Hortonworks Data Platform

Ranger User Guide

(December 21, 2015)

docs.cloudera.com

Hortonworks Data Platform: Ranger User Guide

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, ZooKeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, training and partner-enablement services. All of our technology is, and will remain, free and open source.

Please visit the Hortonworks Data Platform page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. Feel free to contact us directly to discuss your specific needs.

Table of Contents

List of Tables

1. Security Introduction

Security is essential for organizations that store and process sensitive data in the Hadoop ecosystem. Many organizations must adhere to strict corporate security polices.

Hadoop is a distributed framework used for data storage and large-scale processing on clusters using commodity servers. Adding security to Hadoop is challenging because all the interactions do not follow the classic client-server pattern. In Hadoop the file system is partitioned and distributed, requiring authorization checks at multiple points; a submitted job is executed at a later time on nodes different than the node on which the client authenticated and submitted the job; secondary services such as a workflow system access Hadoop on behalf of users; and the system scales to thousands of servers and tens of thousands of concurrent tasks.

A Hadoop-powered "Data Lake" can provide a robust foundation for a new generation of Big Data analytics and insight, but can also increase the number of access points to an organization's data. As diverse types of enterprise data are pulled together into a central repository, the inherent security risks must be understood and addressed.

Hortonworks understands the importance of security and governance for every business. To ensure effective protection for our customers, we use a holistic approach based on five core security features:

- Administration
- Authentication and perimeter security
- Authorization
- Audit
- Data protection

This chapter provides an overview of the security features implemented in the Hortonworks Data Platform (HDP). Subsequent chapters in this guide provide more details on each of these security features.

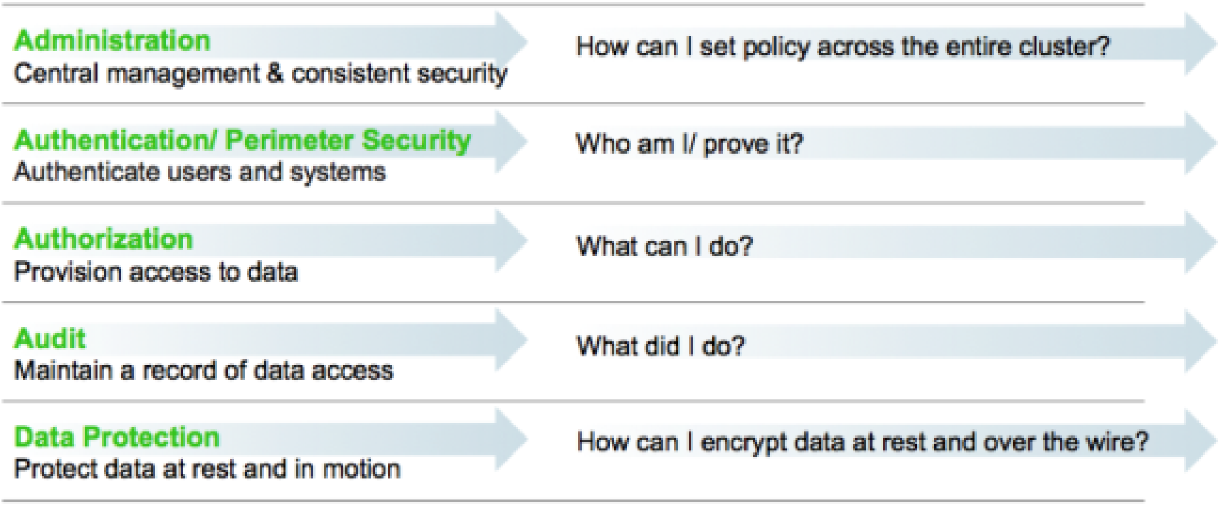
1.1. Understanding Data Lake Security

The general consensus in nearly every industry is that data is an essential new driver of competitive advantage. Hadoop plays a critical role in the modern data architecture by providing low-cost, large-scale data storage and processing. The successful Hadoop journey typically starts with data architecture optimization or new advanced analytic applications, which leads to the formation of a Data Lake. As new and existing types of data from sources such as machine sensors, server logs, clickstream data, and other sources flow into the Data Lake, it serves as a central repository based on shared Hadoop services that power deep organizational insights across a broad and diverse set of data.

The need to protect the Data Lake with comprehensive security is clear. As large and growing volumes of diverse data are channeled into the Data Lake, it will store vital and often highly sensitive business data. However, the external ecosystem of data and operational systems feeding the Data Lake is highly dynamic and can introduce new security threats on a regular basis. Users across multiple business units can access the Data Lake freely and refine, explore, and enrich its data at will, using methods of their own choosing, further increasing the risk of a breach. Any breach of this enterprise-wide data can be catastrophic: privacy violations, regulatory infractions, or the compromise of vital corporate intelligence. To prevent damage to the company's business, customers, finances, and reputation, IT leaders must ensure that their Data Lake meets the same high standards of security as any legacy data environment.

Only as Secure as the Weakest Link

Piecemeal protections are no more effective for a Data Lake than they would be in a traditional repository. Hortonworks firmly believes that effective Hadoop security depends on a holistic approach. Our framework for comprehensive security revolves around five pillars of security: administration, authentication/ perimeter security, authorization, audit, and data protection.



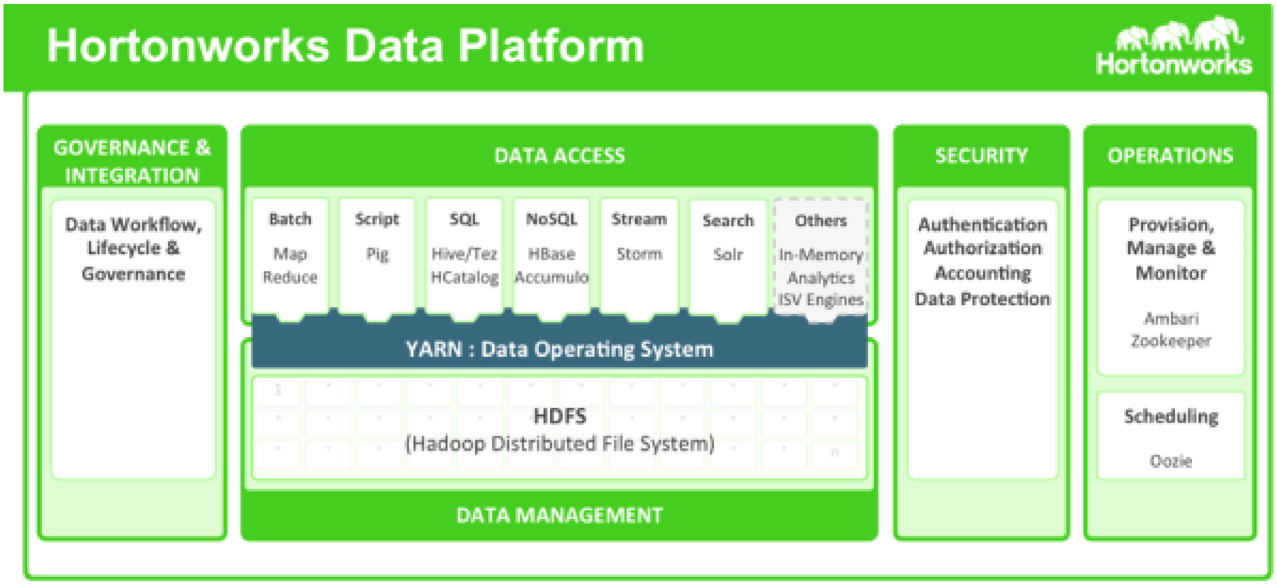
Administration Central management & consistent security	How can I set policy across the entire cluster?
Authentication/ Perimeter Security Authenticate users and systems	Who am I/ prove it?
Authorization Provision access to data	What can I do?
Audit Maintain a record of data access	What did I do?
Data Protection Protect data at rest and in motion	How can I encrypt data at rest and over the wire?

Requirements for Enterrprise-Grade Security

Security administrators must address questions and provide enterprise-grade coverage across each of these areas as they design the infrastructure to secure data in Hadoop. If any of these pillars is vulnerable, it will become a risk vector built into the very fabric of the company's Big Data environment. In this light, your Hadoop security strategy must address all five pillars, with a consistent implementation approach to ensure their effectiveness.

Needless to say, you can't achieve comprehensive protection across the Hadoop stack by using a hodgepodge of point solutions. Security must be an integral part of the platform on which your Data Lake is built. This bottom-up approach makes it possible to enforce and manage security across the stack through a central point of administration, thereby preventubg gaps and inconsistencies. This approach is especially important for Hadoop implementations where new applications or data engines are always on the horizon in the form of new Open Source projects – a dynamic scenario that can quickly exacerbate any vulnerability.

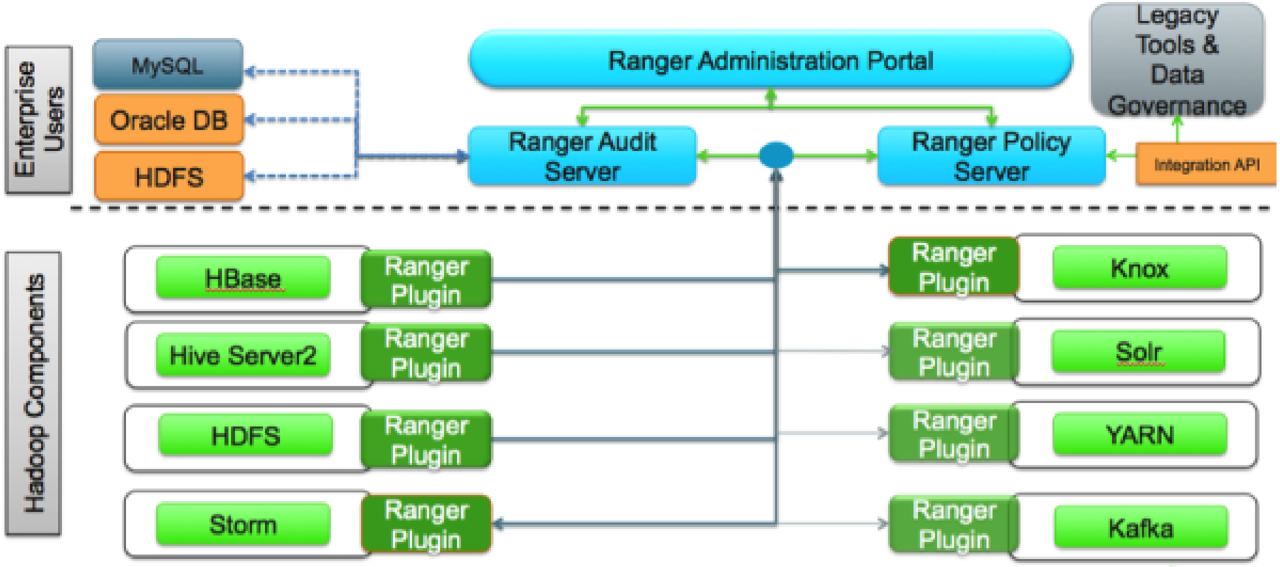
Hortonworks helps customers maintain the high levels of protection for enterprise data by building centralized security administration and management into the infrastructure of the Hortonworks Data Platform. HDP provides an enterprise-ready data platform with rich capabilities spanning security, governance, and operations. HDP includes powerful data security functionality that works across component technologies and integrates with preexisting EDW, RDBMS and MPP systems. By implementing security at the platform level, Hortonworks ensures that security is consistently administered to all of the applications across the stack, and simplifies the process of adding or remoning Hadoop applications.


Hortonworks Data Platform
Hortonworks
GOVERNANCE & INTEGRATION
Data Workflow, Lifecycle & Governance
DATA ACCESS
Batch
Map Reduce
Script
Pig
SQL
Hive/Tez HCatalog
NoSQL
HBase Accumulo
Stream
Storm
Search
Solr
Others
In-Memory Analytics ISV Engines
YARN : Data Operating System
HDFS
(Hadoop Distributed File System)
DATA MANAGEMENT
SECURITY
Authentication Authorization Accounting Data Protection
OPERATIONS
Provision, Manage & Monitor
Ambari Zookeeper
Scheduling
Oozie

The Hortonworks Data Platform

1.2. HDP Security Features

HDP uses Apache Ranger to provide centralized security administration and managemtent. The Ranger Administration Portal is the central interface for security administration. Users can create and update policies, which are then stored in a policy database. Ranger plugins (light-weight Java programs) are embedded within the processes of each cluster component. For example, the Ranger plugin for Apache Hive is embedded within Hiveserver2.

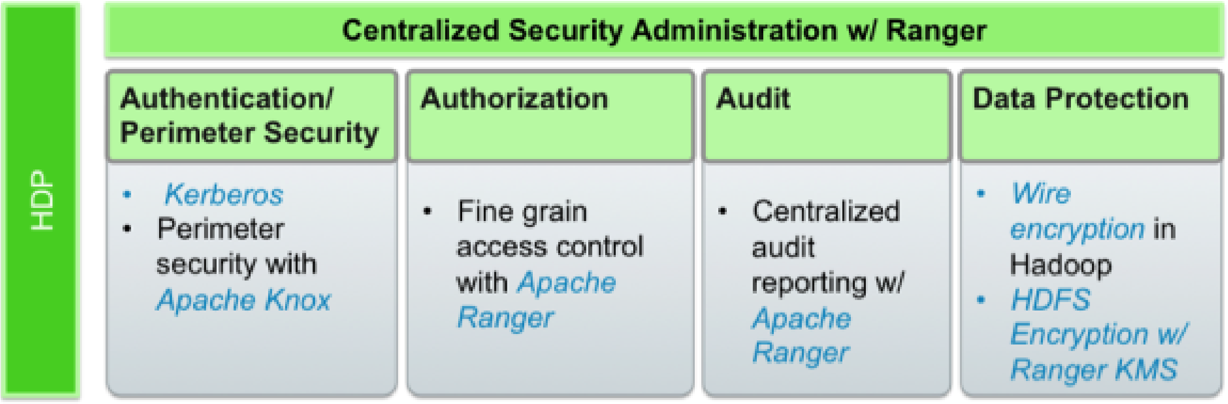


Apache Ranger Architecture

These plugins pull policies from a central server and store them locally in a file. When a user request comes through the component, these plugins intercept the request and evaluate it against the security policy. Plugins also collect data from the user request and follow a separate thread to send this data back to the audit server.

1.2.1. Administration

In order to deliver consistent security administration and management, Hadoop administrators require a centralized user interface that can be used to define, administer and manage security policies consistently across all of the Hadoop stack components.



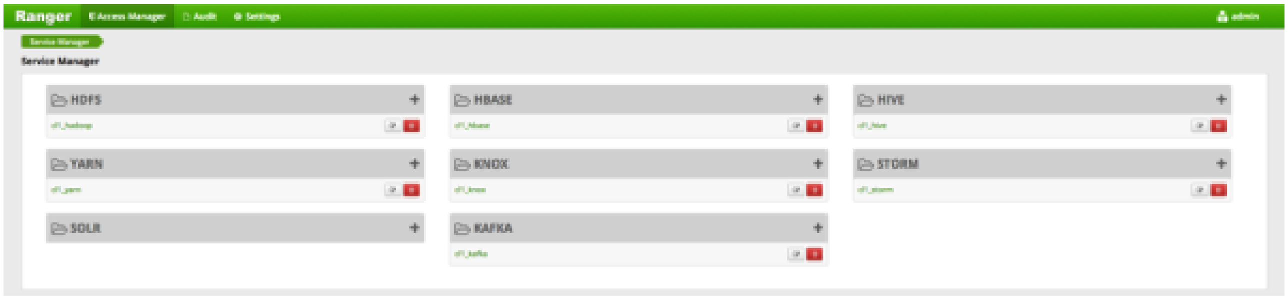
HDP

Centralized Security Administration w/ Ranger

Authentication/ Perimeter Security	Authorization	Audit	Data Protection
- *Kerberos* - Perimeter security with *Apache Knox*	- Fine grain access control with *Apache Ranger*	- Centralized audit reporting w/ *Apache Ranger*	- *Wire encryption* in Hadoop - *HDFS Encryption w/ Ranger KMS*

Ranger Centralized Security Administration

The Apache Ranger administration console provides a central point of administration for the other four pillars of Hadoop security.



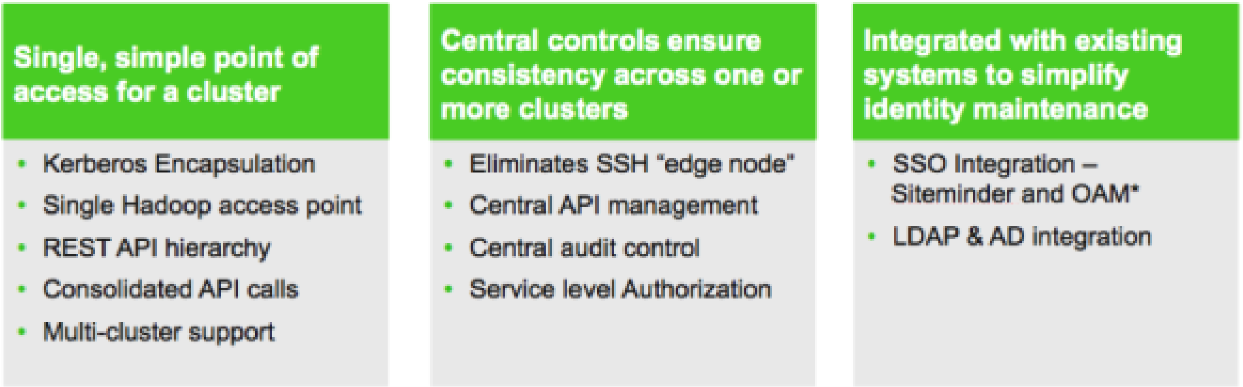
Ranger Admin Console

1.2.2. Authentication and Perimeter Security

Establishing user identity with strong authentication is the basis for secure access in Hadoop. Users need to reliably identify themselves and then have that identity propagated throughout the Hadoop cluster to access cluster resources. Hortonworks uses Kerberos for authentication. Kerberos is an industry standard used to authenticate users and resources within a Hadoop cluster. HDP also includes Ambari, which simplifies Kerberos setup, configuration, and maintenance.

Apache Knox Gateway is used to ensure perimeter security for Hortonworks customers. With Knox, enterprises can confidently extend the Hadoop REST API to new users without

Kerberos complexities, while also maintaining compliance with enterprise security policies. Knox provides a central gateway for Hadoop REST APIs that have varying degrees of authorization, authentication, SSL, and SSO capabilities to enable a single access point for Hadoop.



Single, simple point of access for a cluster

- Kerberos Encapsulation
- Single Hadoop access point
- REST API hierarchy
- Consolidated API calls
- Multi-cluster support

Central controls ensure consistency across one or more clusters

- Eliminates SSH "edge node"
- Central API management
- Central audit control
- Service level Authorization

Integrated with existing systems to simplify identity maintenance

- SSO Integration – Siteminder and OAM*
- LDAP & AD integration

Apache Knox Features

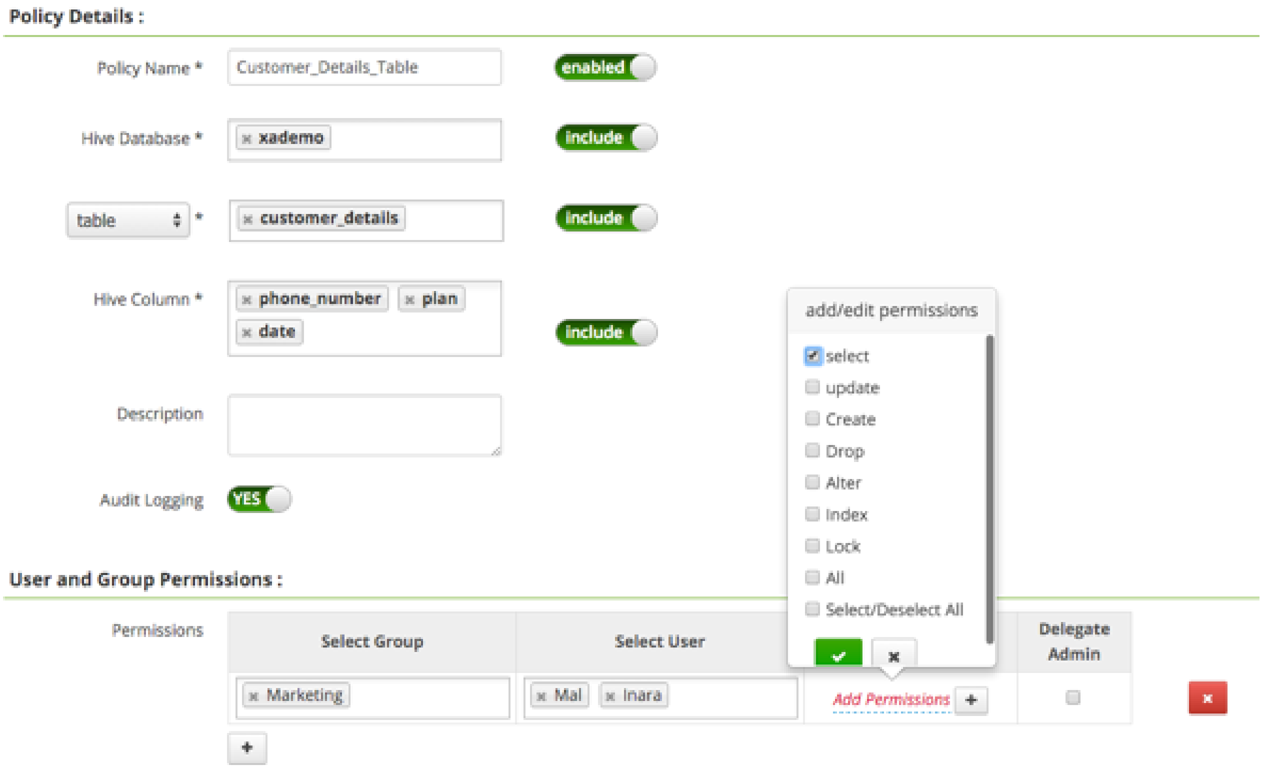
1.2.3. Authorization

Ranger manages fine-grained access control through a rich user interface that ensures consistent policy administration across Hadoop data access components. Security administrators have the flexibility to define security policies for a database, table and column, or a file, and can administer permissions for specific LDAP-based groups or individual users. Rules based on dynamic conditions such as time or geolocation, can also be added to an existing policy rule. The Ranger authorization model is highly pluggable and can be easily extended to any data source using a service-based definition.

Administrators can use Ranger to define a centralized security policy for the following Hadoop components:

- HDFS
- YARN
- Hive
- HBase
- Storm
- Knox
- Solr
- Kafka

Ranger works with standard authorization APIs in each Hadoop component, and is able to enforce centrally administered policies for any method used to access the data lake.



Policy Details :

Policy Name * Customer_Details_Table — enabled

Hive Database * xademo — include

table * customer_details — include

Hive Column * phone_number plan date — include

Description

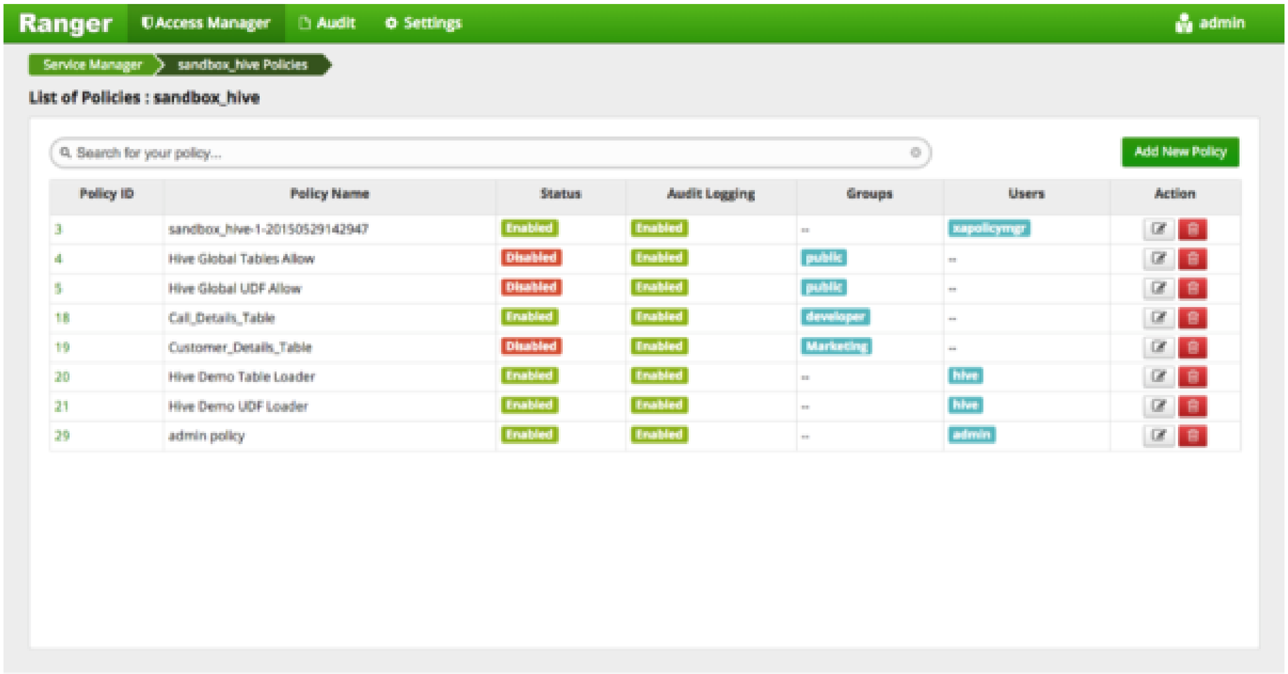
Audit Logging YES

User and Group Permissions :

Permissions	Select Group	Select User		Delegate Admin
	Marketing	Mal Inara	Add Permissions +	□

Ranger Security Policy Definitions

Ranger provides administrators with deep visibility into the security administration process that is required for auditing purposes. The combination of Ranger’s rich user interface with deep audit visibility makes it highly intuitive to use, enhancing productivity for security administrators.



List of Policies : sandbox_hive

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
3	sandbox_hive-1-20150529142947	Enabled	Enabled	--	xapolicymgr	
4	Hive Global Tables Allow	Disabled	Enabled	public	--	
5	Hive Global UDF Allow	Disabled	Enabled	public	--	
18	Call_Details_Table	Enabled	Enabled	developer	--	
19	Customer_Details_Table	Disabled	Enabled	Marketing	--	
20	Hive Demo Table Loader	Enabled	Enabled	--	hive	
21	Hive Demo UDF Loader	Enabled	Enabled	--	hive	
29	admin policy	Enabled	Enabled	--	admin	

Ranger Security Policy Overview

1.2.4. Audit

As customers deploy Hadoop into corporate data and processing environments, metadata and data governance must be vital parts of any enterprise-ready data lake. For these reasons, Hortonworks established the Data Governance Initiative (DGI) with Aetna, Merck, Target, and SAS to introduce a common approach to Hadoop data governance into the open source community. This initiative has since evolved into a new open source project called Apache Atlas. Apache Atlas is a set of core foundational governance services that enables enterprises to effectively and efficiently meet their compliance requirements within Hadoop, and also allows integration with the complete enterprise data ecosystem. These services include:

- Search and Lineage for datasets
- Metadata-driven data access control
- Indexed and searchable centralized auditing operational events
- Data lifecycle management – ingestion to disposition
- Metadata interchange with other tools

Ranger also provides a centralized framework for collecting access audit history and easily reporting this data, including the ability to filter data based on various parameters. HDP enhances audit information that is captured within various components within Hadoop, and provides insights through this centralized reporting capability.

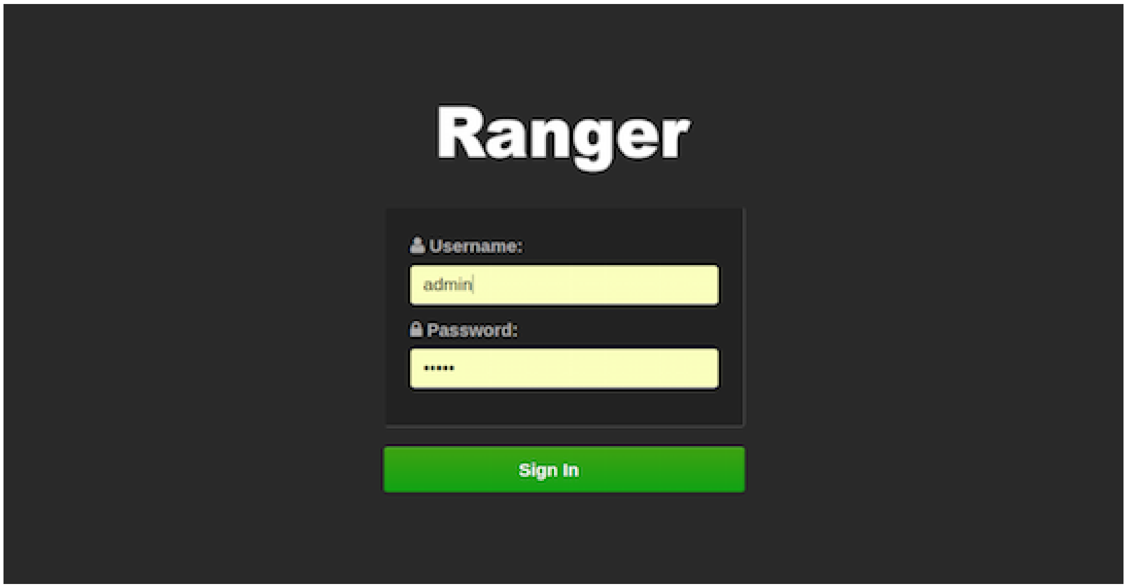
1.2.5. Data Protection

Data protection adds a robust layer of security by making data unreadable in transit over the network or at rest on a disk. HDP fully satisfies enterprise requirements for security and compliance by using transparent data encryption (TDE) to encrypt data for HDFS files, along with a Ranger-embedded open source Hadoop key management store (KMS). Ranger provides security administrators with the ability to manage keys and authorization policies for KMS. Hortonworks is also working extensively with its encryption partners to integrate HDFS encryption with enterprise-grade key management frameworks. With Hortonworks, our customers have the flexibility to leverage an open source key management store (KMS), or use enterprise-wide KMS solutions provided by the partner ecosystem.

Encryption in HDFS, combined with KMS access policies maintained by Ranger, prevents rogue Linux or Hadoop administrators from accessing data, and supports segregation of duties for both data access and encryption.

2. Opening and Closing the Console

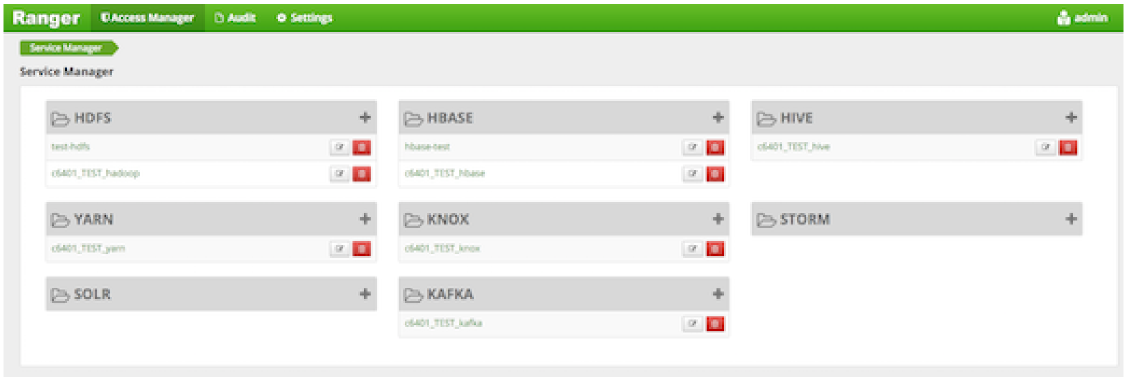
To open the Ranger Console, log in to the Ranger portal at `http://<your_ranger_server_address>:6080`. To log in, enter your username and password, then click **Sign In**.



Username: admin

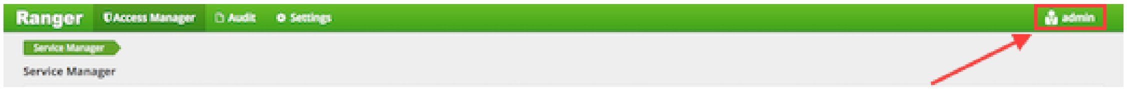
Password: •••••

Ranger Console Home Page

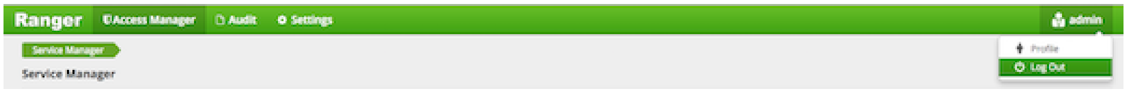


Ranger Login Console

Once you log in, your user name is also displayed on the Ranger Console home page.



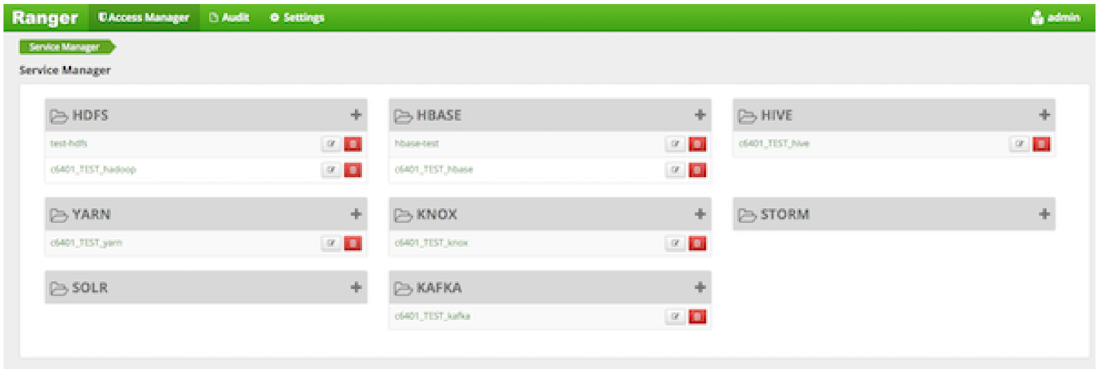
To log out of the Ranger Console, click your user name in the top menu, then select **Log Out**.



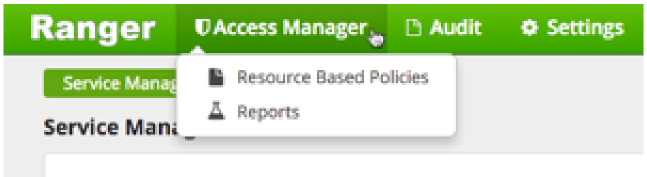
3. Console Operations Summary

The Ranger console controls four types of functions:

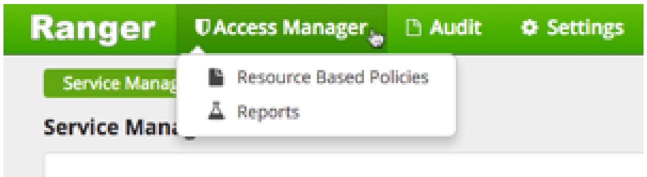
- **Service Manager** - (dispayed when you log in). You can use the Service Manager page to create and administer resource-based services and policies.



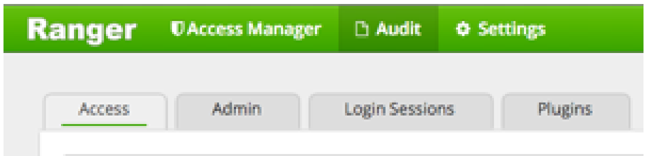
  The Access Manager menu includes the Resource Based Policies and Reports menu options.



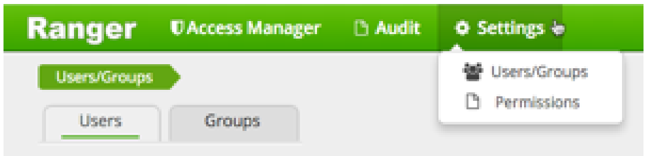
- **Access Manager > Resource Based Policies** - opens the Service Manager page for resource-based policies. You can use the Service Manager page to create and administer resource-based services and policies.



- **The Audit tab** - monitors user activity at the resource level, and conditional auditing based on users, groups, or time. The Audit page includes the Access, Admin, Login Sessions, and Plugins tabs.



- **The Settings tab** - manages users and groups, and assigns policy permissions to users and groups. The dropdown menu includes the Users/Groups and Permissions menu options.

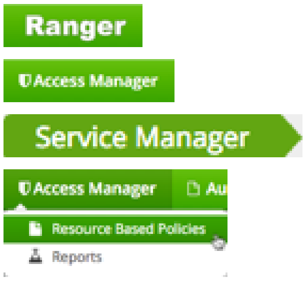

Ranger
Access Manager
Audit
Settings
Users/Groups
Permissions
Users/Groups
Users
Groups

4. Configuring Services

The Ranger Access Manager is open by default in the Ranger Console. To return to the Access Manager from any tab in the Ranger Console, go to the top left corner of the console and click **Ranger, Access Manager, Service Manager**, or **Access Manager>Resource Based Policies**.

Note

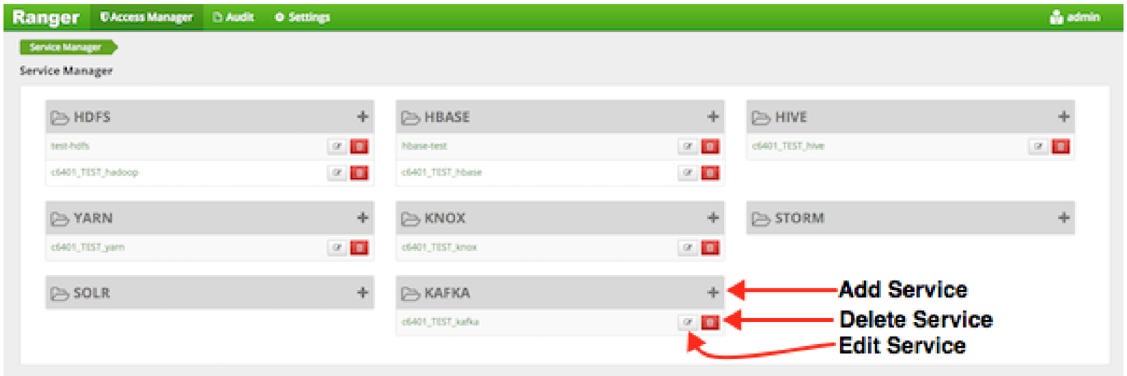
The Ambari Ranger installation procedure automatically configures these services, so there should be no need to a add a service manually.



- **To add a new service** to Resource Based Policies, click the
  ➕
  icon in the applicable box on the Service Manager page. Enter the required configuration settings, then click **Add** (Add).

- **To edit a service**, click
  ✎
  to the right of the entry for that service. Resource Based Policies displays an expanded view of that service, including a list of the policies it contains, their status, and the groups designated to administer those policies.

- **To delete a service** from Resource Based Policies, click
  🗑
  to the right of the entry for that service. Deleting a service also deletes all the policies within that service.



Ranger Service Manager Console

This section describes how to configure services in:

- Configure an HBase Service [12]
- Configure an HDFS Service [14]
- Configure a Hive Service [17]
- Configure a Kafka Service [18]
- Configure a Knox Service [20]
- Configure a Solr Service [22]
- Configure a Storm Service [24]
- Configure a YARN Service [26]

4.1. Configure an HBase Service

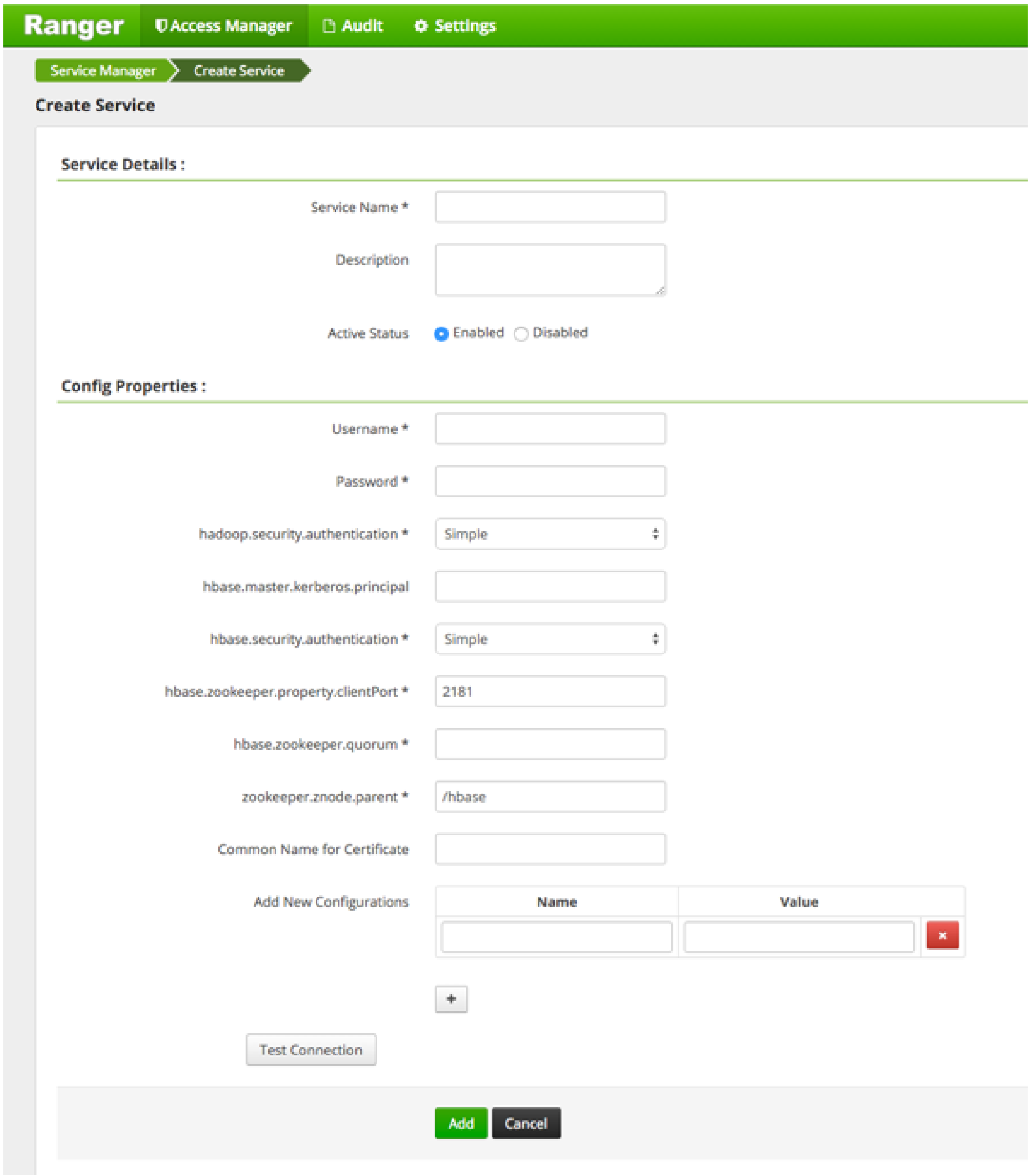
Use the following steps to add a service to HBase:

1. On the Service Manager page, click the

   ➕

   icon next to HBase.

   The Create Service page appears.



Create Service

Service Details :

Service Name *	
Description	
Active Status	● Enabled ○ Disabled

Config Properties :

Username *	
Password *	
hadoop.security.authentication *	Simple
hbase.master.kerberos.principal	
hbase.security.authentication *	Simple
hbase.zookeeper.property.clientPort *	2181
hbase.zookeeper.quorum *	
zookeeper.znode.parent *	/hbase
Common Name for Certificate	
Add New Configurations	Name / Value

2. Enter the following information on the Create Service page:

Table 4.1. Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Table 4.2. Config Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
hadoop.security.authorization	The complete connection URL, including port and database name. (Default port: 10000.) For example, on the sandbox, jdbc:hive2://sandbox:10000/.
hbase.master.kerberos.principal	The Kerberos principal for the HBase Master. (Required only if Kerberos authenticaiton is enabled.)
hbase.security.authentication	As noted in the hadoop configuration file hbase-site.xml.
hbase.zookeeper.property.clientPort	As noted in the hadoop configuration file hbase-site.xml.
hbase.zookeeper.quorum	As noted in the hadoop configuration file hbase-site.xml.
zookeeper.znode.parent	As noted in the hadoop configuration file hbase-site.xml.
Common Name for Certificate	The name of the certificate. This field is interchangeably named **Common Name For Certificate** and **Ranger Plugin SSL CName** in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click **Test Connection**.

4. Click **Add**.

4.2. Configure an HDFS Service

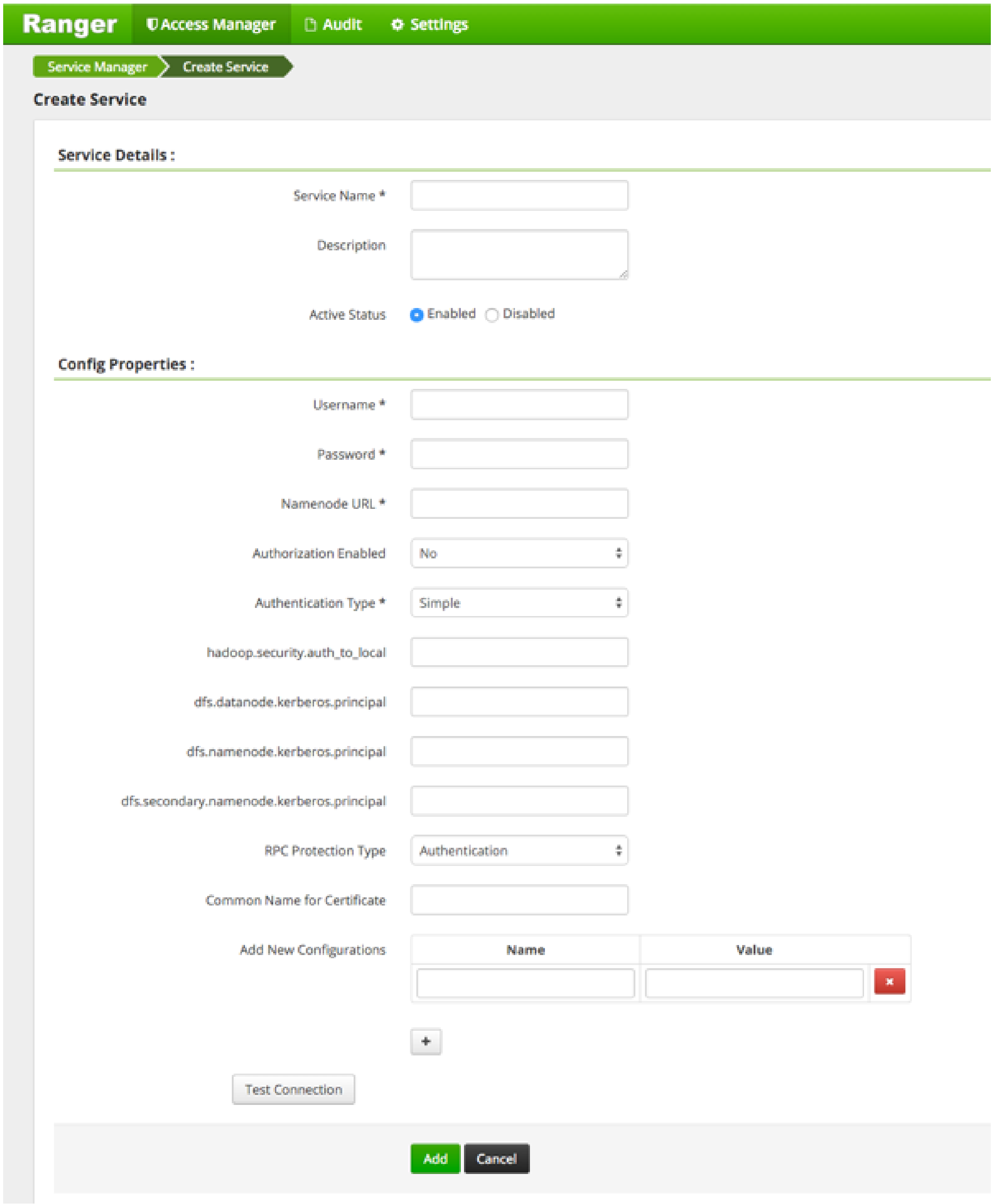
Use the following stetps to add a service to HDFS:

1. On the Service Manager page, click the

   ➕

   icon next to HDFS.

   The Create Service page appears.



Create Service

Service Details :

Service Name *

Description

Active Status ● Enabled ○ Disabled

Config Properties :

Username *

Password *

Namenode URL *

Authorization Enabled: No

Authentication Type *: Simple

hadoop.security.auth_to_local

dfs.datanode.kerberos.principal

dfs.namenode.kerberos.principal

dfs.secondary.namenode.kerberos.principal

RPC Protection Type: Authentication

Common Name for Certificate

Add New Configurations

Name	Value

2. Enter the following information on the Create Service page:

Table 4.3. Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Table 4.4. Config Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
Namenode URL	hdfs://*NAMENODE_FQDN*:*8020* The location of the Hadoop HDFS service, as noted in the hadoop configuration file core-site.xml OR (if this is a HA environment) the path for the primary NameNode. This field was formerly named fs.defaultFS.
Authorization Enabled	Authorization involves restricting access to resources. If enabled, user need authorization credentials.
Authentication Type	The type of authorization in use, as noted in the hadoop configuration file core-site.xml; either `simple` or `Kerberos`. (Required only if authorization is enabled). This field was formerly named hadoop.security.authorization.
hadoop.security.auth_to_local	Maps the login credential to a username with Hadoop; use the value noted in the hadoop configuration file, core-site.xml.
dfs.datanode.kerberos.principal	The principal associated with the datanode where the service resides, as noted in the hadoop configuration file hdfs-site.xml. (Required only if Kerberos authenticaiton is enabled).
dfs.namenode.kerberos.principal	The principal associated with the NameNode where the service resides, as noted in the hadoop configuration file hdfs-site.xml. (Required only if Kerberos authentication is enabled).
dfs.secondary.namenode.kerberos.principal	The principal associated with the secondary NameNode where the service resides, as noted in the hadoop configuration file hdfs-site.xml. (Required only if Kerberos authentication is enabled).
RPC Protection Type	Only authorised user can view, use, and contribute to a dataset. A list of protection values for secured SASL connections. Values: Authentication, Integrity, Privacy
Common Name For Certificate	The name of the certificate. This field is interchangeably named **Common Name For Certificate** and **Ranger Plugin SSL CName** in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click **Test Connection**.

4. Click **Add**.

4.3. Configure a Hive Service

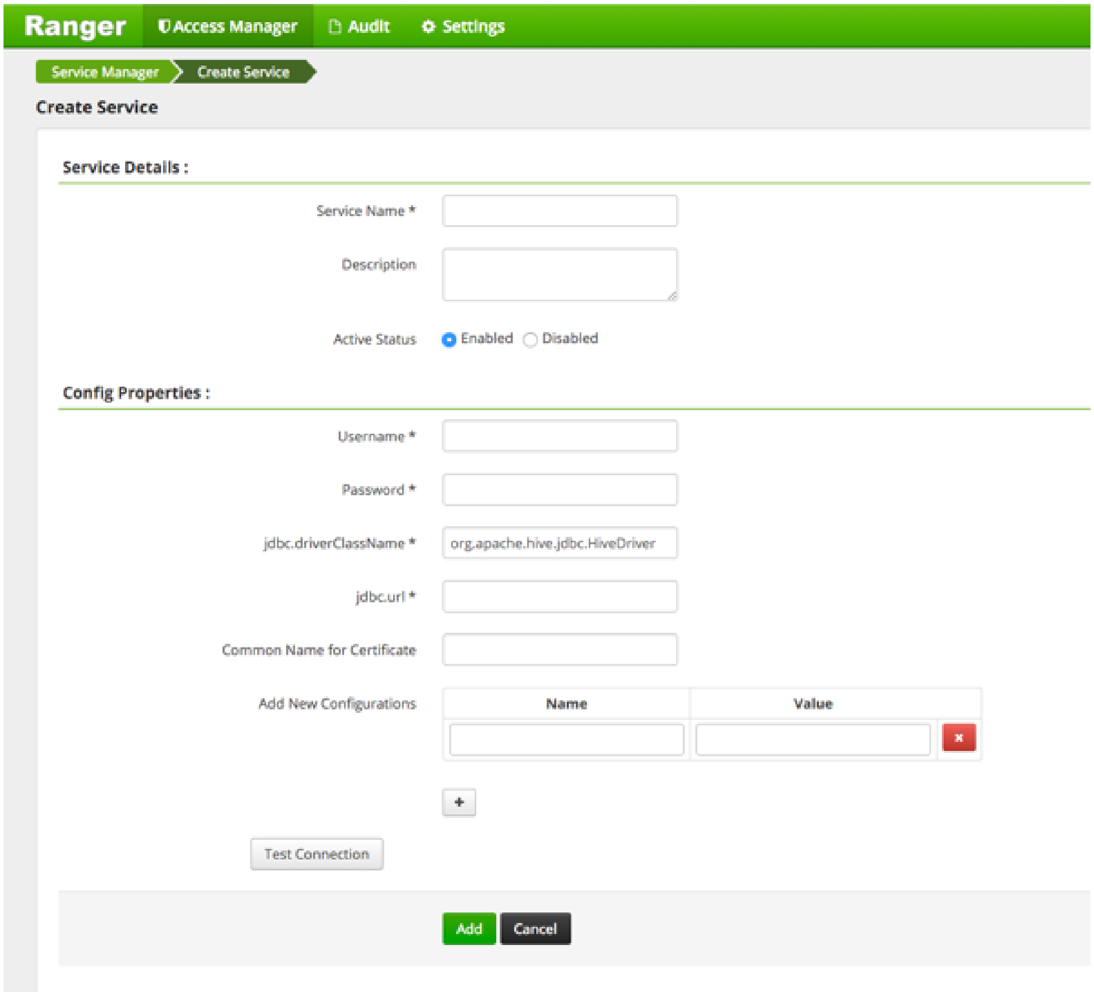
Use the following steps to add a service to Hive:

1. On the Service Manager page, click the

   +

   next to Hive.

   The Create Service page appears.



Create Service

Service Details :

Service Name *

Description

Active Status ● Enabled ○ Disabled

Config Properties :

Username *

Password *

jdbc.driverClassName * org.apache.hive.jdbc.HiveDriver

jdbc.url *

Common Name for Certificate

Add New Configurations

Name	Value

2. Enter the following information on the Create Service page:

Table 4.5. Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Table 4.6. Config Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
jdbc.driver ClassName	The full classname of the driver used for Hive connections. Default: org.apache.hive.jdbc.HiveDriver
jdbc.url	The complete connection URL, including port and database name. (Default port: 10000.) For example, on the sandbox, jdbc:hive2://sandbox:10000/.
Common Name For Certificate	The name of the certificate. This field is interchangeably named **Common Name For Certificate** and **Ranger Plugin SSL CName** in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click **Test Connection**.

4. Click **Add**.

4.4. Configure a Kafka Service

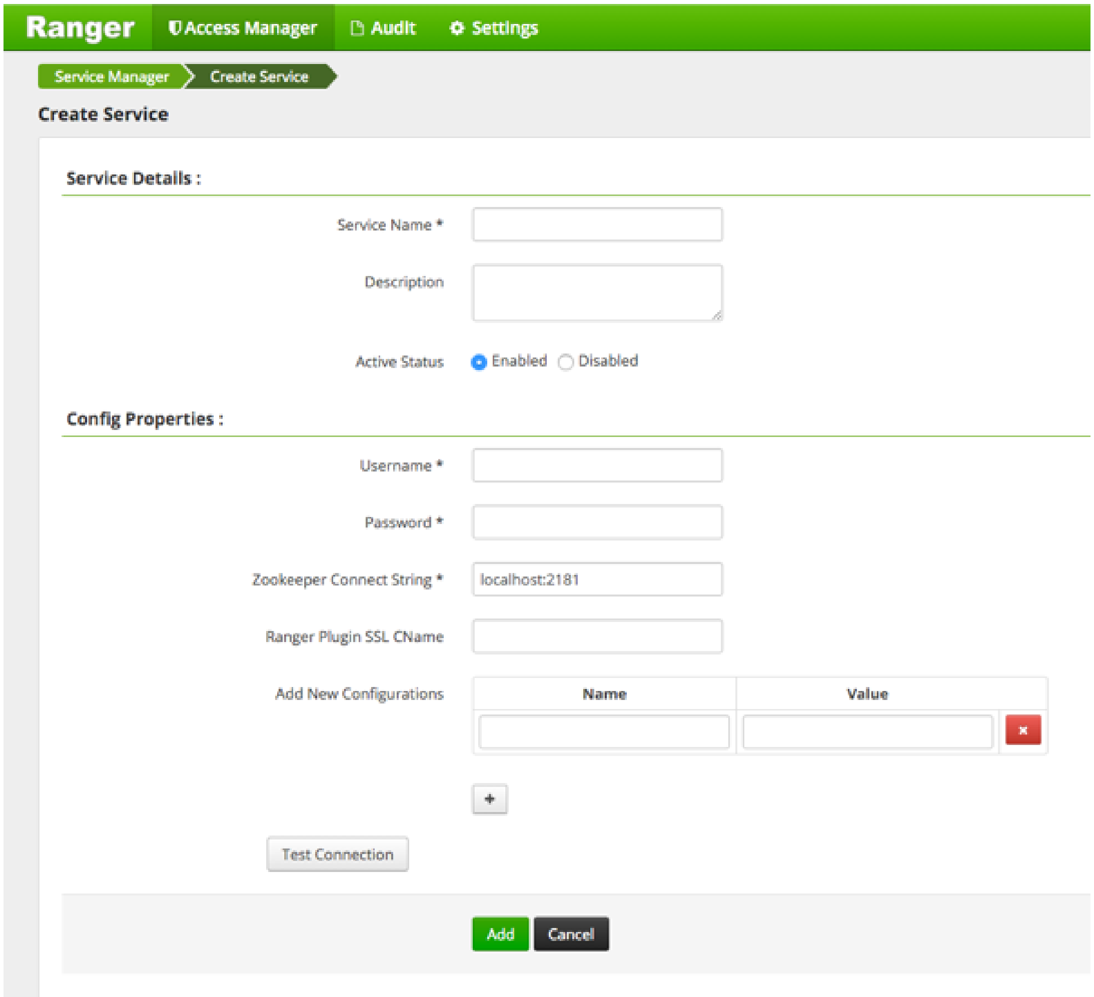
Use the following steps to add a service to Kafka:

1. On the Service Manager page, click the

   +

   next to Kafka.

   The Create Service page appears.



Create Service

Service Details :

Service Name *

Description

Active Status ● Enabled ○ Disabled

Config Properties :

Username *

Password *

Zookeeper Connect String * localhost:2181

Ranger Plugin SSL CName

Add New Configurations

Name	Value

2. Enter the following information on the Create Service page:

Table 4.7. Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Table 4.8. Config Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
ZooKeeper Connect String	Defaults to localhost:2181 (Provide FQDN of zookeeper host : 2181).
Ranger Plugin SSL CName	Provide common.name.for.certificate which is registered with Ranger (in Wire Encryption environment).

Field name	Description
	This field is interchangeably named **Common Name For Certificate** and **Ranger Plugin SSL CName** in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click **Test Connection**.

4. Click **Add**.

4.5. Configure a Knox Service

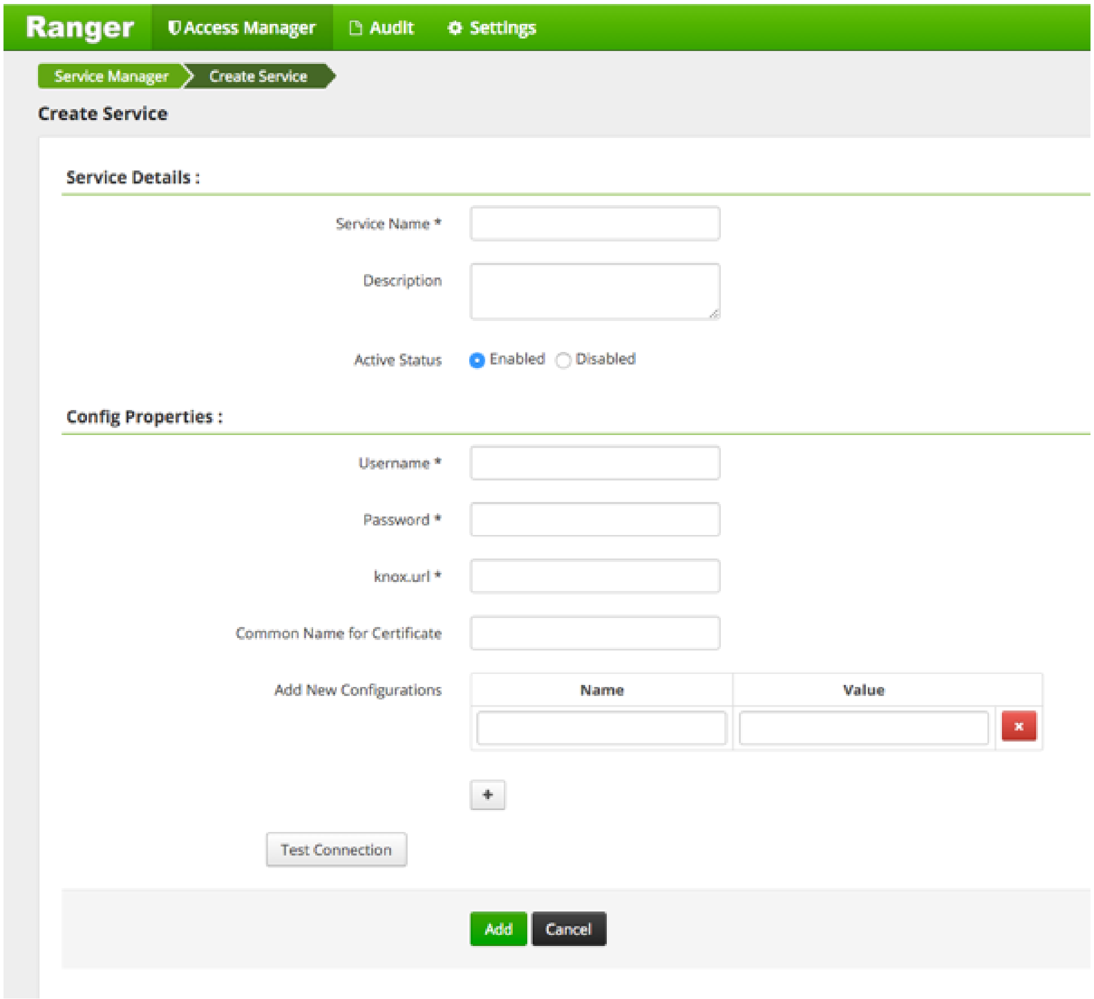
Use the following steps to add a service to Knox:

1. On the Service Manager page, click the

   +

   next to Knox.

   The Create Service page appears.



Create Service

Service Details :

Service Name *

Description

Active Status ● Enabled ○ Disabled

Config Properties :

Username *

Password *

knox.url *

Common Name for Certificate

Add New Configurations

2. Enter the following information on the Create Service page:

Table 4.9. Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Table 4.10. Config Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
knox.url	The Gateway URL for Knox.
Common Name For Certificate	The name of the certificate.

Field name	Description
	This field is interchangeably named **Common Name For Certificate** and **Ranger Plugin SSL CName** in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click **Test Connection**.

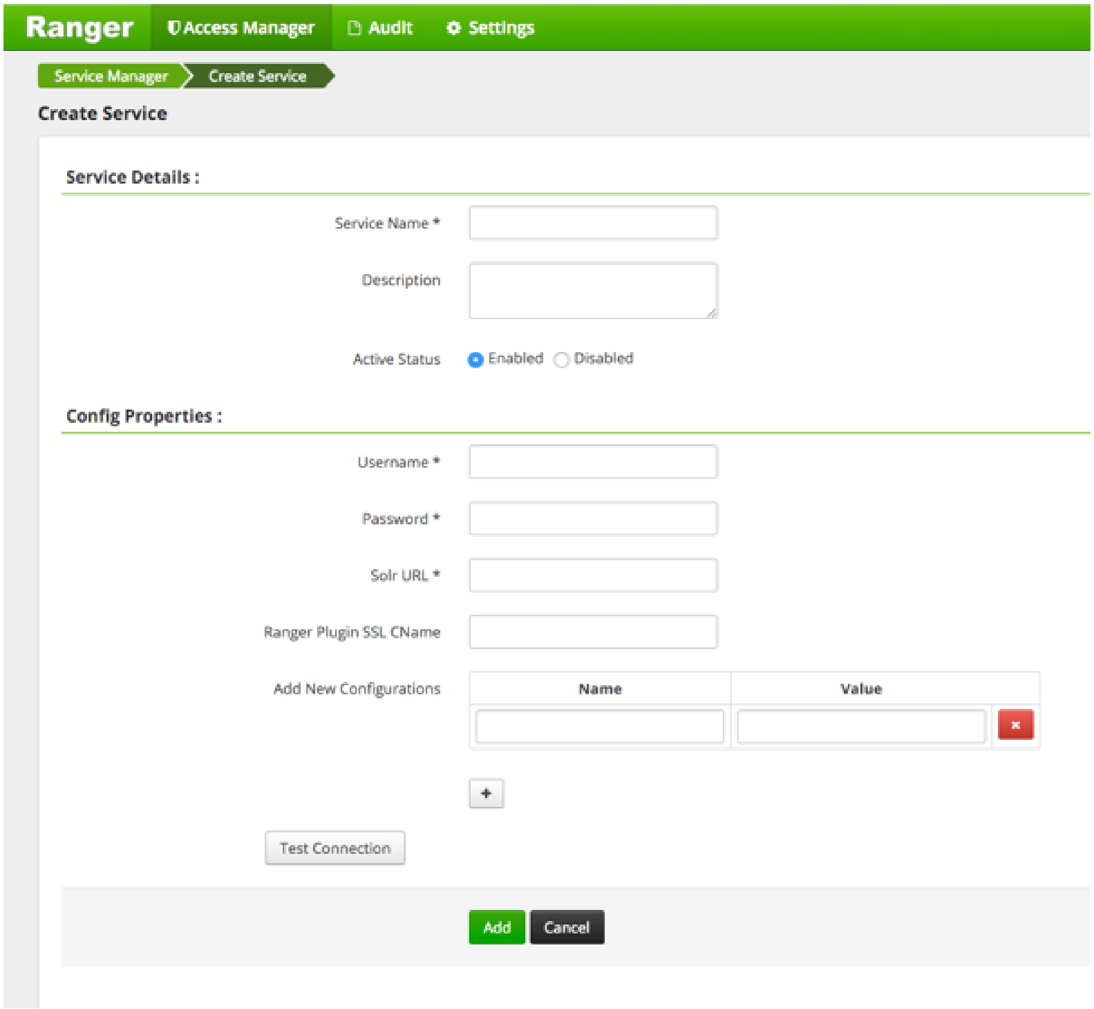
4. Click **Add**.

4.6. Configure a Solr Service

Use the following steps to add a service to Solr:

1. On the Service Manager page, click the
   ➕
   next to Solr.

   The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 4.11. Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Table 4.12. Config Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
Solr URL	http://*Solr_host:6083*
Ranger Plugin SSL CName	Provide common.name.for.certificate which is registered with Ranger (in Wire Encryption environment).

Field name	Description
	This field is interchangeably named **Common Name For Certificate** and **Ranger Plugin SSL CName** in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click **Test Connection**.

4. Click **Add**.

4.7. Configure a Storm Service

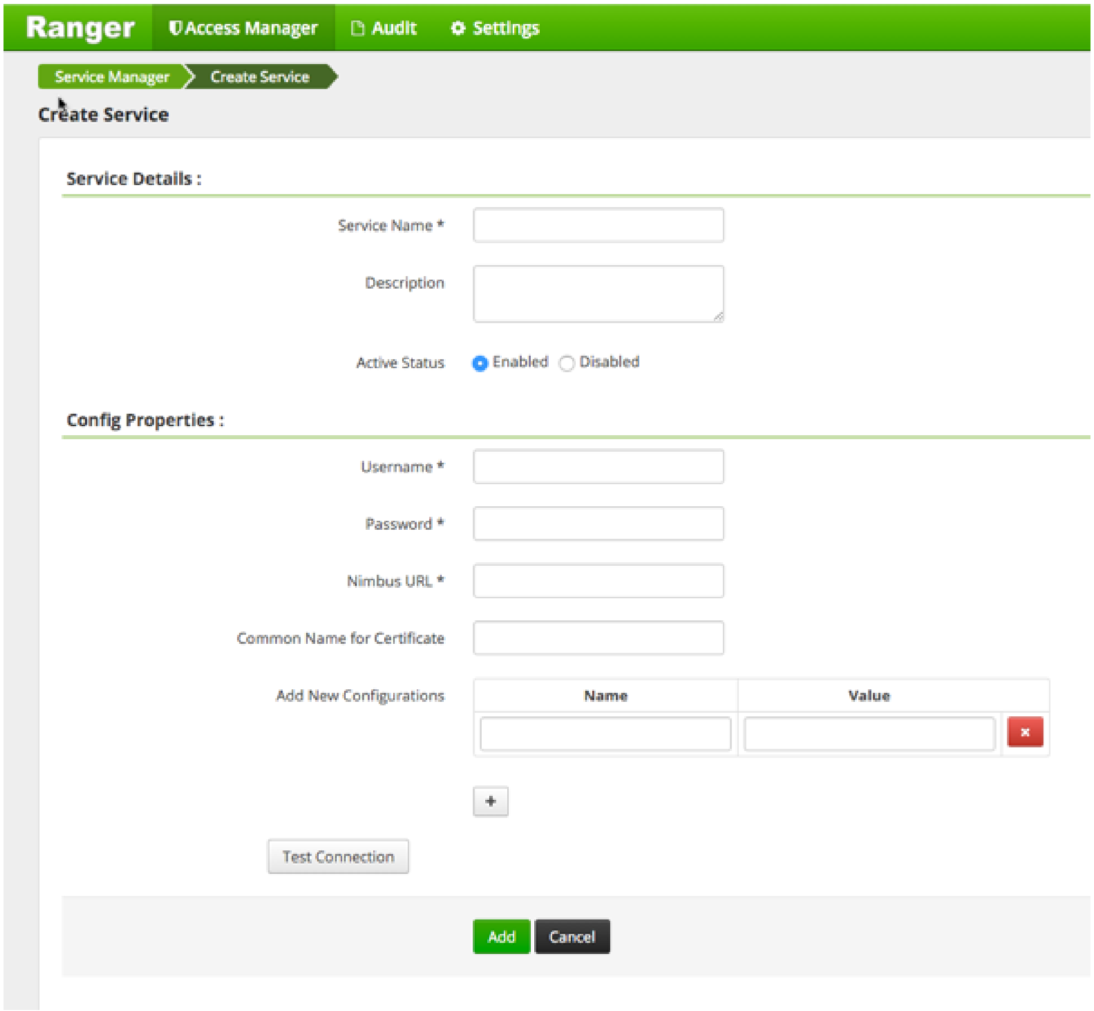
Use the following steps to add a service to Storm:

1. On the Service Manager page, click the

   ➕

   next to Storm.

   The Create Service page appears.



2. Enter the following information on the Create Service page:

Table 4.13. Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Table 4.14. Config Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
Nimbus URL	Hostname of nimbus format, in the form: http://*ipaddress*:*8080*. This field was formerly named nimbus.url.
Common Name For Certificate	The name of the certificate.

Field name	Description
	This field is interchangeably named **Common Name For Certificate** and **Ranger Plugin SSL CName** in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click **Test Connection**.

4. Click **Add**.

4.8. Configure a YARN Service

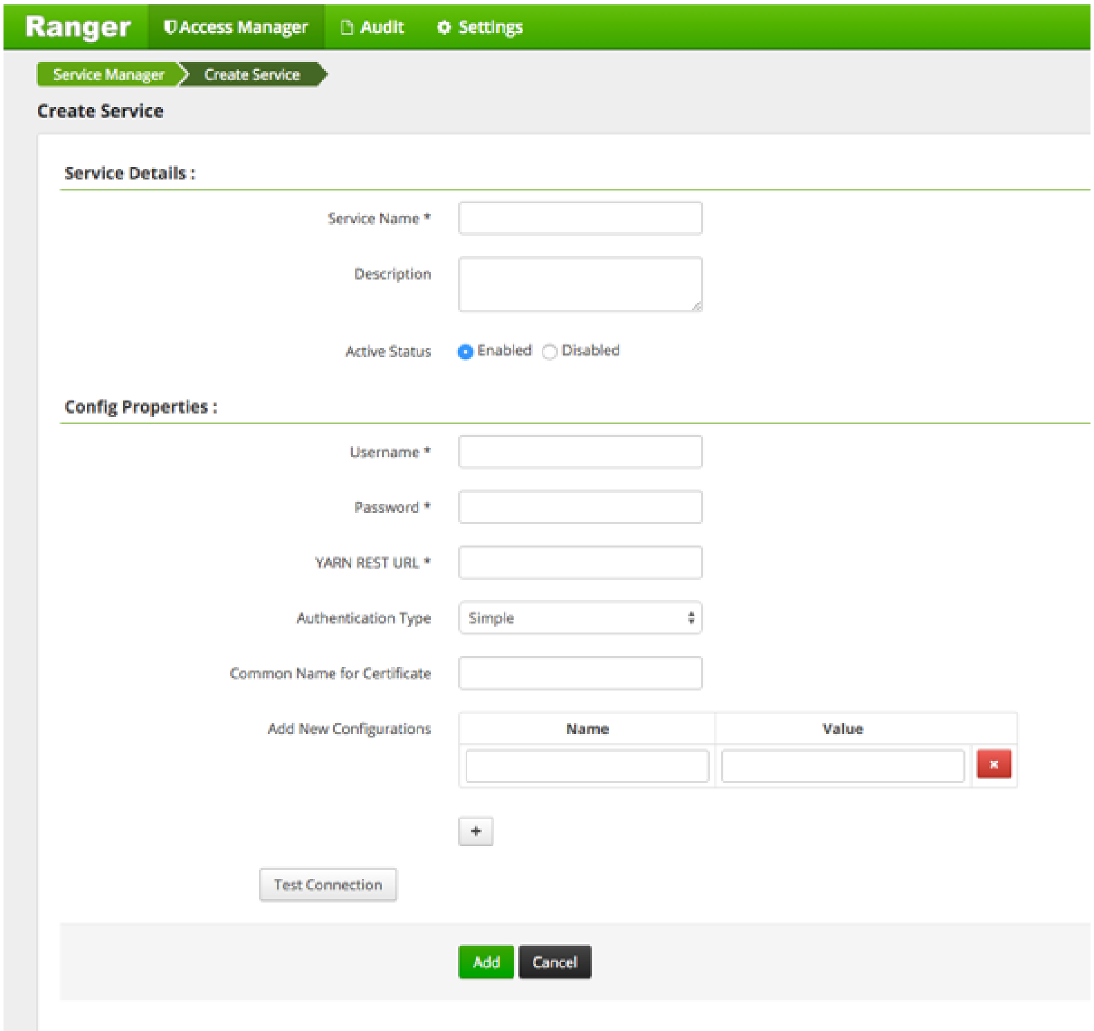
Use the following steps to add a service to YARN:

1. On the Service Manager page, click the

   +

   next to YARN.

   The Create Service page appears.



Create Service

Service Details :

Service Name *

Description

Active Status ● Enabled ○ Disabled

Config Properties :

Username *

Password *

YARN REST URL *

Authentication Type: Simple

Common Name for Certificate

Add New Configurations

Name	Value

2. Enter the following information on the Create Service page:

Table 4.15. Service Details

Field name	Description
Service Name	The name of the service; required when configuring agents.
Description	A description of the service.
Active Status	Enabled or Disabled.

Table 4.16. Config Properties

Field name	Description
Username	The end system username that can be used for connection.
Password	The password for the username entered above.
YARN REST URL	Http or https://*RESOURCEMANAGER_FQDN*:*8088*.
Authentication Type	The type of authorization in use, as noted in the hadoop configuration file core-site.xml; either `simple` or `Kerberos`. (Required only if authorization is enabled).

Field name	Description
	This field was formerly named hadoop.security.authorization.
Common Name For Certificate	The name of the certificate. This field is interchangeably named **Common Name For Certificate** and **Ranger Plugin SSL CName** in Create Service pages.
Add New Configurations	Add any other new configuration(s).

3. Click **Test Connection**.

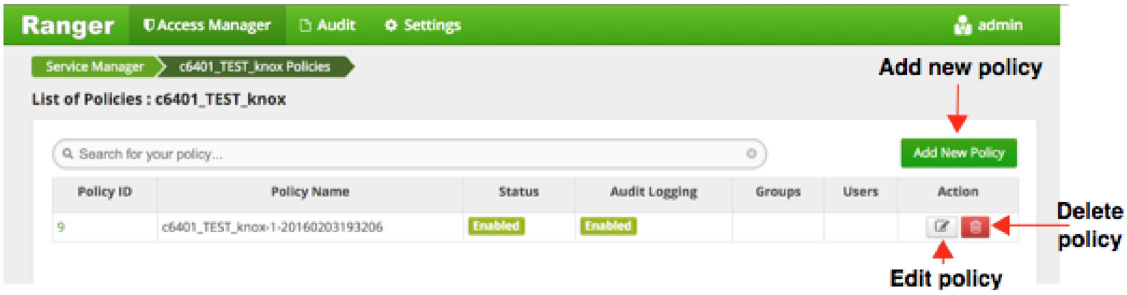
4. Click **Add**.

5. Policy Manager

To examine the policies associated with each service, go to the service where the service resides and click ✎ .

The Ranger Service Manager view appears and an expanded view of that service displays, with the policies listed beneath. The policy view includes a search window.

- **To add a new policy** to the service, click **Add New Policy** (Add New Policy). The form looks slightly different, depending on the type of service to which you are adding the policy.
- **To edit a policy**, click ✎ to the right of the entry for that service. Resource Based Policies displays an expanded view of that policy.
- **To delete a policy**, click 🗑 the Delete icon to the right of the entry for that service.



List of Policies : c6401_TEST_knox

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
9	c6401_TEST_knox-1-20160203193206	Enabled	Enabled			✎ 🗑

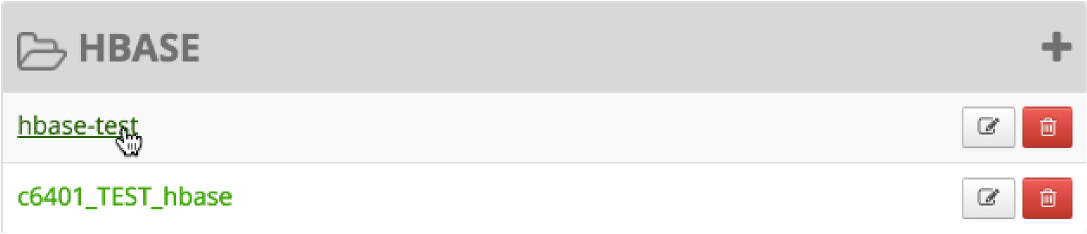
This section describes the requirements for policy creation in

- Create an HBase Policy [30]
- Create an HDFS Policy [33]
- Create a Hive Policy [35]
- Create a Kafka Policy [38]
- Create a Knox Policy [39]
- Create a Solr Policy [41]
- Create a Storm Policy [43]
- Create a YARN Policy [45]

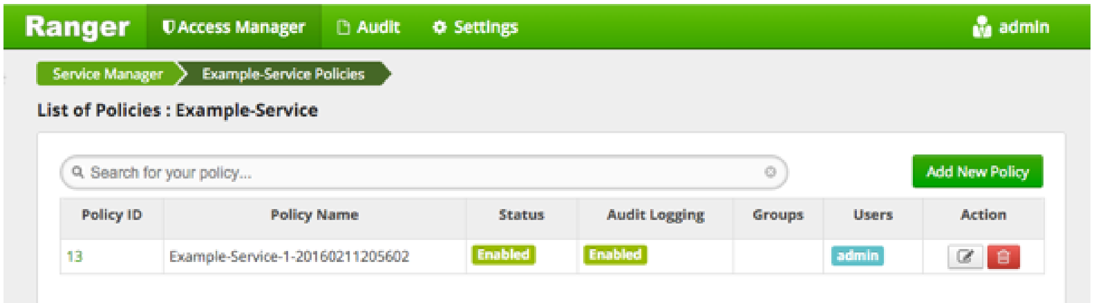
5.1. Create an HBase Policy

To add a new policy to an existing HBase service:

1. On the Service Manager page, select an existing service under HBase.



The List of Policies page appears.

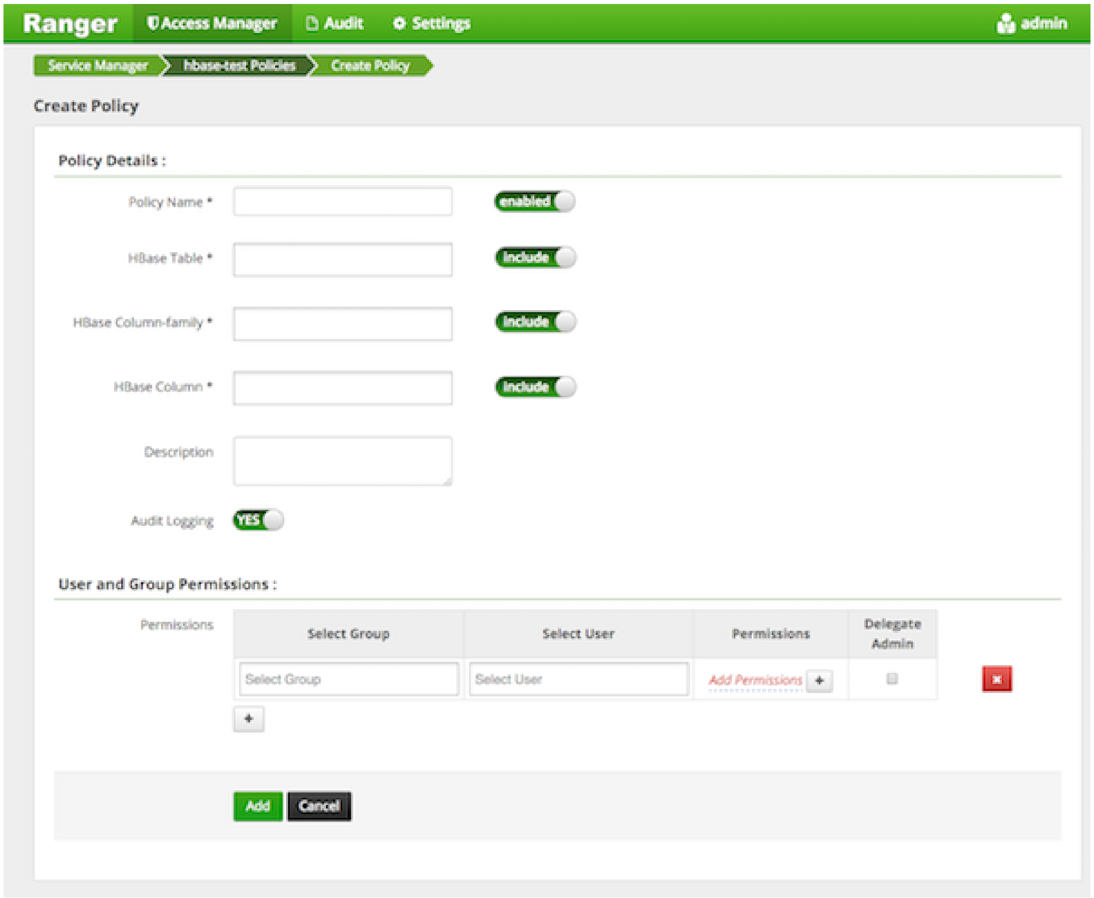


List of Policies : Example-Service

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
13	Example-Service-1-20160211205602	Enabled	Enabled		admin	

2. Click **Add New Policy**.

The Create Policy console appears.



3. Complete the Create Policy page as follows:

Table 5.1. Policy Details

Label	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
HBase Table	Select the appropriate database. Multiple databases can be selected for a particular policy. This field is mandatory.
HBase Column-family	For the selected table, specify the column families to which the policy applies.
HBase Column	For the selected table and column families, specify the columns to which the policy applies.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).

Table 5.2. User and Group Permissions

Label	Description
Select Group	Specify the group to which this policy applies. To designate the group as an Administrator for the chosen resource, specify Admin permissions. (Administrators can create child policies based on existing policies).

Label	Description
Select User	Specify a particular user to which this policy applies (outside of an already-specified group) OR designate a particular user as Admin for this policy. (Administrators can create child policies based on existing policies).
Permissions	Add or edit permissions: Read, Write, Create, Admin, Select/Deselect All.
Delegate Admin	When a policy is assigned to a user or a group of users those users become the delegated admin. The delegated admin can update, delete the policies. It can also create child policies based on the original policy (base policy).

Wild cards can be included in the resource path, in the database name, the table name, or column name:

- ***** indicates zero or more occurrences of characters

- **?** indicates a single character

4. Click **Add**.

5.2. Provide User Access to HBase Database Tables from the Command Line

HBase provides the means to manage user access to HBase database tables directly from the command line. The most commonly-used commands are:

- GRANT

  Syntax: `grant '`<user-or-group>`','`<permissions>`','`<table>

  For example, to create a policy that grants user1 read/write permission on the table usertable, the command is `grant 'user1','RW','usertable'`

  The syntax is the same for granting CREATE and ADMIN rights.

- REVOKE

  Syntax: `revoke '`<user-or-group>','<usertable>'

  For example, to revoke the read/write access of user1 to the table usertable, the command is `revoke 'user1','usertable'`

Note

Unlike Hive, HBase has no specific revoke commands for each user privilege.

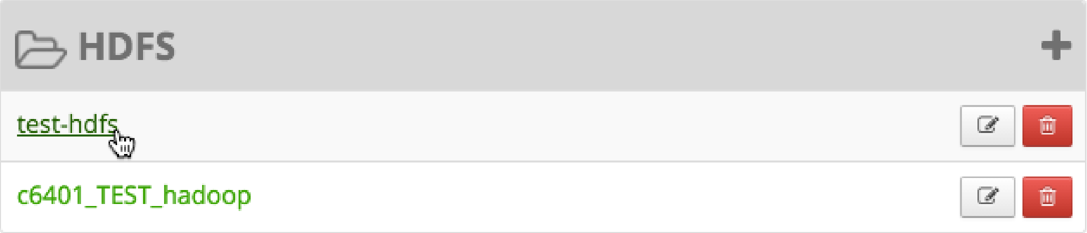
5.3. Create an HDFS Policy

Through configuration, Apache Ranger enables both Ranger policies and HDFS permissions to be checked for a user request. When the NameNode receives a user request, the Ranger plugin checks for policies set through the Ranger Service Manager. If there are no policies, the Ranger plugin checks for permissions set in HDFS.

We recommend that permissions be created at the Ranger Service Manager, and to have restrictive permissions at the HDFS level.

To add a new policy to an existing HDFS service:

1. On the Service Manager page, select an existing service under HDFS.

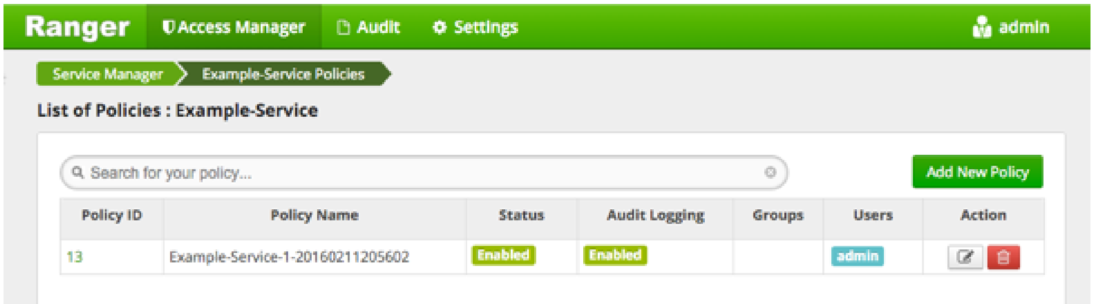


HDFS

test-hdfs

c6401_TEST_hadoop

The List of Policies page appears.

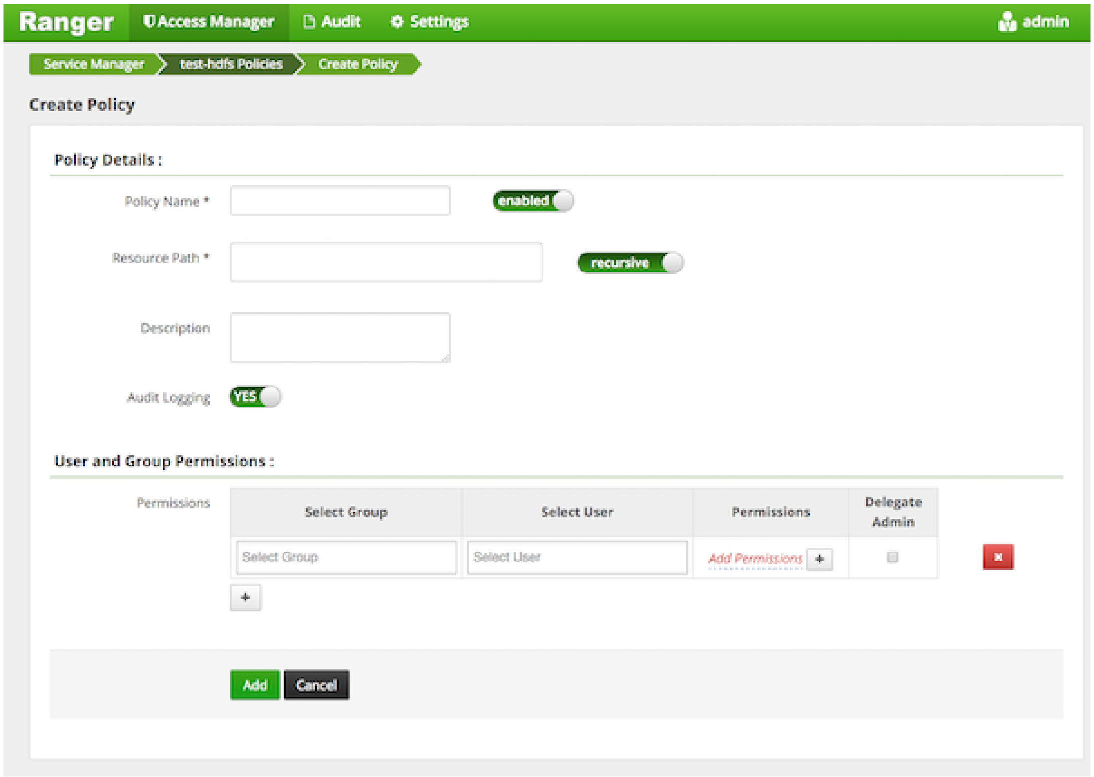


List of Policies : Example-Service

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
13	Example-Service-1-20160211205602	Enabled	Enabled		admin	

2. Click **Add New Policy**.

The Create Policy page appears.



3. Complete the Create Policy page as follows:

Table 5.3. Policy Details

Field	Description
Policy Name	Enter a unique name for this policy. The name cannot be duplicated anywhere in the system.
Resource Path	Define the resource path for the policy folder/file. To avoid the need to supply the full path OR to enable the policy for al subfolders or files, you can either complete this path using wildcards (for example, /home*) or specify that the policy should be recursive. (See below.)
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).

Table 5.4. User and Group Permissions

Label	Description
Select Group	Specify the group to which this policy applies. To designate the group as an Administrator for the chosen resource, specify Admin permissions. (Administrators can create child policies based on existing policies).
Select User	Specify a particular user to which this policy applies (outside of an already-specified group) OR designate a particular user as Admin for this policy. (Administrators can create child policies based on existing policies).
Permissions	Add or edit permissions: Read, Write, Create, Admin, Select/Deselect All.

Label	Description
Delegate Admin	When a policy is assigned to a user or a group of users those users become the delegated admin. The delegated admin can update, delete the policies. It can also create child policies based on the original policy (base policy).

Wild cards can be included in the resource path, in the database name, the table name, or column name:

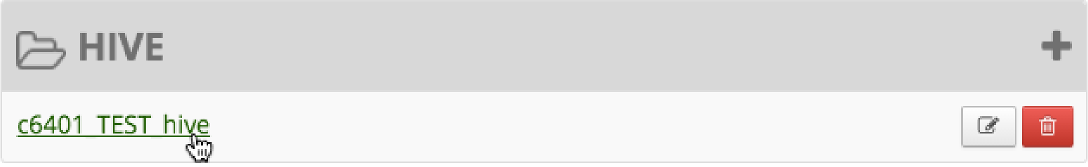
- ***** indicates zero or more occurrences of characters
- **?** indicates a single character

4. Click **Add**.

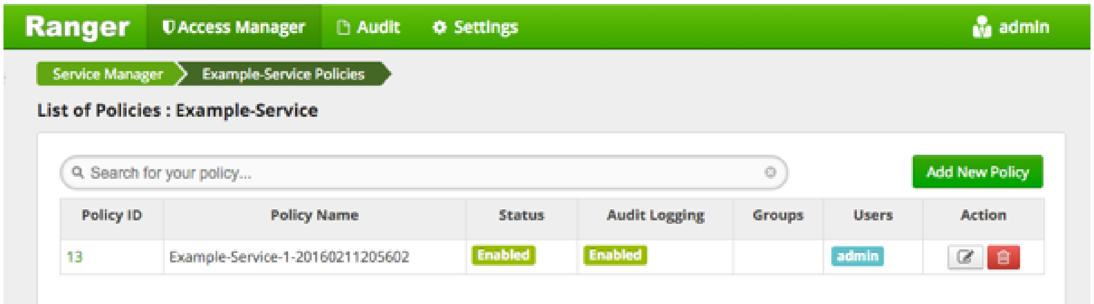
5.4. Create a Hive Policy

To add a new policy to an existing Hive service:

1. On the Service Manager page, select an existing service under Hive.



The List of Policies page appears.

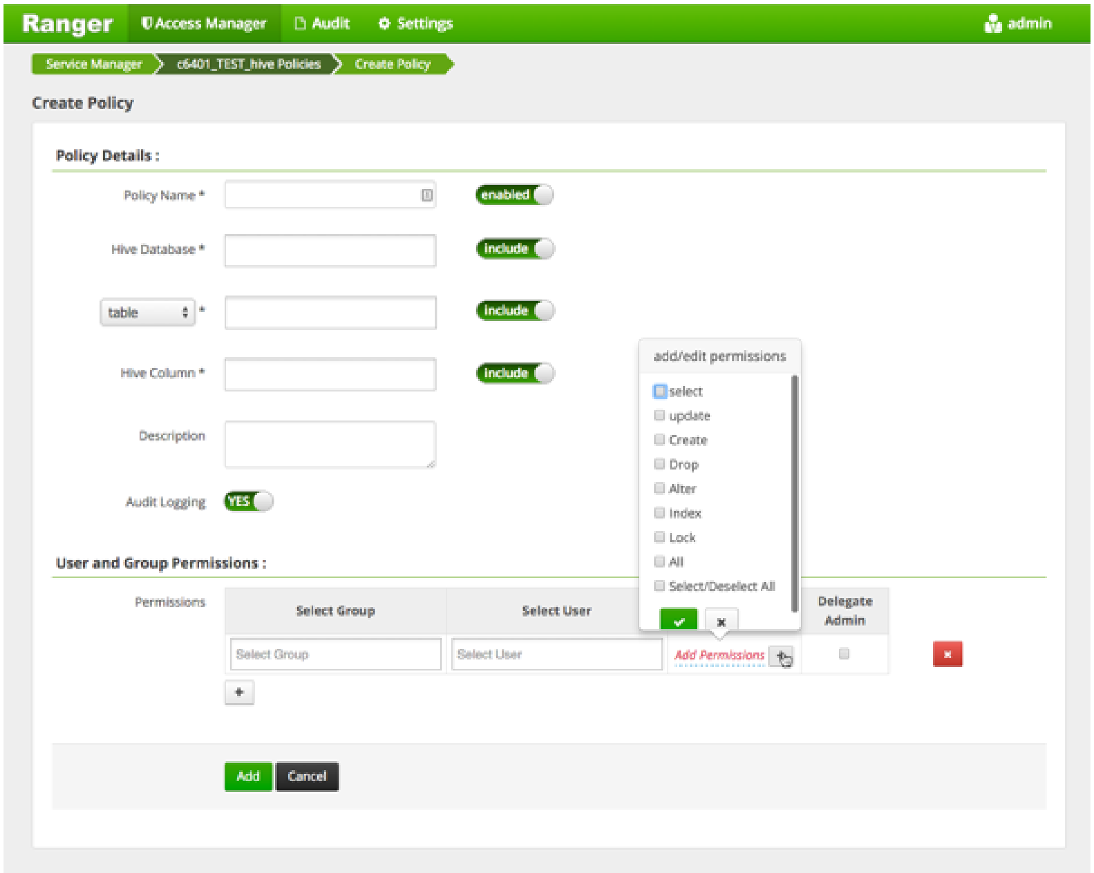


List of Policies : Example-Service

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
13	Example-Service-1-20160211205602	Enabled	Enabled		admin	

2. Click **Add New Policy**.

The Create Policy console appears.



3. Complete the Create Policy page as follows:

Table 5.5. Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
Hive Database	Select the appropriate database. Multiple databases can be selected for a particular policy. This field is mandatory.
Table/UDF Drop-down	To continue adding a table-based policy, keep Table selected. To add a User Defined Function (UDF), select UDF.
Hive Column	For the selected database, select table(s) for which the policy will be applicable.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).

Table 5.6. User and Group Permissions

Label	Description
Select Group	Specify the group to which this policy applies. To designate the group as an Administrator for the chosen resource, specify Admin permissions. (Administrators can create child policies based on existing policies).

Label	Description
Select User	Specify a particular user to which this policy applies (outside of an already-specified group) OR designate a particular user as Admin for this policy. (Administrators can create child policies based on existing policies).
Permissions	Add or edit permissions: Read, Write, Create, Admin, Select/Deselect All.
Delegate Admin	When a policy is assigned to a user or a group of users those users become the delegated admin. The delegated admin can update, delete the policies. It can also create child policies based on the original policy (base policy).

Wild cards can be included in the resource path, in the database name, the table name, or column name:

- ***** indicates zero or more occurrences of characters
- **?** indicates a single character

4. Click **Add**.

5.5. Provide User Access to Hive Database Tables from the Command Line

Hive provides the means to manage user access to Hive database tables directly from the command line. The most commonly-used commands are:

- GRANT

  Syntax: `grant` <permissions> `on table` <table> `to user` <user or group>`;`

  For example, to create a policy that grants user1 SELECT permission on the table default-hivesmoke22074, the command is `grant select on table default.hivesmoke22074 to user user1;`

  The syntax is the same for granting UPDATE, CREATE, DROP, ALTER, INDEX, LOCK, ALL and ADMIN rights.

- REVOKE

  Syntax: `revoke` <permissions> `on table` <table> `from user` <user or group>`;`

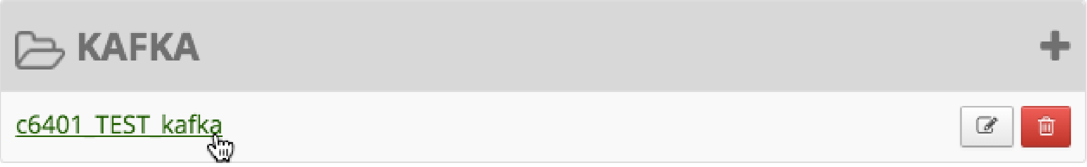
  For example, to revoke the SELECT rights of user1 to the table default.hivesmoke22074, the command is `revoke select on table default.hivesmoke22074 from user user1;`

  The syntax is the same for revoking UPDATE, CREATE, DROP, ALTER, INDEX, LOCK, ALL and ADMIN rights.

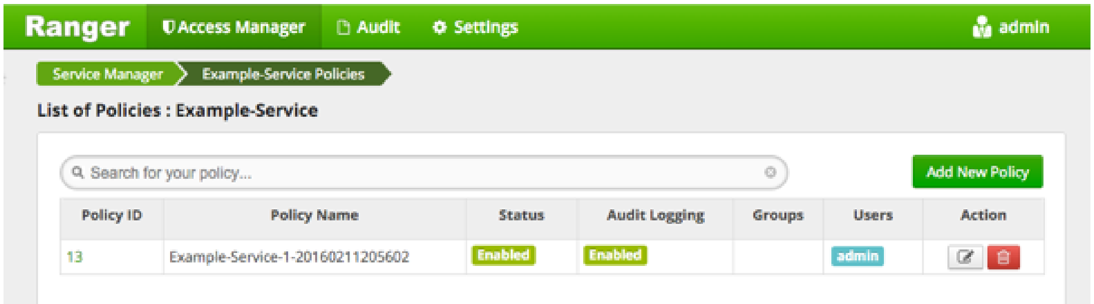
5.6. Create a Kafka Policy

To add a new policy to an existing Kafka service:

1. On the Service Manager page, select an existing service under Kafka.



The List of Policies page appears.

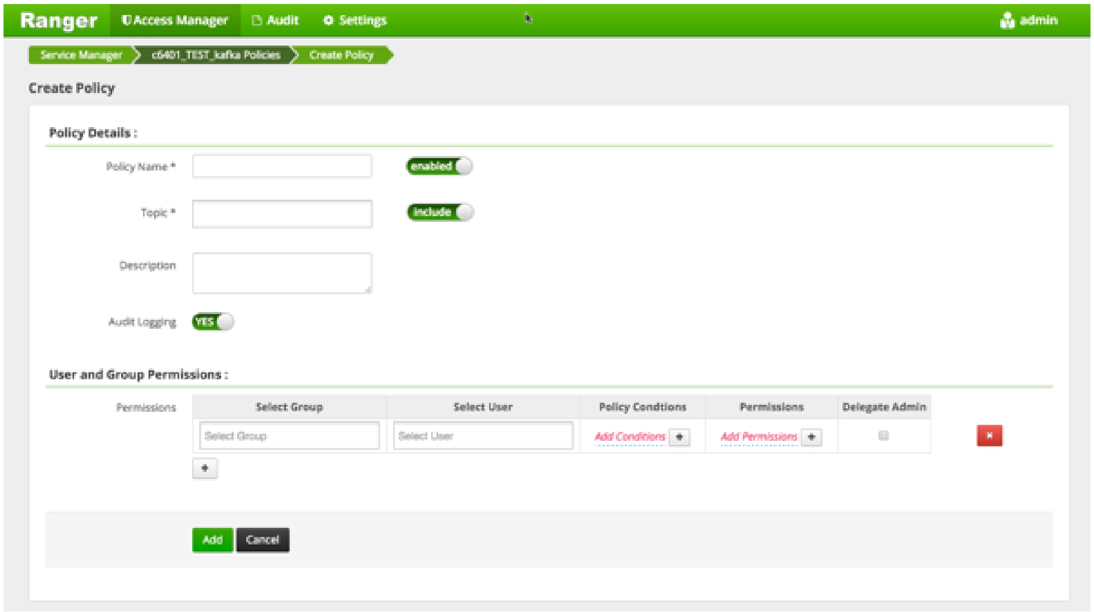


List of Policies : Example-Service

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
13	Example-Service-1-20160211205602	Enabled	Enabled		admin	

2. Click **Add New Policy**.

The Create Policy console appears.



Create Policy

Policy Details :

Policy Name * enabled

Topic * include

Description

Audit Logging YES

User and Group Permissions :

Permissions

Select Group	Select User	Policy Condtions	Permissions	Delegate Admin
Select Group	Select User	Add Conditions +	Add Permissions +	□

3. Complete the Create Policy page as follows:

Table 5.7. Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
Topic	A topic is a category or feed name to which messages are published.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).

Table 5.8. User and Group Permissions

Label	Description
Select Group	Specify the group to which this policy applies. To designate the group as an Administrator for the chosen resource, specify Admin permissions. (Administrators can create child policies based on existing policies).
Select User	Specify a particular user to which this policy applies (outside of an already-specified group) OR designate a particular user as Admin for this policy. (Administrators can create child policies based on existing policies).
Policy Conditions	Specify IP address range,
Permissions	Add or edit permissions: Read, Write, Create, Admin, Select/Deselect All.
Delegate Admin	When a policy is assigned to a user or a group of users those users become the delegated admin. The delegated admin can update, delete the policies. It can also create child policies based on the original policy (base policy).

Wild cards can be included in the resource path, in the database name, the table name, or column name:

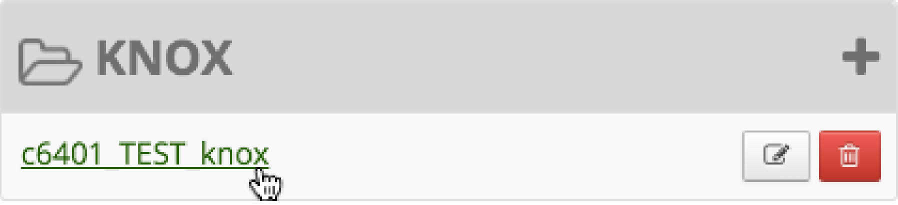
- ***** indicates zero or more occurrences of characters
- **?** indicates a single character

4. Click **Add**.

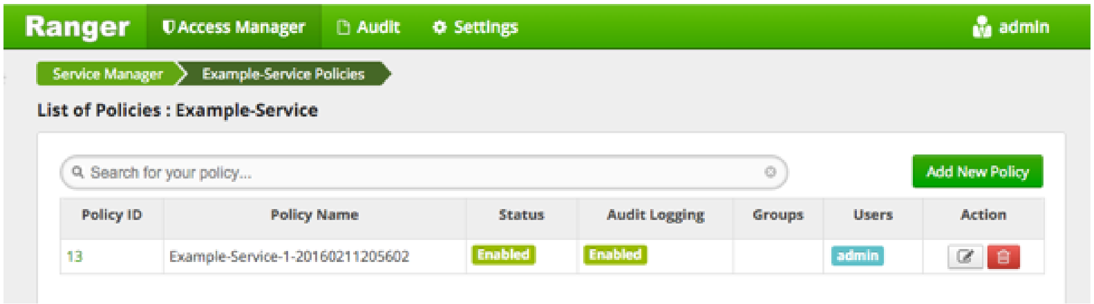
5.7. Create a Knox Policy

To add a new policy to an existing Knox service:

1. On the Service Manager page, select an existing service under Knox.



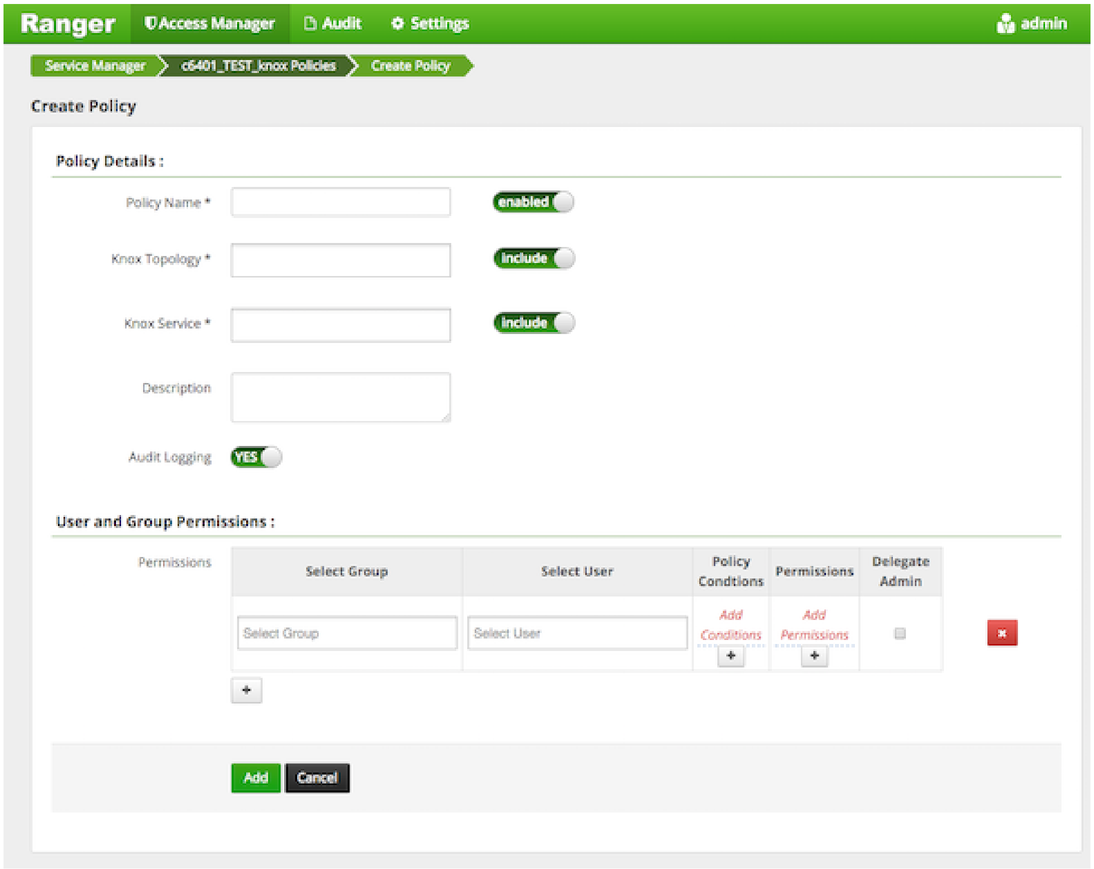
The List of Policies page appears.



Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
13	Example-Service-1-20160211205602	Enabled	Enabled		admin	

2. Click **Add New Policy**.

The Create Policy console appears.



3. Complete the Create Policy page as follows:

Table 5.9. Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
Knox Topology	Enter an appropriate Topology Name.

Field	Description
Knox Service	Enter an appropriate Service Name.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).

Table 5.10. User and Group Permissions

Label	Description
Select Group	Specify the group to which this policy applies. To designate the group as an Administrator for the chosen resource, specify Admin permissions. (Administrators can create child policies based on existing policies).
Select User	Specify a particular user to which this policy applies (outside of an already-specified group) OR designate a particular user as Admin for this policy. (Administrators can create child policies based on existing policies).
Policy Conditions	Specify IP address range,
Permissions	Add or edit permissions: Read, Write, Create, Admin, Select/Deselect All.
Delegate Admin	When a policy is assigned to a user or a group of users those users become the delegated admin. The delegated admin can update, delete the policies. It can also create child policies based on the original policy (base policy).

Since Knox does not provide a command line methodology for assigning privileges or roles to users, the User and Group Permissions portion of the Knox Create Policy form is especially important.

Wild cards can be included in the resource path, in the database name, the table name, or column name:

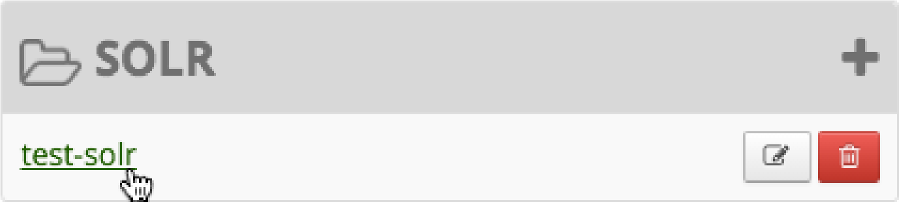
- ***** indicates zero or more occurrences of characters
- **?** indicates a single character

4. Click **Add**.

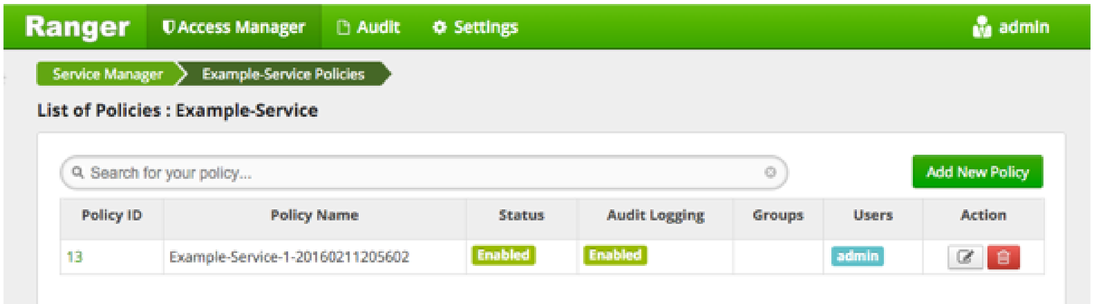
5.8. Create a Solr Policy

To add a new policy to an existing Solr service:

1. On the Service Manager page, select an existing service under Solr.



The List of Policies page appears.



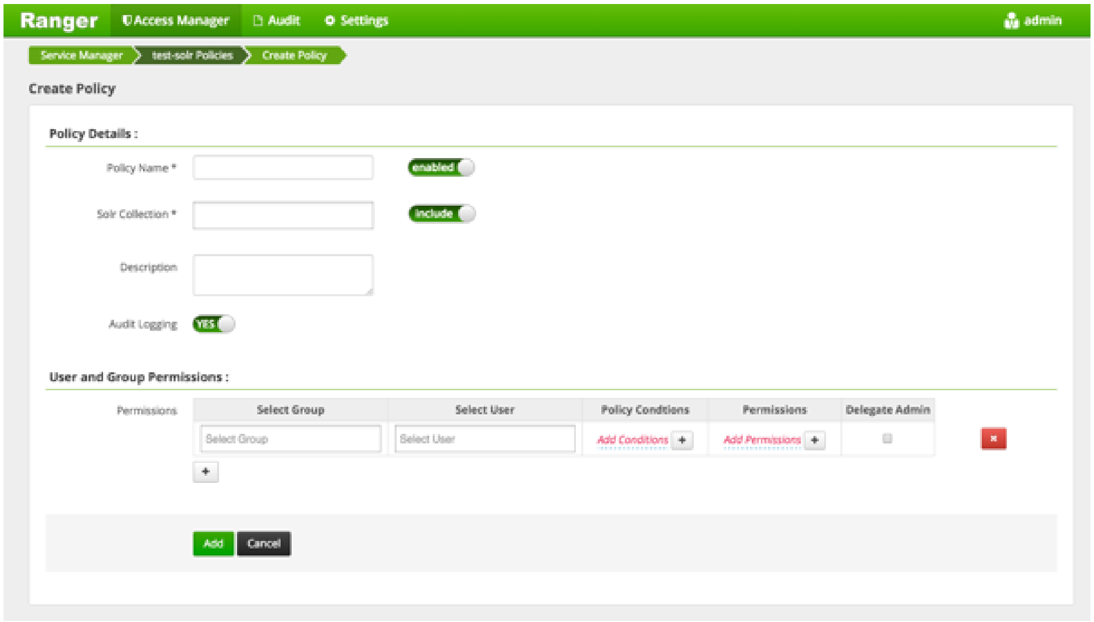
List of Policies : Example-Service

Search for your policy...

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
13	Example-Service-1-20160211205602	Enabled	Enabled		admin	

2. Click **Add New Policy**.

The Create Policy console appears.



Create Policy

Policy Details :

Policy Name *

Solr Collection *

Description

Audit Logging

User and Group Permissions :

Permissions

Select Group	Select User	Policy Condtions	Permissions	Delegate Admin
Select Group	Select User	Add Conditions +	Add Permissions +	□

3. Complete the Create Policy page as follows:

Table 5.11. Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
Solr Collection	http:*host_ip*:*6083*/solr
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).

Table 5.12. User and Group Permissions

Label	Description
Select Group	Specify the group to which this policy applies. To designate the group as an Administrator for the chosen

Label	Description
	resource, specify Admin permissions. (Administrators can create child policies based on existing policies).
Select User	Specify a particular user to which this policy applies (outside of an already-specified group) OR designate a particular user as Admin for this policy. (Administrators can create child policies based on existing policies).
Policy Conditions	Specify IP address range,
Permissions	Add or edit permissions: Read, Write, Create, Admin, Select/Deselect All.
Delegate Admin	When a policy is assigned to a user or a group of users those users become the delegated admin. The delegated admin can update, delete the policies. It can also create child policies based on the original policy (base policy).

Wild cards can be included in the resource path, in the database name, the table name, or column name:

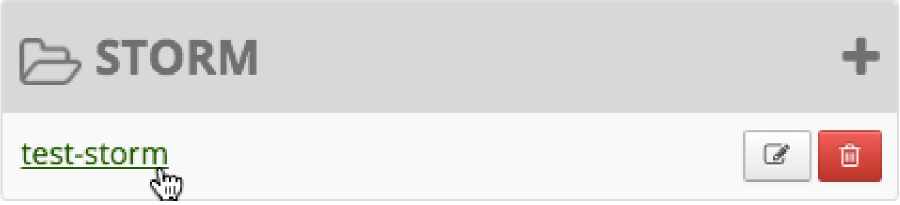
- ***** indicates zero or more occurrences of characters
- **?** indicates a single character

4. Click **Add**.

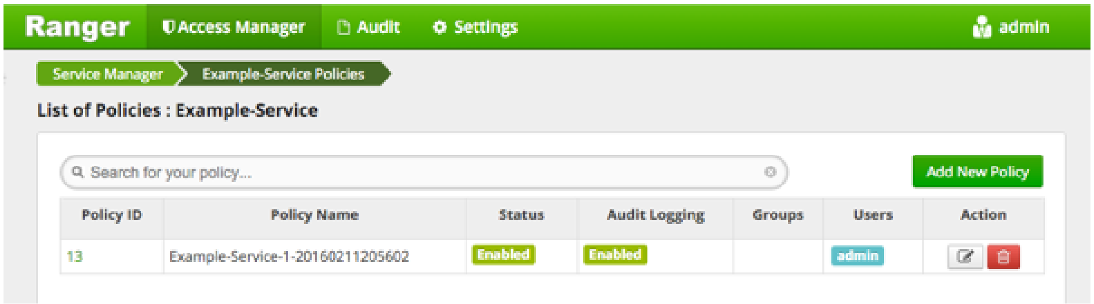
5.9. Create a Storm Policy

To add a new policy to an existing Storm service:

1. On the Service Manager page, select an existing service under Storm.



The List of Policies page appears.

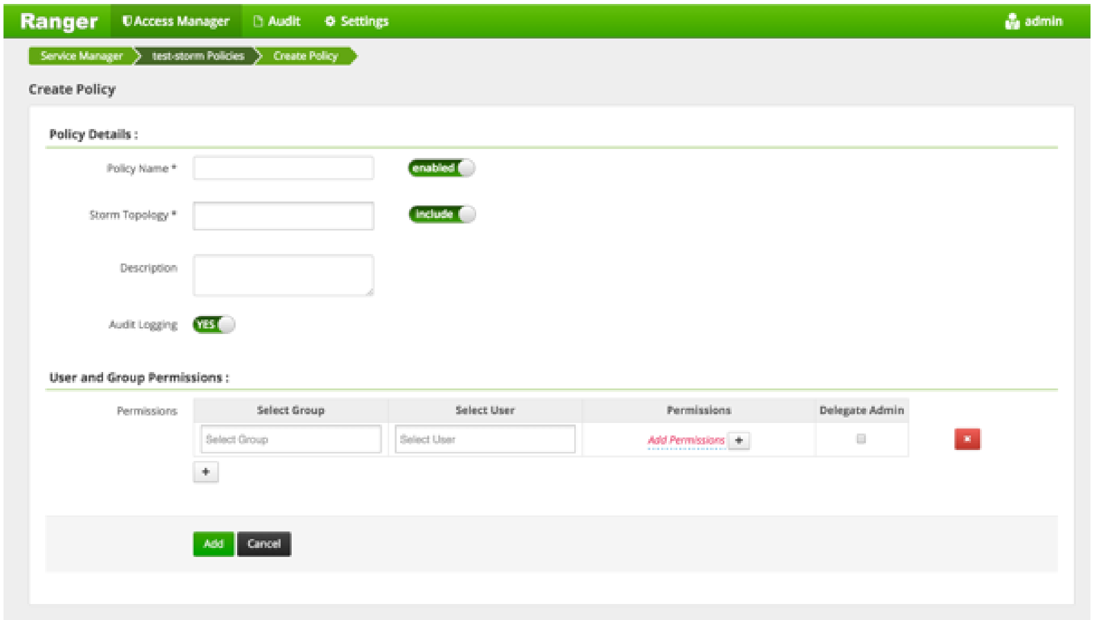


List of Policies : Example-Service

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
13	Example-Service-1-20160211205602	Enabled	Enabled		admin	

2. Click **Add New Policy**.

The Create Policy console appears.



3. Complete the Create Policy page as follows:

Table 5.13. Policy Details

Label	Description
Policy Name	Enter an appropriate policy name. This name is cannot be duplicated across the system. This field is mandatory.
Storm Topology	Enter an appropriate Topology Name.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).

Table 5.14. User and Group Permissions

Label	Description
Select Group	Specify the group to which this policy applies. To designate the group as an Administrator for the chosen resource, specify Admin permissions. (Administrators can create child policies based on existing policies).
Select User	Specify a particular user to which this policy applies (outside of an already-specified group) OR designate a particular user as Admin for this policy. (Administrators can create child policies based on existing policies).
Permissions	Add or edit permissions: Read, Write, Create, Admin, Select/Deselect All.
Delegate Admin	When a policy is assigned to a user or a group of users those users become the delegated admin. The delegated admin can update, delete the policies. It can also create child policies based on the original policy (base policy).

Since Storm does not provide a command line methodology for assigning privileges or roles to users, the User and Group Permissions portion of the Storm Create Policy form is especially important.

Table 5.15. Knox User and Group Permissions

Actions	Description
File upload	Allows a user to upload files.
Get Nimbus Conf	Allows a user to access Nimbus configurations.
Get Cluster Info	Allows a user to get cluster information.
File Download	Allows a user to download files.
Kill Topology	Allows a user to kill the topology.
Rebalance	Allows a user to rebalance topologies.
Activate	Allows a user to activate a topology.
Deactivate	Allows a user to deactivate a topology.
Get Topology Conf	Allows a user to access a topology configuration.
Get Topology	Allows a user to access a topology.
Get User Topology	Allows a user to access a user topology.
Get Topology Info	Allows a user to access topology information.
Upload New Credential	Allows a user to upload a new credential.
Admin	Provides a user with delegated admin access.

Wild cards can be included in the resource path, in the database name, the table name, or column name:

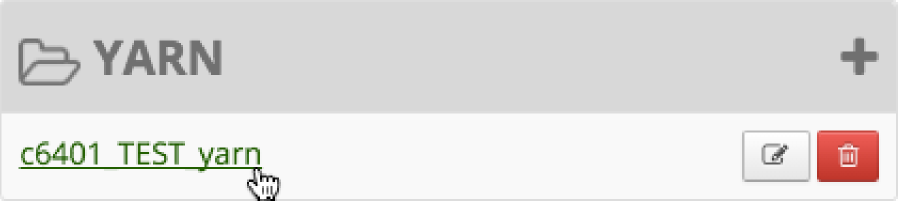
- ***** indicates zero or more occurrences of characters
- **?** indicates a single character

4. Click **Add**.

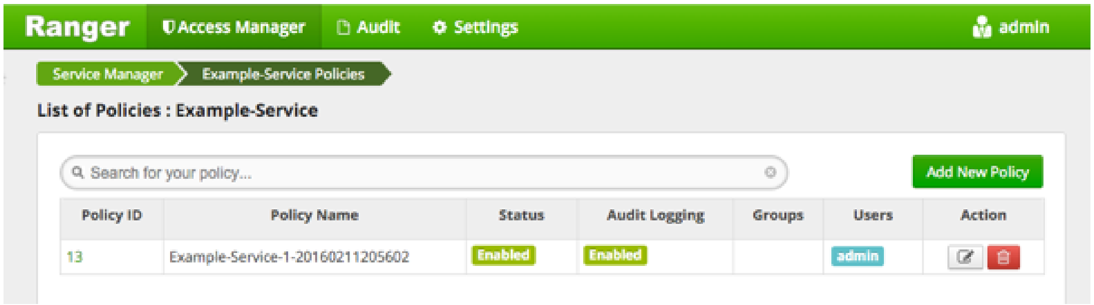
5.10. Create a YARN Policy

To add a new policy to an existing YARN service:

1. On the Service Manager page, select an existing service under YARN.



The List of Policies page appears.

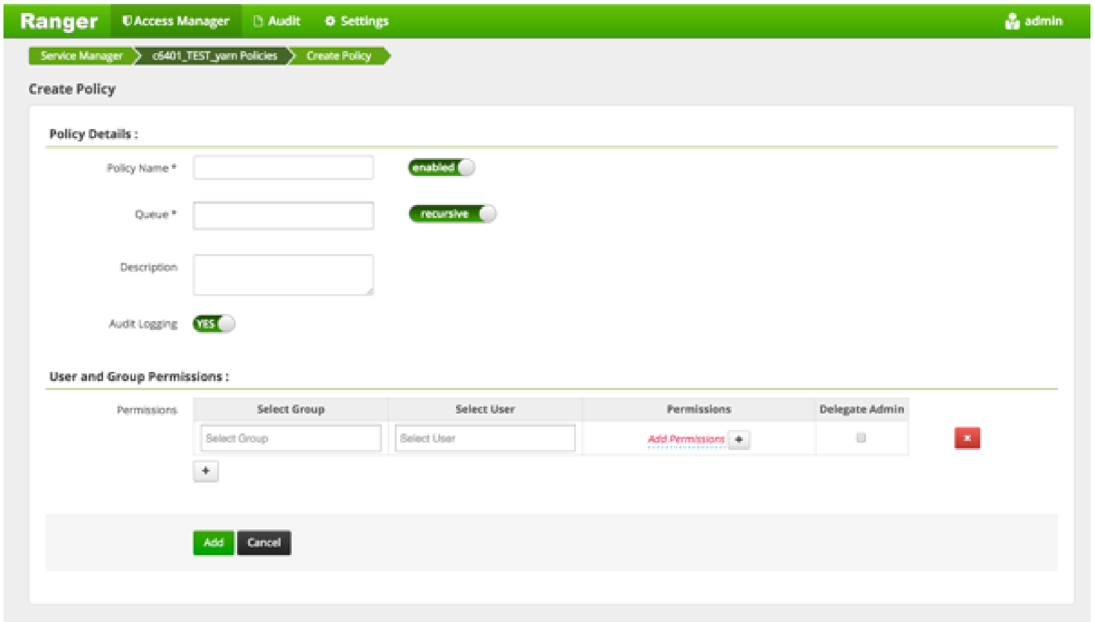


List of Policies : Example-Service

Policy ID	Policy Name	Status	Audit Logging	Groups	Users	Action
13	Example-Service-1-20160211205602	Enabled	Enabled		admin	

2. Click **Add New Policy**.

The Create Policy console appears.



Create Policy

Policy Details :

Policy Name *

Queue *

Description

Audit Logging

User and Group Permissions :

Permissions

Select Group	Select User	Permissions	Delegate Admin
Select Group	Select User	Add Permissions +	□

3. Complete the Create Policy page as follows:

Table 5.16. Policy Details

Field	Description
Policy Name	Enter an appropriate policy name. This name cannot be duplicated across the system. This field is mandatory.
Queue	The fundamental unit of scheduling in yarn.
Recursive	You can indicate whether all files or folders within the existing folder comes under the policy. Can be used instead of wildcard characters.
Description	(Optional) Describe the purpose of the policy.
Audit Logging	Specify whether this policy is audited. (De-select to disable auditing).

Table 5.17. User and Group Permissions

Label	Description
Select Group	Specify the group to which this policy applies. To designate the group as an Administrator for the chosen resource, specify Admin permissions. (Administrators can create child policies based on existing policies).
Select User	Specify a particular user to which this policy applies (outside of an already-specified group) OR designate a particular user as Admin for this policy. (Administrators can create child policies based on existing policies).
Permissions	Add or edit permissions: Read, Write, Create, Admin, Select/Deselect All.
Delegate Admin	When a policy is assigned to a user or a group of users those users become the delegated admin. The delegated admin can update, delete the policies. It can also create child policies based on the original policy (base policy).

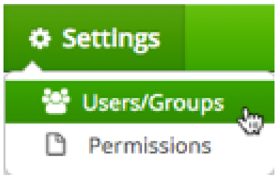
Wild cards can be included in the resource path, in the database name, the table name, or column name:

- ***** indicates zero or more occurrences of characters
- **?** indicates a single character

4. Click **Add**.

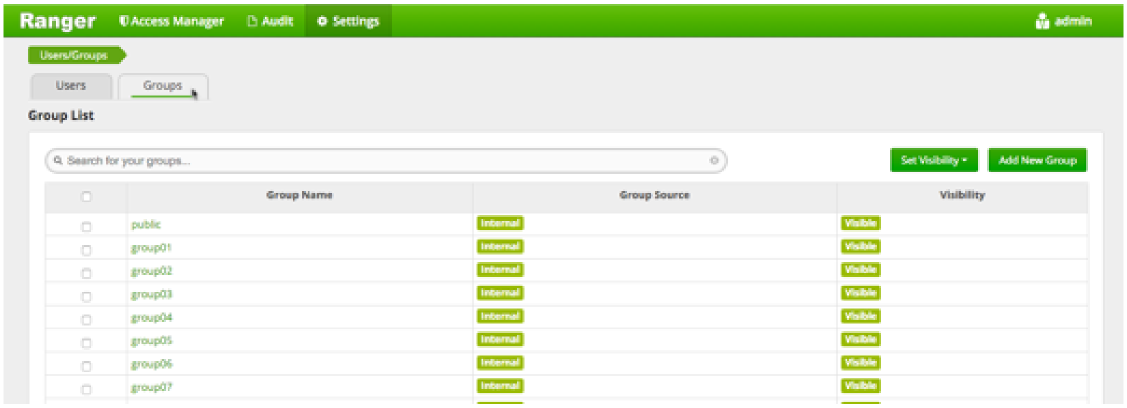
6. Users/Groups and Permissions Administration

To examine the list of users and groups who can access the Ranger portal or its services, click **Settings>Users/Groups** at the top of the Ranger Console.



The User/sGroup view displays:

- Internal users who can log in to the Ranger portal; created by the Ranger console Service Manager
- External users who can access services controlled by the Ranger portal; created at other systems like Active Directory, LDAP or UNIX, and synched with those systems
- Admins who are the only users with permission to create users and create services, run reports, and perform other administrative tasks. Admins can also create child policies based on the original policy (base policy).

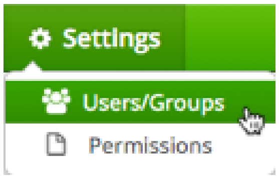


□	Group Name	Group Source	Visibility
□	public	Internal	Visible
□	group01	Internal	Visible
□	group02	Internal	Visible
□	group03	Internal	Visible
□	group04	Internal	Visible
□	group05	Internal	Visible
□	group06	Internal	Visible
□	group07	Internal	Visible

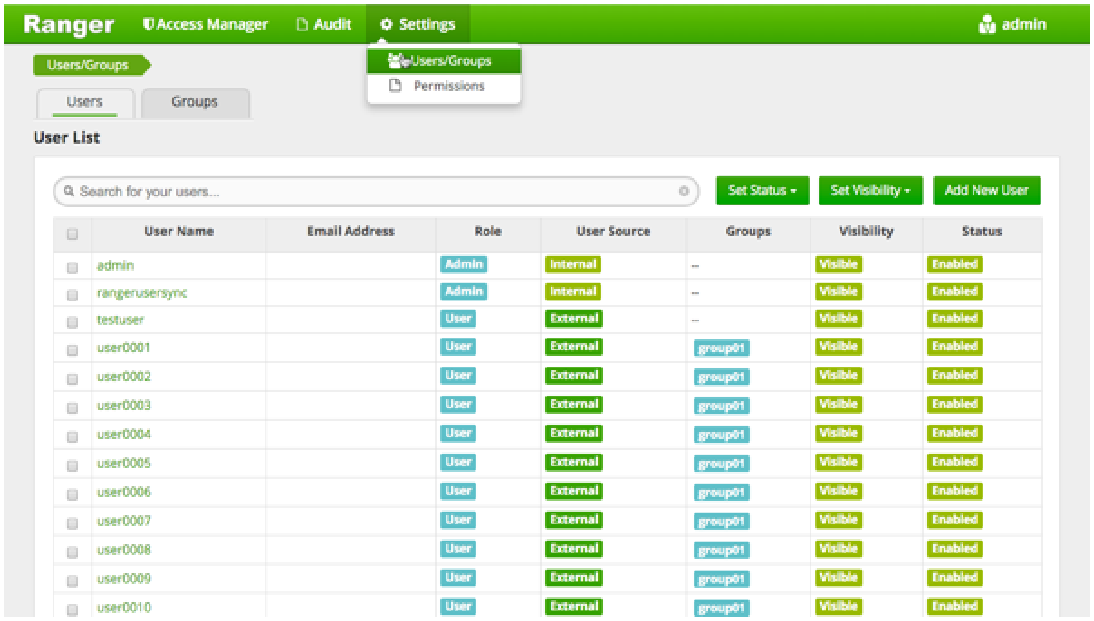
6.1. Add a User

To add a new user to the user list:

1. Click **Settings>Users/Groups**.



The Users/Groups page appears.



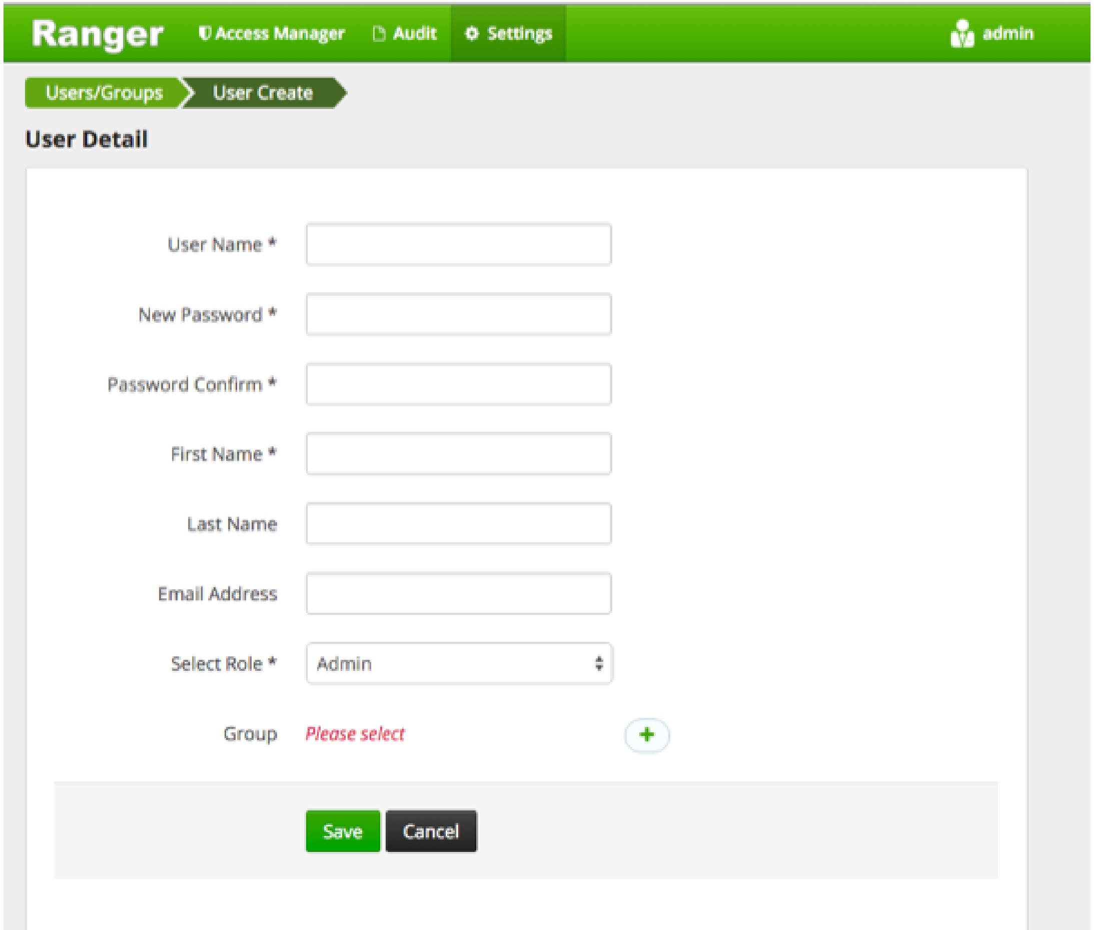
User List

□	User Name	Email Address	Role	User Source	Groups	Visibility	Status
□	admin		Admin	Internal	--	Visible	Enabled
□	rangerusersync		Admin	Internal	--	Visible	Enabled
□	testuser		User	External	--	Visible	Enabled
□	user0001		User	External	group01	Visible	Enabled
□	user0002		User	External	group01	Visible	Enabled
□	user0003		User	External	group01	Visible	Enabled
□	user0004		User	External	group01	Visible	Enabled
□	user0005		User	External	group01	Visible	Enabled
□	user0006		User	External	group01	Visible	Enabled
□	user0007		User	External	group01	Visible	Enabled
□	user0008		User	External	group01	Visible	Enabled
□	user0009		User	External	group01	Visible	Enabled
□	user0010		User	External	group01	Visible	Enabled

2. Click **Add New User** .

Add New User

The Ranger User Detail page appears.



User Detail

User Name *

New Password *

Password Confirm *

First Name *

Last Name

Email Address

Select Role * Admin

Group *Please select*

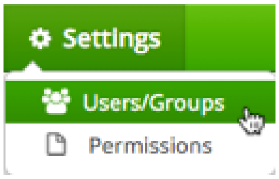
3. Add the appropriate user details, then click **Save**.

The user is immediately added to the list.

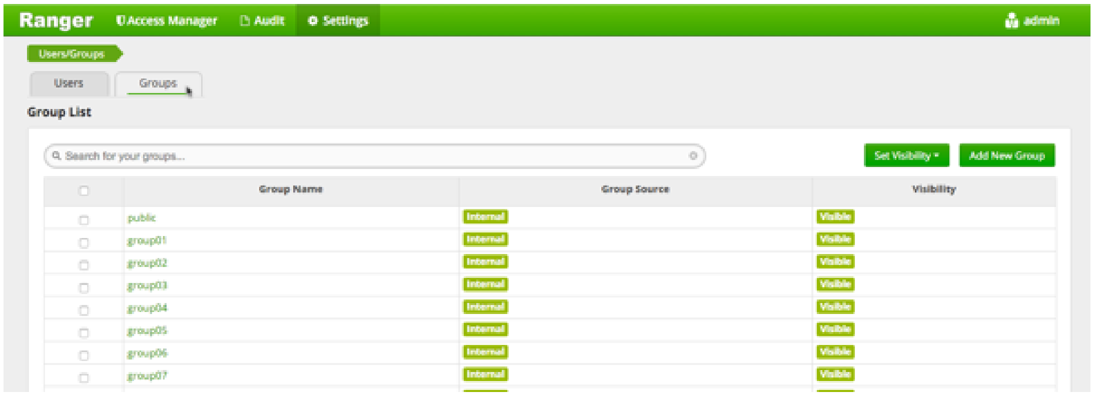
6.2. Edit a User

To edit a user:

1. Click **Settings>Users/Groups**.

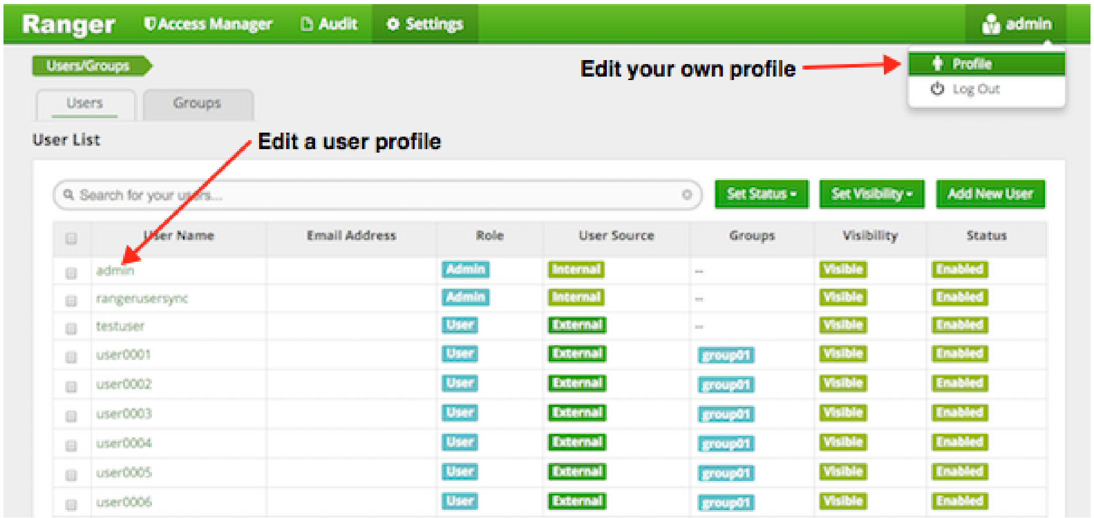


The Users/Groups page opens to the Users tab.



□	Group Name	Group Source	Visibility
□	public	Internal	Visible
□	group01	Internal	Visible
□	group02	Internal	Visible
□	group03	Internal	Visible
□	group04	Internal	Visible
□	group05	Internal	Visible
□	group06	Internal	Visible
□	group07	Internal	Visible

2. Choose to edit yourself or another user:



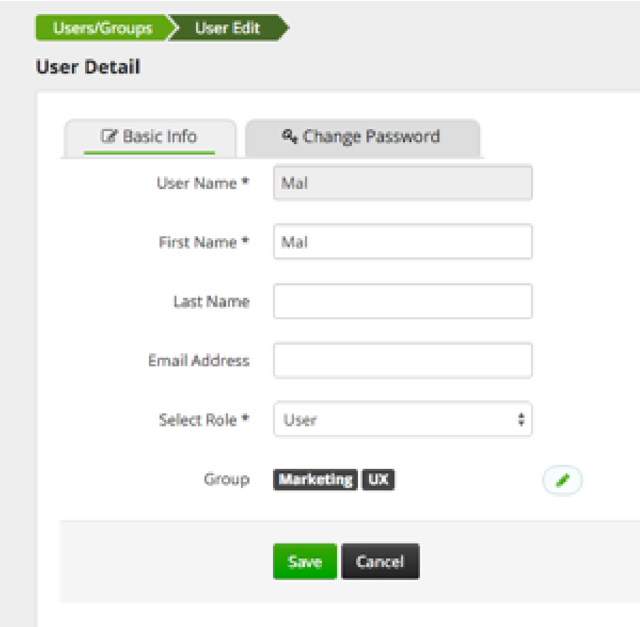
□	User Name	Email Address	Role	User Source	Groups	Visibility	Status
□	admin		Admin	Internal	--	Visible	Enabled
□	rangerusersync		Admin	Internal	--	Visible	Enabled
□	testuser		User	External	--	Visible	Enabled
□	user0001		User	External	group01	Visible	Enabled
□	user0002		User	External	group01	Visible	Enabled
□	user0003		User	External	group01	Visible	Enabled
□	user0004		User	External	group01	Visible	Enabled
□	user0005		User	External	group01	Visible	Enabled
□	user0006		User	External	group01	Visible	Enabled

- To edit another user: in the User Name column, select the user you wish to edit.

  The User Detail page appears.

Note

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



User Name * Mal

First Name * Mal

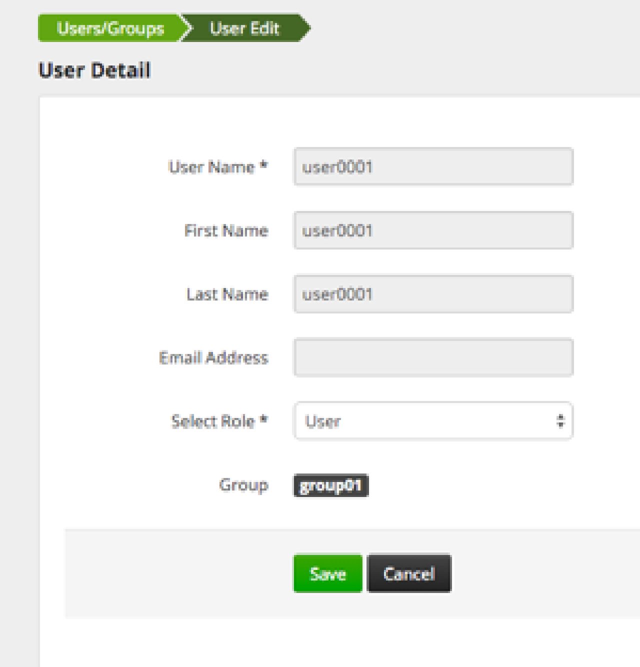
Last Name

Email Address

Select Role * User

Group Marketing UX

You can change everything except the User Name when editing an internal user.



User Name * user0001

First Name user0001

Last Name user0001

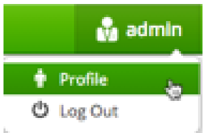
Email Address

Select Role * User

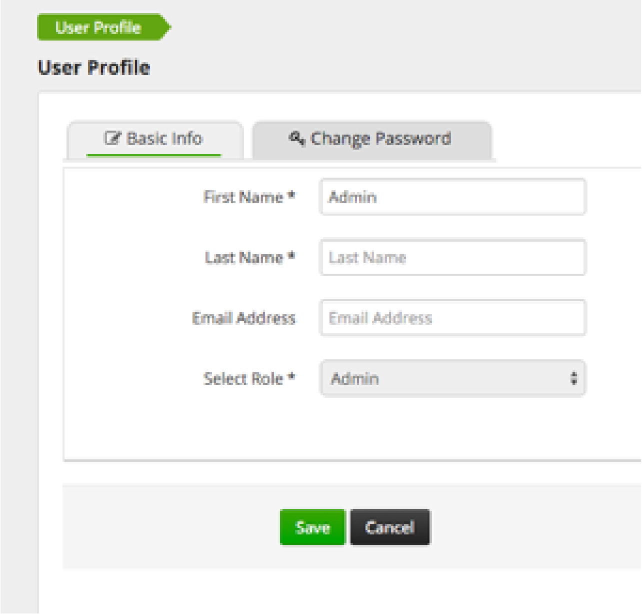
Group group01

You can only change the user role when editing an external user.

- To edit your user profile, click ***Username***>**Profile**.



The User Profile page appears.



User Profile

User Profile

Basic Info | Change Password

First Name * Admin

Last Name * Last Name

Email Address Email Address

Select Role * Admin

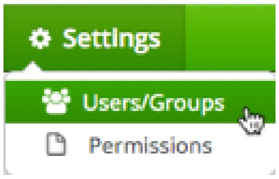
Save | Cancel

3. Edit the appropriate details and click **Save**.

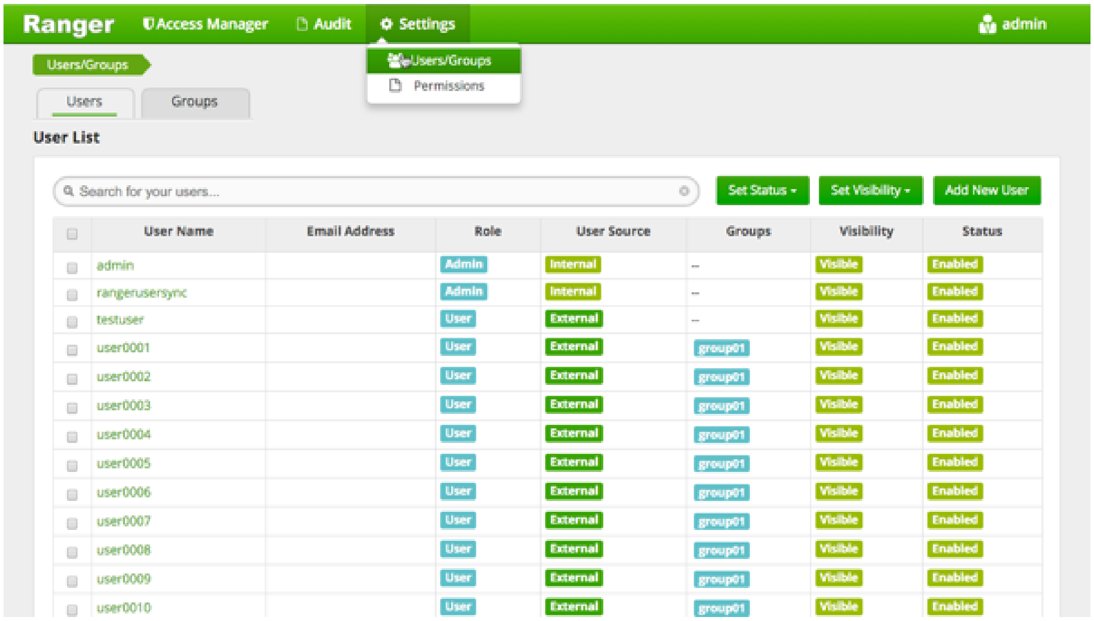
6.3. Add a Group

To add a group:

1. Click **Settings>Users/Groups**.



The Users/Groups page opens to the Users tab.

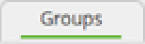


User List

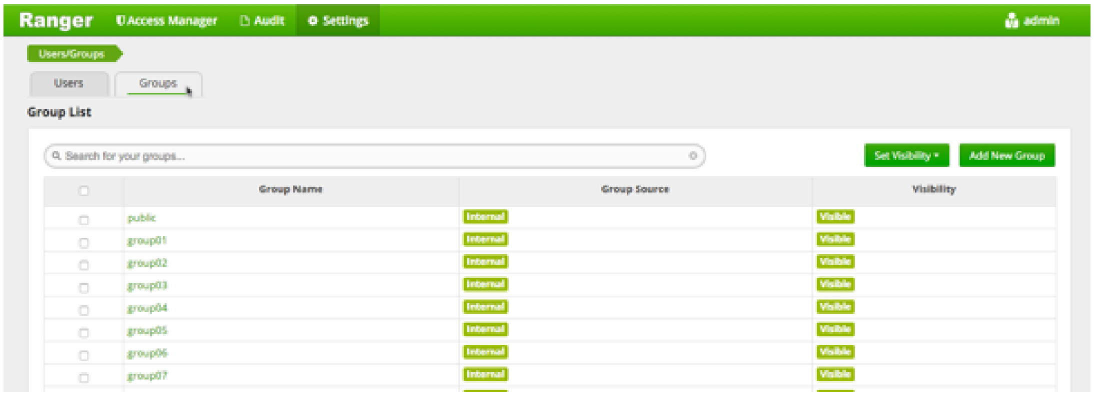
Search for your users...

□	User Name	Email Address	Role	User Source	Groups	Visibility	Status
□	admin		Admin	Internal	--	Visible	Enabled
□	rangerusersync		Admin	Internal	--	Visible	Enabled
□	testuser		User	External	--	Visible	Enabled
□	user0001		User	External	group01	Visible	Enabled
□	user0002		User	External	group01	Visible	Enabled
□	user0003		User	External	group01	Visible	Enabled
□	user0004		User	External	group01	Visible	Enabled
□	user0005		User	External	group01	Visible	Enabled
□	user0006		User	External	group01	Visible	Enabled
□	user0007		User	External	group01	Visible	Enabled
□	user0008		User	External	group01	Visible	Enabled
□	user0009		User	External	group01	Visible	Enabled
□	user0010		User	External	group01	Visible	Enabled

2. Click the **Groups** sub-tab.



The Groups page appears.



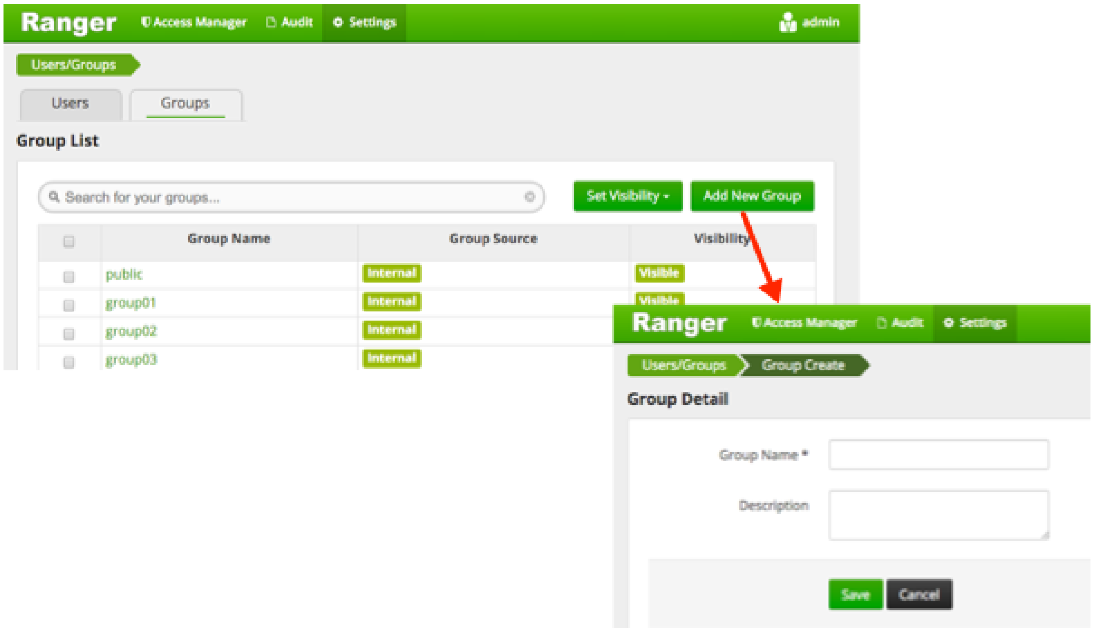
Group List

Search for your groups...

□	Group Name	Group Source	Visibility
□	public	Internal	Visible
□	group01	Internal	Visible
□	group02	Internal	Visible
□	group03	Internal	Visible
□	group04	Internal	Visible
□	group05	Internal	Visible
□	group06	Internal	Visible
□	group07	Internal	Visible

3. Click **Add New Group**

The Group Create page appears.



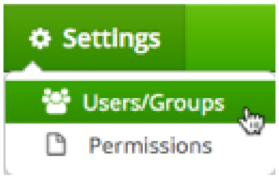
□	Group Name	Group Source	Visibility
□	public	Internal	Visible
□	group01	Internal	Visible
□	group02	Internal	
□	group03	Internal	

4. Enter a unique name for the group, and an optional description, then click **Save**.

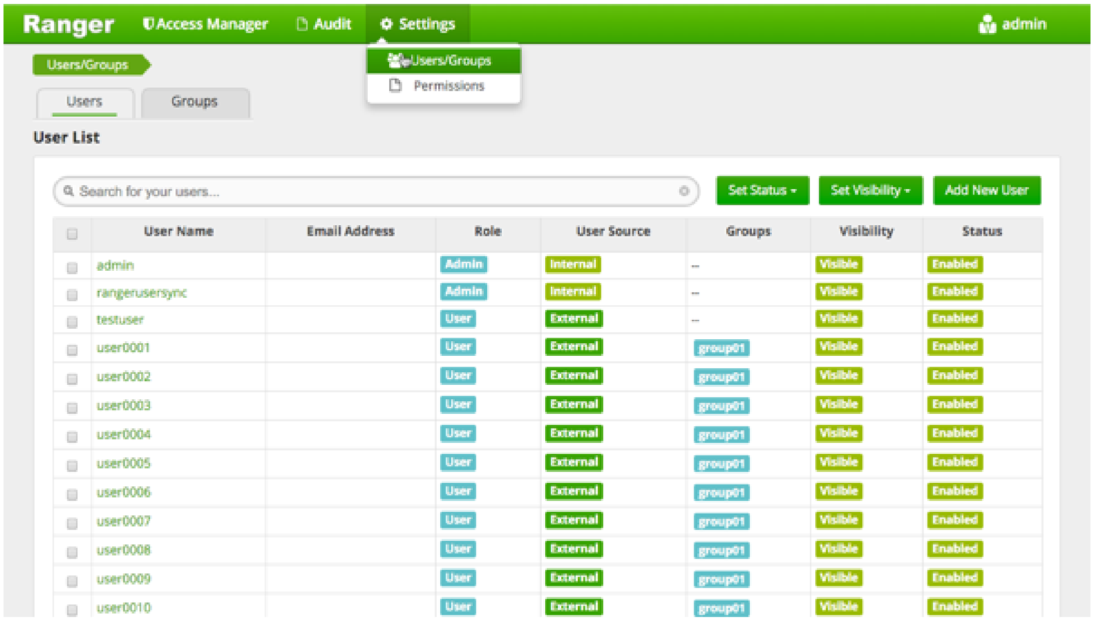
6.4. Edit a Group

To edit a group:

1. Click **Settings>Users/Groups**.



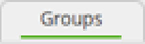
The Users/Groups page opens to the Users tab.



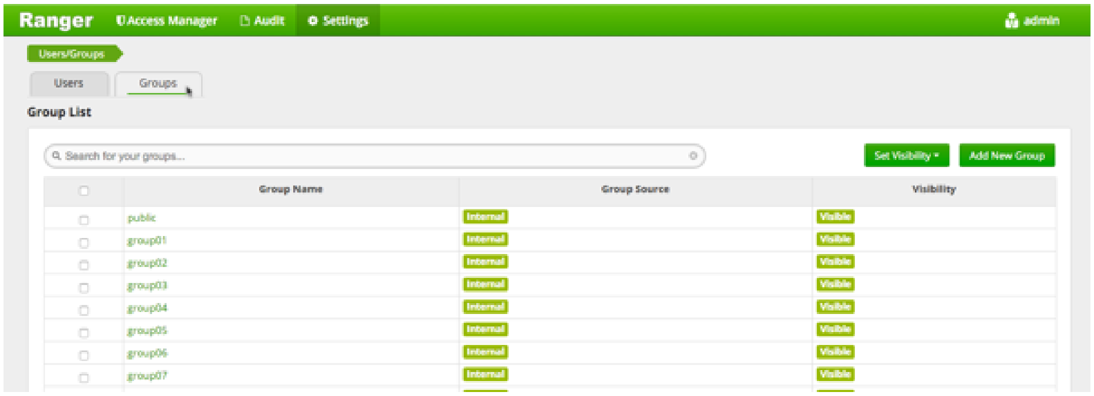
User List

□	User Name	Email Address	Role	User Source	Groups	Visibility	Status
□	admin		Admin	Internal	--	Visible	Enabled
□	rangerusersync		Admin	Internal	--	Visible	Enabled
□	testuser		User	External	--	Visible	Enabled
□	user0001		User	External	group01	Visible	Enabled
□	user0002		User	External	group01	Visible	Enabled
□	user0003		User	External	group01	Visible	Enabled
□	user0004		User	External	group01	Visible	Enabled
□	user0005		User	External	group01	Visible	Enabled
□	user0006		User	External	group01	Visible	Enabled
□	user0007		User	External	group01	Visible	Enabled
□	user0008		User	External	group01	Visible	Enabled
□	user0009		User	External	group01	Visible	Enabled
□	user0010		User	External	group01	Visible	Enabled

2. Click the **Groups** sub-tab.



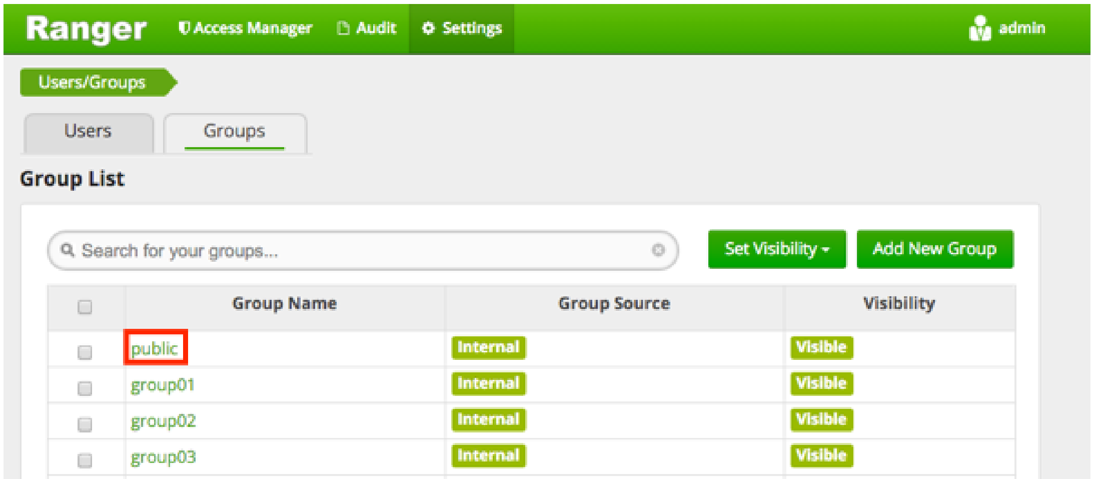
The Groups page appears.



Group List

□	Group Name	Group Source	Visibility
□	public	Internal	Visible
□	group01	Internal	Visible
□	group02	Internal	Visible
□	group03	Internal	Visible
□	group04	Internal	Visible
□	group05	Internal	Visible
□	group06	Internal	Visible
□	group07	Internal	Visible

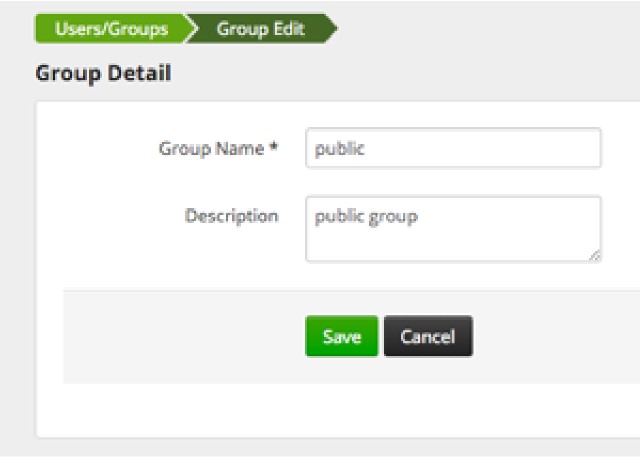
3. Select the group name you wish to edit.



Group List

□	Group Name	Group Source	Visibility
□	public	Internal	Visible
□	group01	Internal	Visible
□	group02	Internal	Visible
□	group03	Internal	Visible

4. The Group Edit page appears.



Group Detail

Group Name * public

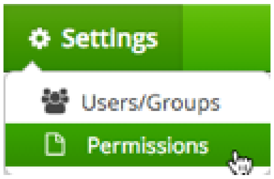
Description public group

5. Edit the appropriate details, then click **Save**.

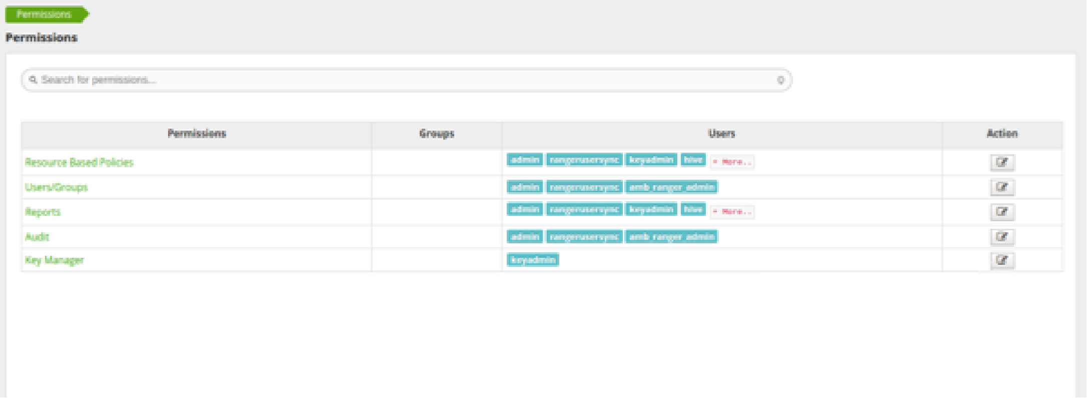
6.5. Add or Edit Permissions

To add or edit user or group:

1. Click **Settings>Permissions**.



The Users/Groups page opens to the Permissions page.



Permissions

Permissions

Search for permissions...

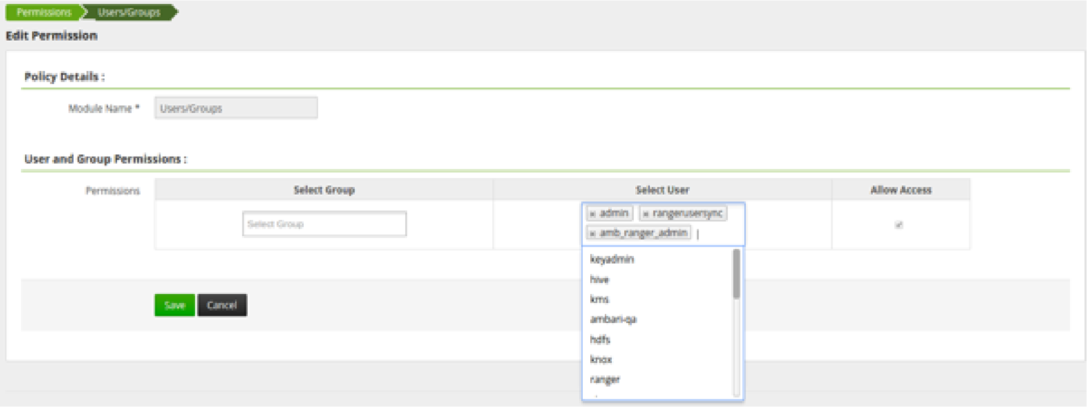
Permissions	Groups	Users	Action
Resource Based Policies		admin rangerusersync keyadmin hive + More..	✎
Users/Groups		admin rangerusersync amb_ranger_admin	✎
Reports		admin rangerusersync keyadmin hive + More..	✎
Audit		admin rangerusersync amb_ranger_admin	✎
Key Manager		keyadmin	✎

2. Click

   ✎

   beside the permission you wish to edit.

   The Edit Permission page appears.



Permissions > Users/Groups

Edit Permission

Policy Details :

Module Name * Users/Groups

User and Group Permissions :

Permissions	Select Group	Select User	Allow Access
	Select Group	× admin × rangerusersync × amb_ranger_admin	☑

3. Edit the necessary fields and click **Save**.

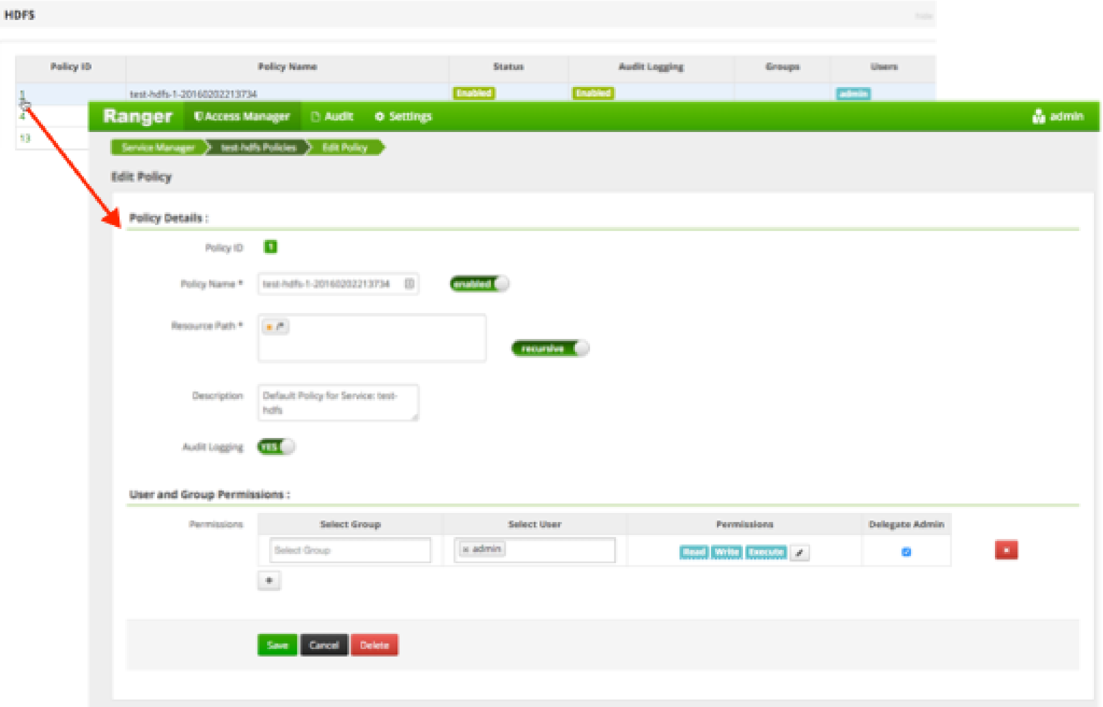
   Save

   You can select multiple users and groups from the dropdown menus.

7. Reports Administration

The Reports module is used to manage the policies more efficiently as the number of policies grow. The page lists all HDFS, HBase, Hive, YARN, Knox, Storm, Solr, and Kafka policies.

You can edit policies from the Reports module by selecting the Policy ID.



HDFS

Policy ID	Policy Name	Status	Audit Logging	Groups	Users
1	test-hdfs-1-20160202213734	Enabled	Enabled		admin
4					
13					

Edit Policy

Policy Details :

Policy ID: 1

Policy Name *: test-hdfs-1-20160202213734 — enabled

Resource Path *: / — recursive

Description: Default Policy for Service: test-hdfs

Audit Logging: YES

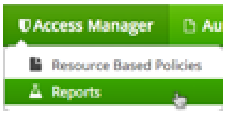
User and Group Permissions :

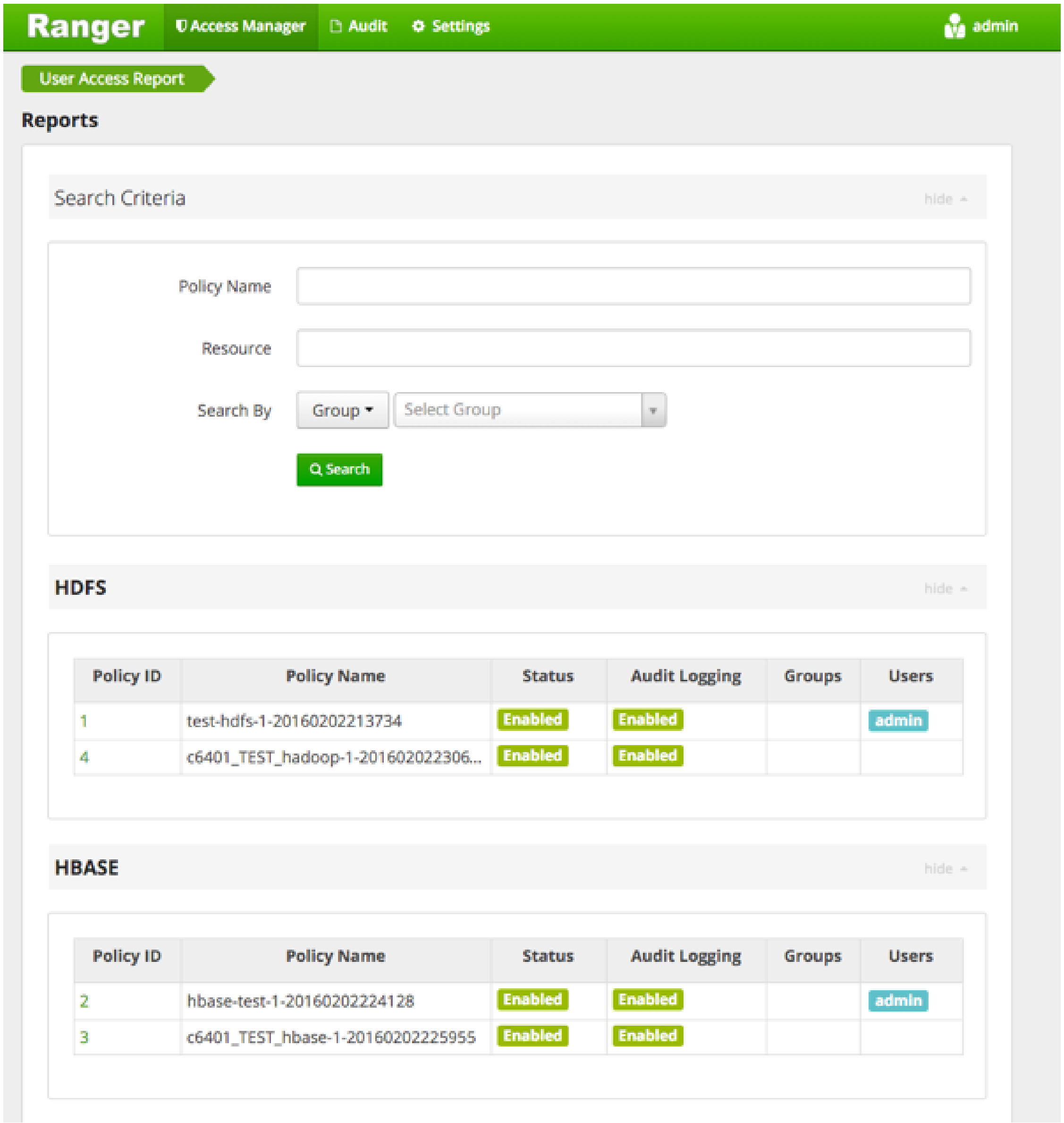
Permissions

Select Group	Select User	Permissions	Delegate Admin
Select Group	admin	Read Write Execute	☑

7.1. View Reports

To view reports on one or more policies, click **Access Manager>Reports**.





Reports

Search Criteria

Policy Name

Resource

Search By Group ▾ Select Group

HDFS

Policy ID	Policy Name	Status	Audit Logging	Groups	Users
1	test-hdfs-1-20160202213734	Enabled	Enabled		admin
4	c6401_TEST_hadoop-1-201602022306...	Enabled	Enabled		

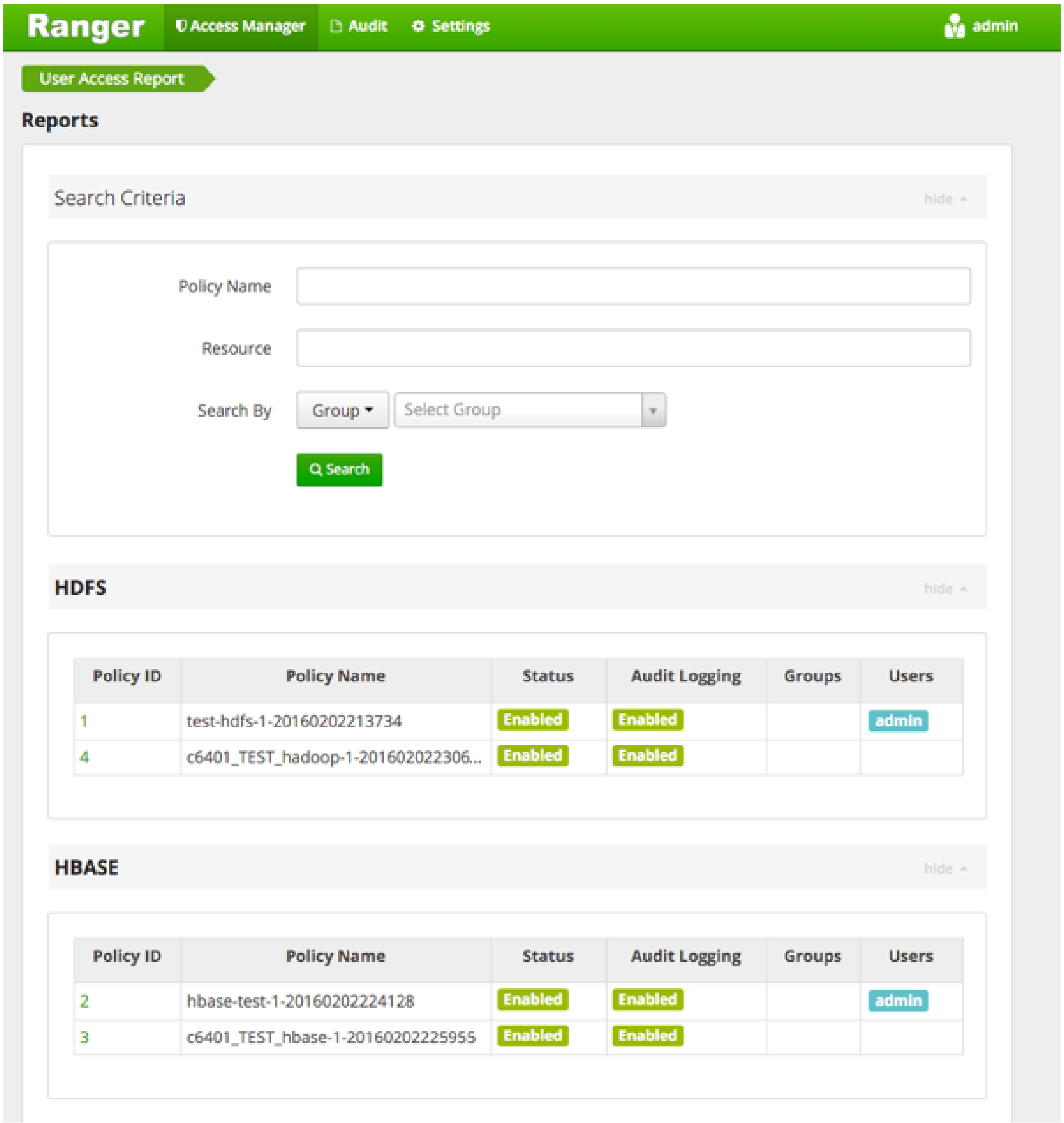
HBASE

Policy ID	Policy Name	Status	Audit Logging	Groups	Users
2	hbase-test-1-20160202224128	Enabled	Enabled		admin
3	c6401_TEST_hbase-1-20160202225955	Enabled	Enabled		

7.2. Search Reports

You can search:

- **Policy Name** - The policy name assigned to the policy while creating it.
- **Resource** - The resource path used while creating the policy.
- **Group, Username** - The group and the users to which the policy is assigned.



HDFS

Policy ID	Policy Name	Status	Audit Logging	Groups	Users
1	test-hdfs-1-20160202213734	Enabled	Enabled		admin
4	c6401_TEST_hadoop-1-201602022306...	Enabled	Enabled		

HBASE

Policy ID	Policy Name	Status	Audit Logging	Groups	Users
2	hbase-test-1-20160202224128	Enabled	Enabled		admin
3	c6401_TEST_hbase-1-20160202225955	Enabled	Enabled		

8. Auditing

To explore options for auditing policies in Ranger, click **Audit** in the top menu.

There are four tabs on the Audit page:

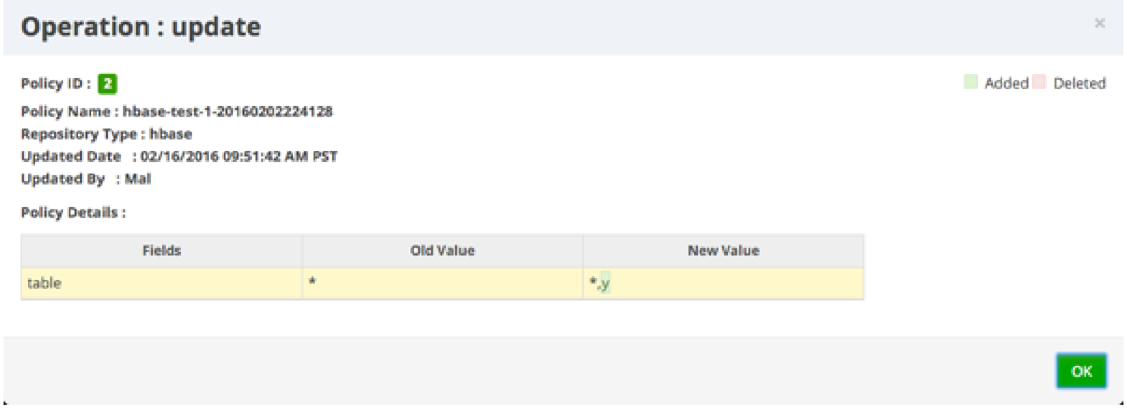
- Access [63]
- Admin [64]
- Login Sessions [65]
- Plugins [66]

Note

To access audit logs from the Ranger Audit UI, enable XAAUDIT.DB.IS_ENABLED in your Ranger plug-in configuration.

8.1. View Operation Details

To view details on a particular operation, click the Policy ID, Operation name, or Session ID.



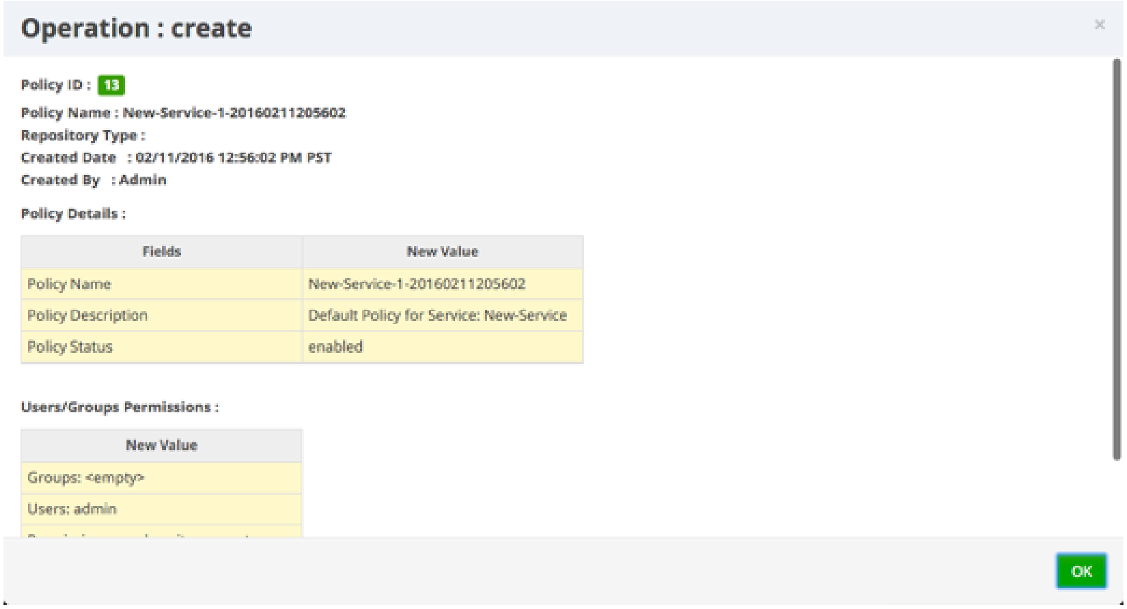
Operation : update

Policy ID : 2

Policy Name : hbase-test-1-20160202224128
Repository Type : hbase
Updated Date : 02/16/2016 09:51:42 AM PST
Updated By : Mal

Policy Details :

Fields	Old Value	New Value
table	*	*,y



Operation : create

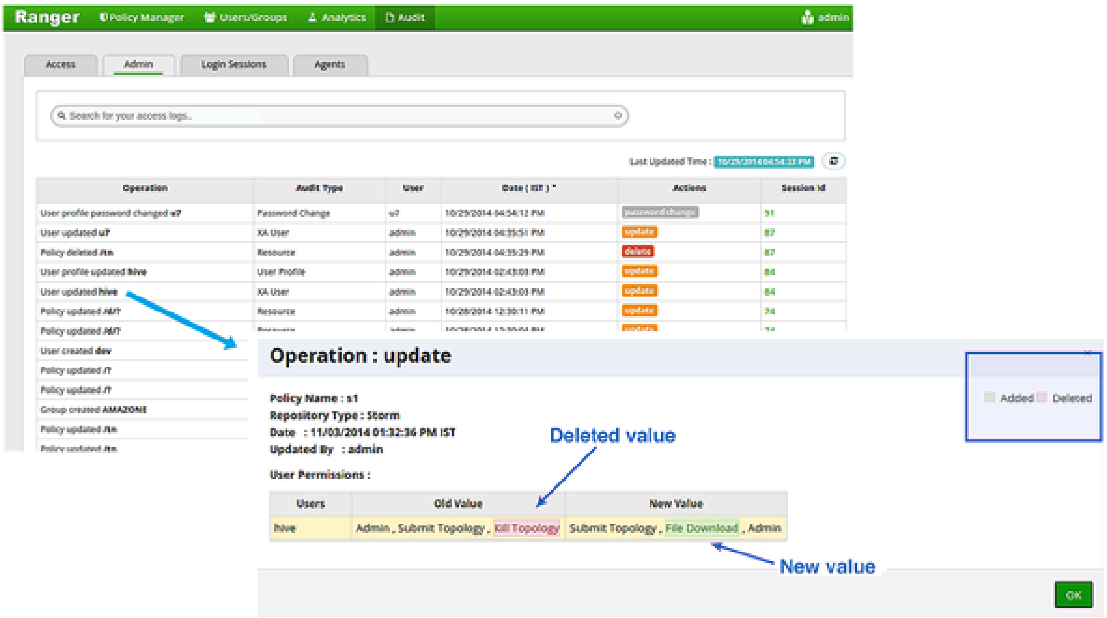
Policy ID : 13
Policy Name : New-Service-1-20160211205602
Repository Type :
Created Date : 02/11/2016 12:56:02 PM PST
Created By : Admin

Policy Details :

Fields	New Value
Policy Name	New-Service-1-20160211205602
Policy Description	Default Policy for Service: New-Service
Policy Status	enabled

Users/Groups Permissions :

New Value
Groups: <empty>
Users: admin



Last Updated Time : 10/29/2014 04:54:33 PM

Operation	Audit Type	User	Date (IST)	Actions	Session Id
User profile password changed u7	Password Change	u7	10/29/2014 04:54:12 PM	password change	91
User updated u7	XA User	admin	10/29/2014 04:35:51 PM	update	87
Policy deleted /tn	Resource	admin	10/29/2014 04:35:29 PM	delete	87
User profile updated hive	User Profile	admin	10/29/2014 02:43:03 PM	update	84
User updated hive	XA User	admin	10/29/2014 02:43:03 PM	update	84
Policy updated /d/?	Resource	admin	10/28/2014 12:30:11 PM	update	74
Policy updated /d/?					
User created dev					
Policy updated /?					
Policy updated /?					
Group created AMAZONE					
Policy updated /tn					
Policy updated /tn					

Operation : update

Policy Name : s1
Repository Type : Storm
Date : 11/03/2014 01:32:36 PM IST
Updated By : admin

User Permissions :

Users	Old Value	New Value
hive	Admin , Submit Topology , Kill Topology	Submit Topology , File Download , Admin

8.2. Access

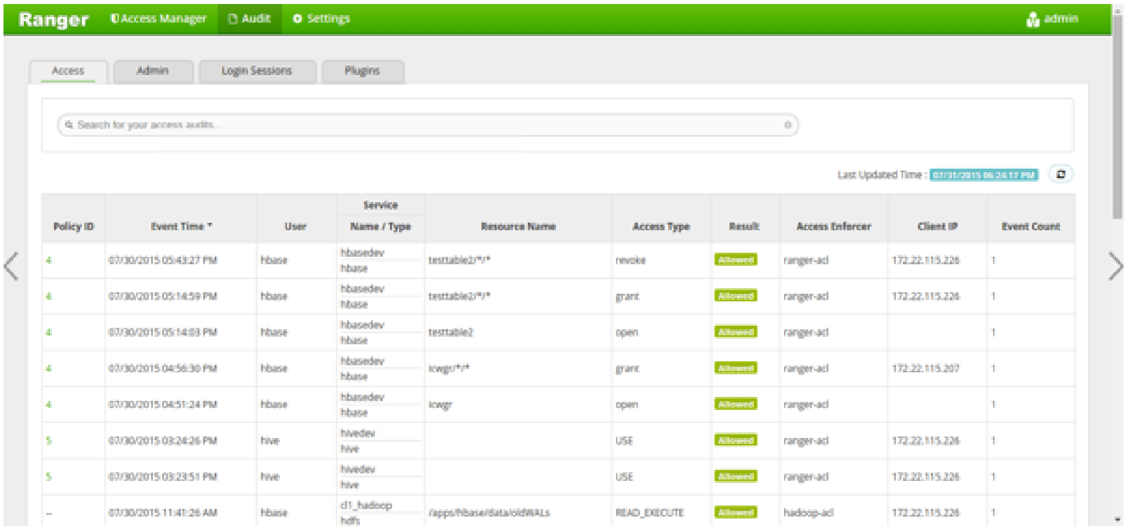
Provides Service activity data for all Policies that have Audit set to **On**. The default service Policy is configured to log all user activity within the Service. This default policy does not contain user and group access rules.

You can filter the data based on the following criteria:

Table 8.1. Search Criteria

Search Criteria	Description
Access Enforcer	Access enforcer indicates who made the decision to allow or deny. In case of HDFS, the enforcer would XA (Ranger) or Hadoop.

Search Criteria	Description
Access Type	Type of access user attempted (E.G., REVOKE, GRANT, OPEN, USE).
Client IP	IP address of the user system that tried to access the resource.
Result	Shows whether the operation was successfull or not.
Service Name	The name of the service that the user tried to access.
Service Type	The type of the service that the user tried to access.
Start Date, End Date	Filters results for a particular date range.
User	Name of the user which tried to access the resource.

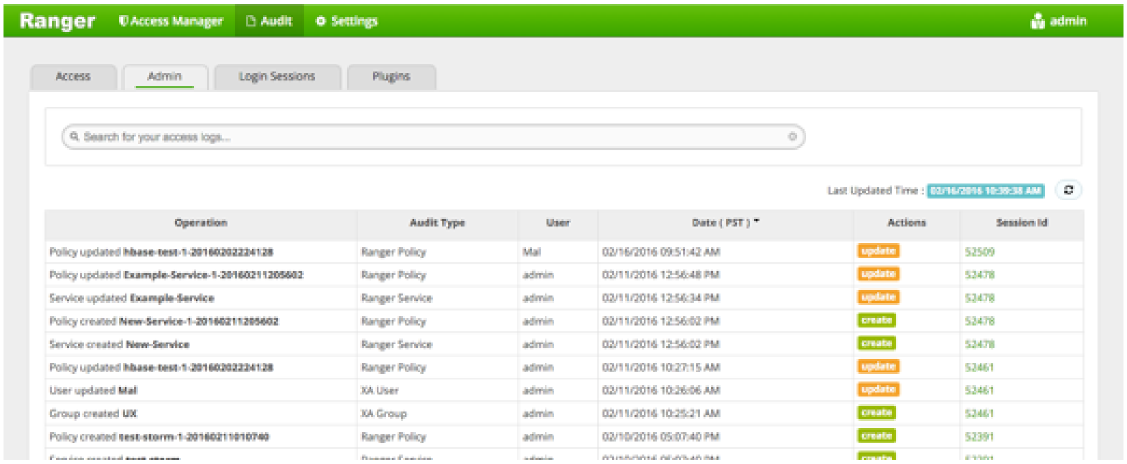


Last Updated Time : 07/31/2015 06:24:17 PM

Policy ID	Event Time *	User	Service Name / Type	Resource Name	Access Type	Result	Access Enforcer	Client IP	Event Count
4	07/30/2015 05:43:27 PM	hbase	hbasedev hbase	testtable2/*/*	revoke	Allowed	ranger-acl	172.22.115.226	1
4	07/30/2015 05:14:59 PM	hbase	hbasedev hbase	testtable2/*/*	grant	Allowed	ranger-acl	172.22.115.226	1
4	07/30/2015 05:14:03 PM	hbase	hbasedev hbase	testtable2	open	Allowed	ranger-acl		1
4	07/30/2015 04:56:30 PM	hbase	hbasedev hbase	icwgr/*/*	grant	Allowed	ranger-acl	172.22.115.207	1
4	07/30/2015 04:51:24 PM	hbase	hbasedev hbase	icwgr	open	Allowed	ranger-acl		1
5	07/30/2015 03:24:26 PM	hive	hivedev hive		USE	Allowed	ranger-acl	172.22.115.226	1
5	07/30/2015 03:23:51 PM	hive	hivedev hive		USE	Allowed	ranger-acl	172.22.115.226	1
--	07/30/2015 11:41:26 AM	hbase	cl1_hadoop hdfs	/apps/hbase/data/oldWALs	READ_EXECUTE	Allowed	hadoop-acl	172.22.115.226	1

8.3. Admin

The Admin tab contains all events for the HDP Security Administration Web UI, including Service, Service Manager, Log in, etc. (actions like create, update, delete, password change).



Last Updated Time : 02/16/2016 10:39:38 AM

Operation	Audit Type	User	Date (PST) *	Actions	Session Id
Policy updated **hbase-test-1-20160202224128**	Ranger Policy	Mal	02/16/2016 09:51:42 AM	update	52509
Policy updated **Example-Service-1-20160211205602**	Ranger Policy	admin	02/11/2016 12:56:48 PM	update	52478
Service updated **Example-Service**	Ranger Service	admin	02/11/2016 12:56:34 PM	update	52478
Policy created **New-Service-1-20160211205602**	Ranger Policy	admin	02/11/2016 12:56:02 PM	create	52478
Service created **New-Service**	Ranger Service	admin	02/11/2016 12:56:02 PM	create	52478
Policy updated **hbase-test-1-20160202224128**	Ranger Policy	admin	02/11/2016 10:27:15 AM	update	52461
User updated **Mal**	XA User	admin	02/11/2016 10:26:06 AM	update	52461
Group created **UX**	XA Group	admin	02/11/2016 10:25:21 AM	create	52461
Policy created **test-storm-1-20160211010740**	Ranger Policy	admin	02/10/2016 05:07:40 PM	create	52391
Service created **test-storm**	Ranger Service	admin	02/10/2016 05:07:40 PM	create	52391

You can filter the data based on the following criteria:

Table 8.2. Search Criteria

Search Criteria	Description
Action	These are operations performed on resources (actions like create, update, delete, password change).
Audit Type	There are three values Resource,asset and xa user according to operations performed on Service,policy and users.
End Date	Login time and date is stored for each session. A date range is used to filter the results for that particular date range.
Session ID	The session count increments each time you try to login to the system
Start Date	Login time and date is stored for each session. A date range is used to filter the results for that particular date range.
User	Username who has performed create,update,delete operation.

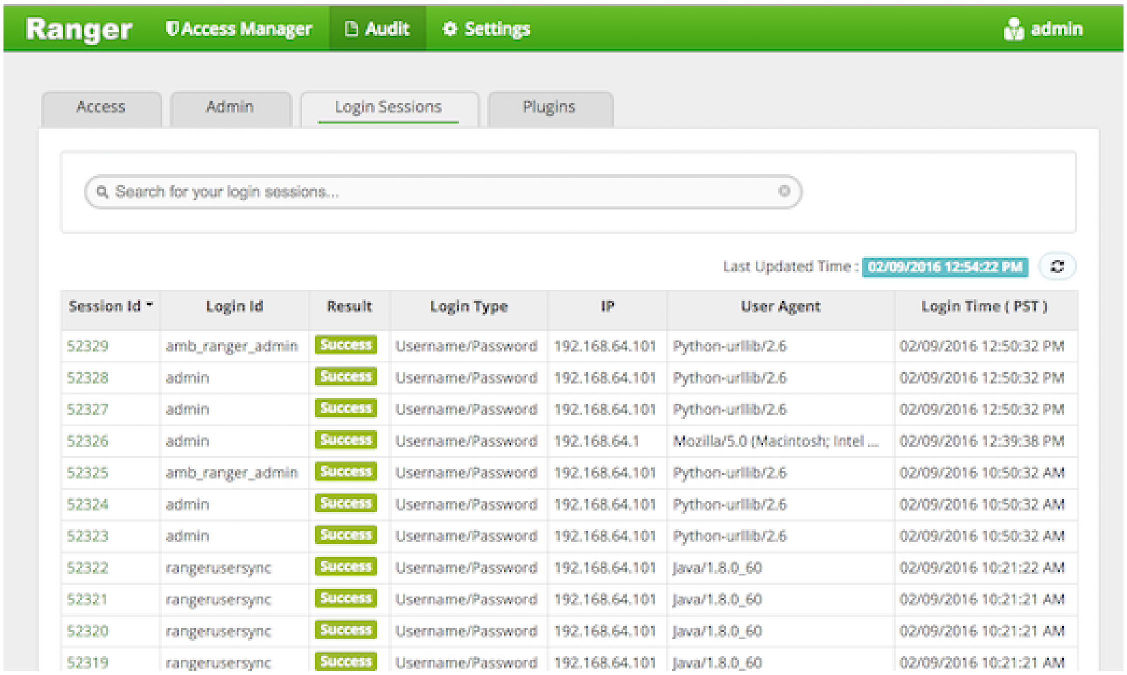
8.4. Login Sessions

The Login Sessions tab logs the information related to the sessions for each login.

You can filter the data based on the following criteria:

Table 8.3. Search Criteria

Search Criteria	Description
Login ID	The username through which someone logs in to the system.
Session-id	The session count increments each time the user tries to log into the system.
Start Date, End Date	Specifies that results should be filtered based on a particular start date and end date.
Login Type	The mode through which the user tries to login (by entering username and password).
IP	The IP address of the system through which the user logged in.
User Agent	The login time and date for each session.
Login time	Logs whether or not the login was successful. Possible results can be Success, Wrong Password, Account Disabled, Locked, Password Expired or User Not Found.



Last Updated Time : 02/09/2016 12:54:22 PM

Session Id	Login Id	Result	Login Type	IP	User Agent	Login Time (PST)
52329	amb_ranger_admin	Success	Username/Password	192.168.64.101	Python-urllib/2.6	02/09/2016 12:50:32 PM
52328	admin	Success	Username/Password	192.168.64.101	Python-urllib/2.6	02/09/2016 12:50:32 PM
52327	admin	Success	Username/Password	192.168.64.101	Python-urllib/2.6	02/09/2016 12:50:32 PM
52326	admin	Success	Username/Password	192.168.64.1	Mozilla/5.0 (Macintosh; Intel ...	02/09/2016 12:39:38 PM
52325	amb_ranger_admin	Success	Username/Password	192.168.64.101	Python-urllib/2.6	02/09/2016 10:50:32 AM
52324	admin	Success	Username/Password	192.168.64.101	Python-urllib/2.6	02/09/2016 10:50:32 AM
52323	admin	Success	Username/Password	192.168.64.101	Python-urllib/2.6	02/09/2016 10:50:32 AM
52322	rangerusersync	Success	Username/Password	192.168.64.101	Java/1.8.0_60	02/09/2016 10:21:22 AM
52321	rangerusersync	Success	Username/Password	192.168.64.101	Java/1.8.0_60	02/09/2016 10:21:21 AM
52320	rangerusersync	Success	Username/Password	192.168.64.101	Java/1.8.0_60	02/09/2016 10:21:21 AM
52319	rangerusersync	Success	Username/Password	192.168.64.101	Java/1.8.0_60	02/09/2016 10:21:21 AM

8.5. Plugins

This tab shows the upload history of the Security Agents.This module displays all of the services exported from the system.You can filter the data based on the following criteria:

Table 8.4. Agents Search Criteria

Search Criteria	Description
Plugin IP	IP Address of the agent that tried to export the service.
Plugin ID	Name of the agent that tried to export the service.
HTTP Response Code	The HTTP code returned when trying to export the service.
Start Date, End Date	Export time and date is stored for each agent. A date range is used to filter the results for that particular date range.
Service Name	The service name we are trying to export.

9. Special Requirements for High Availability Environments

Special Requirements for High Availability Environments In a HA environment, primary and secondary NameNodes must be configured as described in the HDP System Administration Guide.

To enable Ranger in the HDFS HA environment, the HDFS plugin must be set up in each NameNode, and then pointed to the same HDFS service set up in the Security Manager. Any policies created within that HDFS service are automatically synchronized to the primary and secondary NameNodes through the installed Apache Ranger plugin. That way, if the primary NameNode fails, the secondary namenode takes over and the Ranger plugin at that NameNode begins to enforce the same policies for access control.

When creating the service, you must include the `fs.default.name` property must be set to the full hostname of the primary NameNode. If the primary NameNode fails during policy creation, you can then temporarily use the `fs.default.name` of the secondary NameNode in the service details to enable directory lookup for policy creation.

If, while the primary node is down, you wish to create new policies, there is a slight difference in user experience when specifying the resource path. If everything is normal, this is a drop-down menu with selectable paths; however, if your cluster is running from the familover node, there will be no drop-down menu, and you will need to manually enter the path.

Primary NameNode failure does not affect the actual policy enforcement. In this setup for HA, access control is enforced during primary NameNode failure, by the Ranger plugs at the secondary NameNodes.

10. Adding a New Component to Apache Ranger

This document provides a general description of how to add a new component to Apache Ranger.

Apache Ranger has three main components:

- **Admin Tool** – Provides web interface & REST API for managing security policies
- **Custom Authorization Module for components** – Provides custom authorization within the (Hadoop) component to enforce the policies defined in Admin Tool
- **UserGroup synchronizer** – Enables the user/group information in Apache Ranger to synchronize with the Enterprise user/group information stored in LDAP or Active directory.

For supporting new component authorization using Apache Ranger, the component details needs to be added to Apache Ranger as follows:

- Add component details to the Admin Tool
- Develop a custom authorization module for the new component

Adding Component Details to the Admin Tool

The Apache Ranger Admin tool supports policy management via both a web interface (UI) and support for (public) REST API. In order to support a new component in both the UI and the Server, the Admin Tool needs to be modified.

Required UI changes to support the new component:

1. Add a new component template to the Access Manager page (console home page):

   Show new component on the Access Manager page i.e home page[#!/policymanager]. Apache Ranger needs to add table template to Service Manager page and make changes in corresponding JS files. Ranger also needs to create a new service type enum to distinguish the component for which the service/policy is created/updated.

   For example: Add a table template to PolicyManagerLayout_tmpl.html file to view the new component on the Access Manager page and make changes in the PolicyManagerLayout.js file related to the new componen, such as passing knox service collection data to the PolicyManagerLayout_tmpl template. Also create a new service type enum (for example, ASSET_KNOX) in the XAEnums.js file.

2. Add new configuration information to the Service Form:

   Add new configuration fields to Service Form [AssetForm.js] as per new component configuration information. This will cause the display of new configuration fields in the corresponding service Create/Update page. Please note that the AssetForm.js is a common file for every component to create/update the service.

For example: Add new field(configuration) information to AssetForm.js and AssetForm_tmpl.js.

3. Add a new Policy Listing page:

   Add a new policy listing page for the new component in the View Policy list. For example: Create a new KnoxTableLayout.js file and add JS-related changes as per the old component[HiveTableLayout.js] to the View Policy listing. Also create a template page, KnoxTableLayout_tmpl.html.

4. Add a new Policy Create/Update page:

   Add a Policy Create/Update page for the new component. Also add a policy form JS file and its template to handle all policy form-related actions for the new component. For example: Create a new KnoxPolicyCreate.js file for Create/Update Knox Policy. Create a KnoxPolicyForm.js file to add knox policy fields information. Also create a corresponding KnoxPolicyForm_tmpl.html template.

5. Other file changes, as needed:

   Make changes in existing common files as per our new component like Router.js, Controller.js, XAUtils.js, FormInputList.js, UserPermissionList.js, XAEnums.js, etc.

Required server changes for the new component:

Let's assume that Apache Ranger has three components supported in their portal and we want to introduce one new component, Knox:

1. Create New Service Type

   If Apache Ranger is introducing new component i.e Knox, then they will add one new service type for knox. i.e serviceType = “Knox”. On the basis of service type, while creating/updating service/policy, Apache Ranger will distinguish for which component this service/policy is created/updated.

2. Add new required parameters in existig objects and populate objects

   For Policy Creation/Update of any component (i.e HDFS, Hive, Hbase), Apache Ranger uses only one common object, `VXPolicy.` The same goes for the Service Creation/Update of any component: Apache Ranger uses only one common object `VXService.` As Apache Ranger has three components, it will have all the required parameters of all of those three components in `VXPolicy/VXService.` But for Knox, Apache Ranger requires some different parameters which are not there in previous components. Thus, it will add only required parameters into `VXPolicy/VXService` object. When a user sends a request to the Knox create/update policy, they will only send the parameters that are required for Knox to create/update the VXPolicy object.

   After adding new parameters into VXPolixy/VXService, Apache Ranger populates the newly-added parameters in corresponding services, so that it can map those objects with Entity Object.

3. Add newly-added fields (into database table) related parameters into entity object and populate them

As Apache Ranger is using JPA-EclipseLink for database mapping into java, it is necessary to update the Entity object. For example, if for Knox policy Apache Ranger has added two new fields (`topology` and `service`) into db table `x_resource`, it will also have to update the entity object of table (i.e `XXResource`), since it is altering table structure.

After updating the entity object Apache Ranger will populate newly-added parameters in corresponding services (i.e XResourceService), so that it can communicate with the client using the updated entity object.

4. Change middleware code business logic

   After adding and populating newly required parameters for new component, Apache Ranger will have to write business logic into file `AssetMgr`, where it may also need to do some minor changes. For example, if it wants to create a default policy while creating the Service, then on the basis of serviceType, Apache Ranger will create one default policy for the given service. Everything else will work fine, as it is common for all components.

Required database changes for the new component:

For service and policy management, Apache Ranger includes the following tables:

- x_asset (for service)
- x_resource (for service)

As written above, if Apache Ranger is introducing new component then it is not required to create individual table in database for each component. Apache Ranger has common tables for all components.

If Apache Ranger has three components and wants to introduce a fourth one, then it will add required fields into these two tables and will map accordingly with java object. For example, for Knox, Apache Ranger will add two fields (`topology`, `service`) into `x_resource`. After this, it will be able to perform CRUD operation of policy and service for our new component, and also for previous components.

11. Developing a Custom Authorization Module

In the Hadoop ecosystem, each component (i.e., Hive, HBase) has its own authorization implementation and ability to plug in a custom authorization module. To implement the centralized authorization and audit feature for a component, the component should support a customizable (or pluggable) authorization module.

The custom component Authorization Plugin should do the following:

- Provide authorization based on Policies defined in Policy Admin Tool
- Provide audit information based on the authorization decisions

Implementing Custom Component Authorization

To implement the custom component authorization plugin, the Ranger common agent framework provides the following functionalities:

- Ability to read all policies from Policy Manager for a given repository-id
- Ability to log audit information

When the custom authorization module is initialized, the module should do the following:

1. Initiate a REST API call to the “Policy Admin Tool” to retrieve all policies associated with the specific component.
2. Once the policies are available, it should:
   - be built into a custom data structure for enabling the authorization module.
   - kick off the policy updater thread to refresh policies from “Policy Admin Tool” at a regular interval.

When the custom authorization module is called to perform authorization of a component action (such as READ action) on a specific component resource (such as /app folder), the authorization module will:

- **Identify authorization decision** - For each policy:policyList:
  - If (resource in policy <match> auth-requested-resource)
  - If (action-in-policy <match>action-requested
  - If (current-user or current-user-groups or public-group <allowed> for the policy), Return access-allowed
- **Identify auditing needs** - For each policy:policyList
  - If (resource in policy <match> auth-requested-resource), return policy.isAuditEnabled()