Configuring Apache Ranger Authentication with UNIX, LDAP, or AD

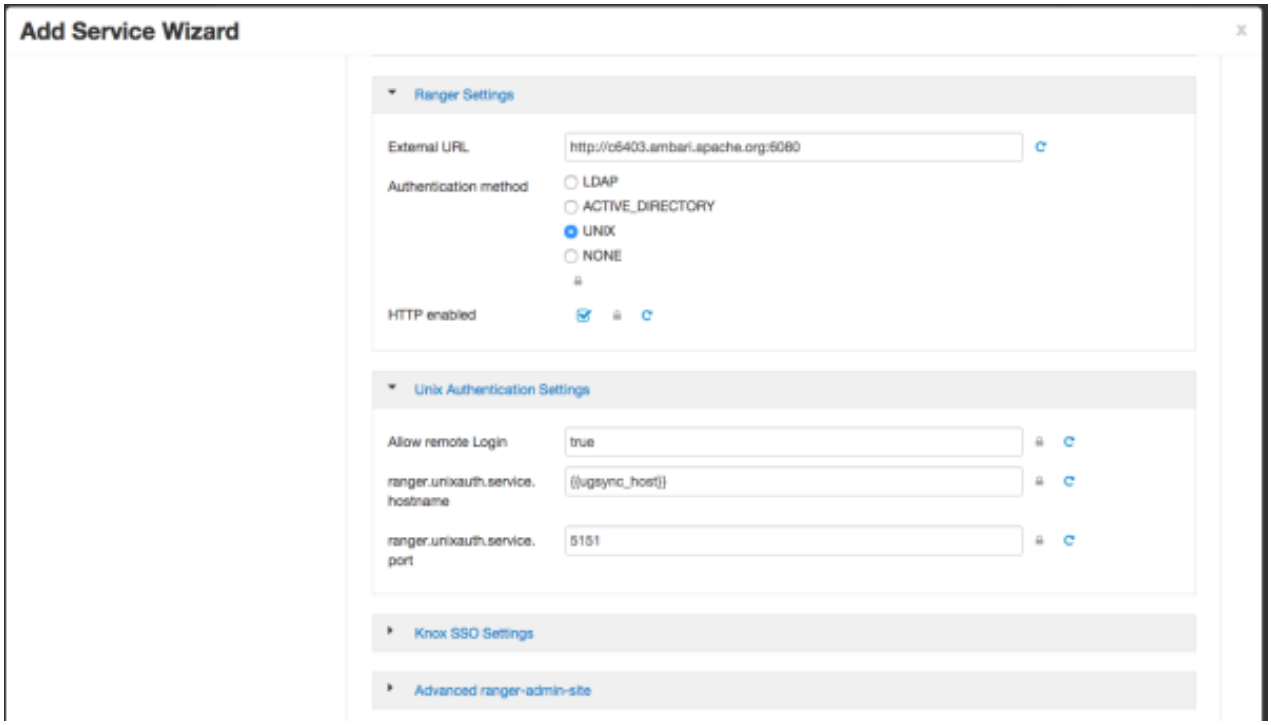
Date of Publish: 2019-08-26

https://docs.hortonworks.com

Contents

Configuring Ranger Authentication with UNIX, LDAP, or AD

This section describes how to configure the authentication method that determines who is allowed to login to the Ranger web interface. The options are local Unix, AD, or LDAP.



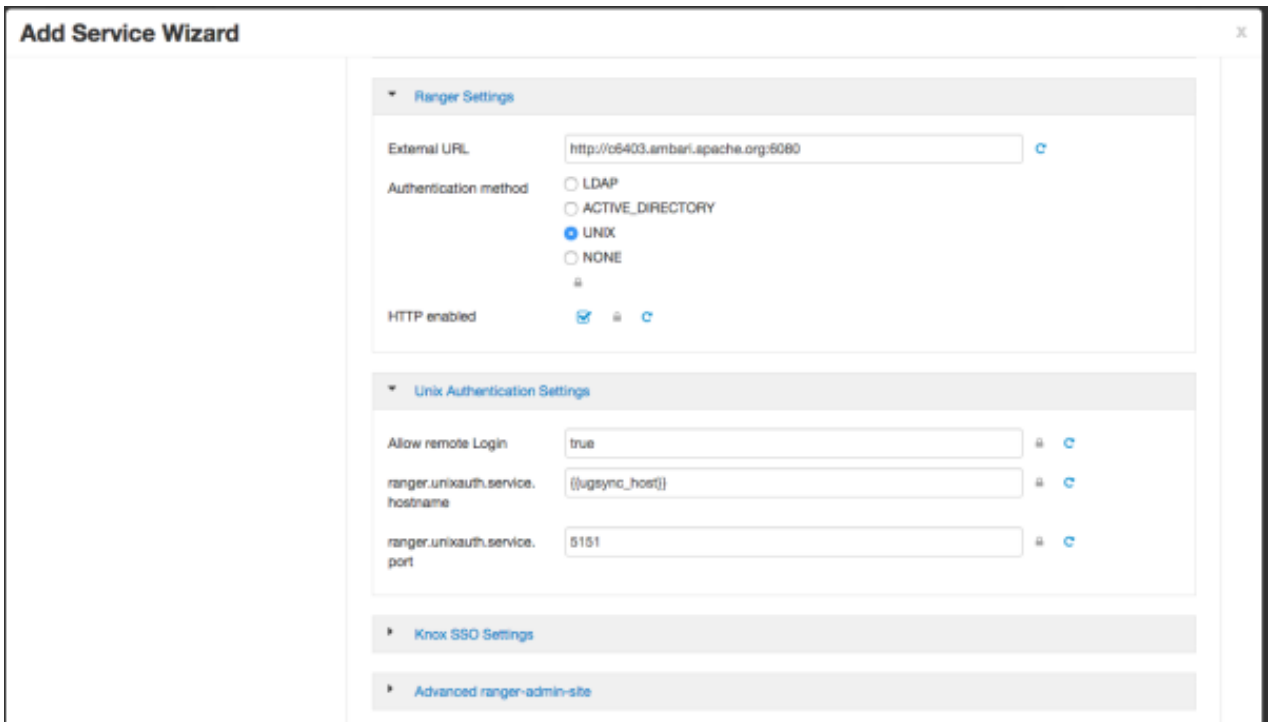
Configure Ranger Authentication for UNIX

How to configure Ranger to use Unix for user authentication.

About this task

You can configure Ranger authentication in two ways:

- During installation: **Ranger Customize Services** > **Advanced tab** > **Ranger Settings**
- After installation: **Ambari** > **Ranger** > **Configs** > **Advanced** > **Ranger Settings**



Procedure

1. From the **Ranger Settings** tab:
   a) Enter the external URL, e.g. http://my-vm.hortonworks.com:6080.
   b) Under **Authentication method**, select **UNIX**.
   c) Under HTTP enabled, make a selection. This option enables you to select HTTP/HTTPS communication for Ranger admin console. If you disable HTTP, only HTTPS is allowed. HTTP is enabled by default.
2. From the **UNIX Authentication Settings** tab, enter the following values:

Table 1: UNIX Authentication Settings

Configuration Property	Description	Default Value	Example Value	Requi
Allow remote Login	Flag to enable/disable remote login via UNIX Authentication Mode.	TRUE	TRUE	No.
ranger.unixauth.service.hostname	The FQDN where the ranger-usersync module is running (along with the UNIX Authentication Service).	localhost	myunixhost.domain.com	Yes, i selecte
ranger.unixauth.service.port	The port number where the ranger-usersync module is running the UNIX Authentication Service.	5151	5151	Yes, i selecte

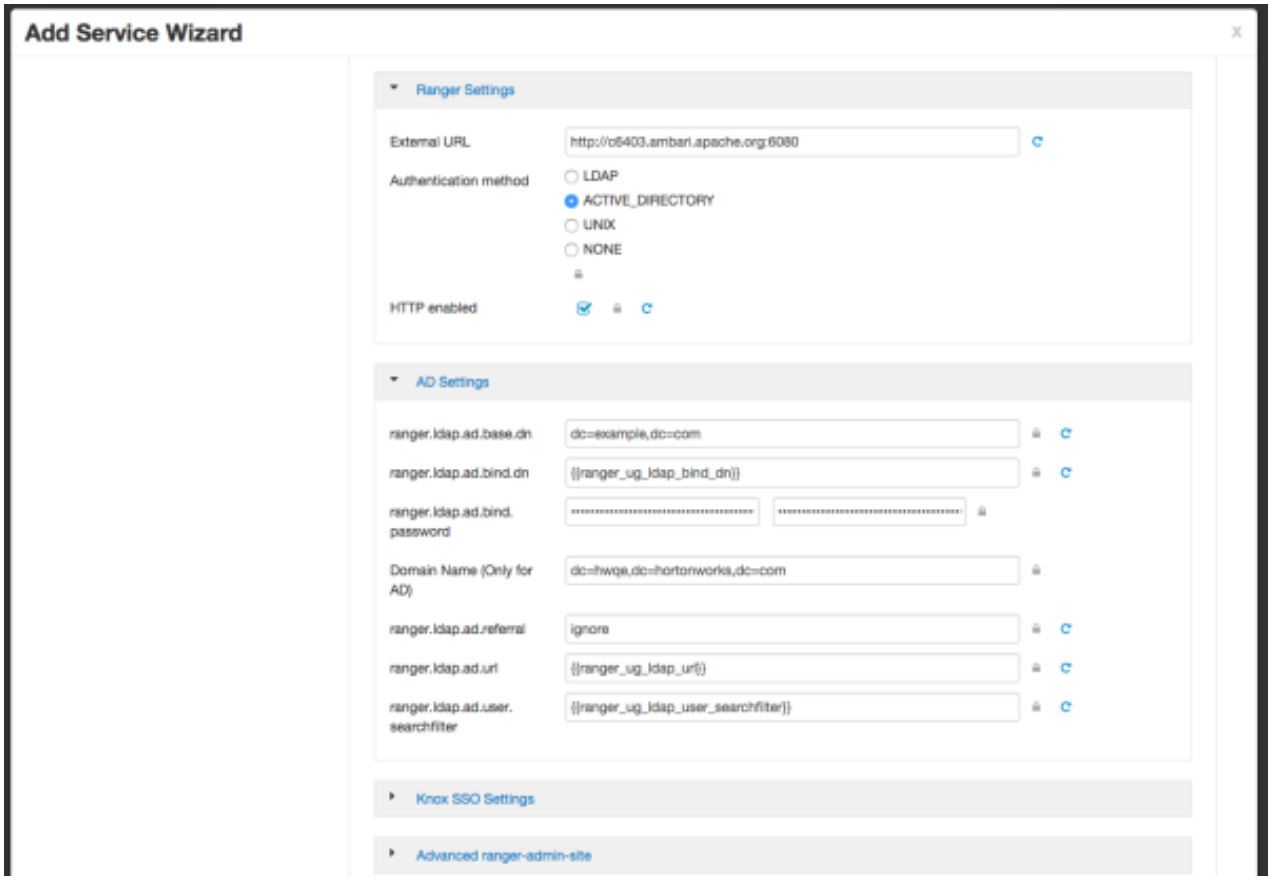
Configure Ranger Authentication for AD

How to configure Ranger to use AD for user authentication.

About this task

You can configure Ranger authentication in two ways:

- During installation: **Ranger Customize Services** > **Advanced tab** > **Ranger Settings**
- After installation: **Ambari** > **Ranger** > **Configs** > **Advanced** > **Ranger Settings**



Procedure

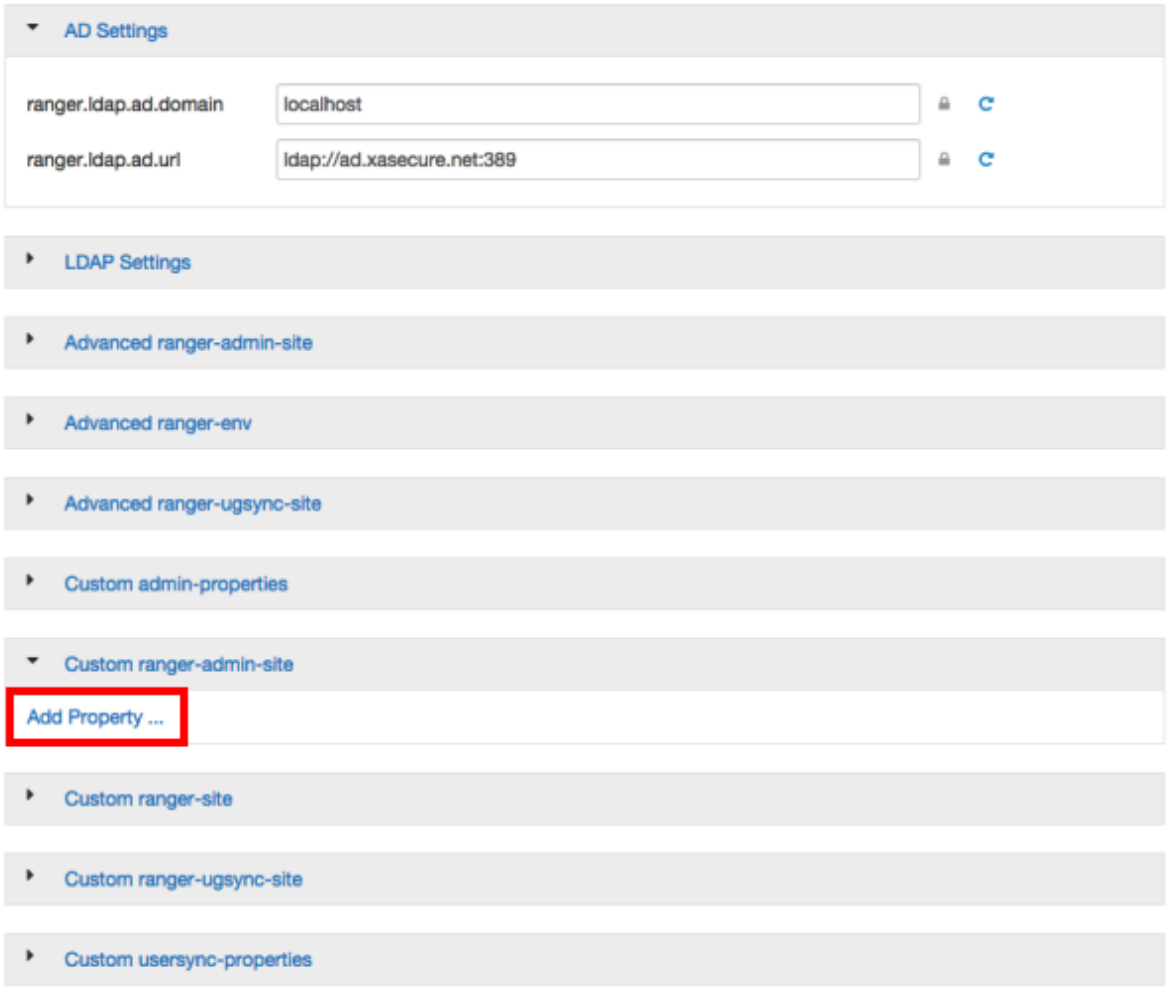
1. From the **Ranger Settings** tab:
   a) Enter the external URL, e.g. http://my-vm.hortonworks.com:6080.
   b) Under **Authentication method**, select **ACTIVE_DIRECTORY**.
   c) Under HTTP enabled, make a selection. This option enables you to select HTTP/HTTPS communication for Ranger admin console. If you disable HTTP, only HTTPS is allowed. HTTP is enabled by default.
2. From the **AD Settings** tab, enter the following values:

Property	Description	Default value	Sample values
ranger.ldap.ad.base. dn	The Distinguished Name (DN) of the starting point for directory server searches.	dc=example,dc=com	dc=example,dc=com
ranger.ldap.ad.bind.dn	The full Distinguished Name (DN), including Common Name (CN) of an LDAP user account that has privileges to search for users. This is a macro variable value that is derived from the Bind User value from Ranger User Info > Common Configs.	{{ranger_ug_ldap_bi nd_dn}}	{{ranger_ug_ldap_bi nd_dn}}
ranger.ldap.ad.bind.password	Password for the bind.dn. This is a macro variable value that is derived from the Bind User Password value from Ranger User Info > Common Configs.		
Domain Name (Only for AD)	The domain name of the AD Authentication service.		dc=example,dc=com

Property	Description	Default value	Sample values
ranger.ldap.ad.referral*	See below.	ignore	follow | ignore | throw
ranger.ldap.ad.url	The AD server URL. This is a macro variable value that is derived from the LDAP/AD URL value from Ranger User Info > Common Configs.	{{ranger_ug_ldap_url }}	{{ranger_ug_ldap_url }}
ranger.ldap.ad.user.searchfilter	The search filter used for Bind Authentication. This is a macro variable value that is derived from the User Search Filter value from Ranger User Info > User Configs.	{{ranger_ug_ldap_user_searchfilter}}	{{ranger_ug_ldap_user_searchfilter}}

3. Optional: Custom ranger-admin-site Settings for Active Directory:

   a) Select Custom ranger-admin-site, then click Add Property.



   b) The following table shows the Custom ranger-admin-site settings required for Active Directory (AD) authentication:

Key	Value
ranger.ldap.ad.base.dn	dc=example,dc=com
ranger.ldap.ad.bind.dn	cn=adadmin,cn=Users,dc=example,dc=com
ranger.ldap.ad.bind.password	Secret123!

Key	Value
ranger.ldap.ad.referral*	follow | ignore | throw

*

There are three possible values for ranger.ldap.ad.referral: follow, throw, and ignore. The recommended setting is follow.

When searching a directory, the server might return several search results, along with a few continuation references that show where to obtain further results. These results and references might be interleaved at the protocol level.

- When this property is set to follow, the AD service provider processes all of the normal entries first, and then follows the continuation references.
- When this property is set to throw, all of the normal entries are returned in the enumeration first, before theReferralException is thrown. By contrast, a "referral" error response is processed immediately when this property is set to follow or throw.
- When this property is set to ignore, it indicates that the server should return referral entries as ordinary entries (or plain text). This might return partial results for the search. In the case of AD, a PartialResultException is returned when referrals are encountered while search results are processed.

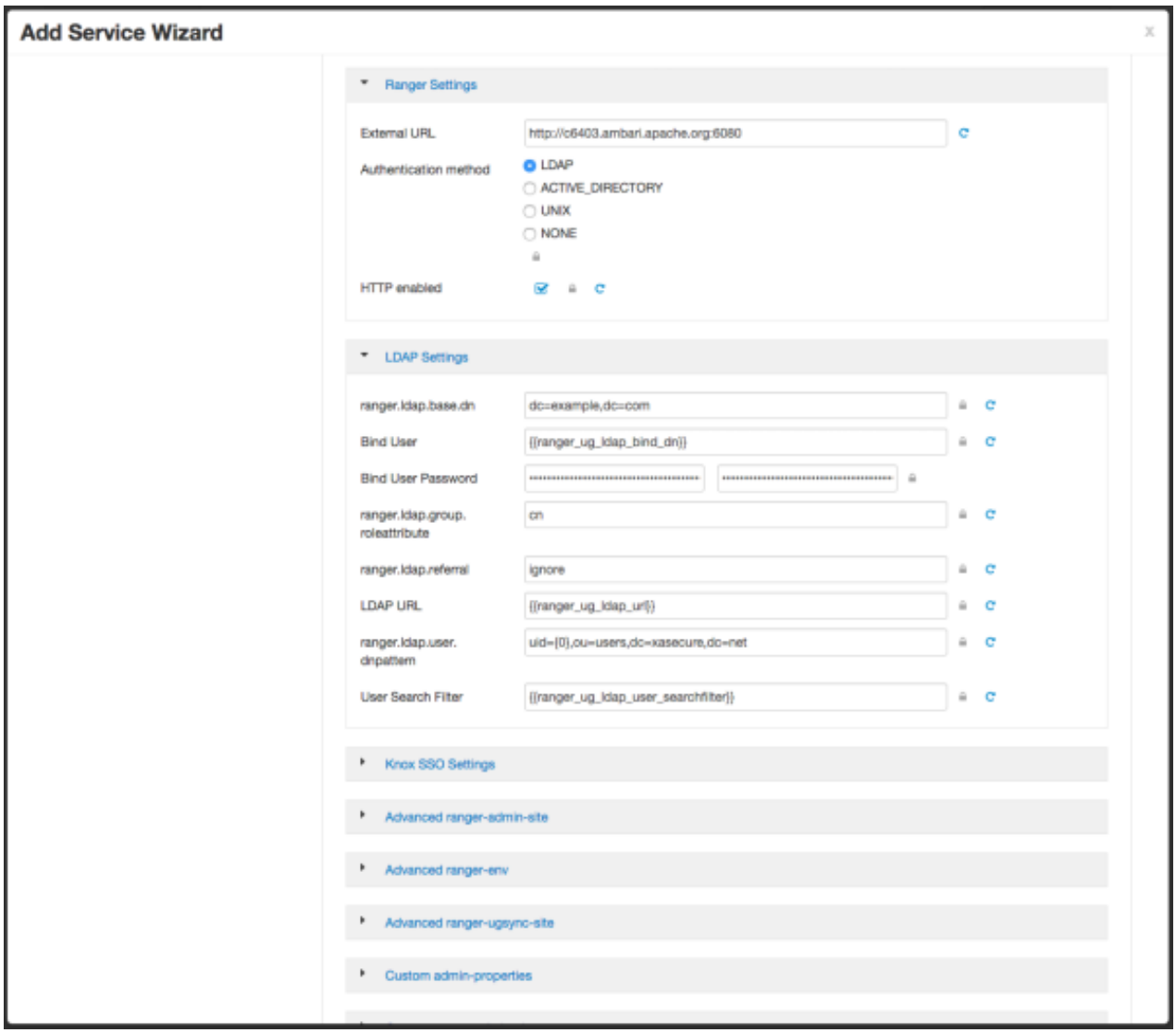
Configure Ranger Authentication for LDAP

How to configure Ranger to use LDAP for user authentication.

About this task

You can configure Ranger authentication in two ways:

- During installation: **Ranger Customize Services** > **Advanced tab** > **Ranger Settings**
- After installation: **Ambari** > **Ranger** > **Configs** > **Advanced** > **Ranger Settings**



Procedure

1. From the **Ranger Settings** tab:
   a) Enter the external URL, e.g. http://my-vm.hortonworks.com:6080.
   b) Under **Authentication method**, select **LDAP**.
   c) Under HTTP enabled, make a selection. This option enables you to select HTTP/HTTPS communication for Ranger admin console. If you disable HTTP, only HTTPS is allowed. HTTP is enabled by default.
2. From the **LDAP Settings** tab, enter the following values:

Property	Description	Default value	Sample values
Group Search Base		{{ranger_ug_ldap_group_searchbase}}	
Group Search Filter		{{ranger_ug_ldap_group_searchfilter}}	
LDAP URL		{{ranger_ug_ldap_url}}	
Bind User		{{ranger_ug_ldap_bind_dn}}	
Bind User Password		N/A	
User Search Filter		(uid={0})	
ranger.ldap.base.dn		dc=example,dc=com	

Property	Description	Default value	Sample values
ranger.ldap.group.roleattribute		cn	
ranger.ldap.referral	See below.	ignore	follow|throw|ignore
ranger.ldap.user.dnpattern		uid={0},ou=users,dc=xasecure,dc=net	

There are three possible values for ranger.ldap.ad.referral: follow, throw, and ignore. The recommended setting is follow.

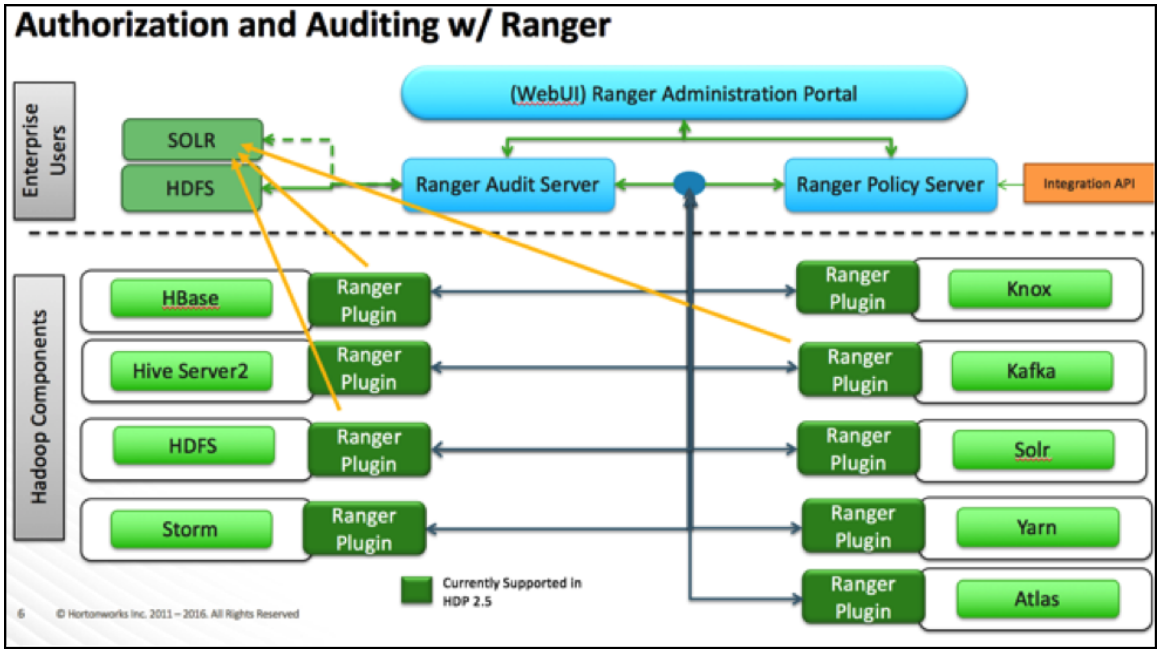
When searching a directory, the server might return several search results, along with a few continuation references that show where to obtain further results. These results and references might be interleaved at the protocol level.

- When this property is set to follow, the AD service provider processes all of the normal entries first, and then follows the continuation references.
- When this property is set to throw, all of the normal entries are returned in the enumeration first, before the ReferralException is thrown. By contrast, a "referral" error response is processed immediately when this property is set to follow or throw.
- When this property is set to ignore, it indicates that the server should return referral entries as ordinary entries (or plain text). This might return partial results for the search. In the case of AD, a PartialResultException is returned when referrals are encountered while search results are processed.

Ranger AD Integration

A conceptual overview of Ranger-AD integration architecture.

Ranger AD Integration: Architecture Overview



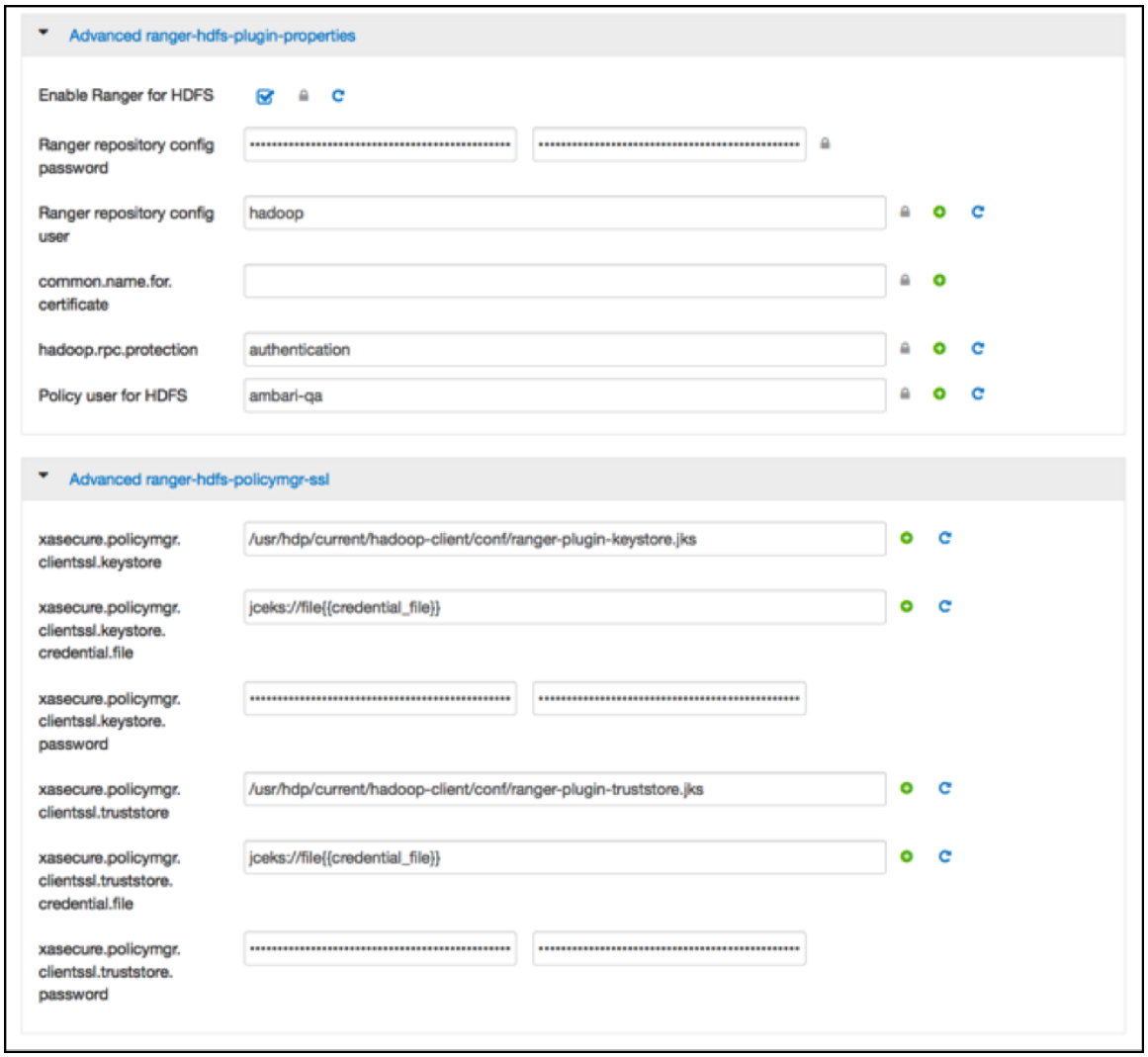
When a Ranger plugin for a component (like HBase or HDFS) is activated, Ranger will be in full control of any access. There is a two-way communication between the Ranger plugin and Ranger (Admin) Policy Server (RPS):

1. Plugins to RPS: Ranger plugins regularly call the RPS to see if new policies were defined in the Ranger Administration Portal (RAP). Generally allow for 30 sec. for a policy to be updated.
2. RPS to components: The RPS queries the component for meta objects that live on the component to base policies upon (this provides the autocomplete and dropdown list when defining policies.)

The first communication channel (Plugins to RPS) is essential for the plugin to function whereas the second (RPS to components) is optional. It would still be possible to define and enforce policies if the second does not work, but you will not have autocomplete during policy definition.

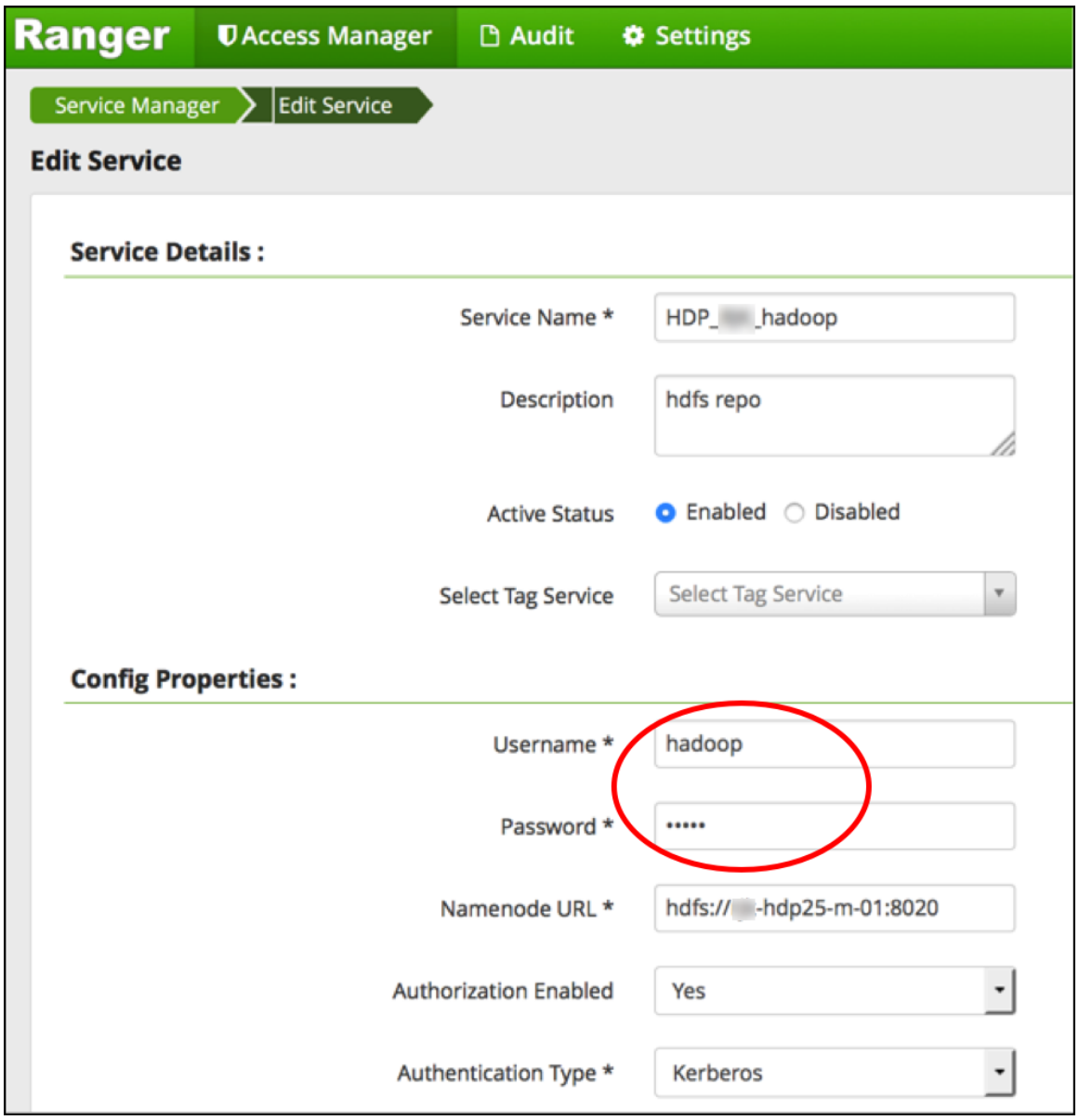
Configuration details on both communication channels are configured on both Ambari configuration for the component and on the RAP.

Example for HDFS plugin:

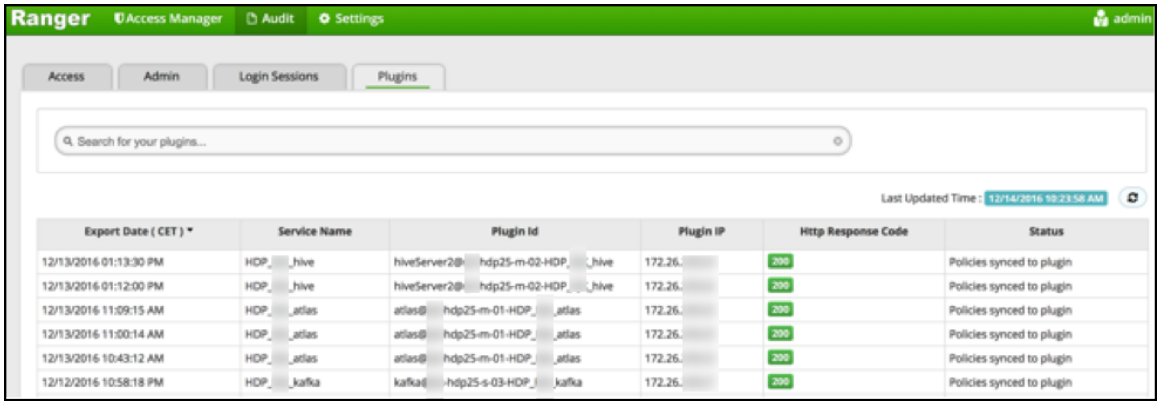


The ‘Ranger repository config user’ is the one that involved the second communication channel (RPS to components) for getting metadata from HDFS (like HDFS folders) across. The settings on the HDFS configuration have to match those set at the Ranger end (Access Manager > Resource Based Policies > HDFS >

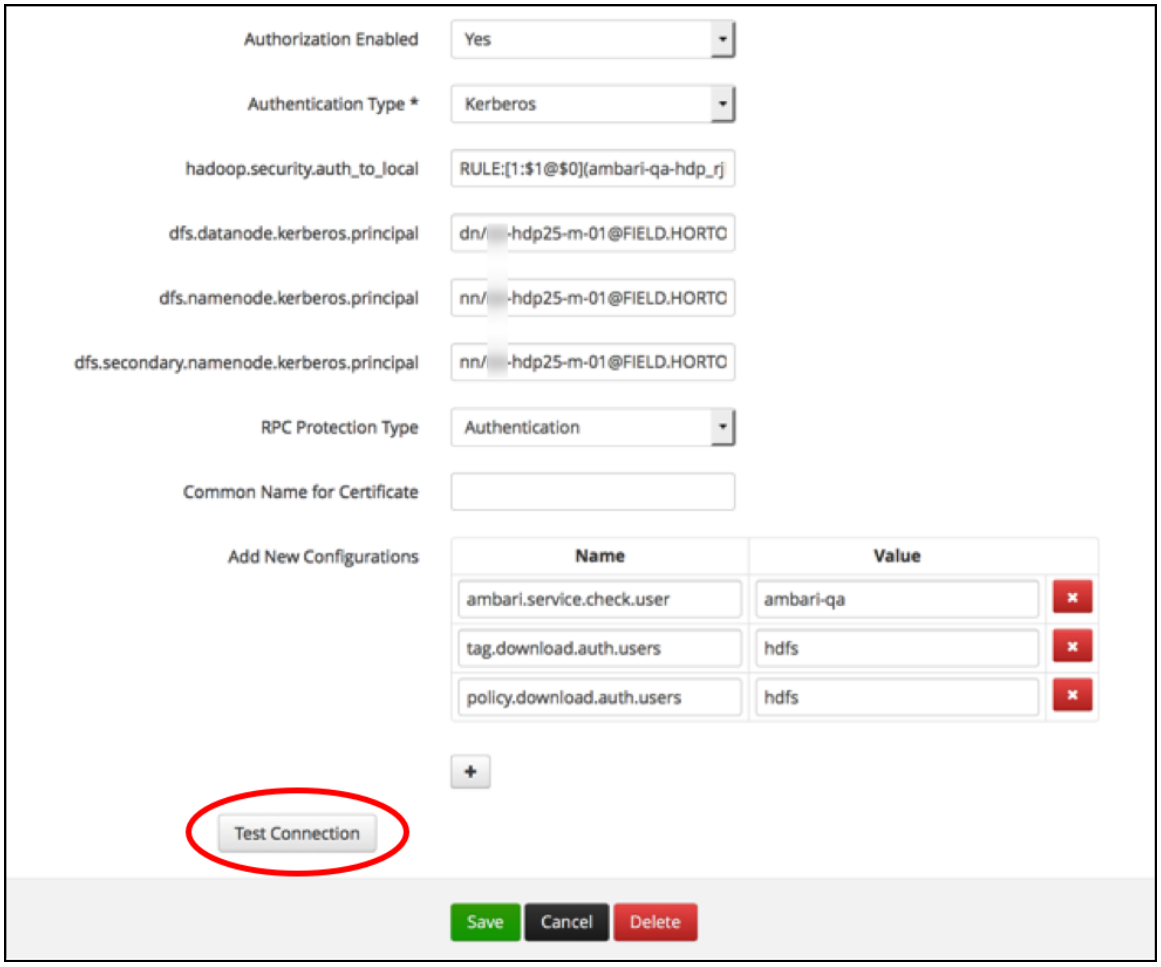
:



To verify if the paramount first communication channel (Plugins to RPS) works can be done by having a look in the RAP at Audit > Plugins:



To verify the second communication channel (RPS to components) press the 'Test Connection' button (Access Manager > Resource Based Policies > HDFS >

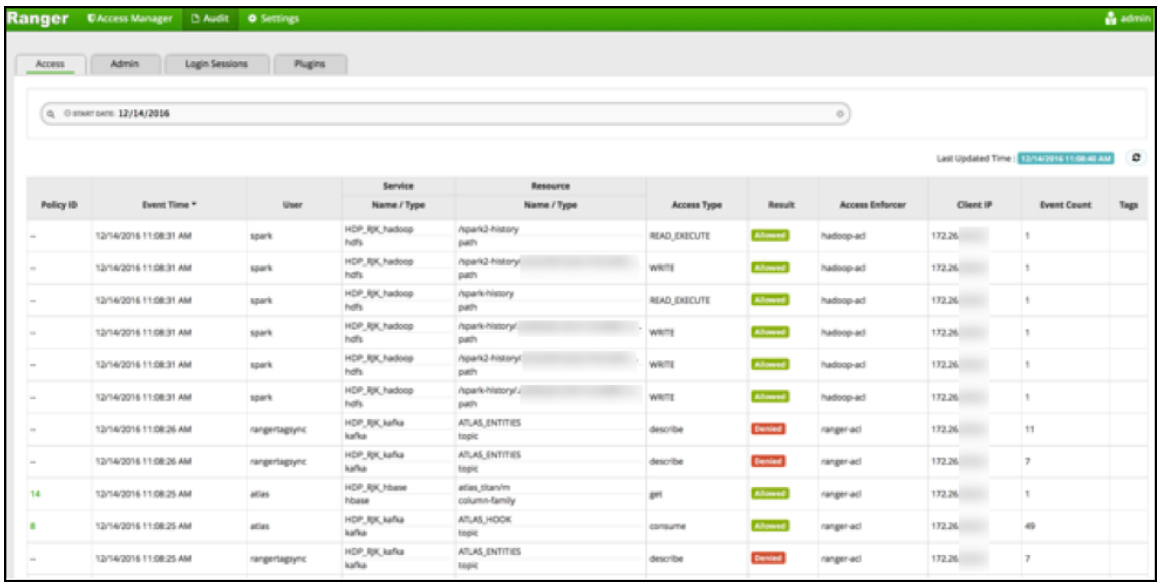


If the settings are right you'll get:

Ranger AD Integration: Ranger Audit

Ranger plugins furthermore send their audit event (whether access was granted or not and based on which policy) directly to the configured sink for audits, which can be HDFS, Solr or both. This is indicated by the yellow arrows in the architectural graph.

The audit access tab on the RAP (Audit > Access) is only populated if Solr is used as sink.



This screen points out an important Ranger feature. When the plugin is enabled AND no specific policy is in place for access to some object, the plugin will fall back to enforcing the standard component level Access Control Lists (ACL's). For HDFS that would be the user : rwx / group : rwx / other : rwx ACL's on folders and files.

Once this defaulting to component ACL's happens the audit events show a ' - ' in the 'Policy ID' column instead of a policy number. If a Ranger policy was in control of allowing/denying the policy number is shown.

Ranger AD Integration: Overview

Rangers AD Integration has 2 levels:

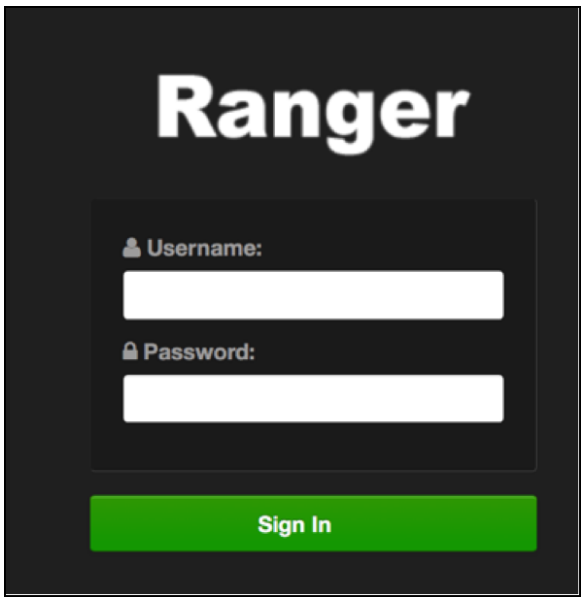
1. Ranger UI authentication (which users may log on to Ranger itself?)
2. Ranger User / group sync (which users / groups to define policies for?)

The configuration of both is done entirely on Ambari.

Ranger UI Authentication

Reference information on Ranger UI authentication, when configuring Ranger AD integration.

This is an extra AD level filter option on top of Kerberos authentication that maps to:

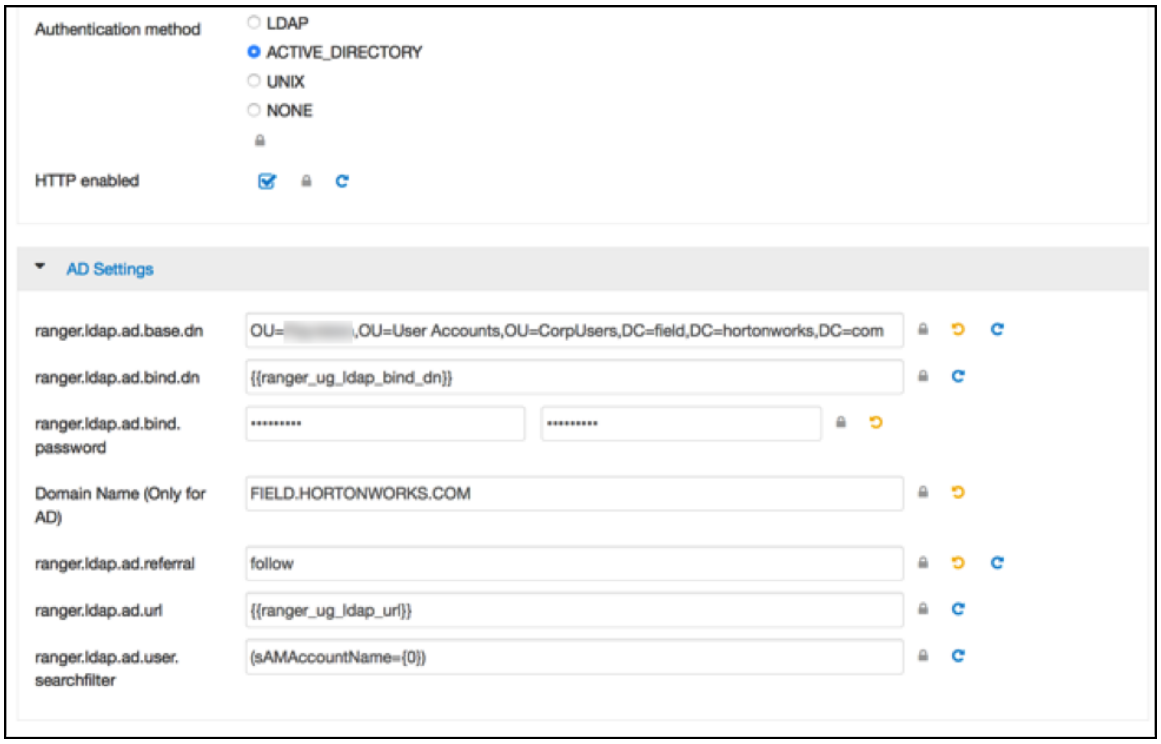


For working with AD there are 2 options for defining who can access the Ranger UI; LDAP or ACTIVE_DIRECTORY. There is not much difference between them, just another set of properties.

Some of the configuration is in fact shared with the configuration of Ranger usersync as can be seen by the property with formats like ranger_ug_ldap_bind_dn. These properties are provided at runtime only with the value of another property by that name.

ACTIVE_DIRECTORY

The configuration for it is on Ambari > Ranger > Configs > Advanced:



The ranger.ldap.ad.base.dn determines the base of any search, so users not on this OU tree path can not be authenticated.

The ranger.ldap.ad.user.searchfilter is a dynamic filter that maps the user name in the Ranger Web UI login screen to sAMAccountName. For example, the AD sAMAccountName property has example values like k.reshi and d.alora so make sure to enter a matching value for 'Username' in the logon dialogue.

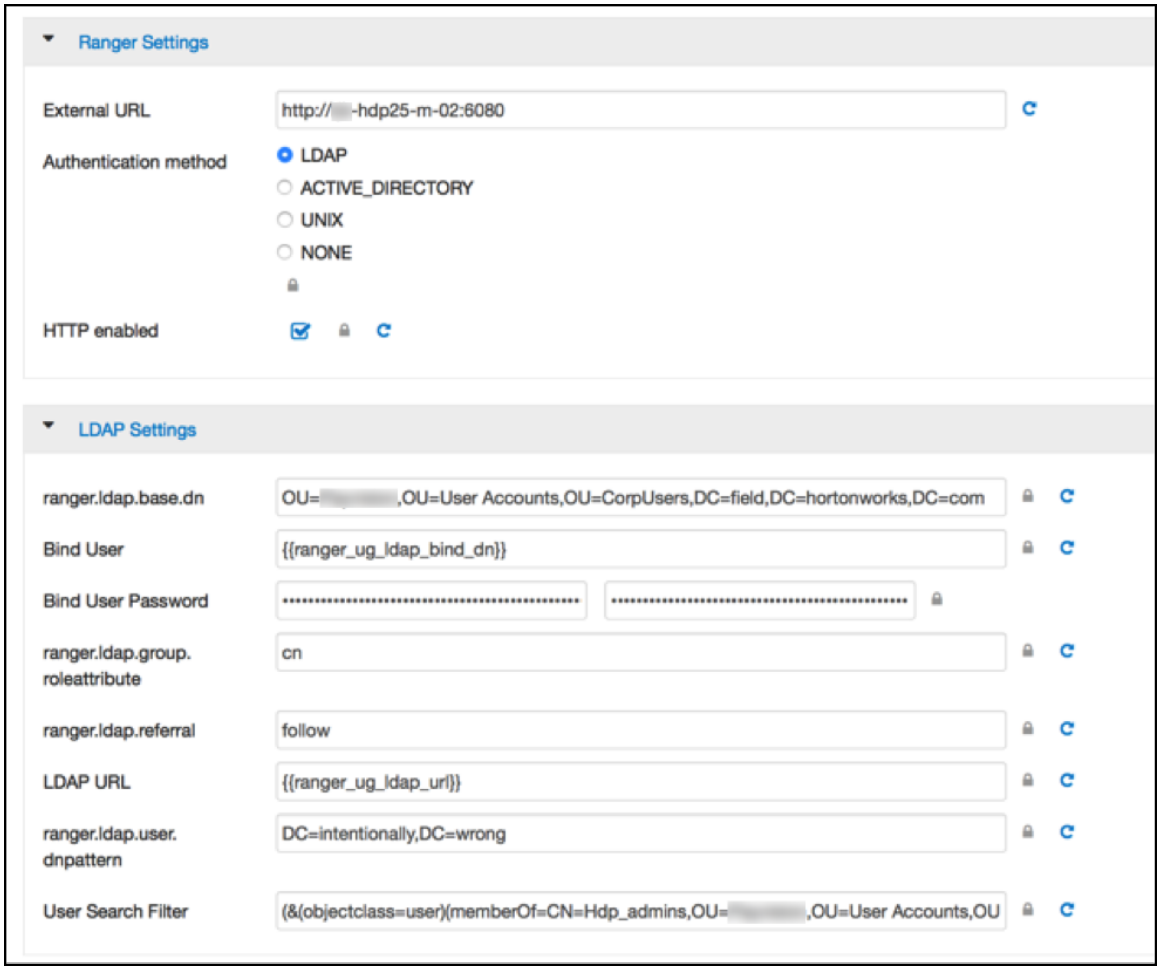
With ACTIVE_DIRECTORY it is not possible to limit the scope of users that can access Ranger UI any further by refining the ranger.ldap.ad.user.searchfilter even further to :

(&(memberOf=CN=Hdp_admins,OU=Company,OU=User Accounts,OU=CorpUsers,DC=field,DC=hortonworks,DC=com)(sAMAccountName={0}))

This does NOT work with the ACTIVE_DIRECTORY option.

LDAP

The other LDAP related properties do allow for more fine tuning:



There is 1 catch though; the ranger.ldap.user.dnpattern is evaluated first, so usually putting a value like:

CN={0},OU=London,OU=Company,OU=User Accounts,OU=CorpUsers,DC=field,DC=hortonworks,DC=com

Would work, but has 2 by-effects; first users would have to log on with their 'long username' (like 'Kvothe Reshi / Denna Alora') which would also mean that policies would have to be updated using that long name in stead of the k.reshi short name variant.

Second traversing AD by DN patterns does not allow for applying group filters at all. In the syntax above only users directly in OU=London would be able to log on.

That adverse behavior can be worked around by intentionally putting a DN pattern (DC=intentionally,DC=wrong) in the ranger.ldap.user.dnpattern property AND a valid filter in **User Search Filter**:

(&(objectclass=user)(memberOf=CN=Hdp_admins,OU=Company,OU=User Accounts,OU=CorpUsers,DC=field,DC=hortonworks,DC=com)(sAMAccountName={0}))

This works because the filter is only applied after the DN pattern query on AD does not return anything. If it does, then the **User Search Filter** is not applied.

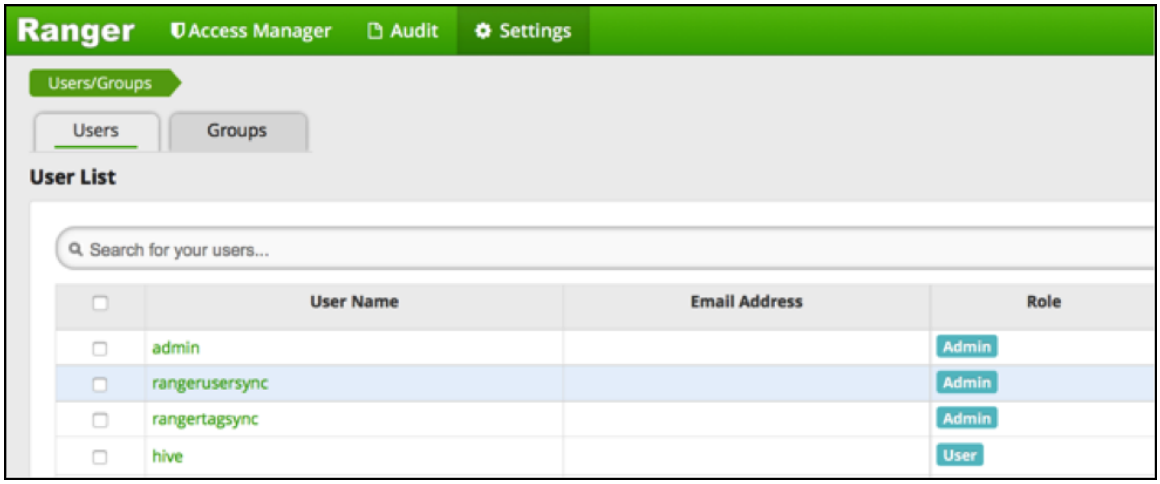
Ranger has a very simple approach to the internal user list that is kept in a relational schema. That list contains all users that were synced with AD ever, and all those users can potentially log on to Ranger UI. But only admin users can really do anything policy related things on the Ranger UI (see next section).

Beware that all this is still only about authentication to Ranger. Someone from the 'Hdp_admins' group would still not have a Ranger admin role.

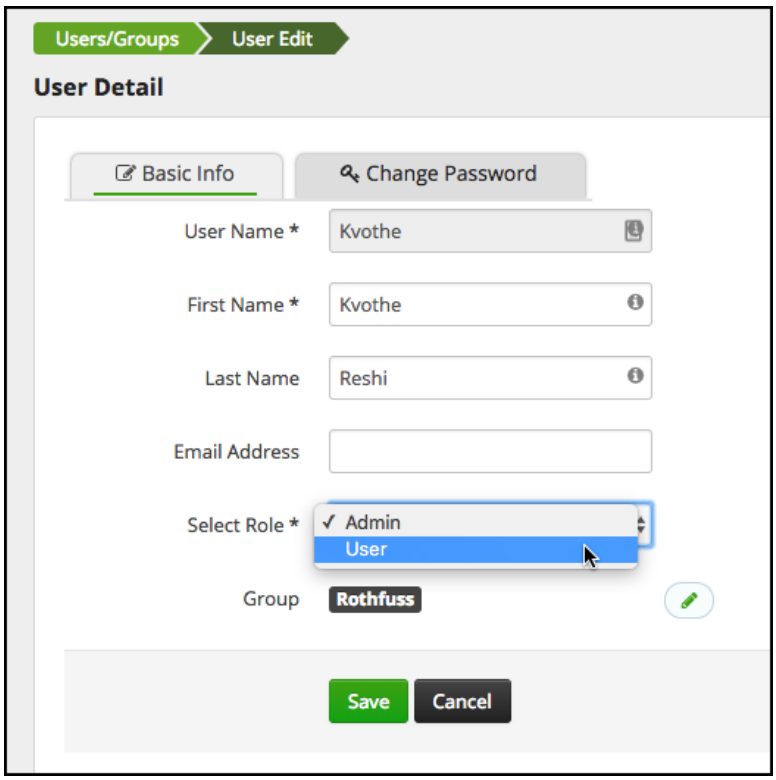
Ranger UI Authorization

Reference information on Ranger UI authorization, when configuring Ranger AD integration.

The Ranger authorization model is quite simple. It is maintained on the Ranger UI at Settings>Users/Groups :



A user can be either a normal user or an admin:



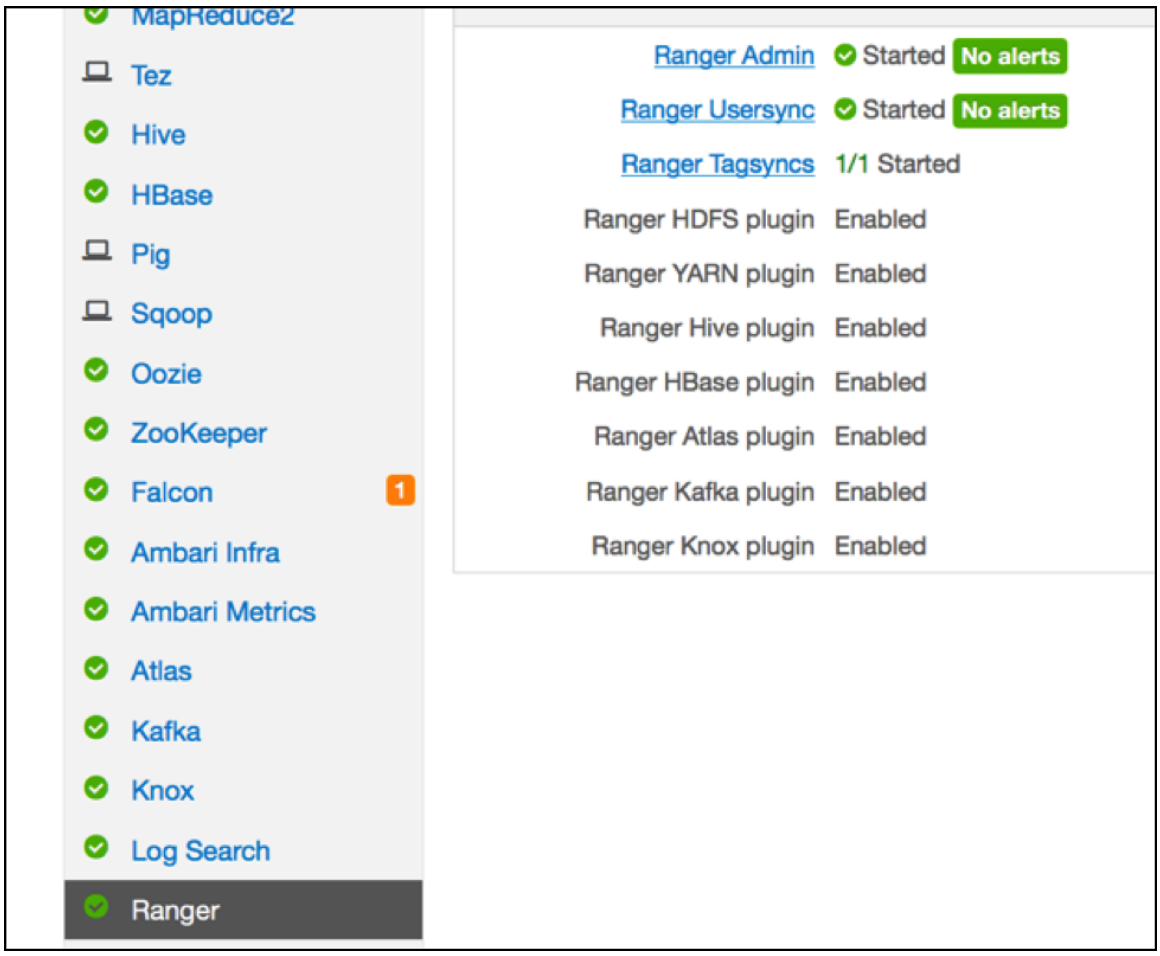
Only user with an Admin role can view or alter policies in Ranger.

Ranger Usersync

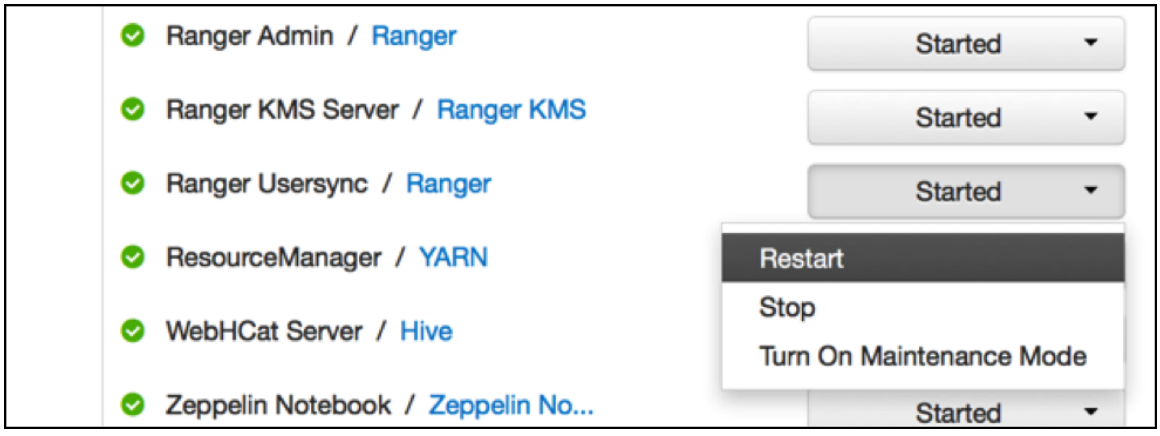
Reference information on Ranger usersync, when configuring Ranger AD integration.

A vital part of the Ranger architecture is the ability to get users and groups from the corporate AD to use in policy definitions.

Ranger usersync runs as separate daemon:



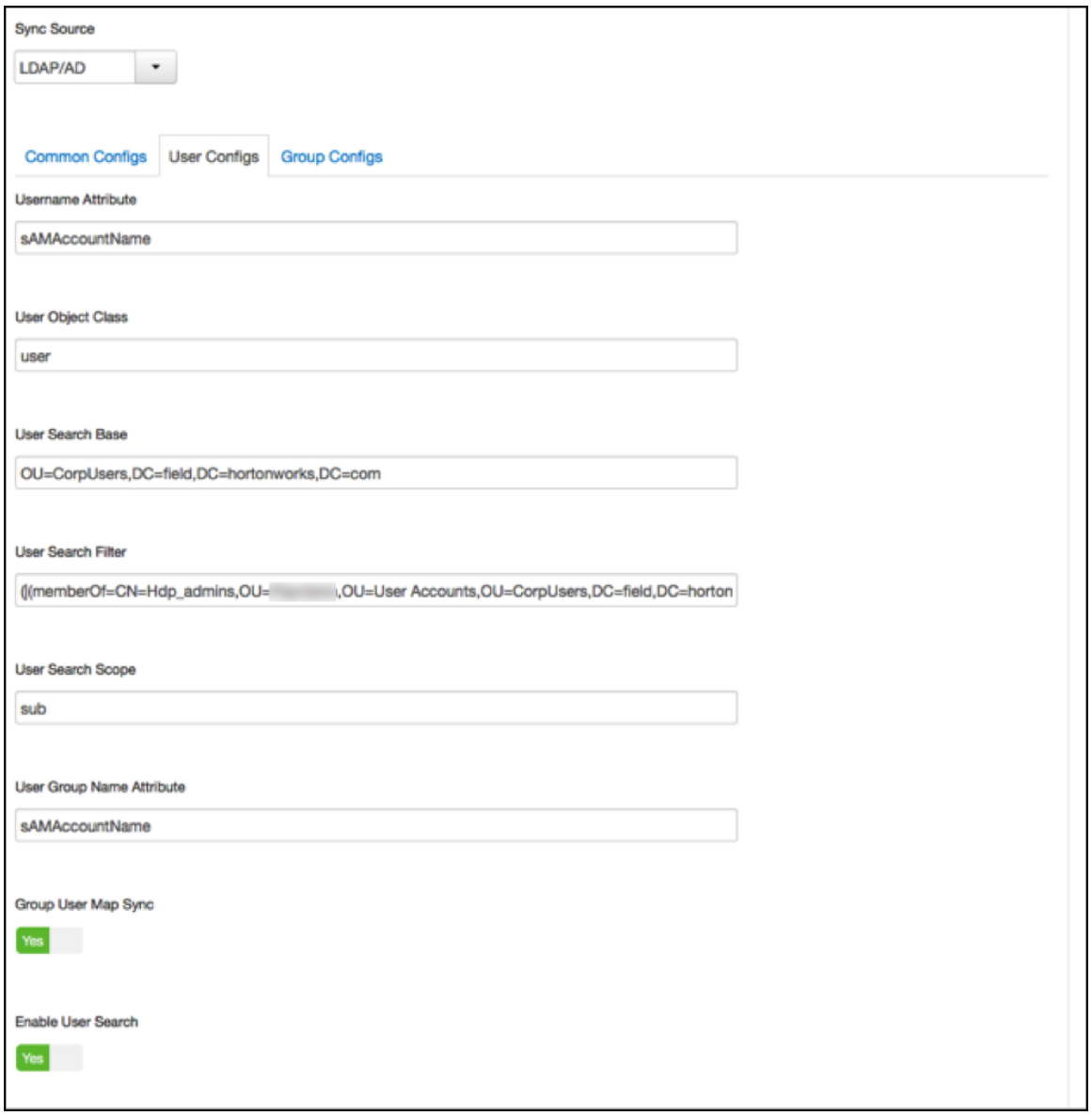
It can also be (re)started separately from Ambari:



Ranger Usersync Configuration

Usersync has a lot of moving parts and can have very different outcomes. Two main sets of properties govern the way users and groups are synchronized.

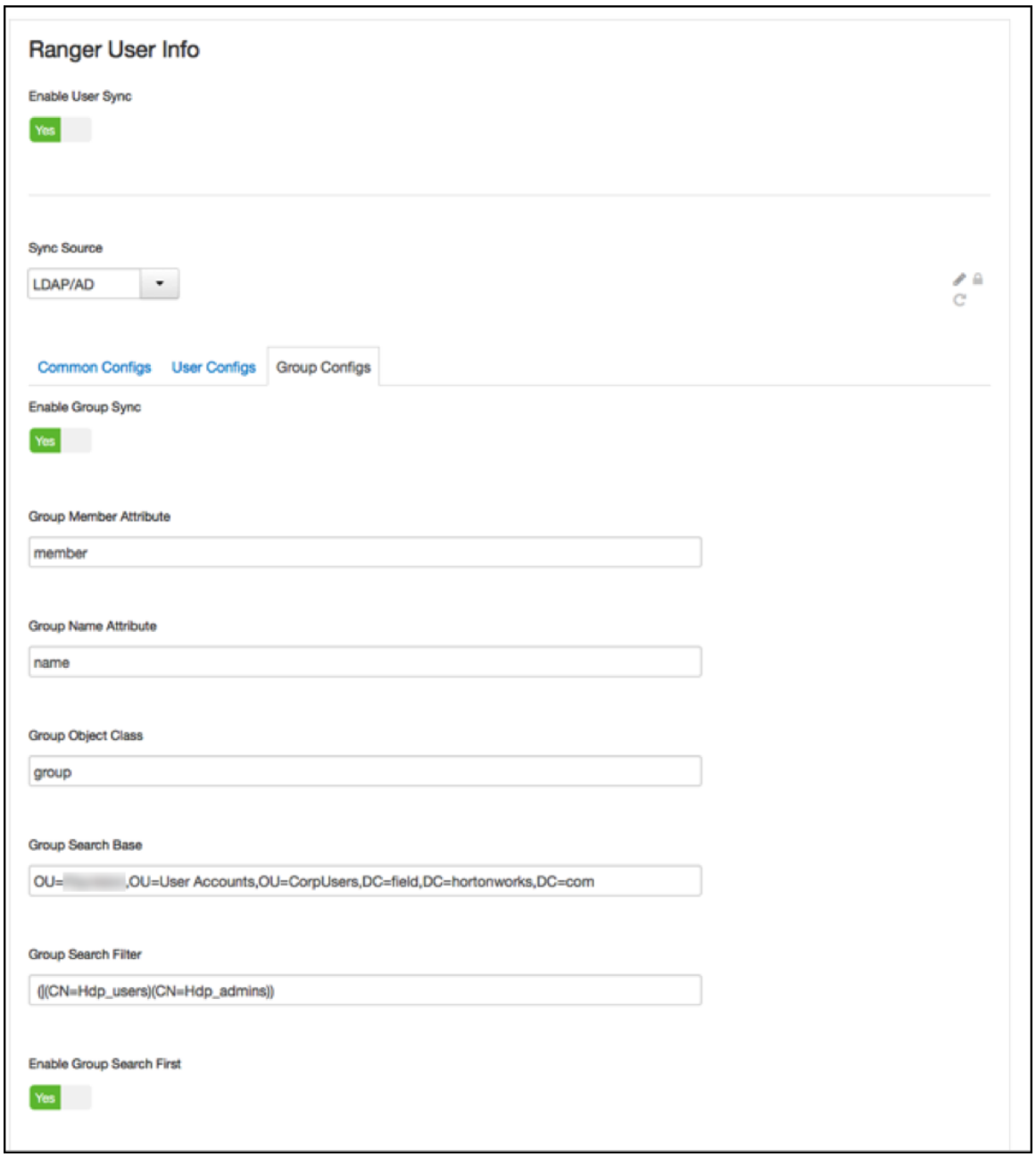
Without **Enable Group Search First** (a setting on the tab **Group Configs**) the primary access pattern is user based and groups will only be searched/added based on the users it finds first. In contrast, with **Enable Group Search First** enabled, the primary access pattern is group based (in turn based on the group search filter) and users will only be searched/added based on the group memberships it finds first



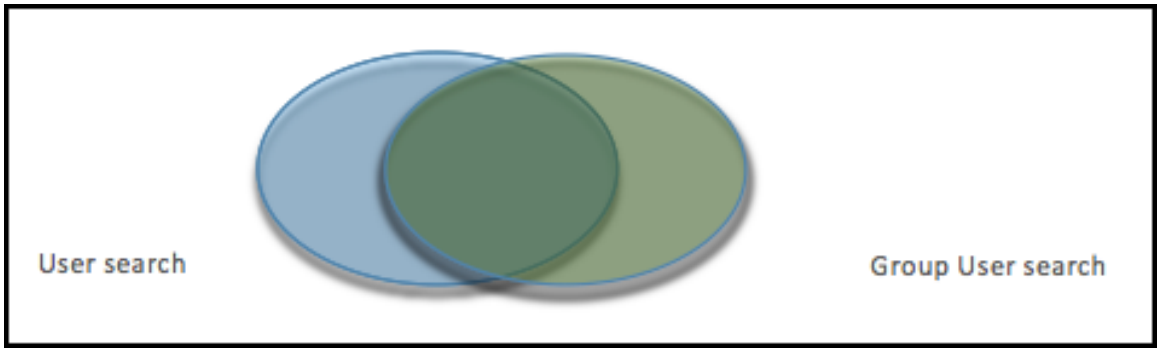

```
Value of ‘User Search Base’:
OU=CorpUsers,DC=field,DC=hortonworks,DC=com

Value of ‘User Search Filter’:
(|(memberOf=CN=Hdp_admins,OU=Company,OU=User
 Accounts,OU=CorpUsers,DC=field,DC=hortonworks,DC=com)
(memberOf=CN=Hdp_users,OU=Company,OU=User
 Accounts,OU=CorpUsers,DC=field,DC=hortonworks,DC=com))

Value of ‘User Group Name Attribute’:
sAMAccountName
```




```
Value of ‘Group Search Base’:
(|(CN=Hdp_users)(CN=Hdp_admins))
```

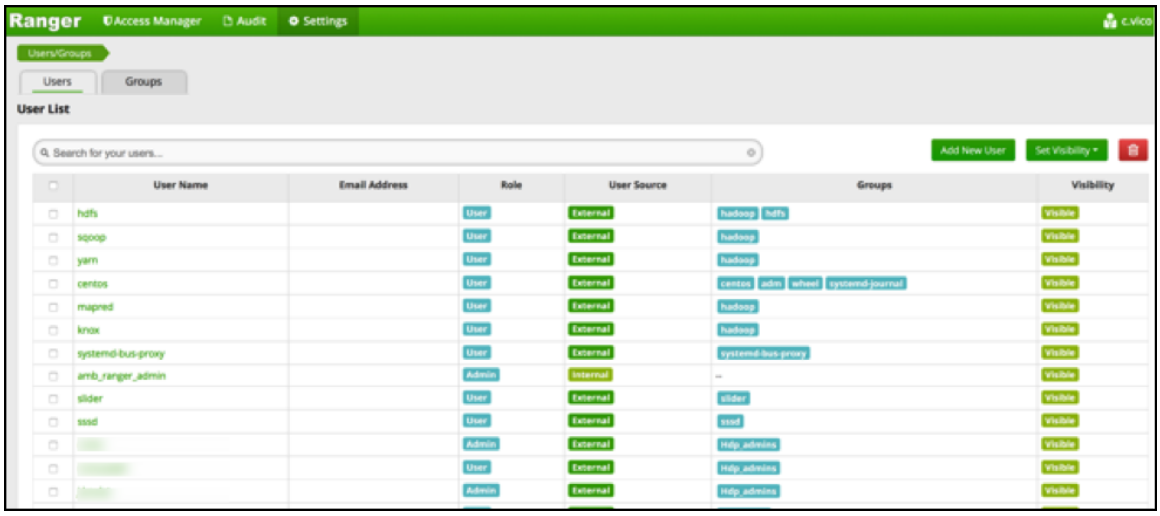


Beware that the filters on group level limit the returns on the user search and vice versa. In the graph above if the left oval would be the results of all users queried by the settings on the **User configs** and the right oval all users queried by **Group configs** settings, the eventual set of users that make it to the Ranger usersync is the overlap between the two.

Hortonworks therefore recommends to have the filters on both ends set exactly the same to potentially have a 100% overlap in the ovals.

In the example configuration given the scope of the usersync would be all members of both the groups ‘Hdp_admins’ and ‘Hdp_users’.

The result in Ranger User list:



Regarding the other switches on the user and group sync pages, best of both worlds is to have **Enable Group Search First** and **Enable User Search** enabled at the same time.

The logging of a run of the usersync deamon can be retrieved from /var/log/ranger/usersync/usersync.log on the server hosting Ranger Admin. A successful run might output logging like below:

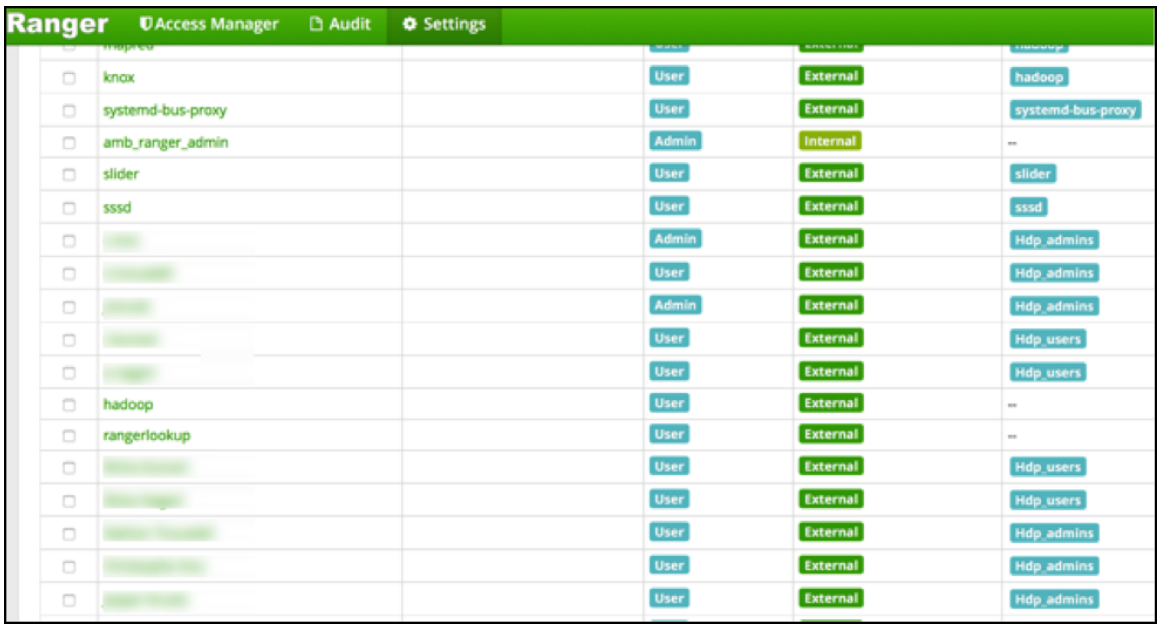
From that log it clearly shows that the groups are synced first and that all users belonging to those groups are then retrieved according to its own settings, after which the user parts are enriched/overwritten by the returns from the user queries.

Beware:

If you don’t enable **Enable User Search** that enrichment does NOT happen. Logging for such a run looks like this:

The result in Ranger UI are other user names (LongUserName) derived from ‘member’ group attributes full DN. You get the long name ‘James Kirk’ in the Ranger userlist in stead of j.kirk.

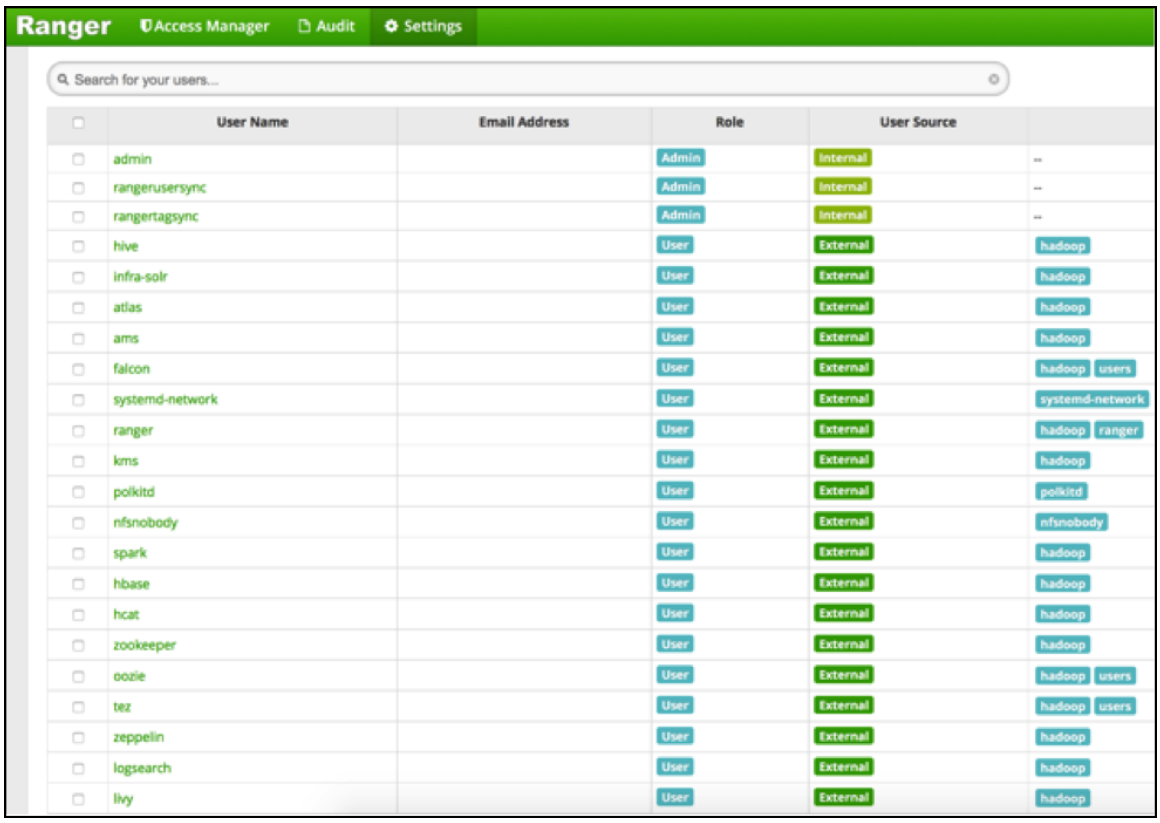
Ranger does not treat those as one and the same user:



Policies that were defined for user ‘k.reshi’ will not map to the user ‘Kvothe Reshi’ and vice versa. To prevent any confusion it is probably best to delete the long username versions from Rangers userlist.

Beware:

On the first page of Rangers user list there are lots of HDP system users. Most of them were put there by the Ranger installer and during the plugins installs:

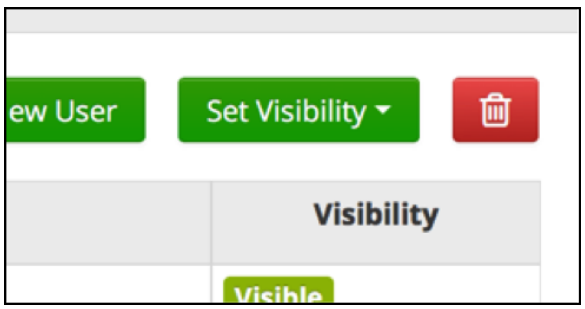


Do NOT remove those system users!

There are basic access policies based on those system users designed to keep a Ranger governed HDP component working after Ranger is given all control over that components authorizations. Without those policies/users many HDP components will be in serious trouble.

Ranger User Management

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



User can be easily remove from Ranger by checking the username in the list and hit the red **Delete** button. Ranger takes care of referential integrity so that user will also be removed from any policy.

Known Issue: Ranger Group Mapping

For Ranger AD integration, there is an issue with Ranger not being able to map a user on a group ‘Hdp_admins’ to a policy that allows/denies access to the group ‘Hdp_admins’. The issue is on the capital characters that might be on a AD group name definition.

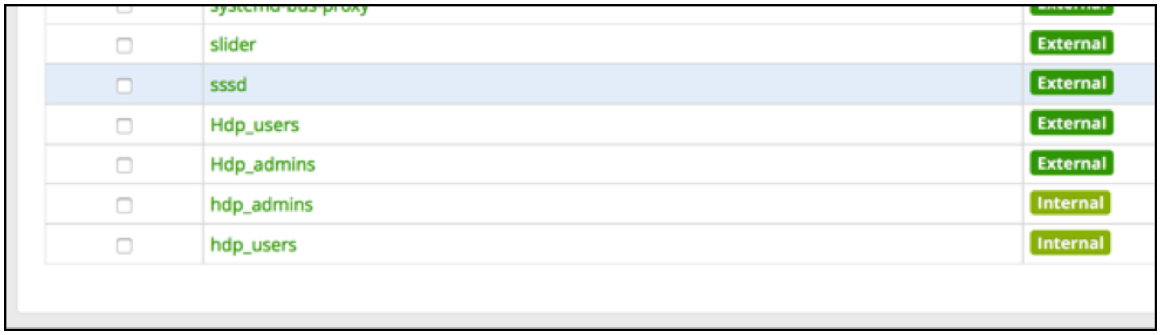
Most HDP components get the group information for a user via the SSSD daemon. When asked for the groups the user ‘d.threpe’ belongs to we get:

```
[centos@rjk-hdp25-m-01 ~]$ groups d.threpe
d.threpe : domain_users hdp_admins hadoop
```

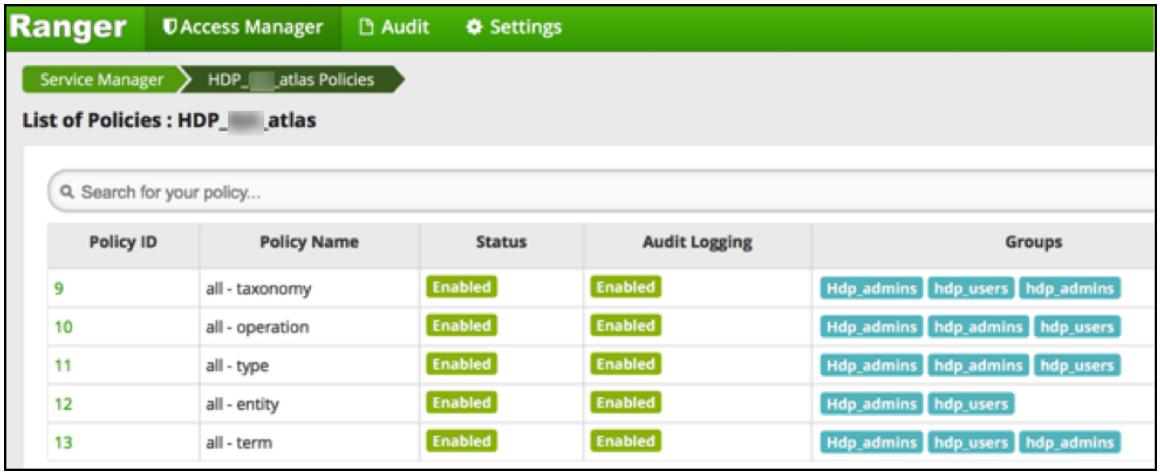
So ‘hdp_admins’ all in lower case. Ranger does not treat this as the same value as ‘Hdp_admins’ which came via the group sync and was applied to some policies.

There is no way to make the group sync write or retrieve the group names all in lower case since there is no AD attribute that rewrites it in lowercase.

This issue can be worked around fortunately (till it gets solved). The solution is to define a local group in Ranger as a shadow group of a real group from AD, but then all in lower case:



If we now create policies and use that lower case ‘shadow’ group literal the result is that policies are correctly mapped to the AD groups again:



*The ‘Hdp_admins’ entry does not have to be there, it is shown for clarification only. ‘hdp_admins’ is necessary to make it work.