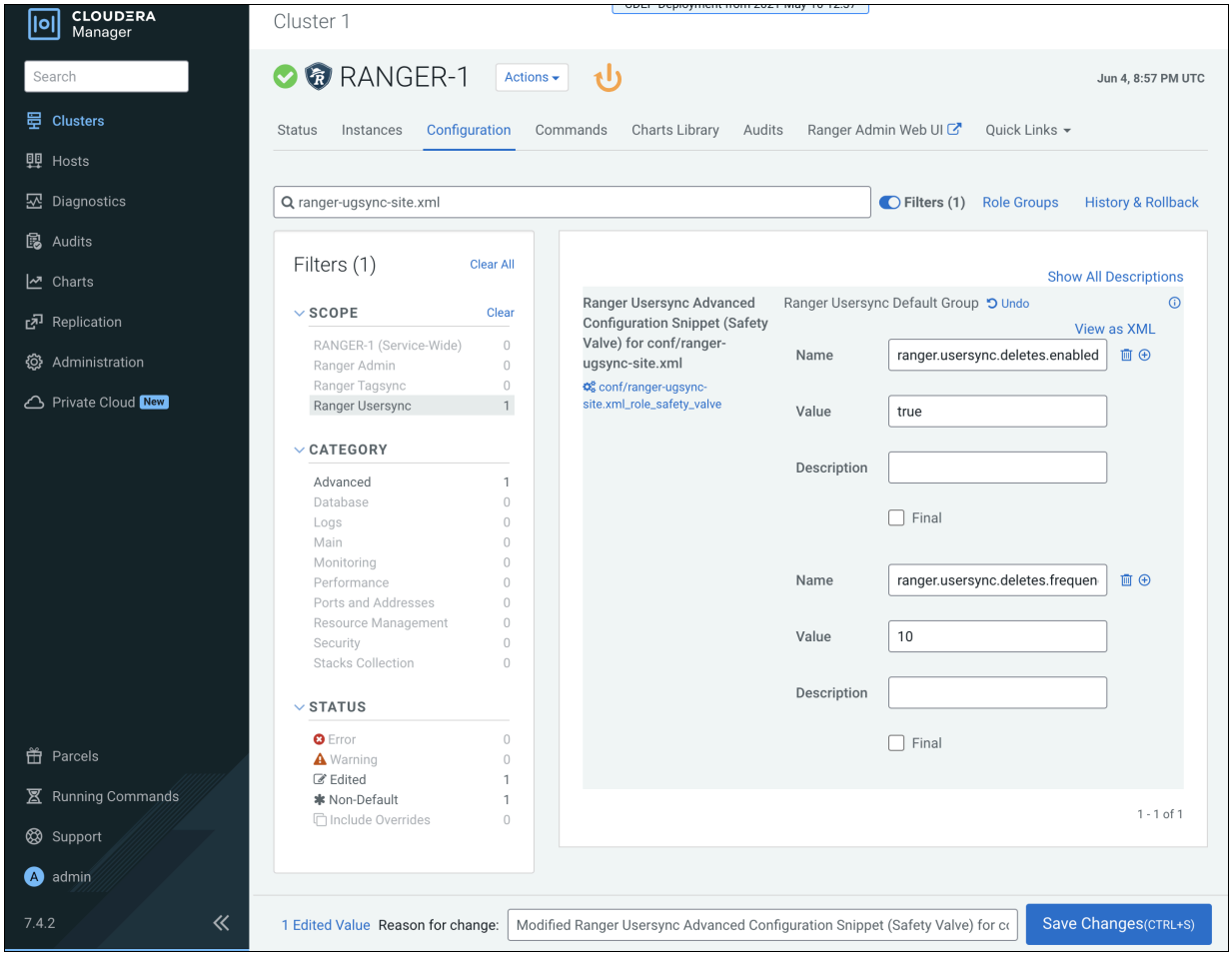
About this task

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.

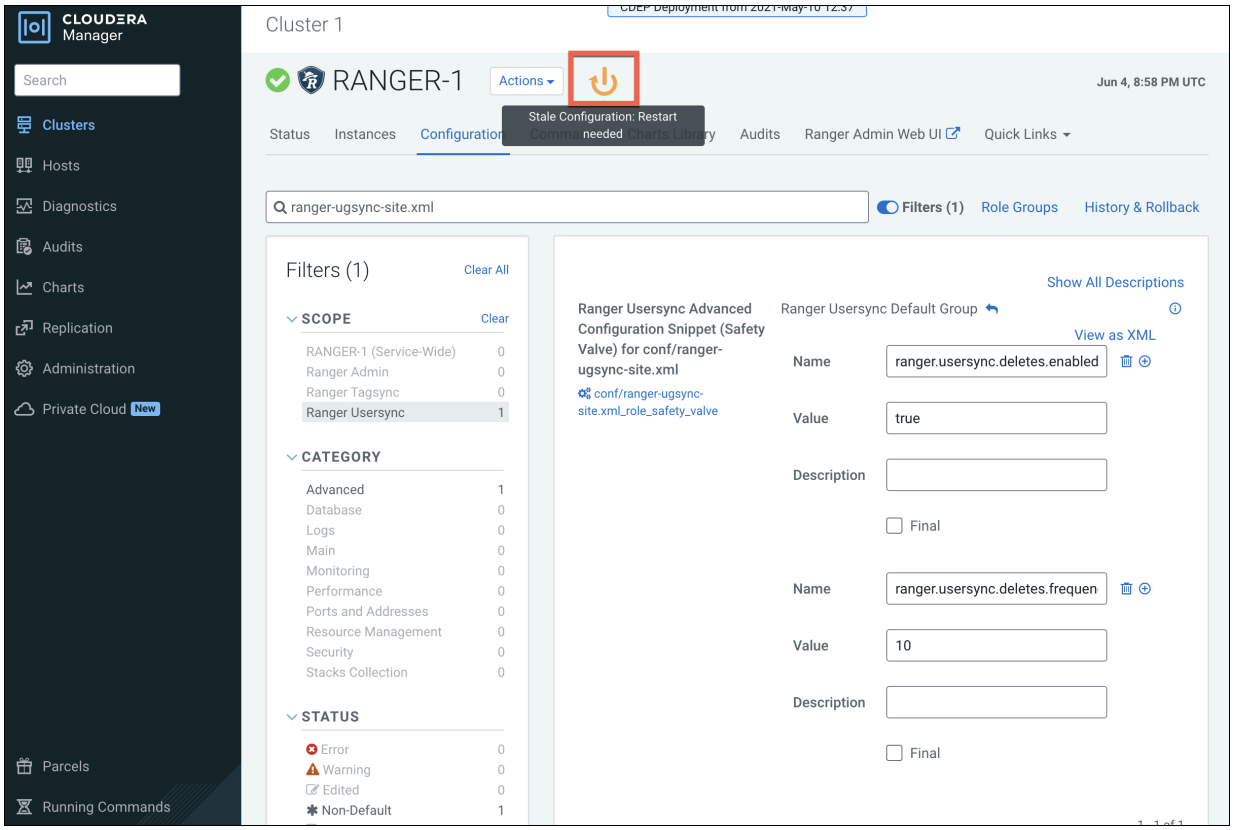
Procedure

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.

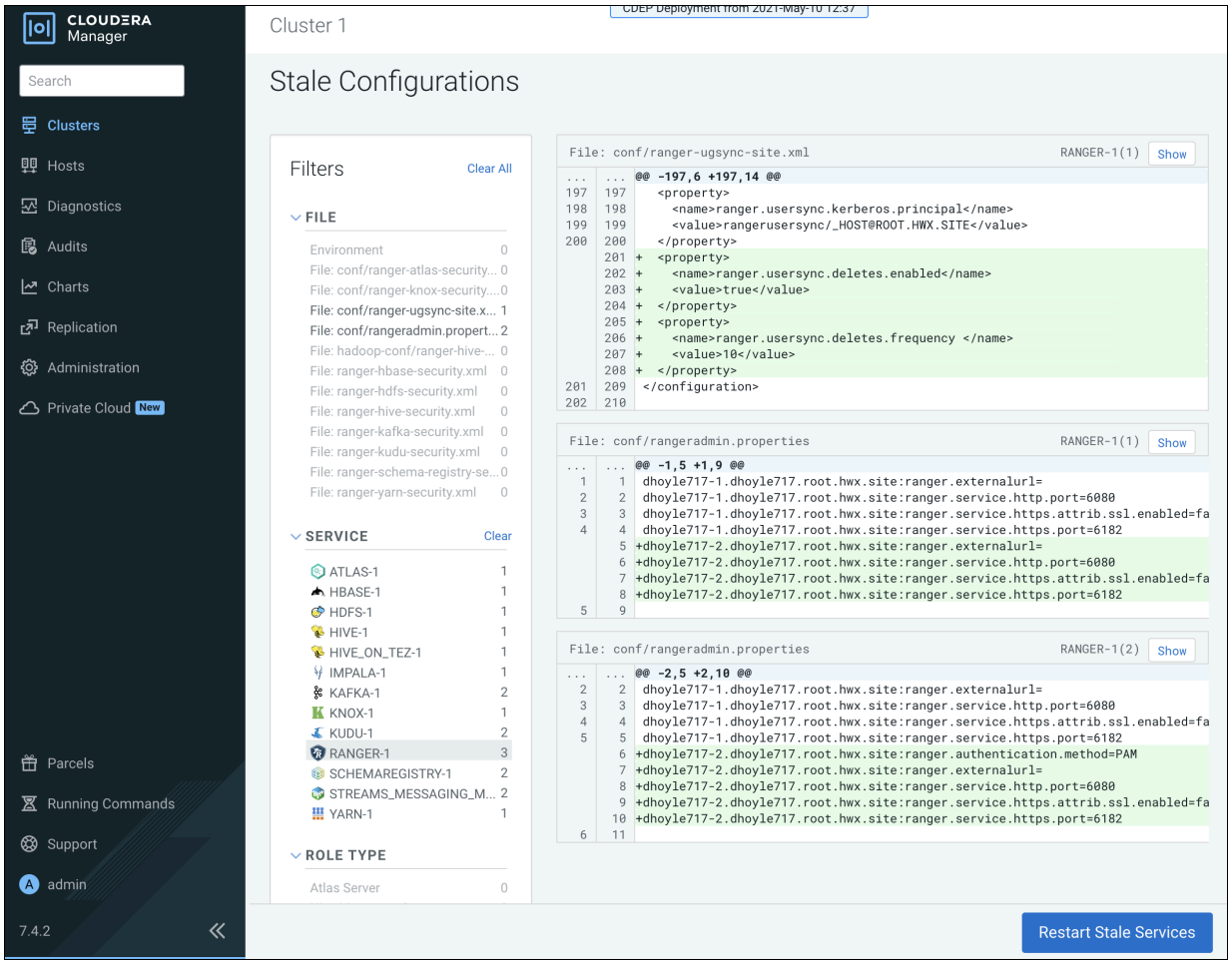
Name	Value	Description
ranger.usersync.deletes.enabled	true	Enables deleted users and groups synchronization. The default setting is false (disabled).
ranger.usersync.deletes.frequency	10	Sets the frequency of delete synchronization. The default setting is 10, or once every 10 Usersync cycles. Delete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



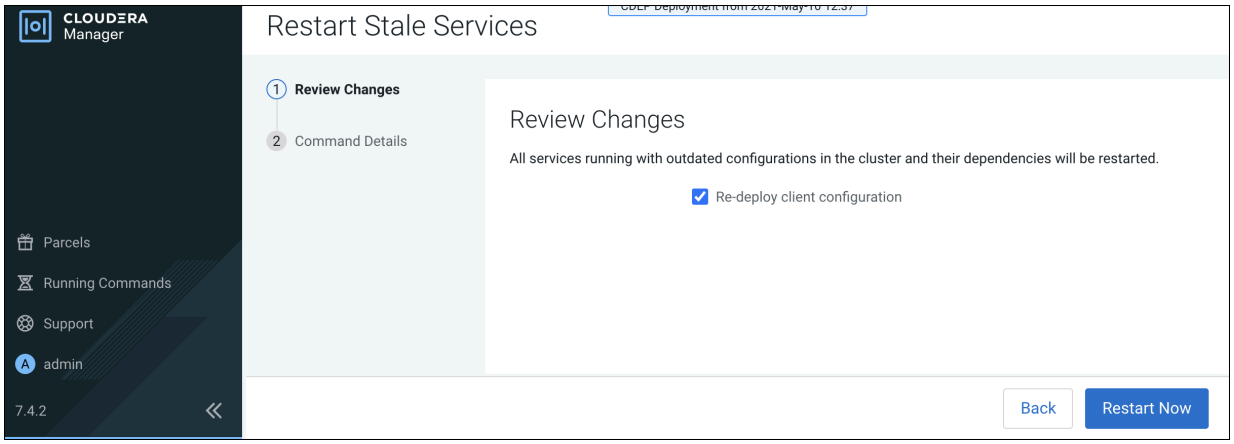
2. Click the Ranger Restart icon.



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



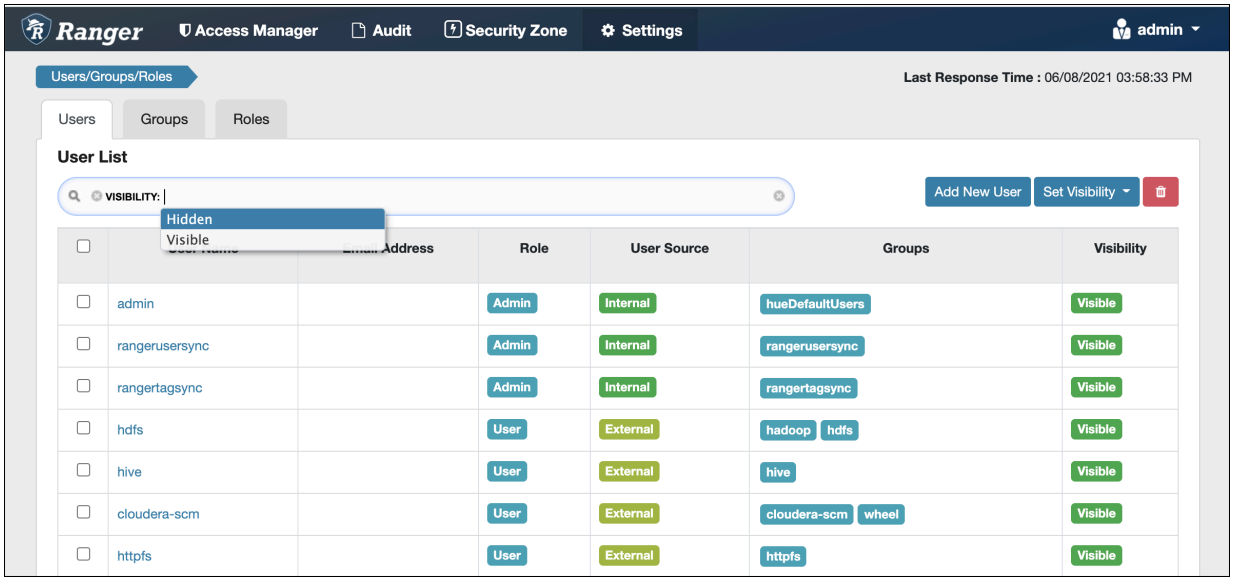
4. On the Restart Stale Services page, select the Re-deploy client configuration check box, then click Restart Now.



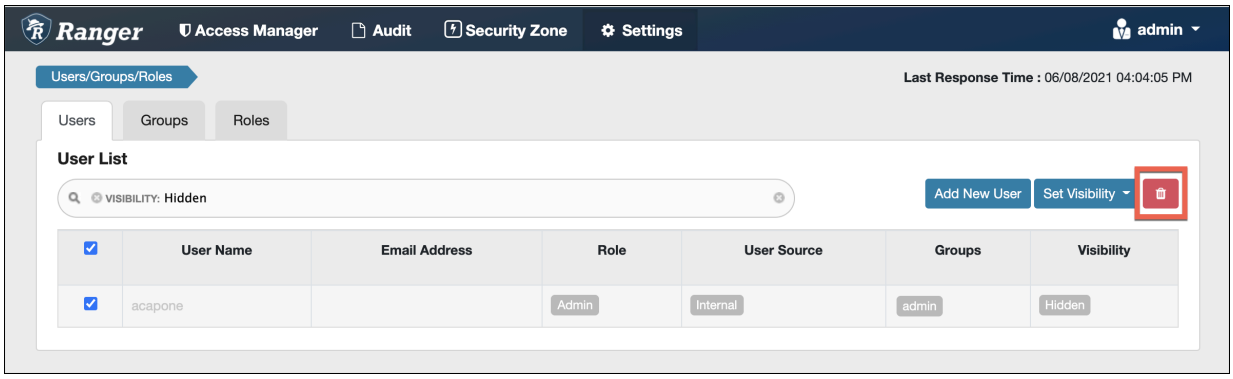
5. A progress indicator page appears while the services are being restarted. When the services have restarted, click Continue.

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.



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.



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

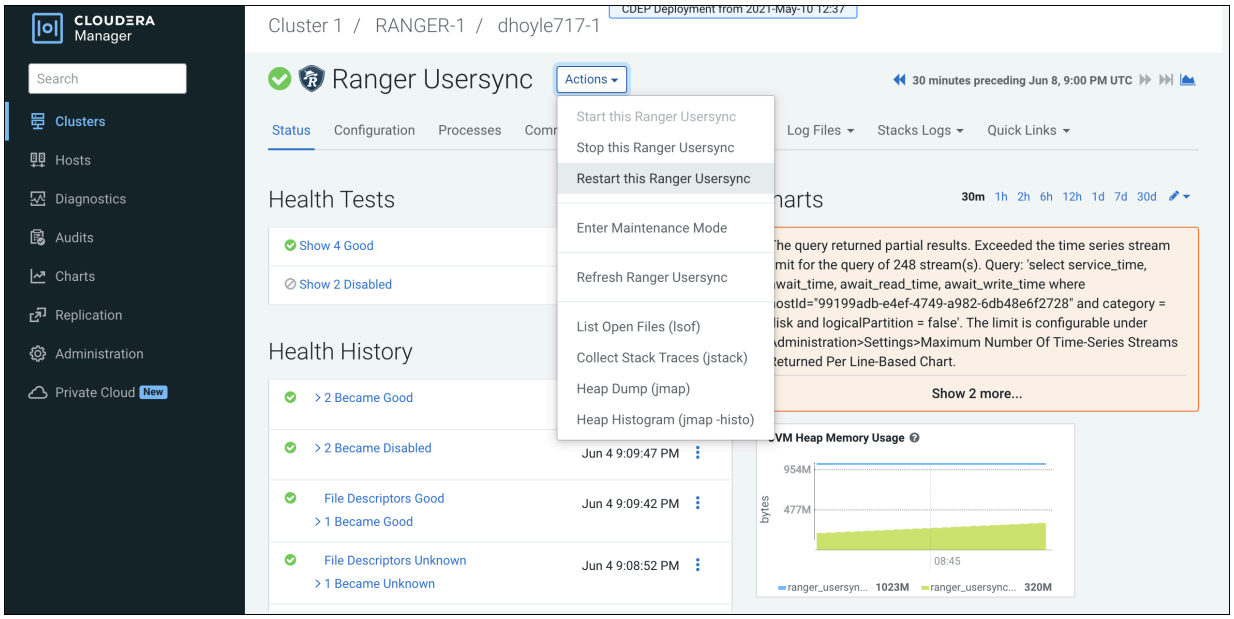
For example:

```
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.



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.

Configure 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.

Adding default service users and roles for Ranger

Cloudera Manager creates default Ranger Admin roles for the minimum set of service users by default.

Runtime releases 7.1.8 and 7.2.16 introduce a new configuration property:

Name

usersyncranger.usersync.whitelist.users.role.assignment.rules

Default Value

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

This property uses same format as ranger.usersync.group.based.role.assignment.rules. It is populated by Cloudera Manager with default service usernames. For custom principals, this configuration must be updated accordingly for the role assignments rules to be applied appropriately by Ranger usersync. 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 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 and will reset in the next Ranger usersync sync cycle.

Set credentials for Ranger Usersync

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

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 CDP 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.

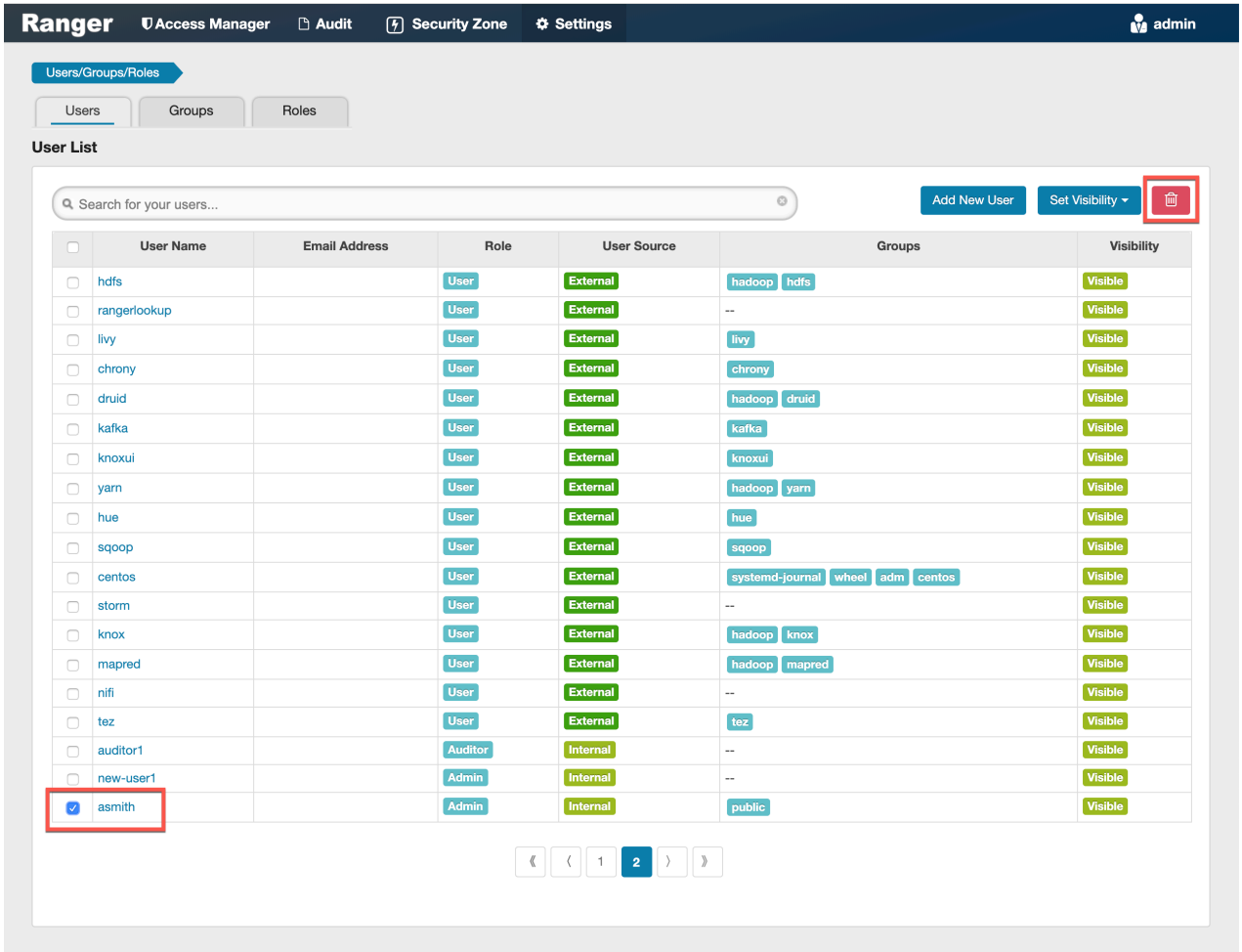
Results

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

Ranger user management

Reference information on Ranger user management, when configuring Ranger AD integration.

To delete a user, select the check box for the user in the User Name list, then click the red Delete button. Ranger removes the user from all policies.


Ranger
Access Manager
Audit
Security Zone
Settings
admin
Users/Groups/Roles
Users
Groups
Roles
User List
Search for your users...
Add New User
Set Visibility
User Name
Email Address
Role
User Source
Groups
Visibility
hdfs User External hadoop hdfs Visible
rangerlookup User External -- Visible
livy User External livy Visible
chrony User External chrony Visible
druid User External hadoop druid Visible
kafka User External kafka Visible
knoxui User External knoxui Visible
yarn User External hadoop yarn Visible
hue User External hue Visible
sqoop User External sqoop Visible
centos User External systemd-journal wheel adm centos Visible
storm User External -- Visible
knox User External hadoop knox Visible
mapred User External hadoop mapred Visible
nifi User External -- Visible
tez User External tez Visible
auditor1 Auditor Internal -- Visible
new-user1 Admin Internal -- Visible
asmith Admin Internal public Visible
1
2