

Cloudera Runtime 7.1.6

## Ranger Auditing

Date published: 2019-11-01

Date modified:

# CLOUDERA

<https://docs.cloudera.com/>

# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

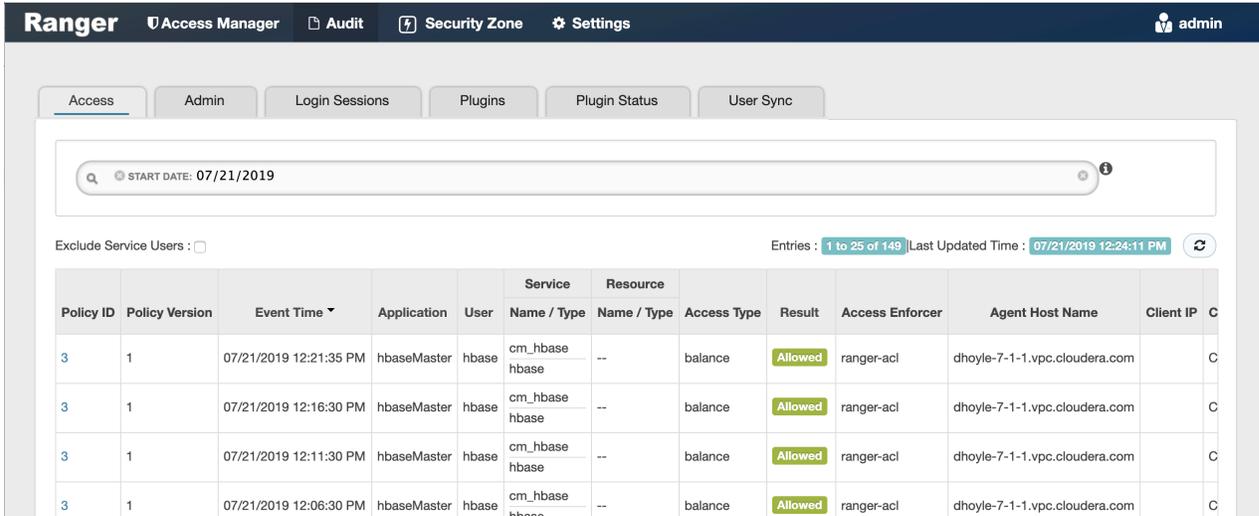
<b>Audit Overview.....</b>	<b>4</b>
<b>Managing Auditing with Ranger.....</b>	<b>4</b>
View audit details.....	4
Create a read-only Admin user (Auditor).....	7
Update Ranger audit configuration parameters.....	8
<b>Ranger Audit Filters.....</b>	<b>9</b>
<b>Changing Ranger audit storage location and migrating data.....</b>	<b>13</b>

## Audit Overview

Apache Ranger provides a centralized framework for collecting access audit history and reporting data, including filtering on various parameters. Ranger enhances audit information obtained from Hadoop components and provides insights through this centralized reporting capability.

## Managing Auditing with Ranger

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



The screenshot shows the Ranger web interface with the 'Audit' tab selected. The page displays a search bar with 'START DATE: 07/21/2019' and a table of audit entries. The table has columns for Policy ID, Policy Version, Event Time, Application, User, Service Name / Type, Resource Name / Type, Access Type, Result, Access Enforcer, Agent Host Name, Client IP, and a 'C' column. The 'Result' column shows 'Allowed' for all entries.

Policy ID	Policy Version	Event Time	Application	User	Service Name / Type	Resource Name / Type	Access Type	Result	Access Enforcer	Agent Host Name	Client IP	C
3	1	07/21/2019 12:21:35 PM	hbaseMaster	hbase	cm_hbase hbase	--	balance	Allowed	ranger-acl	dhoyle-7-1-1.vpc.cloudera.com		C
3	1	07/21/2019 12:16:30 PM	hbaseMaster	hbase	cm_hbase hbase	--	balance	Allowed	ranger-acl	dhoyle-7-1-1.vpc.cloudera.com		C
3	1	07/21/2019 12:11:30 PM	hbaseMaster	hbase	cm_hbase hbase	--	balance	Allowed	ranger-acl	dhoyle-7-1-1.vpc.cloudera.com		C
3	1	07/21/2019 12:06:30 PM	hbaseMaster	hbase	cm_hbase hbase	--	balance	Allowed	ranger-acl	dhoyle-7-1-1.vpc.cloudera.com		C

There are six tabs on the Audit page:

- Access
- Admin
- Login sessions
- Plugins
- Plugin Status
- User Sync

## View audit details

How to view operation details in Ranger audits.

### Procedure

To view details for a particular operation, click any tab, then Policy ID, Operation name, or Session ID.





## Audit > User Sync: Sync details

The screenshot shows the Ranger Admin console with the 'User Sync' tab selected. The table below displays the sync events, and the 'Sync Details' modal is open for one of the entries.

User Name	Sync Source	Number Of New		Number Of Modified		Event Time	Sync Details
		Users	Groups	Users	Groups		
rangerusersync	Unix	0	0	0	0	07/21/2019 01:22:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:21:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:20:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:19:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:18:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:17:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:16:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:15:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:14:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:13:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:12:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:11:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:10:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:09:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:08:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:07:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:06:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:05:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:04:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:03:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:02:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:01:48 PM	[Eye Icon]
rangerusersync	Unix	0	0	0	0	07/21/2019 01:00:47 PM	[Eye Icon]

Name	Value
Unix	nss
File Name	/etc/passwd
Sync time	07/21/2019 10:21:48 AM
Last modified time	12/31/1969 04:00:00 PM
Minimum user id	500
Minimum group id	0
Total number of users synced	35
Total number of groups synced	39

## Create a read-only Admin user (Auditor)

Creating a read-only Admin user (Auditor) enables compliance activities because this user can monitor policies and audit events, but cannot make changes.

### About this task

When a user with the Auditor role logs in, they see a read-only view of Ranger policies and audit events. An Auditor can search and filter on access audit events, and access and view all tabs under Audit to understand access events. They cannot edit users or groups, export/import policies, or make changes of any kind.

### Procedure

1. Select Settings > Users/Groups/Roles.
2. Click Add New User.

- Complete the **User Detail** section, selecting Auditor as the role:

The screenshot shows the Ranger 'User Create' interface. The 'User Detail' section includes the following fields:

- User Name \*: auditor1
- New Password \*: [masked]
- Password Confirm \*: [masked]
- First Name \*: Audrey
- Last Name: [empty]
- Email Address: [empty]
- Select Role \*: Auditor
- Group: audit

Buttons: Save, Cancel

- Click Save.

## Update Ranger audit configuration parameters

How to change the default time settings that control how long Ranger keeps audit data collected by solr.

### About this task

You can configure parameters that control how much data collected by solr that Ranger will store for auditing purposes.

**Table 1: Ranger Audit Configuration Parameters**

Parameter Name	Description	Default Setting	Units
ranger.audit.solr.config.ttl	Time To Live for Solr Collection of Ranger Audits	90	days
ranger.audit.solr.config.delete.trigger	Auto Delete Period in seconds for Solr Collection of Ranger Audits for expired documents	1	days (configurable)



**Note:** "Time To Live for Solr Collection of Ranger Audits" is also known as the Max Retention Days attribute.

### Procedure

- From Cloudera Manager choose Ranger Configuration .
- In Search, type ranger.audit.solr.config, then press Return.
- In ranger.audit.solr.config.ttl, set the the number of days to keep audit data.
- In ranger.audit.solr.config.delete.trigger set the number and units (days, minutes, hours, or seconds) to keep data for expired documents

5. Refresh the configuration, using one of the following two options:
  - a) Click Refresh Configuration, as prompted or, if Refresh Configuration does not appear,
  - b) In Actions, click Update Solr config-set for Ranger, then confirm.

## Ranger Audit Filters

You can use Ranger audit filters to control the amount of audit log data collected and stored on your cluster.

### About Ranger audit filters

Ranger audit filters allow you to control the amount of audit log data for each Ranger service. Audit filters are defined using a JSON string that is added to each service configuration. The audit filter JSON string is a simplified form of the Ranger policy JSON. Audit filters appear as rows in the Audit Filter section of the Edit Service view for each service. The set of audit filter rows defines the audit log policy for the service. For example, the default audit log policy for the Hadoop SQL service appears in the in the Ranger Admin web UI Service Manager Edit Service when you scroll down to Audit Filter. Audit filter is checked (visible) by default. In this example, the top row defines an audit filter that causes all instances of "access denied" to appear in audit logs. The lower row defines a filter that causes no metadata operations to appear in audit logs. These two filters comprise the default audit filter policy for the Hadoop SQL service.

The screenshot shows the Ranger Admin web UI for editing a service's audit filters. The interface includes a navigation bar with 'Ranger', 'Access Manager', 'Audit', 'Security Zone', and 'Settings'. The main content area is titled 'Service Manager > Edit Service' and shows the 'Audit Filter' section. The table below has the following structure:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions	Select User	Select Group	Select Role
No	Select Value	--	x METADATA OPERATION	Add Permissions	Select User	Select Group	Select Role

Below the table, there is a '+', a 'Test Connection' button, and 'Save', 'Cancel', and 'Delete' buttons at the bottom.

### Default audit filters

Default audit filters for the following Ranger service appear in the Edit Services and can then be modified as needed by Admin users.

HDFS service:

Audit Filter:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
Yes	Select Value	--	<input type="checkbox"/> delete <input type="checkbox"/> rename	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	--	<input type="checkbox"/> listStatus <input type="checkbox"/> getFileInfo <input type="checkbox"/> listCachePools <input type="checkbox"/> listCacheDirectives <input type="checkbox"/> listCompustFileBlocks <input type="checkbox"/> monitorHealth <input type="checkbox"/> rollEditLog <input type="checkbox"/> open	Add Permissions +	<input type="checkbox"/> hdfs	Select Group	Select Role
No	Select Value	path/user/ocde/share/no recursive	Type Action Name	Add Permissions +	<input type="checkbox"/> hdfs	Select Group	Select Role
No	Select Value	path/user/spark/applicationHistory recursive	Type Action Name	Add Permissions +	<input type="checkbox"/> spark	Select Group	Select Role
No	Select Value	path/user/hue recursive	Type Action Name	Add Permissions +	<input type="checkbox"/> hue	Select Group	Select Role
No	Select Value	path/hbase recursive	Type Action Name	Add Permissions +	<input type="checkbox"/> hbase	Select Group	Select Role
No	Select Value	path/user/history recursive	Type Action Name	Add Permissions +	<input type="checkbox"/> mapped	Select Group	Select Role
No	Select Value	--	<input type="checkbox"/> getFileInfo	Add Permissions +	Select User	Select Group	Select Role

HBase service:

Audit Filter:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	table~"ROOF"; "META"; "_acl"; hbase:meta; hbase:default; hbase	Type Action Name	Add Permissions +	<input type="checkbox"/> hbase	Select Group	Select Role
No	Select Value	table~"atlas_janus, ATLAS_ENTITY_AUDIT_EVENTS column-family=" column:"	Type Action Name	Add Permissions +	<input type="checkbox"/> atlas <input type="checkbox"/> hbase	Select Group	Select Role
No	Select Value	--	<input type="checkbox"/> balance	Add Permissions +	<input type="checkbox"/> hbase	Select Group	Select Role

Hadoop SQL service:

Audit Filter:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	--	<input type="checkbox"/> METADATA OPERATION	Add Permissions +	Select User	Select Group	Select Role

Knox service

Audit Filter:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	--	Type Action Name	Add Permissions +	<input type="checkbox"/> knox	Select Group	Select Role

Solr service

Audit Filter:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	--	Type Action Name	Add Permissions +	<input type="checkbox"/> hive <input type="checkbox"/> hdfs <input type="checkbox"/> kafka <input type="checkbox"/> hbase <input type="checkbox"/> sqoop <input type="checkbox"/> ranger-sz <input type="checkbox"/> knox <input type="checkbox"/> atlas	Select Group	Select Role

Kafka service:

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	topic:ATLAS_ENTITIES, ATLAS_HOOK, ATLAS_SPARK_HOOK	describe, publish, consume	Add Permissions +	atlas	Select Group	Select Role
No	Select Value	topic:ATLAS_HOOK	publish, describe	Add Permissions +	hive, hbase, impala, nifi	Select Group	Select Role
No	Select Value	topic:ATLAS_ENTITIES	consume, describe	Add Permissions +	rangertagsync	Select Group	Select Role
No	Select Value	consumergroup:*	consume	Add Permissions +	atlas, rangertagsync	Select Group	Select Role
No	Select Value	--	Type Action Name	Add Permissions +	kafka	Select Group	Select Role

### KMS service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	--	read	Add Permissions +	keyadmin	Select Group	Select Role

### Atlas service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	--	Type Action Name	Add Permissions +	atlas	Select Group	Select Role

### Ozone service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role
No	Select Value	--	Type Action Name	Add Permissions +	om	Select Group	Select Role

### Tag-based service

Is Audited	Access Result	Resources	Operations	Permissions	Users	Groups	Roles
Yes	DENIED	--	Type Action Name	Add Permissions +	Select User	Select Group	Select Role

Default audit filter policies do not exist for Yarn, NiFi, NiFi Registry, Kudu, or schema registry services.

## Ranger audit filter policy configuration

To configure an audit filter policy, click the Edit icon for either a resource-, or tag-based service in the Ranger Admin web UI. You configure a Ranger audit filter policy by adding (+), deleting (X), or modifying each audit filter row for the service. The preceding example shows the Add and Delete icons for each filter row. To configure each filter in the policy, use the controls in the filter row to edit filter properties. For example, you can configure:

#### Is Audited: choose Yes or No

to include or not include a filter in the audit logs for a service

#### Access Result: choose DENIED, ALLOWED, or NOT\_DETERMINED

to include that access result in the audit log filter

#### Resources: Add or Delete a resource item

to include or remove the resource from the audit log filter

#### Operations: Add or Remove an action name

to include the action/operation in the audit log filter

(click x to remove an existing operation)

**Permissions: Add or Remove permissions**

1. Click + in Permissions to open the Add dialog.
2. Select/Unselect required permissions.

For example, in HDFS service select read, write, execute, or All permissions.

**Users: click Select User to see a list of defined users**

to include one or multiple users in the audit log filter

**Groups: click Select Group to see a list of defined groups**

to include one or multiple groups in the audit log filter

**Roles: click Select Role to see a list of defined roles**

to include one or multiple roles in the audit log filter

**Audit filter details**

- When you save the UI selections described in the preceding list, audit filters are defined as a JSON list. Each service references a unique list.
- For example, ranger.plugin.audit.filters for the HDFS service includes:

```
[
  {
    "accessResult": "DENIED",
    "isAudited": true
  },
  {
    "users": [
      "unaudited-user1"
    ],
    "groups": [
      "unaudited-group1"
    ],
    "roles": [
      "unaudited-role1"
    ],
    "isAudited": false
  },
  {
    "actions": [
      "listStatus",
      "getFileinfo"
    ],
    "accessTypes": [
      "execute"
    ],
    "isAudited": false
  },
  {
    "resources": {
      "path": {
        "values": [
          "/audited"
        ],
        "isRecursive": true
      }
    },
    "isAudited": true
  },
]
```

```

    {
      "resources": {
        "path": {
          "values": [
            "/unaudited"
          ],
          "isRecursive": true
        }
      },
      "isAudited": false
    }
  ]

```

- Each value in the list is an audit filter, which takes the format of a simplified Ranger policy, along with access results fields.
- Audit filters are defined with rules on Ranger policy attributes and access result attributes.
  - Policy attributes: resources, users, groups, roles, accessTypes
  - Access result attributes: isAudited, actions, accessResult
- The following audit filter specifies that accessResult=DENIED will be audited.

The isAudited flag specifies whether or not to audit.

```
{ "accessResult": "DENIED", "isAudited": true }
```

- The following audit filter specifies that “resource => /unaudited” will not be audited.

```
{ "resources": { "path": { "values": [ "/unaudited" ], "isRecursive": true } }, "isAudited": false }
```

- The following audit filter specifies that access to resource database=> sys table=> dump by user “use2” will not be audited.

```
{ "resources": { "database": { "values": [ "sys" ] }, "table": { "values": [ "dump" ] } }, "users": [ "user2" ], "isAudited": false }
```

- The following audit filter specifies that access result in actions => listStatus, getFileInfo and accessType => execute will not be audited.

```
{ "actions": [ "listStatus", "getFileinfo" ], "accessTypes": [ "execute" ], "isAudited": false }
```

- The following audit filter specifies that access by user “superuser1” and group “supergroup1” will not be audited.

```
{ "users": [ "superuser1" ], "groups": [ "supergroup1" ], "isAudited": false }
```

- The following audit filter specifies that access to any resource tagged as NO\_AUDIT will not be audited.

```
{ "resources": { "tag": { "values": [ "NO_AUDIT" ] } }, "isAudited": false }
```

## Changing Ranger audit storage location and migrating data

How to change the location of existing and future Ranger audit data collected by Solr from HDFS to a local file system or from a local file system to HDFS.

### Before you begin

- Stop Atlas from Cloudera Manager.

- If using Kerberos, set the SOLR\_PROCESS\_DIR environment variable.

```
# export SOLR_PROCESS_DIR=$(ls -ldtr /var/run/cloudera-scm-agent/process/
*SOLR_SERVER | tail -1)
```

### About this task

Starting with Cloudera Runtime version 7.1.4 / 7.2.2, the storage location for ranger audit data collected by Solr changed to local file system from HDFS, as was true for previous versions. The default storage location Ranger audit data storage location for Cloudera Runtime-7.1.4+ and Cloudera Runtime-7.2.2+ installations is local file system. After upgrading from an earlier Cloudera platform version, follow these steps to backup and migrate your Ranger audit data and change the location where Solr stores your future Ranger audit records.

- The default value of the index storage in the local file system is /var/lib/solr-infra. You can configure this, using Cloudera Manager Solr Configuration "Solr Data Directory" .
- The default value of the index storage in HDFS is /solr-infra. You can configure this, using Cloudera Manager Solr Configuration "HDFS Data Directory" .

### Procedure

1. Create HDFS Directory to store the collection backups.

As an HDFS super user, run the following commands to create the backup directory:

```
# hdfs dfs -mkdir /solr-backups
# hdfs dfs -chown solr:solr /solr-backups
```

2. Obtain valid kerberos ticket for Solr user.

```
# kinit -kt solr.keytab solr/$(hostname -f)
```

3. Download the configs for the collection.

```
# solrctl instancedir --get ranger_audits /tmp/ranger_audits
# solrctl instancedir --get atlas_configs /tmp/atlas_configs
```

4. Modify the solrconfig.xml for each of the configs for which data needs to be stored in HDFS.

In /tmp/<config\_name>/conf created during Step 3., edit properties in the solrconfig.xml file as follows:

- When migrating your data storage location from a local file system to HDFS, replace these two lines:

```
<directoryFactory name="DirectoryFactory"
  class="{solr.directoryFactory:solr.NRTCachingDirectoryFactory}">
<lockType>{solr.lock.type:native}</lockType>
```

with

```
<directoryFactory name="DirectoryFactory"
  class="{solr.directoryFactory:org.apache.solr.core.HdfsDirectoryFactory}">
<lockType>{solr.lock.type:hdfs}</lockType>
```

- When migrating your data storage location from HDFS to a local file system, replace these two lines:

```
<directoryFactory name="DirectoryFactory"
  class="{solr.directoryFactory:org.apache.solr.core.HdfsDirectoryFactory}">
<lockType>{solr.lock.type:hdfs}</lockType>
```

with

```
<directoryFactory name="DirectoryFactory"
  class="{solr.directoryFactory:solr.NRTCachingDirectoryFactory}">
<lockType>{solr.lock.type:native}</lockType>
```

## 5. Update the modified configs in Zookeeper.

```
# solrctl --jaas $SOLR_PROCESS_DIR/jaas.conf instancedir --update
atlas_configs /tmp/atlas_configs
# solrctl --jaas $SOLR_PROCESS_DIR/jaas.conf instancedir --update
ranger_audits /tmp/ranger_audits
```

## 6. Backup the Solr collections.

- When migrating your data storage location from a local file system to HDFS, run:

```
# curl -k --negotiate -u : "https://$(hostname
-f):8995/solr/admin/collections?action=BACKUP&name=vertex_backup&col
lection=vertex_index&
location=hdfs://<Namenode_Hostname>:8020/solr-backups"
```

In the preceding command, the important points are name, collection, and location:

### **name**

specifies the name of the backup. It should be unique per collection

### **collection**

specifies the collection name for which the backup will be performed

### **location**

specifies the HDFS path, where the backup will be stored

Repeat the curl command for different collections, modifying the parameters as necessary for each collection.

The expected output would be -

```
"responseHeader": {
  "status": 0,
  "QTime": 10567},
"success": {
  "Solr_Server_Hostname:8995_solr": {
    "responseHeader": {
      "status": 0,
      "QTime": 8959}}}}
```

- When migrating your data storage location from HDFS to a local file system:

Refer to Back up a Solr collection for specific steps, and make the following adjustments:

- If TLS is enabled for the Solr service, specify the trust store and password by using the ZKCLI\_JVM\_FLAGS environment variable before you begin the procedure.

```
# export ZKCLI_JVM_FLAGS="-Djavax.net.ssl.trustStore=/path/to/
truststore.jks -Djavax.net.ssl.trustStorePassword="
```

- Create Snapshot

```
# solrctl --jaas $SOLR_PROCESS_DIR/jaas.conf collection --create-
snapshot <snapshot_name> -c <collection_name>
```

- or use the Solr API to take the backup:

```
curl -i -k --negotiate -u : "https://(hostname -f):8995/solr/admin/
collections?"
```

```
action=BACKUP&name=ranger_audits_bkp&collection=ranger_audits&location=/
path/to/solr-backups"
```

- Export Snapshot

```
# solrctl --jaas $SOLR_PROCESS_DIR/jaas.conf collection
--export-snapshot <snapshot_name> -c <collection_name> -d
<destination_directory>
```



**Note:** The <destination\_directory> is a HDFS path. The ownership of this directory should be solr:solr.

## 7. Delete the collections from the original location.

All instances of Solr service should be up, running, and healthy before deleting the collections. Use Cloudera Manager to check for any alerts or warnings for any of the instances. If alerts or warnings exist, fix those before deleting the collection.

```
# solrctl collection --delete edge_index
# solrctl collection --delete vertex_index
# solrctl collection --delete fulltext_index
# solrctl collection --delete ranger_audits
```

## 8. Verify that the collections are deleted from the original location.

```
# solrctl collection --list
```

This will give an empty result.

## 9. Verify that no leftover directories for any of the collections have been deleted.

- When migrating your data storage location from a local file system to HDFS:

```
# cd /var/lib/solr-infra
```

Get the value of "Solr Data Directory, using Cloudera Manager Solr Configuration .

```
# ls -ltr
```

- When migrating your data storage location from HDFS to a local file system, replace these two lines:

```
# hdfs dfs -ls /solr/<collection_name>
```



**Note:** If any directory name which starts with the collection name deleted in Step 7. exists, delete/ move the directory to another path.

## 10. Restore the collection from backup to the new location.

Refer to Restore a Solr collection, for more specific steps.

```
# curl -k --negotiate -u : "https://$(hostname
-f):8995/solr/admin/collections?
action=RESTORE&name=<Name_of_backup>&location=hdfs:/
<Namenode_Hostname>:8020/solr-backups&collection=<Collection_Name>"
```

```
# solrctl collection --restore ranger_audits
-l hdfs://<Namenode_Hostname>:8020/solr-backups
```

```
-b ranger_backup -i ranger1
```

The request id must be unique for each restore operation, as well as for each retry.

To check the status of restore operation:

```
# solrctl collection --request-status <requestId>
```



**Note:** If the Atlas Collections (vertex\_index, fulltext\_index and edge\_index) restore operations fail, restart the solr service and rerun the restore command. Now, the restart operations should complete successfully.

## 11. Verify the Atlas & Ranger functionality.

Verify that both Atlas and Ranger audits functions properly, and that you can see the latest audits in Ranger Web UI and latest lineage in Atlas.

- To verify Atlas audits, create a test table in Hive, and then query the collections to see if you are able to view the data.
- You can also query the collections every 20-30 seconds (depending on how other services utilize Atlas/Ranger), and verify if the "numDocs" value increases at every query.

```
# curl -k --negotiate -u : "https://$(hostname -f):8995/solr/edge_index/  
select?q=%3A*&wt=json&ident=true&rows=0"  
# curl -k --negotiate -u : "https://$(hostname -f):8995/solr/vertex_index/  
select?q=%3A*&wt=json&ident=true&rows=0"  
# curl -k --negotiate -u : "https://$(hostname -f):8995/solr/  
fulltext_index/select?q=%3A*&wt=json&ident=true&rows=0"  
# curl -k --negotiate -u : "https://$(hostname -f):8995/solr/  
ranger_audits/select?q=%3A*&wt=json&ident=true&rows=0"
```