Cloudera Runtime 7.2.12

Schema Registry Security

Date published: 2020-06-12
Date modified: 2021-10-25

https://docs.cloudera.com/

Legal Notice

Contents

Schema Registry Authorization through Ranger Access Policies

User and group access to various Schema Registry functions is controlled through Apache Ranger.

Pre-defined access policies for Schema Registry allow the administrator to quickly add a user or user group to specify:

- Who can add/evolve schemas to a schema metadata.
- Who can view and edit schemas within a schema metadata.
- Who can upload the ser/des jar files.

If a higher level of granularity is necessary, the administrator can create an access policy and add the user or user-group to this custom policy.

Related Information

Pre-defined Access Policies for Schema Registry

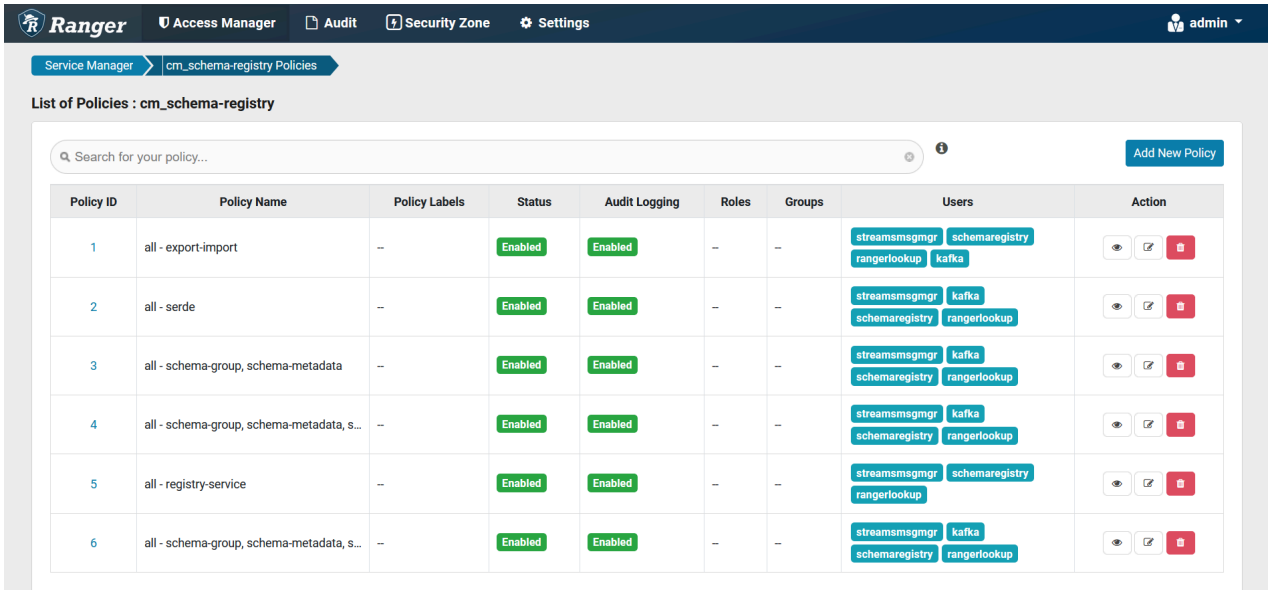
Add the user or group to a pre-defined access policy

Create a Custom Access Policy

Pre-defined Access Policies for Schema Registry

Based on a user's responsibilities, you can add users or a user group to one or more of the following pre-defined access policies for Schema Registry and you can specify the type of permission such as Create, Read, Update, and Delete.

The following image shows the pre-defined access policies for Schema Registry:



The following table describes the pre-defined access policies for Schema Registry:

Access Policy	Description
all - export-import	Allows users to import and export schemas to/from the Schema Registry service. For example, a user can import a .json file with schemas from a Confluent Kafka topic to Cloudera's Schema Registry.

Access Policy	Description
all - serde	Allows users to store metadata for the format of how data should be read and how it should be written. Users can store JAR files for serializers and deserializers and then map the serdes to the schema.
all - schema-group, schema-metadata	Allows users to access the schema groups and schema metadata.
all - schema-group, schema-metadata, schema-branch	Allows users to access the schema groups, schema metadata, and schema branch.
all - registry-service	Allows users to access the schema registry service. If a user is added to this policy, the user can access all Schema Registry entities.
all - schema-group, schema-metadata, schema-branch, schema-version	Allows users to access the schema groups, schema metadata, schema branch, and schema version.

Related Information

Schema Registry Authorization through Ranger Access Policies
Add the user or group to a pre-defined access policy
Create a Custom Access Policy

Add the user or group to a pre-defined access policy

When an authenticated user attempts to view, create, edit, or delete a Schema Registry entity, the system checks whether the user has privileges to perform that action. These privileges are determined by the Ranger access policies that a user is associated with.

Before you begin

For Ranger policies to work, you must have a user group named schemaregistry. If you use UNIX PAM, the sche maregistry user group must be on the node that hosts Schema Registry.

About this task

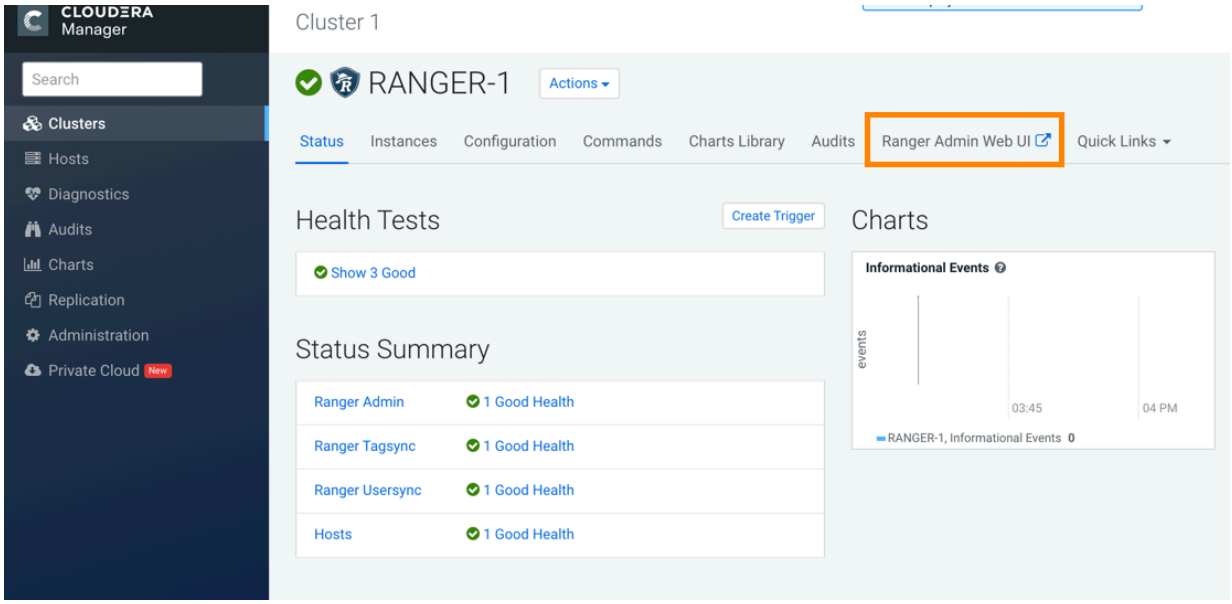
Determine the permissions required by a user or user group and accordingly add the user or group to the appropriate pre-defined access policy.

Each pre-defined access policy controls access to one or more Schema Registry entities.

Procedure

1. From the Cloudera Manager home page, click the Ranger link.
   The **Ranger** management page appears.

2. Click the Ranger Admin Web UI link.

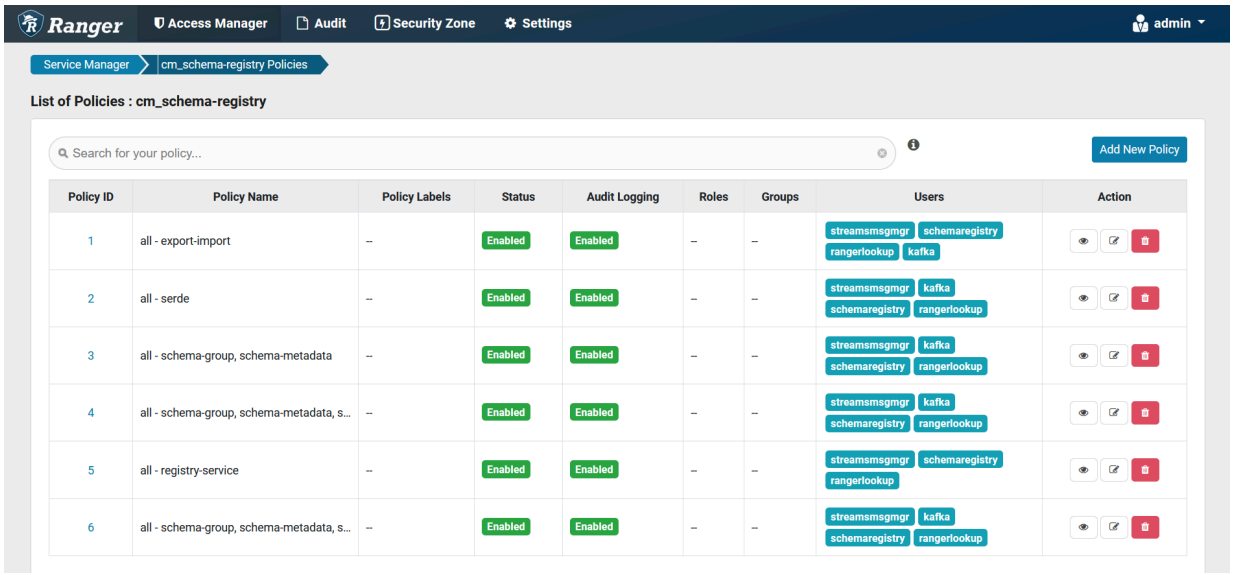


The **Ranger Log In** page appears.

3. Enter your user name and password to log in.
The **Ranger Service Manager** page appears.

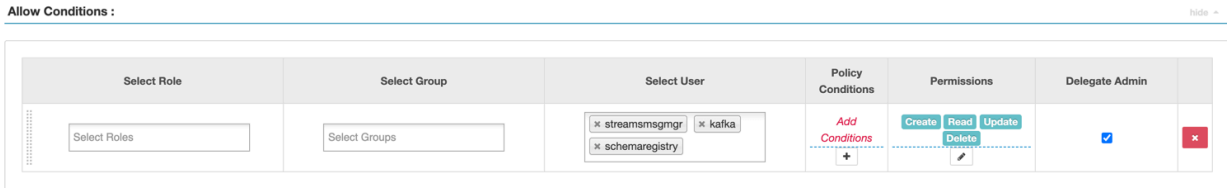
The page is organized by service. Each cluster is listed under its respective service. For example, the Schema Registry clusters in the environment are listed under Schema Registry.

4. Select a cluster from the Schema Registry section.
The **List of Policies** page appears.



5. Click the ID for a policy.
The **Edit Policy** page appears.

6. In the Allow Conditions section, add the user or group to the respective Select User or Select Group field.



7. From the Policy Conditions field, enter the appropriate IP address.
8. From the Permissions field, select the appropriate permission.
9. Click Save.

Results

The user now has the rights according to the policy and the permission you assigned to the user. These rights apply to all objects in the entities unless you specified otherwise in the Policy Conditions field.

Related Information

Schema Registry Authorization through Ranger Access Policies

Pre-defined Access Policies for Schema Registry

Create a Custom Access Policy

Create a Custom Access Policy

You can create a custom access policy for a specific Schema Registry entity, specify an access type, and add a user or user-group to the policy.

Before you begin

Determine the following information:

- The schema registry entity that the user needs access to.
- Whether the user requires all objects in the entity or specific objects.
- Whether the user needs read, view, edit, or delete permissions to the entity.
- If there are any IP addresses to include or exclude from the user's access.

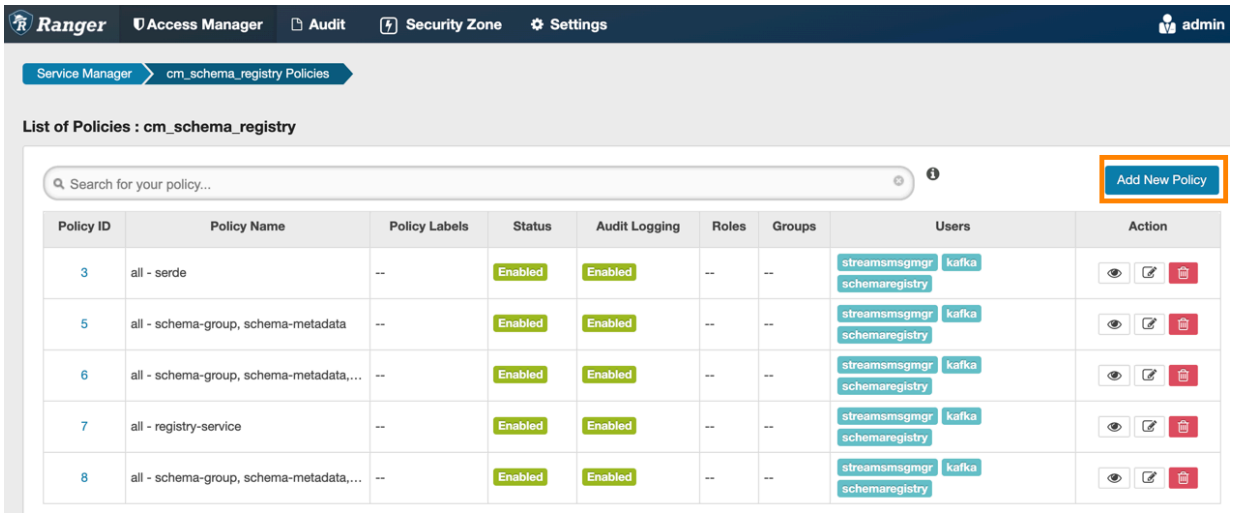
About this task

With a custom policy you can specify the Schema Registry entity and the type of access the user requires.

Procedure

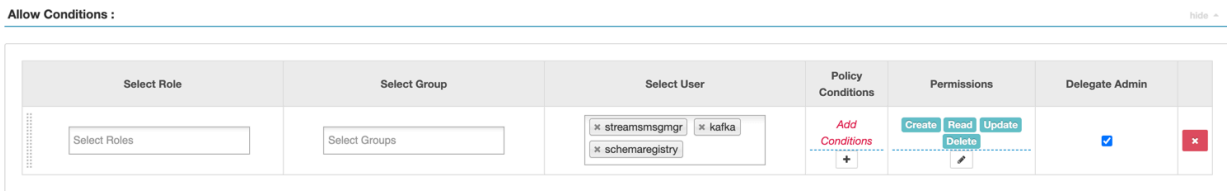
1. Go to the **Ranger List of Policies** page.

2. Click Add New Policy.



The **Create Policy** page appears.

3. Enter a unique name for the policy.
4. Optionally, enter a keyword in the Policy Label field to aid in searching for a policy.
5. Select a Schema Registry entity. You can choose the Schema Registry service, schema group, or serde. Then, do one of the following tasks:
   - If you want the user to access all the objects in the entity, enter *.
   - If you want to specify the objects in the entity that a user can access, enter the name of the object in the text field.
6. Optionally, enter a description.
7. In the Allow Conditions section, add the user or group to the respective Select User or Select Group field.



8. Optionally, from the Policy Conditions field, enter the appropriate IP address.
9. From the Permissions field, select the appropriate permission.
10. Click Save.

Results

The user now has the rights according to the policy and the permission you assigned to the user.

Related Information

Schema Registry Authorization through Ranger Access Policies

Pre-defined Access Policies for Schema Registry

Add the user or group to a pre-defined access policy

TLS Encryption

Transport Layer Security (TLS) is an industry standard set of cryptographic protocols for securing communications over a network. To encrypt sensitive information between the Cloudera Manager Server and cluster hosts, you must enable TLS.

You can choose to enable Auto-TLS or manually configure TLS.

Auto-TLS simplifies the process of enabling and managing TLS encryption on your cluster. When you enable Auto-TLS, an internal certificate authority (CA) is created and certificates deployed automatically across all cluster hosts. For more information on Auto-TLS, see *Configuring TLS Encryption for Cloudera Manager Using Auto-TLS*.

If you choose to enable TLS manually, you must create the TLS certificates making sure the certificates meet the requirements. Then configure Cloudera Manager and Schema Registry.

Related Information

Configuring TLS Encryption for Cloudera Manager Using Auto-TLS

TLS Certificate Requirements and Recommendations

If you choose to manually configure TLS, you must use your own certificates. The certificates must meet the requirements listed here.

Certificate Requirements

Verify the following minimum requirements:

- The KeyStore must contain only one PrivateKeyEntry. Using multiple private keys in one KeyStore is not supported.
- The KeyStore password and key/certificate password must be the same or no password should be set on the certificate.
- The unique KeyStores used on each cluster node must use the same KeyStore password and key/certificate password. Ambari and Cloudera Manager do not support defining unique passwords per host.
- The X509v3 ExtendedKeyUsages section of the certificate must have the following attributes:
  - clientAuth - This attribute is for TLS web client authentication.
  - serverAuth - This attribute is for TLS web server authentication.
- The signature algorithm used for the certificate must be sha256WithRSAEncryption (SHA-256).
- The certificates must not use wildcards. Each cluster node must have its own certificate.
- Subject Alternate Names (SANs) are mandatory and should at least include the FQDN of the host.
- Additional names for the certificate/host can be added to the certificate as SANs.
  - Add the FQDN used for the CN as a DNS SAN entry.
  - If you are planning to use a load balancer, include the FQDN for the load balancer as a DNS SAN entry.
- The X509v3 KeyUsage section of the certificate must include the following attributes:
  - DigitalSignature
  - Key_Encipherment

Cloudera Recommendations

Cloudera recommends the following security protocols:

- Use certificates that are signed by a CA. Do not issue self-signed certificates.
- Generate a unique certificate per host.

Configure TLS Encryption Manually for Schema Registry

If you do not want to enable Auto-TLS because for example, you need to use your own enterprise-generated certificates, you can manually enable TLS for Schema Registry.

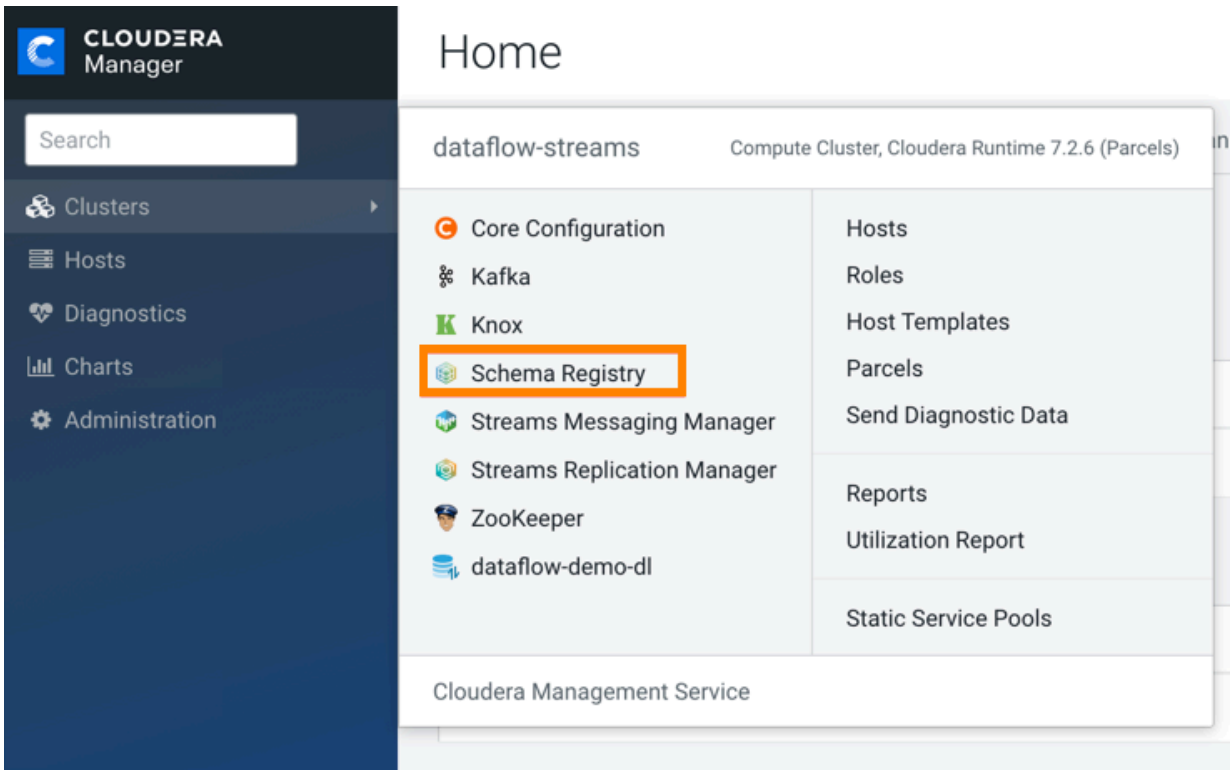
Before you begin

Ensure you have set up TLS for Cloudera Manager:

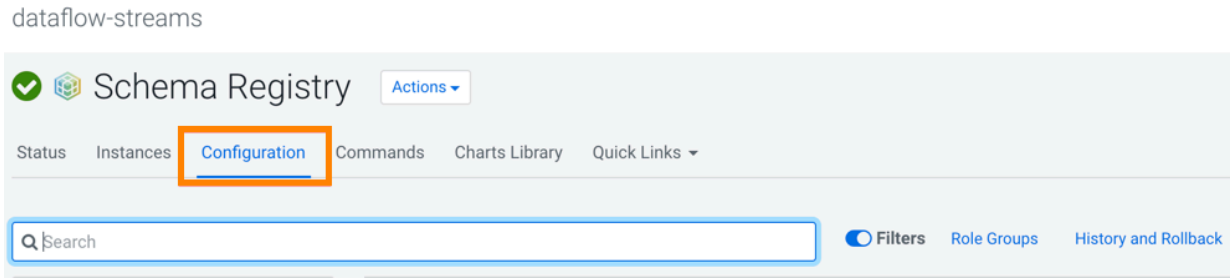
1. Review the requirements and recommendations for the certificates. See *TLS Certificate Requirements and Recommendations*.
2. Generate the TLS certificates and configure Cloudera Manager. See *Manually Configuring TLS Encryption for Cloudera Manager*.

Procedure

1. From the Cloudera Manager UI, click Cluster Schema Registry .

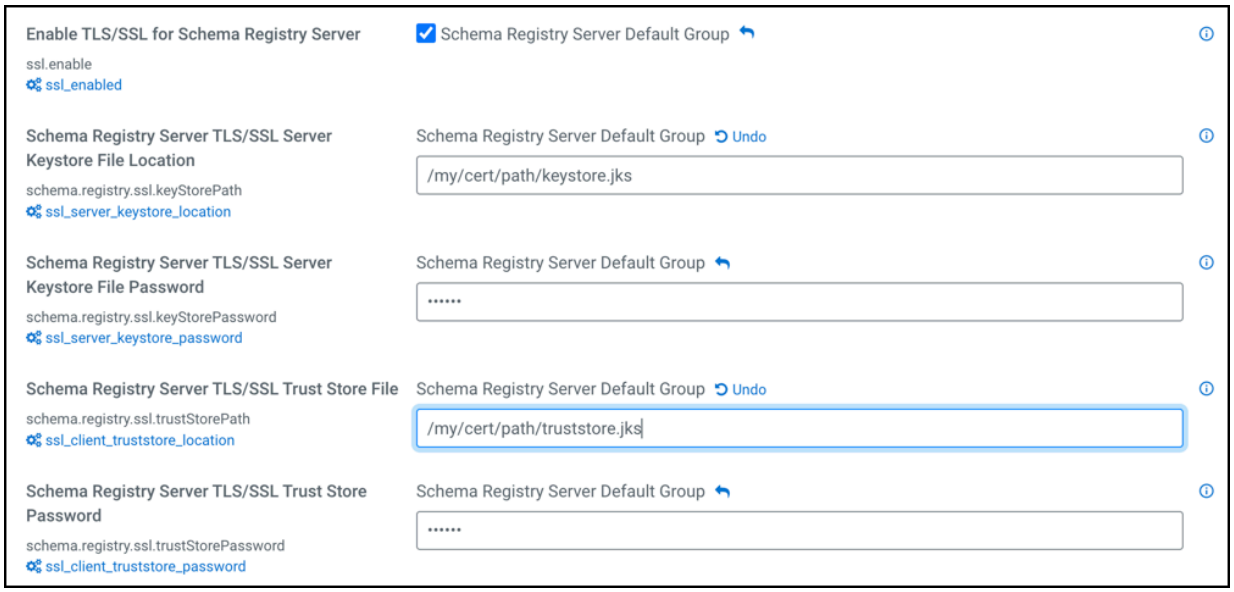


2. Click the **Configuration** tab.



3. Enter ssl in the Search field.
   The Security properties for Schema Registry appear.

4. Edit the Security properties.
   For example,



5. Click Save Changes.
6. Restart the Schema Registry service.

Related Information

TLS Certificate Requirements and Recommendations

Manually Configuring TLS Encryption for Cloudera Manager

Schema Registry TLS Properties

To enable and configure TLS manually for Schema Registry, edit the security properties according to the cluster configuration.

The following table lists the Security properties for Schema Registry:

Property	Description
Schema Registry Port (SSL) `schema.registry.ssl.port`	HTTPS port Schema Registry node runs on when SSL is enabled.
Schema Registry Admin Port (SSL) `schema.registry.ssl.adminPort`	HTTPS admin port Schema Registry node runs on when SSL is enabled.
SSL Keystore Type `schema.registry.ssl.keyStoreType`	The keystore type. It is blank by default but required if schema registry's ssl is enabled. e.g. PKCS12 or JKS. If it is left empty then this keystore type will come from CM settings.
SSL TrustStore Type `schema.registry.ssl.trustStoreType`	The truststore type. It is blank by default but required if schema registry's ssl is enabled. e.g. PKCS12 or JKS. If it is left empty then this keystore type will come from CM settings.

Property	Description
SSL ValidateCerts `schema.registry.ssl.validateCerts`	Whether or not to validate TLS certificates before starting. If enabled, it will refuse to start with expired or otherwise invalid certificates.
SSL ValidatePeers `schema.registry.ssl.validatePeers`	Whether or not to validate TLS peer certificates.
Version of oracle.net.ssl `schema.registry.oracle.net.ssl_version`	Oracle net ssl version.
Oracle TLS javax.net.ssl.keyStore `schema.registry.javax.net.ssl.keyStore`	Path to keystore file if enabling TLS using Oracle DB.
Oracle TLS javax.net.ssl.keyStoreType `schema.registry.javax.net.ssl.keyStoreType`	KeyStoreType type if enabling TLS using Oracle DB.
Oracle TLS javax.net.ssl.keyStorePassword `schema.registry.javax.net.ssl.keyStorePassword`	KeyStorePassword if enabling TLS using Oracle DB.
Oracle TLS javax.net.ssl.trustStore `schema.registry.javax.net.ssl.trustStore`	Required Path to truststore file if enabling TLS using Oracle DB.
Oracle TLS javax.net.ssl.trustStoreType `schema.registry.javax.net.ssl.trustStoreType`	Required Truststore type if enabling TLS using Oracle DB.
Oracle TLS javax.net.ssl.trustStorePassword `schema.registry.javax.net.ssl.trustStorePassword`	TrustStorePassword type if enabling TLS using Oracle DB.
Oracle TLS oracle.net.ssl_cipher_suites `schema.registry.oracle.net.ssl_cipher_suites`	Oracle net ssl cipher suites if enabling TLS using Oracle DB e.g. SSL_DH_DSS_WITH_DES_CBC_SHA.
Oracle TLS oracle.net.ssl_server_dn_match `schema.registry.oracle.net.ssl_server_dn_match`	Oracle ssl server domain name match if enabling TLS using Oracle DB.

Property	Description
Enable TLS/SSL for Schema Registry Server `ssl.enable`	Encrypt communication between clients and Schema Registry Server using Transport Layer Security (TLS) (formerly known as Secure Socket Layer (SSL)).
Schema Registry Server TLS/SSL Server JKS Keystore File Location `schema.registry.ssl.keyStorePath`	The path to the TLS/SSL keystore file containing the server certificate and private key used for TLS/SSL. Used when Schema Registry Server is acting as a TLS/SSL server.
Schema Registry Server TLS/SSL Server JKS Keystore File Password `schema.registry.ssl.keyStorePassword`	The password for the Schema Registry Server keystore file.
Schema Registry Server TLS/SSL Client Trust Store File `schema.registry.ssl.trustStorePath`	The location on disk of the trust store, in .jks format, used to confirm the authenticity of TLS/SSL servers that Schema Registry Server might connect to. This is used when Schema Registry Server is the client in a TLS/SSL connection. 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.
Schema Registry Server TLS/SSL Client Trust Store Password `schema.registry.ssl.trustStorePassword`	The password for the Schema Registry 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.

Schema Registry Authorization through Ranger Access Policies

User and group access to various Schema Registry functions is controlled through Apache Ranger.

Pre-defined access policies for Schema Registry allow the administrator to quickly add a user or user group to specify:

- Who can add/evolve schemas to a schema metadata.
- Who can view and edit schemas within a schema metadata.
- Who can upload the ser/des jar files.

If a higher level of granularity is necessary, the administrator can create an access policy and add the user or user-group to this custom policy.

Related Information

Pre-defined Access Policies for Schema Registry

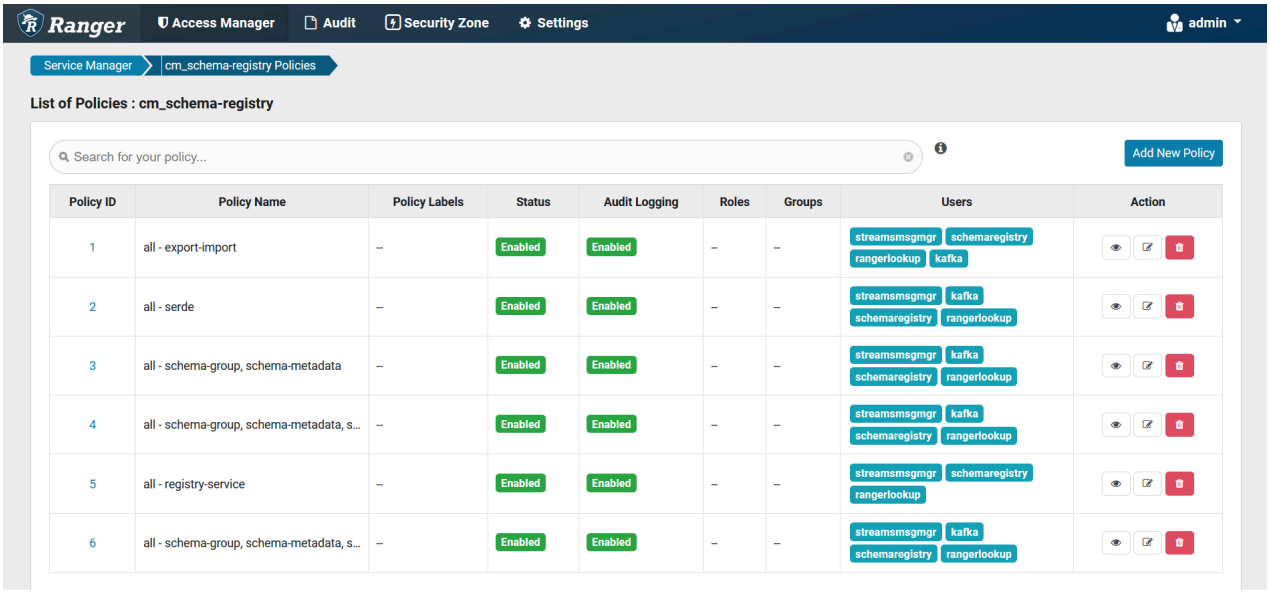
Add the user or group to a pre-defined access policy

Create a Custom Access Policy

Pre-defined Access Policies for Schema Registry

Based on a user's responsibilities, you can add users or a user group to one or more of the following pre-defined access policies for Schema Registry and you can specify the type of permission such as Create, Read, Update, and Delete.

The following image shows the pre-defined access policies for Schema Registry:



The following table describes the pre-defined access policies for Schema Registry:

Access Policy	Description
all - export-import	Allows users to import and export schemas to/from the Schema Registry service. For example, a user can import a .json file with schemas from a Confluent Kafka topic to Cloudera's Schema Registry.
all - serde	Allows users to store metadata for the format of how data should be read and how it should be written. Users can store JAR files for serializers and deserializers and then map the serdes to the schema.
all - schema-group, schema-metadata	Allows users to access the schema groups and schema metadata.
all - schema-group, schema-metadata, schema-branch	Allows users to access the schema groups, schema metadata, and schema branch.
all - registry-service	Allows users to access the schema registry service. If a user is added to this policy, the user can access all Schema Registry entities.
all - schema-group, schema-metadata, schema-branch, schema-version	Allows users to access the schema groups, schema metadata, schema branch, and schema version.

Related Information

Schema Registry Authorization through Ranger Access Policies
Add the user or group to a pre-defined access policy
Create a Custom Access Policy

Add the user or group to a pre-defined access policy

When an authenticated user attempts to view, create, edit, or delete a Schema Registry entity, the system checks whether the user has privileges to perform that action. These privileges are determined by the Ranger access policies that a user is associated with.

Before you begin

For Ranger policies to work, you must have a user group named schemaregistry. If you use UNIX PAM, the sche maregistry user group must be on the node that hosts Schema Registry.

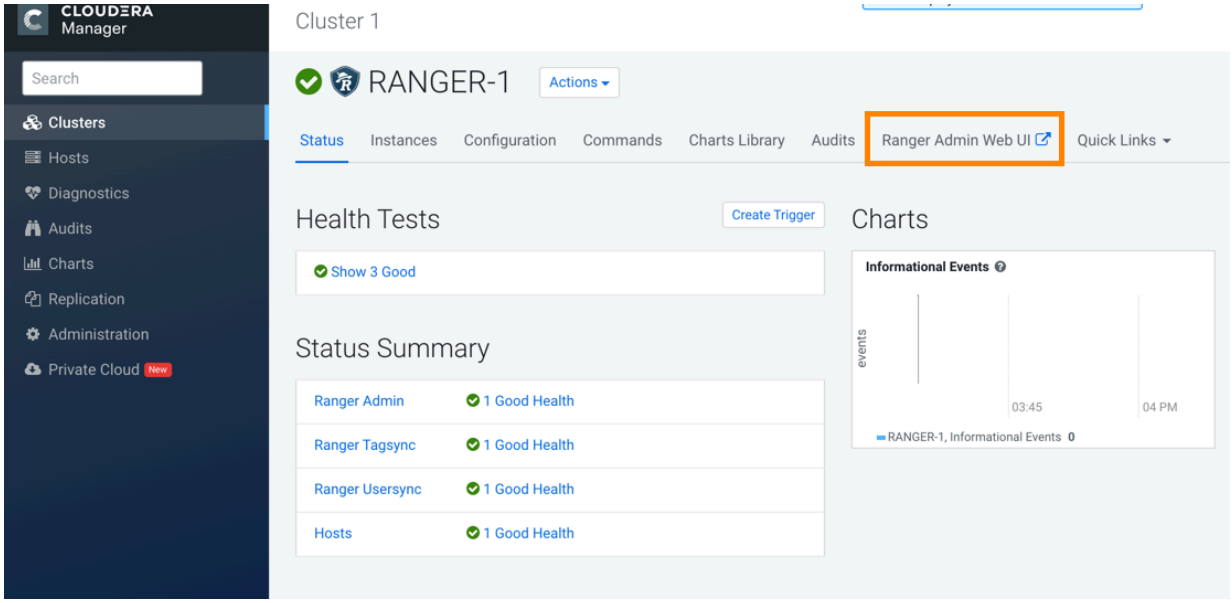
About this task

Determine the permissions required by a user or user group and accordingly add the user or group to the appropriate pre-defined access policy.

Each pre-defined access policy controls access to one or more Schema Registry entities.

Procedure

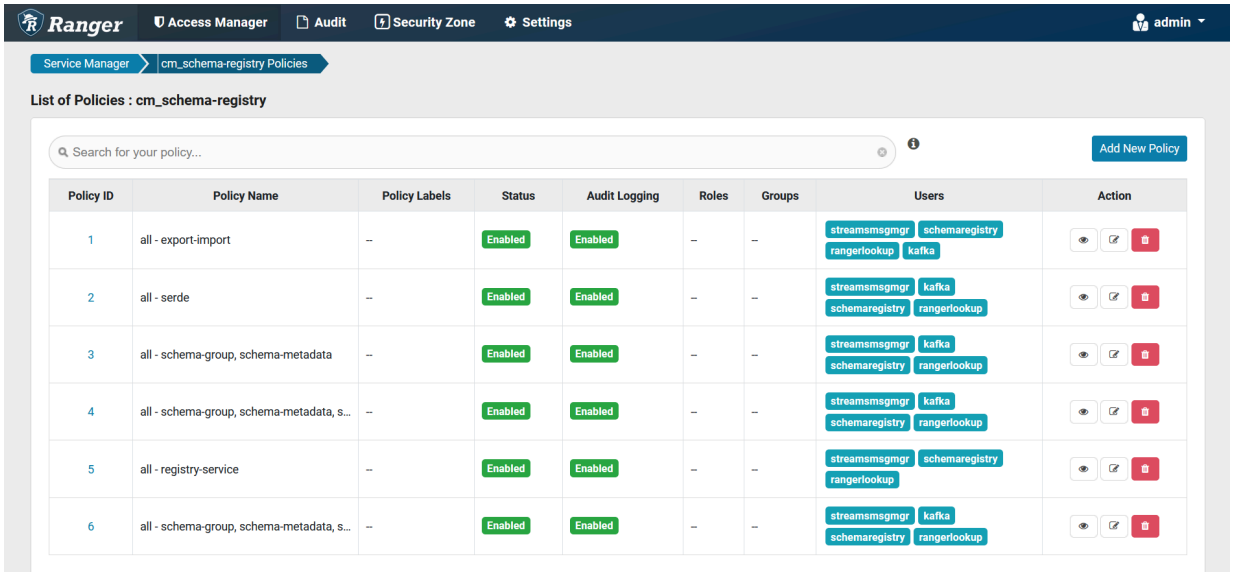
1. From the Cloudera Manager home page, click the Ranger link.
   The **Ranger** management page appears.
2. Click the Ranger Admin Web UI link.



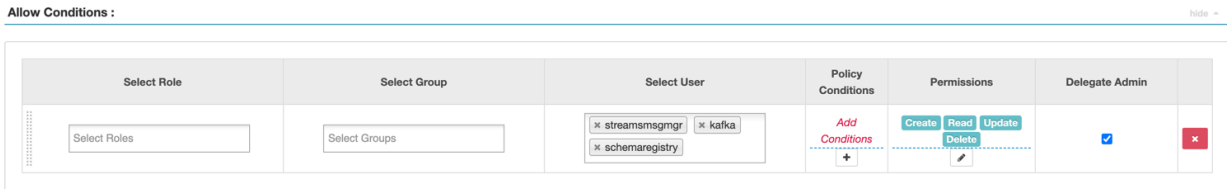
   The **Ranger Log In** page appears.
3. Enter your user name and password to log in.
   The **Ranger Service Manager** page appears.

   The page is organized by service. Each cluster is listed under its respective service. For example, the Schema Registry clusters in the environment are listed under Schema Registry.

4. Select a cluster from the Schema Registry section.
   The **List of Policies** page appears.



5. Click the ID for a policy.
   The **Edit Policy** page appears.
6. In the Allow Conditions section, add the user or group to the respective Select User or Select Group field.



7. From the Policy Conditions field, enter the appropriate IP address.
8. From the Permissions field, select the appropriate permission.
9. Click Save.

Results

The user now has the rights according to the policy and the permission you assigned to the user. These rights apply to all objects in the entities unless you specified otherwise in the Policy Conditions field.

Related Information

Schema Registry Authorization through Ranger Access Policies

Pre-defined Access Policies for Schema Registry

Create a Custom Access Policy

Create a Custom Access Policy

You can create a custom access policy for a specific Schema Registry entity, specify an access type, and add a user or user-group to the policy.

Before you begin

Determine the following information:

- The schema registry entity that the user needs access to.
- Whether the user requires all objects in the entity or specific objects.
- Whether the user needs read, view, edit, or delete permissions to the entity.

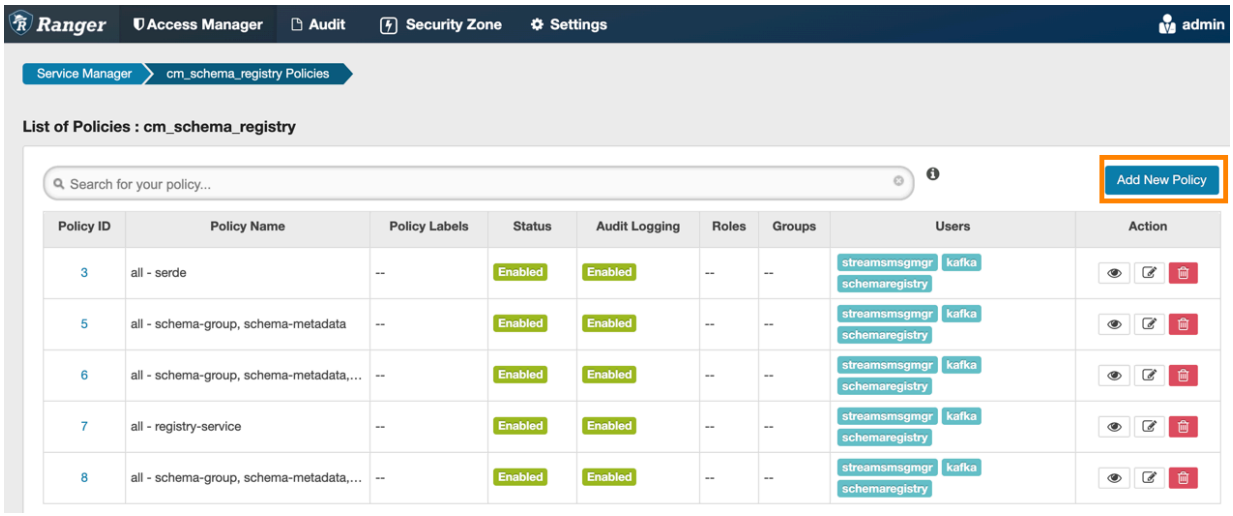
- If there are any IP addresses to include or exclude from the user's access.

About this task

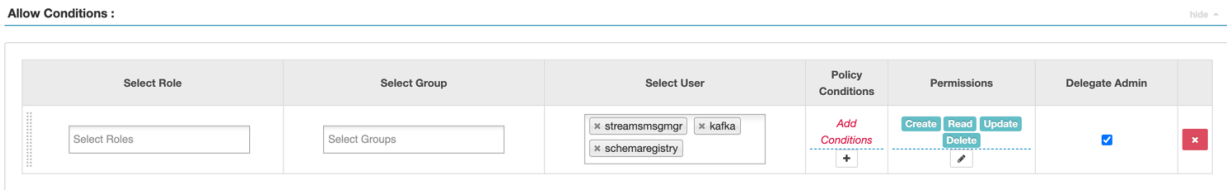
With a custom policy you can specify the Schema Registry entity and the type of access the user requires.

Procedure

1. Go to the **Ranger List of Policies** page.
2. Click Add New Policy.



   The **Create Policy** page appears.
3. Enter a unique name for the policy.
4. Optionally, enter a keyword in the Policy Label field to aid in searching for a policy.
5. Select a Schema Registry entity. You can choose the Schema Registry service, schema group, or serde. Then, do one of the following tasks:
   - If you want the user to access all the objects in the entity, enter *.
   - If you want to specify the objects in the entity that a user can access, enter the name of the object in the text field.
6. Optionally, enter a description.
7. In the Allow Conditions section, add the user or group to the respective Select User or Select Group field.



8. Optionally, from the Policy Conditions field, enter the appropriate IP address.
9. From the Permissions field, select the appropriate permission.
10. Click Save.

Results

The user now has the rights according to the policy and the permission you assigned to the user.

Related Information

Schema Registry Authorization through Ranger Access Policies

Pre-defined Access Policies for Schema Registry

Add the user or group to a pre-defined access policy

Customizing the Kerberos principal for Schema Registry

The Kerberos principal for Schema Registry is configured by default to use the same service principal as the default process user. However, you can change the default setting by providing a custom principal in Cloudera Manager.

About this task

Note: By default, Cloudera Manager configures CDP services to use the default Kerberos principal names. While it is possible to customize the Kerberos principal names for most cluster services by setting various configuration properties, it requires extensive custom configuration and, if absolutely required, we highly recommend working closely with Cloudera Professional services in doing so.

Procedure

1. Go to your cluster in Cloudera Manager.
2. Select Schema Registry from the list of services.
3. Select the **Configuration** tab.
4. Search for the Kerberos Principal by entering kerberos in the search field.
5. Enter a custom name in the Kerberos Principal field.
6. Click Save changes.
7. Click on Action Restart next to the Schema Registry service name to restart the service.