Cloudera Runtime 7.0.3

Securing Apache Atlas

Date published: 2019-11-22
Date modified:

https://docs.cloudera.com/

Legal Notice

Contents

Configuring and Securing Atlas

Cloudera Manager manages Atlas as a service, automatically ensuring Atlas can communicate securely with its clients and the services it depends on. Use Cloudera Manager to manage additional Atlas settings; use Ranger to control user access in Atlas.

When you include Atlas as a service in a cluster, Cloudera Manager automatically configures the following settings:

- Ranger plugin. Atlas uses Ranger to determine which users have access to perform actions in Atlas.
- TagSync with Ranger. Atlas passes entity metadata for classifications to Ranger using a Kafka topic.
- Access Policies in Ranger. Default policies are configured for the following users:
  - rangertagsync: the TagSync service users has read access to entity metadata, specifically to entity classifications to be used in Ranger tag-based policies.
  - beacon: the Data Plane service user has full access to entity metadata, classification and relationship creation, and the ability to import and export metadata from Atlas. These privileges allow integration between the Data Catalog (Data Steward Studio) and Atlas.
  - admin: the initial Cloudera Manager superuser has full access to all Atlas actions, including full access to entity metadata, classification and relationship creation, the ability to import and export metadata from Atlas, and the ability to save searches.
  - public: all users are granted access to read Atlas entity metadata and relationships (such as lineage).
  - {USER}: any user who successfully logs in to Atlas can save searches so they are available in subsequent Atlas sessions.

  You will probably want to update and add to these policies to include users and groups in your organization who will need access to Atlas actions.
- TLS-enabled clusters. Cloudera Manager configures:
  - The option to enable TLS for Atlas (atlas.enableTLS)
  - Keystore file locations and passwords for encrypting client-server communication
  - Trust store location and password for the Atlas server to communicate as a client to other services such as HBase and Solr
  - Trust store location and password for the Atlas gateway role that passes information through Kafka topics.
- Kerberos-enabled clusters. Cloudera Manager configures:
  - Principals for Atlas service users
  - Ranger policies to support authentication for Atlas server and hook communication to Kafka
  - Ranger policies to support authentication for the Atlas server to communicate with Solr and HBase

Configure TLS/SSL for Apache Atlas

How to configure TLS/SSL for Apache Atlas if you don't choose to use Cloudera Manager's Auto-TLS

About this task

Typically you would configure TLS for Atlas by using Cloudera Manager's Auto-TLS option. If you need to change these settings or want to understand more about the values set by Cloudera Manager, follow these steps.

Procedure

1. In Cloudera Manager, select the Atlas service, then click the Configuration tab.
2. Under Category, select Security.

3. Set or update the following properties.

Table 1: Apache Atlas TLS/SSL Settings

Grouping	Configuration Property	Description
Turn on TLS	Enable TLS/SSL for Atlas Server atlas.enableTLS	Select this check box to encrypt Atlas server communication using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)). This option enables TLS for communication between clients and Atlas Server, between the Atlas Server as a client and services it depends on, such as HBase, and between the Atlas Gateway server and Kafka messaging topics.
Private Key Management	Atlas Server TLS/SSL Server JKS Keystore File Location keystore.file	The path to the TLS/SSL keystore file containing the server certificate and private key Atlas uses to prove its own identity when it receives communication from other applications using the public key identified in Atlas Server TLS/SSL Client Trust Store File. The keystore must be in JKS format.
	Atlas Server TLS/SSL Server JKS Keystore File Password keystore.password	The password that allows access to the Atlas Server JKS keystore file.
	Atlas Server TLS/SSL Server JKS Keystore File Password password	(Optional) The password that protects the Atlas Server private key contained in the JKS keystore.
Public Key Management	Atlas Server TLS/SSL Client Trust Store File truststore.file	The path to a TLS/SSL public key trust store file, in .jks format, that contains Atlas server public key. This trust store must contain the certificate(s) used to sign the connected service(s). If this parameter is not provided, the default list of well-known certificate authorities is used instead.
	Atlas Server TLS/SSL Client Trust Store Password truststore.password	(Optional) The password for the Atlas Server TLS/SSL Certificate trust store file. This password is not required to access the trust store; this field can be left blank. This password provides optional integrity checking of the file. The contents of trust stores are certificates, and certificates are public information.
	Gateway TLS/SSL Client Trust Store File atlas.kafka.ssl.truststore.location	The path to the trust store file, in .jks format, used to confirm the authenticity of TLS/SSL servers that the Atlas Gateway might connect to. This trust store must contain the certificate(s) used to sign the service(s) connected to. If this parameter is not provided, the default list of well-known certificate authorities is used instead. Typically, this file is the same file listed in the truststore.file.
	Gateway TLS/SSL Client Trust Store Password atlas.kafka.ssl.truststore.password	The password for the Gateway TLS/SSL Certificate trust store file. This password is not required to access the trust store; this field can be left blank. This password provides optional integrity checking of the file. The contents of trust stores are certificates, and certificates are public information.

4. Click Save Changes.
5. Restart the Atlas service.

Related Information

Configuring TLS Encryption for Cloudera Manager and CDH Using Auto-TLS

Configuring Atlas Authentication

This section describes how to configure the authentication methods that determine who is allowed to log in to the Atlas web UI. The authentication options are Kerberos, LDAP—including AD, or file-based.

Atlas allows more than one authentication method to be enabled at one time. If more than one authentication method is enabled, users failing the first method are authenticated against the second method. The priority order of the methods is Kerberos, LDAP, then file-based authentication. For example if both Kerberos and LDAP authentication are enabled, a request without a Kerberos principal and keytab are authenticated using LDAP.

Specifying more than one authentication method allows you to setup useful production and development scenarios:

- In a Production environment, you might configure Kerberos for service account access to the Atlas server while also supporting LDAP authentication for users logging in through the UI.
- In a Development environment, you might configure Kerberos for service account access while leaving file-based authentication enabled to allow a limited number of administrator to access the Atlas UI.

Note: By default, Cloudera Manager installs Atlas with file-based authentication with full Atlas access given to the username you configure for this authentication method. This configuration ensures a smooth end-to-end installation experience. Be sure to disable file-based authentication when you configure your production authentication method.

Configure Kerberos authentication for Apache Atlas

How to configure Kerberos Authentication for Apache Atlas

Kerberos authentication for Apache Atlas is automatically configured when you use Cloudera Manager to enable Kerberos authentication for the cluster (typically using the Cloudera Manager Kerberos Wizard).

A prerequisite for using Kerberos authentication is that your cluster is configured for TLS/SSL encryption. There are Atlas-specific steps to complete to enable TLS for Atlas.

Related Information

Configuring Authentication in Cloudera Manager
Configure TLS/SSL for Apache Atlas
Configuring TLS Encryption for Cloudera Manager and CDH Using Auto-TLS

Configure Atlas authentication for AD

How to configure Atlas to use AD for user authentication when using AD cluster-wide.

About this task

The settings indicated in these steps apply to Atlas authentication and it is likely that the values will be the same as you use to configure other services on the cluster.

Procedure

1. In Cloudera Manager, select the Atlas service, then open the Configuration tab.
2. To display the authentication settings, type "authentication" in the Search box. You may need to scroll down to see all of the AD settings.
3. Configure the following settings for AD authentication:

Property	Description	Sample values
Enable LDAP Authentication atlas.authentication.method.ldap	Determines whether LDAP is used for authentication.	true

Property	Description	Sample values
LDAP Authentication Type atlas.authentication.method.ldap.type	The LDAP type (ldap, ad, or none).	ad
AD URL atlas.authentication.method.ldap.ad.url	The AD server URL.	
AD Bind DN Username atlas.authentication.method.ldap.ad.bind.dn	Full distinguished name (DN), including common name (CN), of an AD user account that has privileges to search.	cn=admin,dc=example,dc=com
AD Bind DN Password atlas.authentication.method.ldap.ad.bind.password	Password for the account that can search in AD.	
AD Domain Name (Only for AD) atlas.authentication.method.ldap.ad.domain	AD domain, only used if Authentication method is AD.	
AD User Search Filter atlas.authentication.method.ldap.ad.user.searchfilter	The AD user search filter.	
AD Base DN atlas.authentication.method.ldap.ad.base.dn	The Distinguished Name (DN) of the starting point for directory server searches.	dc=example,dc=com
AD User Default Role atlas.authentication.method.ldap.ad.default.role	AD User default Role.	
AD Referral atlas.authentication.method.ldap.ad.referral*	See below. Defaults to ignore.	follow

* There are three possible values for atlas.authentication.method.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.

4. Click Save Changes.
5. Restart the Atlas service.

Related Information

Cloudera Manager Authentication Overview

Configure Atlas authentication for LDAP

How to configure Atlas to use LDAP for user authentication.

About this task

The settings indicated in these steps apply to Atlas authentication and it is likely that the values will be the same as you use to configure other services on the cluster.

Procedure

1. In Cloudera Manager, select the Atlas service, then open the Configuration tab.
2. To display the authentication settings, type "authentication" in the Search box. You may need to scroll down to see all of the LDAP settings.
3. Configure the following settings for LDAP authentication:

Grouping	Property	Description	Sample values
Enable LDAP Authentication	Enable LDAP Authentication atlas.authentication.method.ldap	Determines whether LDAP is used for authentication.	true
	LDAP Authentication Type atlas.authentication.method.ldap.type	The LDAP type (ldap, ad, or none).	ldap
LDAP Server Location	LDAP Server URL atlas.authentication.method.ldap.url	The LDAP server URL.	ldap://localhost:389 or ldaps://localhost:636
Bind Credentials	LDAP Bind Username atlas.authentication.method.ldap.bind.dn	Full distinguished name (DN), including common name (CN), of an LDAP user account that has privileges to query the LDAP database of user accounts on behalf of Atlas. This could be a read-only LDAP user.	cn=admin,ou=people,dc=example,dc=com
	LDAP Bind DN Password atlas.authentication.method.ldap.bind.password	Password for the account that can search for users.	Secret123!
Group Lookup	LDAP Group-Search Base atlas.authentication.method.ldap.groupSearchBase	The organizational unit (OU) and domain component (DC) properties for the LDAP search tree where Atlas searches for groups.	(|(CN=Hdp_users)(CN=Hdp_admins))
	LDAP Group-Search Filter atlas.authentication.method.ldap.groupSearchFilter	(Optional) Refine the scope of LDAP group search. The Groups-Search Filter is combined with the Group-Search Base to define the group lookup.	
Role Assignment	LDAP Group-Role Attribute atlas.authentication.method.ldap.groupRoleAttribute	The attribute stored in the LDAP Group object to use to map LDAP groups to Atlas roles.	cn

Grouping	Property	Description	Sample values
	LDAP User Default Role atlas.authentication.method.ldap.default.role	Atlas role to assign to LDAP users.	
LDAP Search-Bind Authentication Mode	LDAP DN atlas.authentication.method.ldap.base.dn	The Distinguished Name (DN) of the starting point of the LDAP search tree for directory server searches. You can also specify a User Search Filter to further reduce the scope of the search.	dc=example,dc=com
	LDAP User Search Filter atlas.authentication.method.ldap.user.searchfilter	The LDAP user search filter. Used with the User Search Base to further limit the scope of the search for a directory entry that matches the credentials of the user logging into Atlas. Use a user search filter along with a DN pattern so that the search filter provides a fallback if the DN pattern search fails.	
LDAP Direct-Bind Authentication Mode	LDAP User DN Pattern atlas.authentication.method.ldap.userDNpattern	Direct-bind authentication can be used if search is not required to determine the DN needed to bind to the LDAP server. Leave this property blank if LDAP DN is set. To use this authentication mode, all users must be under a single branch in the LDAP directory.	To search for a distinguished name where the uid attribute is the username at login, you might provide a pattern such as: uid={0},ou=users,dc=xasecure,dc=net where {0} indicates the username of the authenticating user. If a user provides the username "foo" at the login page, Atlas searches for the DN: uid=foo,ou=People,dc=corp,dc=com
LDAP Referral	LDAP Referral atlas.authentication.method.ldap.referral*	See below. Defaults to ignore.	follow

* There are three possible values for atlas.authentication.method.ldap.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 LDAP 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 LDAP, a PartialResultException is returned when referrals are encountered while search results are processed.

4. Click Save Changes.
5. Restart the Atlas service.

Related Information

Cloudera Manager Authentication Overview

Configure Atlas file-based authentication

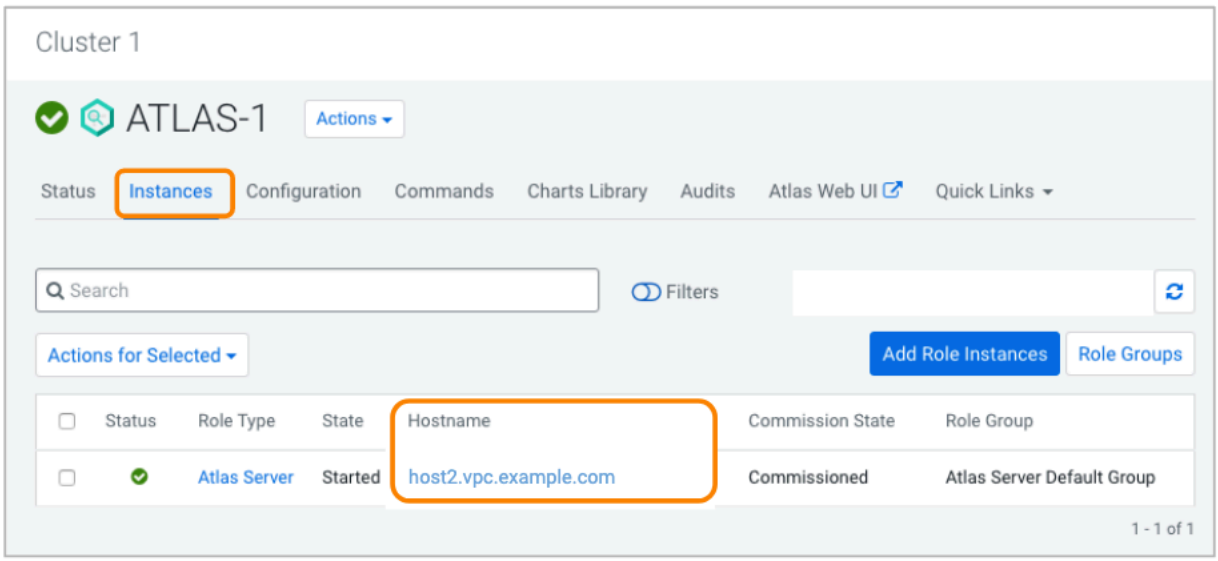
How to manage Atlas user authentication when using user credentials from a file.

Atlas uses file-based authentication to allow a smooth installation experience: the installation process allows you to set an admin user who then has access to all Atlas operations. These steps describe two of the most likely tasks you would perform related to file-based authentication for Atlas:

- Add to the list of authorized users for use in a non-production environment
- Disable file-based authorization when you transition to a production-quality authentication method, such as Kerberos or Kerberos and a version of LDAP

To add to the users list

1. In Cloudera Manager, select the Atlas service, then open the Instances tab.
2. Note the Hostname where the Atlas service is running. You'll need to be able to SSH into that host to update the user list.



3. Open the Configuration tab.
4. Type "file" in the search box to filter the property list.

5. Find and update the user file.
   - Look for the location of the users-credentials.properties file as set in the Path to Credentials for File-based Login property.

     The default directory location indicated by ATLAS_USER_CREDENTIALS_CONF_PATH is the Atlas configuration directory in the Cloudera Runtime installation location, typically

     ```
     /opt/cloudera/parcels/CDH-VERSION/etc/atlas/conf.dist
     ```
   - SSH into the Atlas host to edit the users file to include additional users in the format:

     ```
     USERNAME=GROUP::SHA256-PASSWORD-HASH
     ```

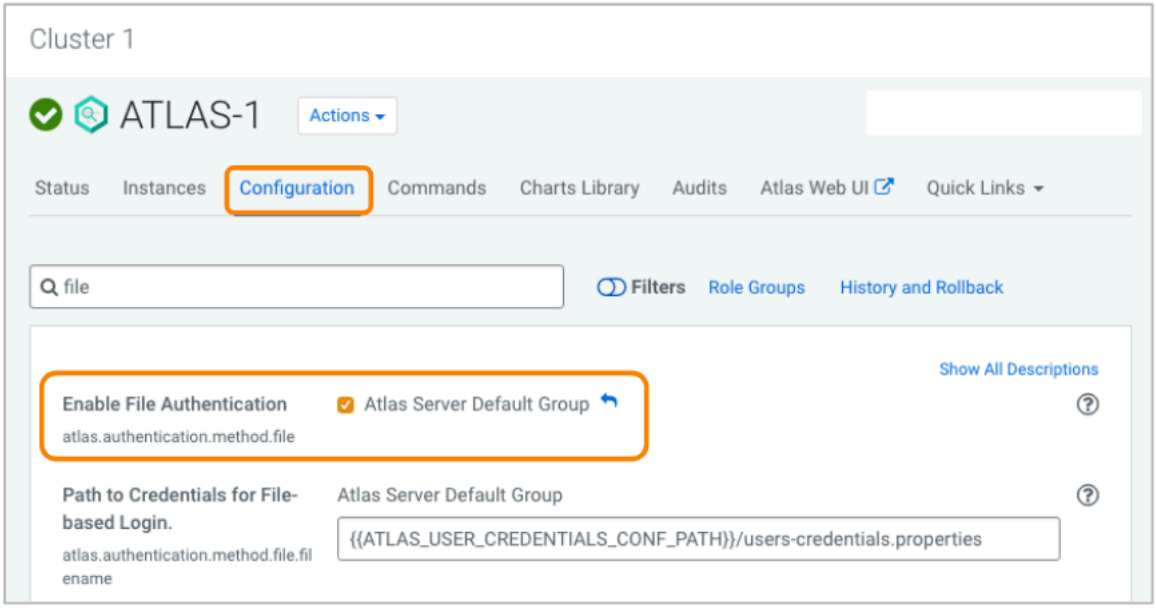
     where the *USERNAME* is the string used in the login page, the *GROUP* is one of ADMIN, DATA_STE WARD, or DATA_SCIENTIST, and the password is encoded with the sha256 encoding method. To generate a sha256 encrypted password, use:

     ```
     echo -n string | sha256sum
     ```

     The output includes the hash followed by a space and a hyphen (-); include only the hash in the users file.
6. Restart the Atlas service.

To disable file-based authentication

1. In Cloudera Manager, select the Atlas service, then open the Configuration tab.
2. To display the authentication settings, type "file" in the Search box.
3. To disable file-based authentication, uncheck the option Enable File Authentication.



4. Click Save Changes.
5. Restart the Atlas service.

Configuring Atlas Authorization

Atlas in CDP uses Ranger policies to control access to metadata managed by Atlas and to Atlas administrative operations.

Ranger provides authorization to access the following metadata and operations:

Types

Atlas "types" are the entity model definitions, whether provided in Atlas or added in your environment. Types include these "categories":

- Entity
- Classification
- Relationship
- Struct
- Enum

Ranger authorization allows you to configure access for users and groups to perform the following operations on types:

- Create
- Update
- Delete

The policies can be configured to apply to one or more types or all types. For example, the Atlas administrator user has access to create, update, and delete all type categories (type-category *).

Entities

Atlas "entities" are instances of entity types: entities represent assets and processes on your cluster. Ranger authorization allows you to configure access to users and groups to perform the following operations on entities:

- Read
- Create
- Update
- Delete
- Read classification
- Add classification
- Update classification
- Remove classification

Note that the classification operations are those that involve associating a classification to an entity; operations on a classification definition are controlled by authorization on the classification category of type described previously. Use the entity authorization to give a user the ability to associate an existing classification with any entity (entity-type *); use the type authorization to give a user the ability to create new classifications (type-category classification).

Some Atlas features, such as saved searches, are modeled as entities. You can control access to these features using entity policies. For example, a default policy allows any authenticated user to save Atlas searches (entity-type __AtlasUserProfile, __AtlasUserSavedSearch).

Relationships

Atlas "relationships" describe connections between two entities, including, but not limited to, the input and output relationships that are used to build lineage graphs. Ranger authorization allows you to configure access to users and groups to perform the following operations on relationships:

- Add relationship
- Update relationship
- Remove relationship

These operations are required to build rich models among entities and are granted to administrative users and system users. Relationships cannot be updated by users through the Atlas UI.

Admin operations

Atlas administrative operations include:

- Import entities

- Export entities

These operations encompass all the privileges needed to create new and update existing entities. Typically, this access is granted to administrative users and system users such as RangerLookup and the Data Plane profiler user (DPProfiler).

Configuring Ranger Authorization for Atlas

Atlas is configured to use Ranger for authorization by default. You might need to change configuration settings to disable Ranger as the source of authorization in a development environment; Ranger authorization is highly recommended in a production environment. In addition, there are some configuration values that you might need to change should you make significant changes to how Atlas and Ranger are installed in your cluster.

About this task

Atlas behaves like any other service when it comes to integrating with Ranger for access control: turn on Ranger authorization from the Cloudera Manager configuration page for the Atlas service. This integration allows Atlas to use Ranger policies to determine authorization for user actions in Atlas; Atlas also reports success or failure against the policies back to Ranger. In addition to this standard integration for authorization, Atlas integrates with Ranger to send metadata updates to Ranger using a Kafka topic. The configuration properties that support the two integration paths include specifying HDFS locations for caching Ranger policies, storing Atlas audits, and storing other metadata exchanged between the two services. In rare circumstances, you may need to relocate these storage locations.

Before you begin

Minimum Required Role in Cloudera Manager: Full Administrator.

Procedure

1. In Cloudera Manager, select the Atlas service, then open the Configuration tab.
2. To display the Ranger configuration settings, type "ranger" in the Search box.
3. To modify the behavior of the Atlas to Ranger integration, update the following properties:

   To disable Atlas authentication through Ranger

   Uncheck the RANGER Service property. Leave the supporting properties as is so you can re-instate Ranger easily if needed.

   To set where Atlas stores intermediate data for Ranger audits and cached authorization policies

   Review and potentially change the path value for these properties:

   - Ranger Atlas Plugin Hdfs Audit Directory (HDFS location)
   - Ranger Atlas Plugin Audit Hdfs Spool Directory Path (local file system location)
   - Ranger Atlas Plugin Audit Solr Spool Directory Path (local file system location)
   - Ranger Atlas Plugin Policy Cache Directory Path (local file system location)

   To re-instate Ranger after disabling it

   Enable the RANGER Service property. The supporting properties should still be available.
4. Restart the Atlas service.

Configure Atlas Authorization using Ranger

Use the Ranger Admin UI to add or update policies to control Atlas access.

About this task

You can use Ranger policies to control user-access to Atlas metadata and to actions that users can perform in Atlas. The following policies are defined by default:

- rangertagsync: the TagSync service users has read access to entity metadata, specifically to entity classifications to be used in Ranger tag-based policies.

- beacon: the Data Plane service user has full access to entity metadata, classification and relationship creation, and the ability to import and export metadata from Atlas. These privileges allow integration between the Data Catalog (Data Steward Studio) and Atlas.
- admin: the initial Cloudera Manager superuser has full access to all Atlas actions, including full access to entity metadata, classification and relationship creation, the ability to import and export metadata from Atlas, and the ability to save searches.
- public: all users are granted access to read Atlas entity metadata and relationships (such as lineage).
- {USER}: any user who successfully logs in to Atlas can save searches so they are available in subsequent Atlas sessions.

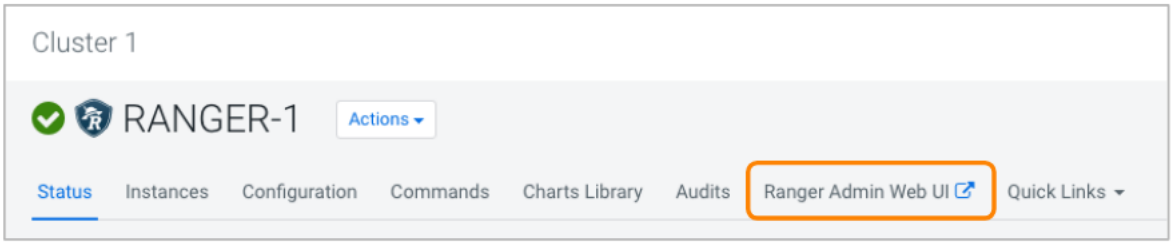
The following video summarizes the steps in Ranger.

Before you begin

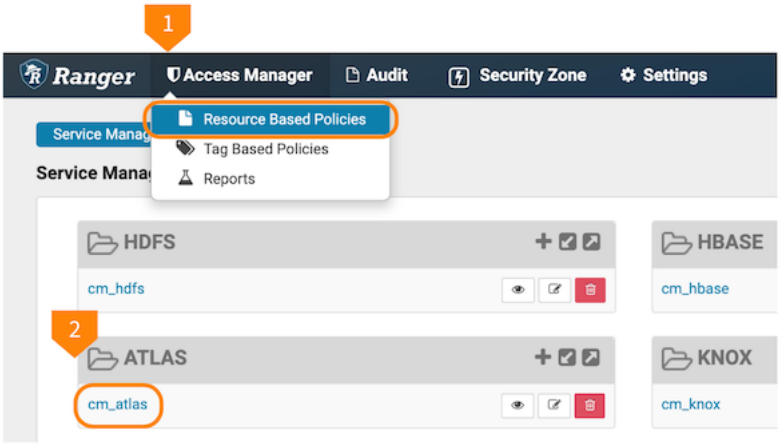
To change Ranger policies for Atlas, your user needs privileges in Ranger to change Resource Based Policies.

Procedure

1. Open the Ranger service that is running in the same cluster as Atlas.
   One way to do this is to open the Ranger Admin Web UI from Cloudera Manager.



2. Open Access Manager Resource Based Policies and select Atlas policies (cm_atlas).



3. On the List of Policies page, click Add New Policy.
4. Use the Create Policy page to specify the Atlas authorization policy.

Policy Type	Access. There are no other policy types available for an Atlas service.
Policy Name	255 character name that appears in the list of policies. Roles, users, and groups also show up in the list, so it helps if your name includes the operations or metadata that the policy controls.
Policy Label	Metadata you can include in the policy definition to help organize the policies for a given service. The same label can be added to any number of policies for the service. There is no limit to the number of characters in a label, but only 28 characters display in the policy list.
type-category selection	The metadata or operation type ("resources" in Ranger terms) that the policy applies to, including: • type-category • entity-type • atlas-service • relationship-type After selecting the resource type, add the type category in the category list. To apply the policy to all categories for the selected type, use *. For details, see Configuring Atlas Authorization on page 11. This selection can be set to "include" the selected resources or "exclude" the selected resources. An include policy for type-category and category entity would apply only to entities. An exclude policy for the same type-category and entity would apply to all metadata types other than entity.
Type Name	Further refinement within the metadata or operation category specified in the type-category selection. You can limit the policy to metadata that matches the name or names you specify here. For example, if you chose type-category and the category entity, you could use the Type Name to apply the policy to entities with names that start with "fy2020*". This field supports ? and * wildcards for single and multiple character replacement.
Description	Information that you add to help you remember the value of this policy.
Audit Logging	Enables Ranger's audit logging for this policy. There are other options in Ranger's configuration that can conflict with this option, but generally if you turn off this setting, Ranger will enforce the policy but will not audit success or failed actions against the policy.

Allow Conditions	Choose the roles, users, and/or groups and the permissions they can access for the resources defined in the policy. If you need to include parts of overlapping groups, add an exclude condition in addition to the allow condition.
Deny Conditions	Choose the roles, users, and/or groups and the permissions they cannot access for the resources defined in the policy.

5. Click Add.

Results

You should be able to validate the policy almost immediately after saving a valid policy.

Related Information

Configure a resource-based service: Atlas

Add or edit permissions in Ranger