

cloudera[®]

Cloudera Data Management

Important Notice

© 2010-2021 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, and any other product or service names or slogans contained in this document are trademarks of Cloudera and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of Cloudera or the applicable trademark holder. If this documentation includes code, including but not limited to, code examples, Cloudera makes this available to you under the terms of the Apache License, Version 2.0, including any required notices. A copy of the Apache License Version 2.0, including any notices, is included herein. A copy of the Apache License Version 2.0 can also be found here: <https://opensource.org/licenses/Apache-2.0>

Hadoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation. All other trademarks, registered trademarks, product names and company names or logos mentioned in this document are the property of their respective owners. Reference to any products, services, processes or other information, by trade name, trademark, manufacturer, supplier or otherwise does not constitute or imply endorsement, sponsorship or recommendation thereof by us.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Cloudera.

Cloudera may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Cloudera, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property. For information about patents covering Cloudera products, see <http://tiny.cloudera.com/patents>.

The information in this document is subject to change without notice. Cloudera shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

Cloudera, Inc.

**395 Page Mill Road
Palo Alto, CA 94306
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-362-0488
www.cloudera.com**

Release Information

Version: Cloudera Navigator 2.11.x
Date: February 3, 2021

Table of Contents

About Cloudera Data Management.....	5
--	----------

Cloudera Navigator Analytics.....	6
--	----------

Data Stewardship Dashboard.....	8
---------------------------------	---

<i>Dashboard.....</i>	<i>9</i>
-----------------------	----------

<i>Data Explorer.....</i>	<i>13</i>
---------------------------	-----------

Cloudera Navigator and the Cloud.....	14
--	-----------

Using Cloudera Navigator with Altus Clusters.....	14
---	----

<i>How it Works: Background to the Setup Tasks.....</i>	<i>15</i>
---	-----------

<i>Configuring Extraction for Altus Clusters on AWS.....</i>	<i>15</i>
--	-----------

Configuring Extraction for Amazon S3.....	20
---	----

<i>AWS Credentials Requirements.....</i>	<i>21</i>
--	-----------

<i>Default Configuration.....</i>	<i>21</i>
-----------------------------------	-----------

<i>Custom Configurations.....</i>	<i>23</i>
-----------------------------------	-----------

<i>Using Cloudera Navigator with Amazon S3.....</i>	<i>29</i>
---	-----------

Cloudera Navigator Metadata Architecture.....	33
--	-----------

Defining Properties for Managed Metadata.....	35
---	----

<i>Creating Custom Properties with the Cloudera Navigator Console</i>	<i>35</i>
---	-----------

<i>Using Cloudera Navigator Console to Manage Properties.....</i>	<i>37</i>
---	-----------

<i>Navigator Built-in Classes.....</i>	<i>38</i>
--	-----------

<i>Defining Metadata with the Navigator API and Navigator SDK.....</i>	<i>39</i>
--	-----------

Adding and Editing Metadata.....	40
----------------------------------	----

Metadata Extraction and Indexing.....	47
---------------------------------------	----

Searching Metadata.....	48
-------------------------	----

<i>Using the Cloudera Navigator Console.....</i>	<i>48</i>
--	-----------

Performing Actions on Entities.....	53
-------------------------------------	----

Metadata Policies.....	54
------------------------	----

<i>Metadata Policy Expressions.....</i>	<i>57</i>
---	-----------

Cloudera Navigator Auditing Architecture.....	65
--	-----------

Exploring Audit Data Using the Cloudera Navigator Console.....	66
--	----

<i>Viewing Audit Events.....</i>	<i>67</i>
----------------------------------	-----------

<i>Filtering Audit Events.....</i>	<i>67</i>
------------------------------------	-----------

<i>Monitoring Navigator Audit Service Health</i>	68
Service Audit Events.....	70
<i>Operations by Component</i>	72
<i>Navigator Metadata Server Sub Operations</i>	73
Cloudera Navigator Audit Event Reports.....	73
<i>Creating Audit Event Reports</i>	73
<i>Editing Audit Event Reports</i>	74
<i>Downloading Audit Event Reports</i>	74
Downloading HDFS Directory Access Permission Reports.....	76
Cloudera Navigator Auditing Use Cases.....	76

Cloudera Navigator Provenance Use Case.....80

Cloudera Navigator Lineage Diagram Reference.....83

Manipulating Lineage Diagrams.....	86
Displaying a Template Lineage Diagram.....	90
Displaying an Instance Lineage Diagram.....	92
Displaying the Template Lineage Diagram for an Instance Lineage Diagram.....	92
Using Lineage to Display Table Schema.....	92
<i>Displaying Hive, Impala, and Sqoop Table Schema</i>	92
<i>Displaying Pig Table Schema</i>	93
<i>Displaying HDFS Dataset Schema</i>	93

Search: Syntax and Properties Reference.....96

Search Syntax.....	96
Searchable Properties Reference.....	97

Cloudera Navigator Administration Tasks.....102

Maintaining Metadata Store Using Purge.....	102
<i>Scheduling the Purge Process</i>	102

Troubleshooting Navigator Data Management.....105

Cloudera Navigator-Cloudera Altus	105
Navigator Audit Server.....	105
Navigator Metadata Server.....	107

Appendix: Apache License, Version 2.0.....109

About Cloudera Data Management

This guide shows you how to use Cloudera Navigator Data Management component for comprehensive data governance, compliance, data stewardship, and other data management tasks. Data management tasks include auditing all access to data stored in the cluster, in HDFS and Hive metastores; reviewing, customizing, and updating metadata about the objects contained in the cluster; and tracing the lineage of data objects.



Important: Cloudera Navigator Data Management requires a Cloudera Enterprise license. This feature is not available in Cloudera Express. See for [Managing Licenses](#) for details.

Cloudera Navigator Analytics

Cloudera Navigator's powerful metadata management capabilities underpin its Analytics features. Improved significantly and fully-enabled as of Cloudera Navigator 2.10, the Cloudera Navigator console lets you view metadata and audit analytics through several graphically rich dashboards, including the [Data Stewardship Dashboard](#) and Data Explorer. This section focuses on the HDFS Analytics menu of the Cloudera Navigator console and how to use it to

Viewing Metadata Analytics

Required Role: [Lineage Viewer](#) and [Policy Administrator](#) (or **Full Administrator**)

1. Open your browser.
2. Navigate to the host within the cluster running the Cloudera Navigator Metadata Server role, replacing the URL with the appropriate path.

```
http://fqdn-1.example.com:7187/login.html
```

The login page displays.

3. Log in to the Cloudera Navigator console using the [credentials](#) assigned by your administrator.
4. Click the **Analytics** tab. The Metadata analytics tab displays.
5. Click the **Source** button and select an HDFS service instance from the drop-down list.
6. The Metadata tab displays a set of bar graphs that list the number of files that satisfy groups of values for last access time, created time, size, block size, and replication count.
 - To display the files at the right, click a bar. This draws a blue selection outline around the bar and selects the property checkbox.
 - To select more than one value, grab a bar edge and brush a range of values.
 - To change a range, click a bar, drag to a different range of values, and drop.
 - To reduce a range, grab a bar edge and contract the range.
 - To clear a property, clear the checkbox. The previous selection is indicated with a gray outline.
 - When you select a previously selected property, the previous selection is reused. For example, if you had previously selected one and three for replication count, and you reselect the replication count checkbox, the values one and three are reselected.
 - To clear all selections, present and previous, click **Clear all selections**.
7. In the listing on the right, select an option to display the number of files by directory, owner, or tag. In the listing:
 - Filter the selections by typing strings in the search box and pressing **Enter** or **Return**.
 - Add categories (directory, owner, or tag) to a search query and display the Search tab by doing one of the following:
 - Clicking a directory, owner, or tag name link.
 - Selecting **Actions > Show in search**. To further refine the query, select one or more checkboxes, and select **Actions > Show selection in search**.
 - **Required Role:** [Policy Administrator](#) (or **Full Administrator**)

Add categories to the search query of a new policy and display the Policies tab by selecting **Actions > Create a policy**. To further refine the query, select one or more checkboxes, and select **Actions > Create a policy from selection**.

Viewing Audit Analytics

Required Role: [Auditing Viewer](#) (or **Full Administrator**)

The screenshot displays the Cloudera Navigator Analytics interface. At the top, there is a navigation bar with 'Search', 'Audits', 'Analytics', 'Policies', 'Administration', and 'admin'. Below this, the 'HDFS-1 Analytics' section is visible, with a 'Source' dropdown menu. The 'Activity' tab is selected, showing 'Top Users' and 'Top Commands' charts. The 'Top Users' chart shows four users from 'E.CLOUDERA.COM' with counts of 20, 13, 3, and 1. The 'Top Commands' chart shows five commands: getFileinfo (998), listStatus (880), setPermission (793), create (752), and setOwner (87).

1. [Accessing the Cloudera Navigator console.](#)
2. Click the **Analytics** tab. If the logged-in user has a role that permits access to metadata analytics, the Metadata analytics tab displays.
3. Click the **Source** button and select an HDFS service instance from the drop-down list.
4. If not already displayed, click the **Audit** tab. The Activity tab displays a bar graph that lists the number of files that have been read the number of times listed in the x-axis.
 - To display at the right the directories containing the files that have been read, click an activity bar. This draws a blue selection outline around the bar and selects the Activity checkbox.
 - To select more than one value, grab a bar edge and brush a range of values.
 - To change a range, click a bar, drag to a different range of values, and drop.
 - To reduce a range, grab a bar edge and contract the range.
 - To clear Activity, clear the checkbox. The previous selection is indicated with a gray outline.
 - When you select Activity and the graph had a previous selection, the previous selection is reused. For example, if you had previously selected values spanning six through nine for the number of times files have been read, and you select the checkbox, six through nine will be reselected.
5. In the directory listing on the right:
 - Filter the directories by typing directory strings in the search box and pressing **Enter** or **Return**.
 - **Required Role:** [Lineage Administrator](#) (or **Metadata Administrator**, **Full Administrator**)

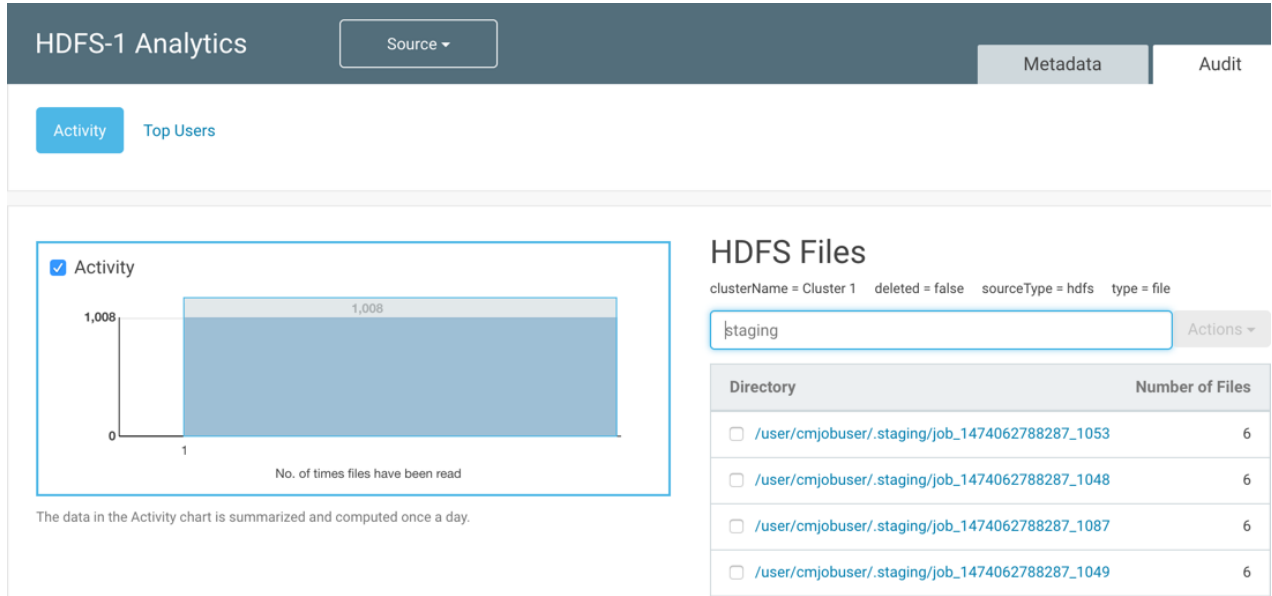
Add selected directories to a search query and display the Search tab by doing one of the following:

 - Clicking a directory name link.
 - Selecting one or more directory checkboxes and selecting **Actions > Show selection in search**.
 - **Required Role:** [Metadata Viewer](#) (or **Metadata Administrator**, **Full Administrator**)

Required Role: [Lineage Administrator](#) (or **Metadata Administrator**, **Full Administrator**)

Add selected directories to the search query of a new policy and display the Policies tab by selecting one or more directory checkboxes and selecting **Actions > Create a policy from selection**.

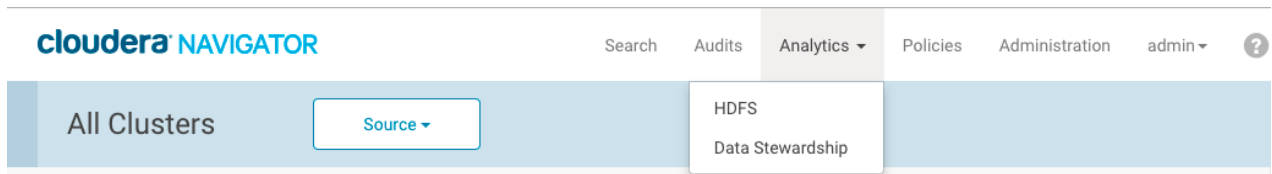
For example, the following screenshot shows files that have been accessed once and match the string `staging`. Each directory has six files that has been accessed.



Data Stewardship Dashboard

The Cloudera Navigator Data Stewardship dashboard captures a variety of information about data, metadata, and user jobs that process the data. The Data Stewardship dashboard provides information and metrics to help you understand the state of the data and data usage, and allows you to visualize trends and averages for a variety of data sources and actions.

Access the dashboard by clicking **Analytics** and then choosing **Data Stewardship** on the navigation bar. Specify the source clusters by clicking **Source** and clicking a cluster name or **All Clusters**.



On the Data Stewardship dashboard, select a tab for the information you want to view:

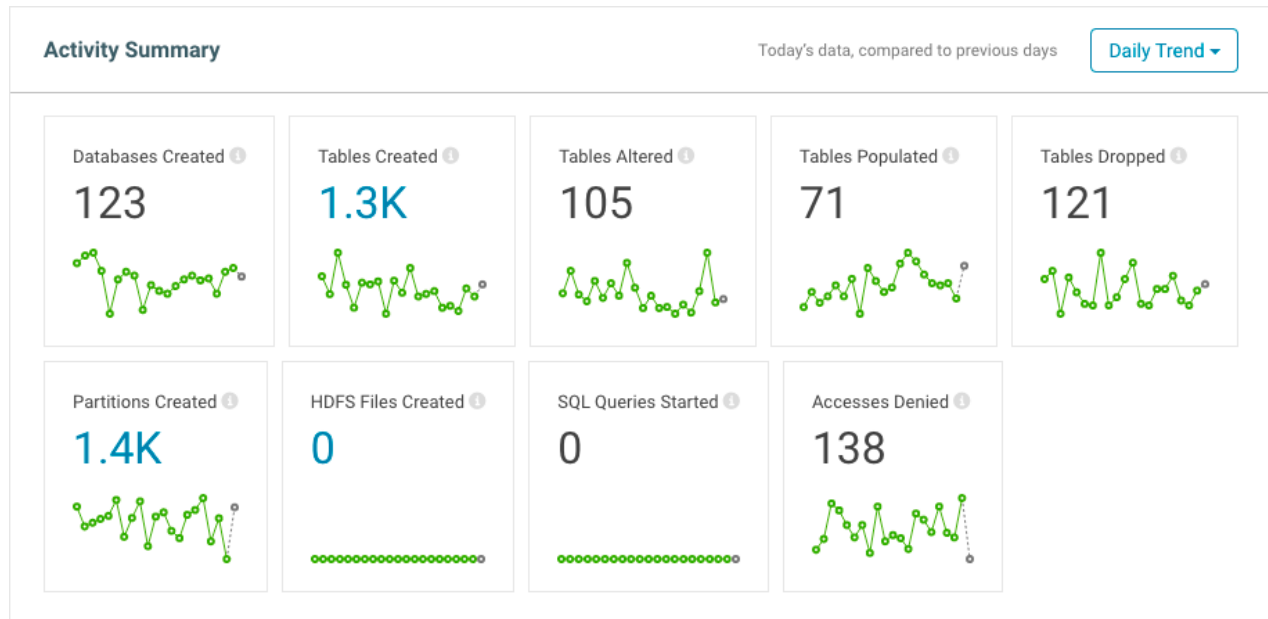
- **Dashboard.** Provides "at-a-glance" information about databases, tables, operations, and files and directories
- **Data Explorer.** Allows you to select various cluster actions to view and compare, for a specified period of time, as well as chart averages and trendlines.

The dashboard is divided into the following major information areas:

- [Activity Summary](#)
- [Databases](#)
- [Hive Tables](#)
- [Files and Directories](#)
- [Operations and Operation Executions](#)

Dashboard

Activity Summary



Each tile in the Activity Summary section provides summary information for actions on a particular entity type and includes the following:

- The name of the activity
- The number of occurrences for that activity for a time period that you select (daily, weekly, monthly, quarterly, all time)
- A line graph showing activity trends based on the time period that you select

A graphical representation of the time-lapse summary for each activity tile is located at the bottom of the tile. Hovering over a point displays the value for that entity on a particular date. For example, if you select Daily Trend, the number in the graph shows number of occurrences for the day so far (since midnight), and hovering over a graph point shows the number of occurrences for that full day as well as the average for the 20-day period represented by the graph.











The Activity Summary area includes the following information:

Databases Created	Number of new databases that were added to the cluster.
Tables Created	Number of new tables that were added to the cluster. Click the value to link to the Search page that shows the search results of the query defined. You can apply filters to narrow the search results and perform any other search actions.
Tables Altered	Number of tables that were changed.
Tables Populated	Number of tables that were populated with data.
Tables Dropped	Number of tables that were deleted.
Partitions Created	Number of partitions added. You can apply filters to narrow the search results and perform any other search actions.
HDFS Files Created	Files that were created. Click the new files value to link the Search page that shows the results of the query defined. You can apply filters to narrow the search results and perform any other search actions.
SQL Queries Started	Number of SQL queries that were run.
Accesses Denied	Number of access attempts by users that were denied.

Databases

114 Databases

Top Databases by Table Count

	Database	Table Count	
1	clusterstats	2K	
2	netsuite	1.2K	
3	support	1K	
4	jira	626	
5	sfdc	434	
6	warehouse_main	374	
7	moodle	340	
8	as_adventure	258	
9	default	167	
10	w_house_satellite	141	

The Databases area of the Dashboard shows the total number of databases in the source clusters. The top 10 databases, by table count, are displayed in the bar graph.

Click the value next to the Database heading (in this case, 1.5K) to open a Search page showing results from the query "Deleted=Do not show deleted" "Type=Database". Apply or remove filters to refine or broaden the results.

Search Actions ▾

Filters Add Filters Clear All Filters

Add Filter...

DELETED ×

- Show Deleted only
- Do not show Deleted
- Show when there is a value for Deleted

SOURCE TYPE

- HDFS 972
- Hive 259
 - STARTED
 - ENDED
- Impala 103
- S3 21
- Pig 18
- YARN 14
- Spark 12
- Oozie 3

TYPE

OWNER

CLUSTER NAME

259 results Show full query

Deleted = Do not show Deleted Source Type = Hive

	Hive title	Type Field	Data Type string	Parent Path /movieLens/movies	Source HIVE-1	View in Hue
	Hive genres	Type Field	Data Type array<string>	Parent Path /movieLens/movies	Source HIVE-1	View in Hue
	Hive occupations	Type Table	Parent Path /movieLens	Path s3a://cloudera-nav-demo/movieLens/occupations	Owner admin	View in Hue
		Created May 10, 2017 10:29 AM	Source HIVE-1			
	Hive id	Type Field	Data Type int	Parent Path /movieLens/occupations	Source HIVE-1	View in Hue
	Hive occupation	Type Field	Data Type string	Parent Path /movieLens/occupations	Source HIVE-1	View in Hue
	Hive ratings	Type Table	Parent Path /movieLens	Path s3a://cloudera-nav-demo/movieLens/ratings	Owner admin	View in Hue
		Created May 10, 2017 10:29 AM	Source HIVE-1			
	Hive userid	Type Field	Data Type int	Parent Path /movieLens/ratings	Source HIVE-1	View in Hue
	Hive movielid	Type Field	Data Type int	Parent Path /movieLens/ratings	Source HIVE-1	View in Hue
	Hive rating	Type Field	Data Type int	Parent Path /movieLens/ratings	Source HIVE-1	View in Hue
	Hive tstamp	Type Field	Data Type string	Parent Path /movieLens/ratings	Source HIVE-1	View in Hue

Hover over the bar in the graph to see information about that database, and click the bar to open the Details page for that database. The following figure shows the Details page for the database **nav_policy_db**.

movielens Actions ▾ Details

hdfs://node-5.example.com:7187/user/hive/warehouse/movielens.db

Database
Hive

4 Tables

0 Views

▼ Technical Metadata

Source Type	Hive
Type	Database
Parent	(no parent)
Path	hdfs://node-5.example.com:7187/user/hive/warehouse/movielens.db
Source	HIVE-1
Classname	Hive Database
Package Name	nav

Description Add

Tags Add

Managed Metadata Add

Custom Metadata Add

▼ Tables (4)

- [movies](#)
- [occupations](#)
- [ratings](#)
- [users](#)

► Views (0)

Hive Tables

8.2K Hive Tables and Views

Top Tables by Partition Count

	Table	Database	Partition Count	
1	customer_logs	default	120.8K	
2	metrics_production	default	98.1K	
3	usage_scrapes	product_usage	77.7K	
4	metrics_staging	default	23.1K	
5	clusterid_mostrecentcollection...	u_mulyadi	10.4K	
6	opportunity_history	sfdc	7.7K	
7	usage_final	product_usage	7K	
8	dlstats_parquet	default	6.2K	
9	case_history	sfdc	5.2K	
10	jira__c_history	sfdc	4.9K	

The Tables area of the dashboard shows the total number of Hive tables in the cluster. The top 10 tables, by partition count, are displayed in the bar graph.

Click the value next to the Hive Tables heading (in this case, 103.9K) to view matching tables in Search.

Click the bar to open the Details page for that table.

Files and Directories

802 Files 168 Directories

Top by Size

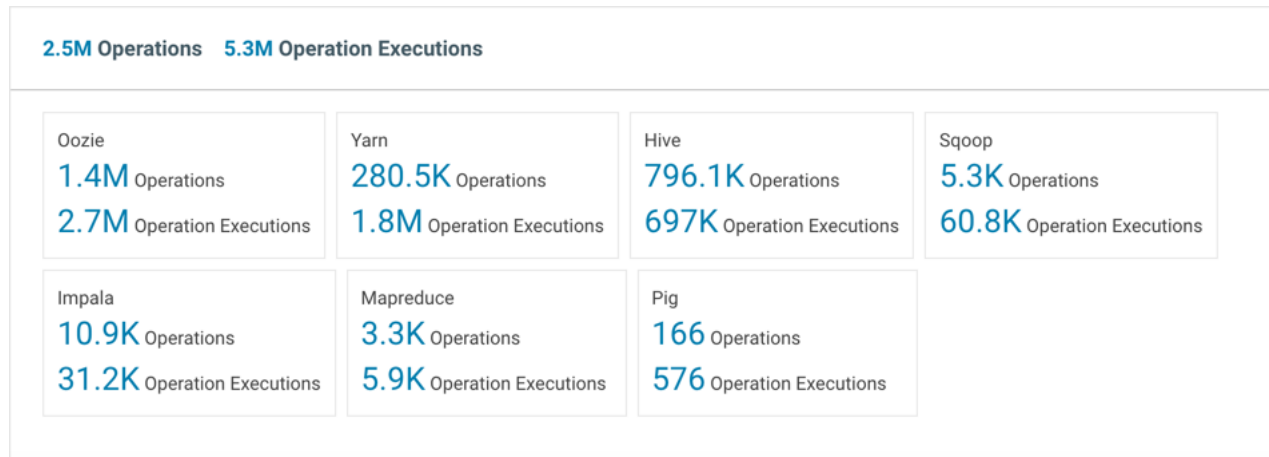
	File	Path	Size	
1	access.log	/dualcore/web_logs	106.3M	
2	ad_data1.txt	/dualcore	31.5M	
3	part-m-00000	/dualcore/ad_data1	28.1M	
4	ad_data2.txt	/dualcore	23.8M	
5	part-r-00000	/dualcore/ad_data2	22.4M	
6	hive-exec.jar	/user/oozie/share/lib/lib_20170509205421/hive	19.5M	
7	hive-exec.jar	/user/oozie/share/lib/lib_20170509205421/hive2	19.5M	
8	hive-exec.jar	/user/oozie/share/lib/lib_20170509205421/spark	19.5M	
9	hive-exec.jar	/user/oozie/share/lib/lib_20170509205421/sqoop	19.5M	
10	part-m-00001	/dualcore/orders	15.8M	

The Files and Directories area of the Dashboard shows the total number of files and directories in the cluster.

Clicking the value next to the Files or Directory heading (in this case, 64.6M or 16.2M, respectively) to show matching files or directories in Search.

The bar graph displays the top 20 files, based on size. Hover over the bar in the graph to see information about that file. Click the bar to open the Details page for that file.

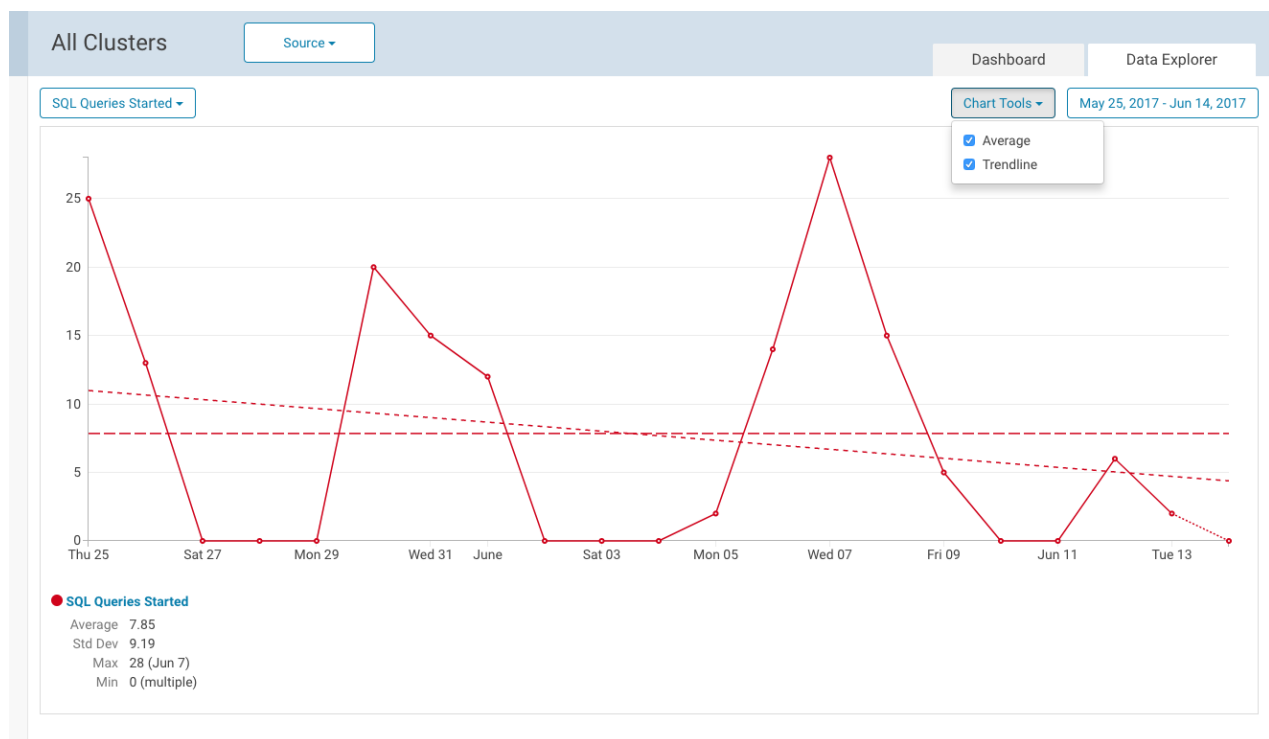
Operations and Operation Executions



The Operations and Operation Executions area of the Dashboard shows the total number of operations and operation executions that occurred in the cluster for the specified period of time.

Click the value next to the Operations or Operations Executions for a service to view matching operations or operation executions in Search.

Data Explorer



Cloudera Navigator and the Cloud

Market-leading organizations today are deploying clusters to the cloud to increase efficiency and reduce costs. Using public cloud infrastructure lets companies create optimal solutions that combine both on-premises and cloud clusters in persistent or transient deployment model. With Cloudera Altus, transient clusters are provisioned quickly, expand and shrink in response to varying workloads, and can be terminated just as easily when no longer needed. See [Cloudera Enterprise in the Cloud](#) for an overview.

In addition to full-scale cluster deployments to the cloud, on-premises Cloudera clusters can leverage cloud resources, such as Amazon Simple Storage Service (S3). For example, an Amazon S3 storage bucket can be used as the source or target for BDR (backup and disaster recovery), enabling organizations to replicate HDFS files and Hive data to and from the S3 bucket. As another example, an S3 bucket is often used as a persistent data store, with data copied to HDFS on a transient cluster for interactive or iterative workloads. Other supported cloud-based functionality includes using Amazon Simple Queue Service (SQS) to support notification from changes to Amazon S3 bucket objects.

Cloudera Navigator can collect metadata from clusters running on Cloudera Altus and from objects stored on Amazon S3. That means that the entities resulting from Hive jobs, for example, running on clusters instantiated by Cloudera Altus users can be seen in lineage diagrams in a centralized on-premises Cloudera Navigator instance running on a persistent or long-running Cloudera Manager cluster.

The screenshot displays the Cloudera Navigator interface for the table `hotel_cust_books_props`. The main area shows a lineage diagram with nodes for `hotel_customers`, `hotel_bookings`, `hotel_cust_bookings`, and `hotel_cust_books_props`. The `hotel_cust_books_props` node is highlighted with a dashed box. On the right, the 'Lineage Options' panel is expanded, showing options for 'Operations', 'Control Flow Relations', 'Only Upstream', 'Downstream', 'Latest Partition and Operation Execution' (which is checked), and 'Deleted Entities'. Below this is a search bar and a 'Data Flow Summary' section showing the selected relation and the operation 'insert into table hotel_cust_bo...' performed by the cluster 'cloudera-eng-navigator-hive-cca-altu'.



Note: The Cloudera Navigator instance runs in an on-premises or persistent cluster.

This chapter provides the details for setting up Cloudera Navigator to extract from cloud-based resources, including Cloudera Altus clusters and Amazon S3 buckets. After configuring support, data stewards, data engineers, and others use the Cloudera Navigator console to explore metadata, relationships, and trace data lineage to its source as with any other type of entity.

Using Cloudera Navigator with Altus Clusters

Cloudera Navigator can extract metadata and lineage from Cloudera Altus clusters. That means that data engineers running jobs on transient clusters and processing jobs using Hive on Altus clusters can view metadata and lineage in

Cloudera Navigator. Because Cloudera Navigator identifies `cluster`, `cluster group`, and other properties related to Altus deployed clusters, data engineers can use the lineage diagrams available in the Cloudera Navigator console to trace data back to its source, even when data processing spans different clusters.

Support for Cloudera Altus metadata and lineage extraction from Cloudera Navigator is not enabled by default. Configuration tasks must be completed in the Cloudera Altus console and in the Cloudera Manager Admin Console to enable this capability. Completing the configuration tasks requires your organization to have a Cloudera Altus account, and assume that you are successfully running jobs on clusters according to the processes detailed in the [Cloudera Altus documentation](#).



Note: Cloudera Altus requires an account on the cloud service provider. For more information about Cloudera Altus and for a demonstration, visit [Introducing Cloudera Altus](#). To request a Cloudera Altus account, go to the [Altus registration page](#).

How it Works: Background to the Setup Tasks

Every cluster deployed using Cloudera Altus is deployed as a Cloudera Manager cluster, that is, it has its own Cloudera Manager Server. This Cloudera Manager Server instance is read-only, but an Altus user—for example, a data engineer developing a new Hive or MapReduce2 script—can use Cloudera Manager to view execution and other details. This Cloudera Manager Server also includes the Telemetry Publisher role instance (daemon) that publishes data from the transient cluster to other systems. The Telemetry Publisher stores metadata and lineage details from the running transient cluster to a cloud-based storage location (for example, an Amazon S3 bucket) specified in the Altus Environment used to instantiate the cluster.

It is the Telemetry Publisher role instance running in the Cloudera Manager Server of a Cloudera Altus cluster and the cloud-based storage mechanism that enables Cloudera Navigator to obtain metadata and lineage information from single-user transient clusters. Before the cluster shuts down, an internal process checks to ensure that all metadata and lineage from that transient cluster has been successfully published to the specified cloud-based storage resource.

Meanwhile, the on-premises Cloudera Navigator instance has been collecting the metadata and lineage information by reading from the same cloud-based storage location. Because cloud storage is a persistent store that runs independently and is not part of the transient cluster, it remains a viable source of data for Cloudera Navigator.

These system interactions are all but transparent to both Cloudera Altus users and Cloudera Navigator data stewards and other users. However, enabling the interaction involves several preliminary tasks across both Cloudera Altus and Cloudera Navigator. The specifics vary, depending on the cloud provider.

Configuring Extraction for Altus Clusters on AWS

Follow the steps below to configure Cloudera Navigator to extract metadata and lineage from single-user transient clusters deployed to Amazon Web Services using Cloudera Altus. The Cloudera Navigator extraction process for clusters launched by Cloudera Altus works as follows:

- Any HDFS paths in a job, query, or data entity are extracted as proxy entities for the path, similar to how Hive entities are extracted. That means that HDFS is not bulk extracted from an Altus cluster.
- Hive Metastore (HMS) entities are also not bulk extracted. Cloudera Navigator extracts Hive entities used in queries that generate lineage, such as databases, tables, and so on.

Requirements

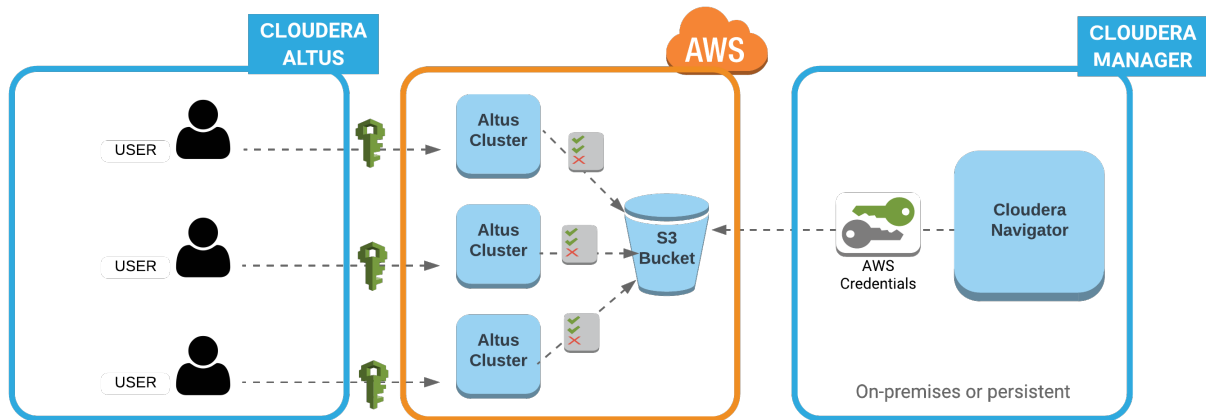
Cloudera Navigator collects metadata and lineage entities from transient clusters deployed to AWS by Cloudera Altus users. The metadata and lineage data is not collected directly from the transient clusters but rather from an Amazon S3 bucket that serves as the storage mechanism for the Telemetry Publisher running in the cluster (see [How it Works: Background to the Setup Tasks](#) on page 15 for details).


As detailed in the steps below, successful integration of Cloudera Navigator and Cloudera Altus clusters depends on correctly:


- Identifying the Amazon S3 bucket, and

Cloudera Navigator and the Cloud

- Configuring correct access permissions to it from both Cloudera Altus and Cloudera Navigator. Transient clusters instantiated by Altus users must have read and write permissions to the Amazon S3 bucket used by Telemetry Publisher. The on-premises centralized Cloudera Navigator instance must have read permissions on the same Amazon S3 bucket.



 — Altus user account (with cross-account privileges to AWS account). Privileges to launch EC2 clusters and use other AWS resources, including Amazon S3 buckets identified in the Environment (data input, data output, logs, and the bucket for Telemetry Publisher).

 — Read and write privileges to the Amazon S3 bucket configured in the Environment for the Altus user.

 -- AWS access key ID and AWS secret key for the AWS account associated with the Amazon S3 bucket.

The steps below assume that you have:

- An Amazon Web Services account.
- A Cloudera Altus account.
- A Cloudera Altus user account that can run jobs on transient clusters deployed to AWS.
- Access to the on-premises or persistent Cloudera Manager cluster running Cloudera Navigator. The Cloudera Manager user role of Full Administrator and the ability to log in to the Cloudera Manager Admin Console is required.
- AWS Credentials for the AWS account hosting the Amazon S3 bucket that serves as the storage mechanism for metadata and lineage data from clusters on AWS launched by Cloudera Altus.

Obtaining AWS Credentials for the Amazon S3 Bucket

AWS Credentials are available to be downloaded whenever you create an IAM user account through the AWS Management Console. If you are configuring an existing Amazon S3 bucket and you do not have the AWS Credentials for it, you can generate new AWS Credentials from the AWS account using either the AWS Management Console or the AWS CLI.



Important: The AWS credentials must have read and write access to the Amazon S3 bucket.

Generating new AWS Credentials deactivates any previously issued credentials and makes the newly generated credentials **Active** for the AWS account. Keep that in mind if you obtain new AWS Credentials to use for the Cloudera Navigator-Cloudera Altus integration.



Note: If you have the AWS Credentials obtained when the account was created, do not regenerate a new set of AWS Credentials unless you want to change the credentials.

These steps assume you have an AWS account and that an Amazon S3 bucket exists on that account that you want to use as the storage location for metadata and lineage.

- Log in to the [AWS Management Console](#) using the account associated with the Amazon S3 bucket.
- Navigate to the **Security credentials** section of the Users page in IAM for this account. For example:

The screenshot shows the AWS IAM console interface. On the left is a navigation menu with options like Dashboard, Groups, Users, Roles, Policies, etc. The main content area is titled 'Summary' for the user 'cust-input'. It displays details such as User ARN, Path, and Creation time. Below this, there are tabs for Permissions, Groups (1), Security credentials (selected), and Access Advisor. The 'Security credentials' section is expanded to show 'Sign-in credentials' with fields for Console password (Enabled), Console login link, Last login, Assigned MFA device, and Signing certificates. Below that is the 'Access keys' section, which includes a 'Create access key' button and a table of existing access keys. The table has columns for Access key ID, Created, Last used, and Status. One access key is listed with ID AKIAISWGLTGFN2ORDH4A, created on 2017-09-13 10:30 PDT, last used on 2017-09-14 12:19 PDT with iam in us-east-1, and is currently Active.

- Click the **Create access key** button to generate new AWS Credentials. Extract the credentials (the Access Key ID and Secret Key) from the user interface or download the `credentials.csv` for later use.

New credentials can be created by using the AWS CLI rather than the AWS Management Console. See [Amazon documentation](#) for details.

Cloudera Altus Configuration

Cloudera Altus instantiates single-user transient clusters focused on data engineering workloads that use compute services such as Hive, Impala, or MapReduce2. The typical deployment scenario involves running scripts that invoke the Cloudera Altus CLI to instantiate the cluster, in this case, using Amazon Web Services according to the details specified in the Altus Environment. An Altus environment specifies all resources needed by the cluster, including the AWS account that will be used to instantiate the cluster. The Cloudera Altus user account is configured to provide cross-account access to the AWS account that has permissions to launch AWS Elastic Compute Cloud (EC2) instances and use other AWS resources, including Amazon S3 buckets.

Although the Altus Environment can be created using Quickstart, Cloudera recommends using the [Environment Wizard](#) or the Altus CLI instead. The Wizard provides better control over configuring resources, including letting you specify the Amazon S3 bucket that clusters will use to store metadata and lineage information for collection by Cloudera Navigator. Specifically, the **Resources & Policies** page of the Configuration Wizard lets you enable integration with Cloudera Navigator and specify the Amazon S3 bucket that will hold collected metadata and lineage information.

On the **Resources & Policies** page of the Configuration Wizard, under the **EC2 Instance Settings**:

- Click the **Enable Cloudera Navigator** checkbox to enable the feature.
- In the **Cloudera Navigator S3 Data Bucket** field, enter the path to the Amazon S3 bucket, including the final `/`, which identifies the target as an S3 bucket. For example:

```
s3a://cluster-lab.example.com/cust-input/
```

To provide the correct access to the S3 bucket, you must also create the appropriate policy in the [AWS Management Console](#) and apply the policy to the Amazon S3 bucket. See [Cloudera Altus documentation](#) for details.

Cloudera Navigator Configuration

The Cloudera Navigator runs in the context of Cloudera Manager Server. Its two role instances, the Navigator Audit Server and Navigator Metadata Server, run on the Cloudera Management Service. The Navigator Metadata Server role instance is the component that extracts metadata and lineage from the Amazon S3 bucket using the AWS Credentials configured for connectivity in the steps below:

- Follow the steps in [Adding AWS Credentials and Configuring Connectivity](#) on page 18 to add new or regenerated AWS Credentials to the Cloudera Manager Server and then configure connectivity.
- Follow the steps in [Configuring Connectivity for AWS Credentials](#) on page 19 to configure connectivity for AWS Credentials that are already available to be used for the Amazon S3 bucket but have not yet been configured for connectivity.



Important: Cloudera Navigator extracts metadata and lineage for clusters deployed using Altus from one Amazon S3 bucket only. In addition, for any given Amazon S3 bucket collecting metadata and lineage from Altus clusters, configure only one Cloudera Navigator instance to extract from that Amazon S3 bucket. Using multiple Cloudera Navigator instances to extract from the same Amazon S3 bucket is not supported and has unpredictable results.

Adding AWS Credentials and Configuring Connectivity

Cloudera Manager Required Role: [Full Administrator](#)

The AWS Credentials must be added to the Cloudera Manager Server for use by Cloudera Navigator. These credentials must be from the AWS account hosting the Amazon S3 bucket that is configured in the [Altus environment](#).



Note: The AWS account associated with these credentials must have cross-account access permissions from the Altus user account that will launch clusters on AWS and run jobs. These credentials must also have read and write permissions on the S3 bucket because the clusters launched must be able to write metadata and lineage information to the Amazon S3 bucket as jobs run.

1. Log in to the Cloudera Manager Admin Console.
2. Select **Administration > AWS Credentials**.
3. Click the **Add Access Key Credentials** button on the AWS Credentials page.
 - a. Enter a meaningful name for the AWS Credential, such as the type of jobs the associated clusters will run (for example, **etl-processing**). This name is for your own information and is not checked against any Cloudera Altus or AWS attributes.
 - b. Enter the AWS Access Key ID and the AWS Secret Key.

Add Access Key Credentials

Name *

Enter a friendly name to identify this credential.

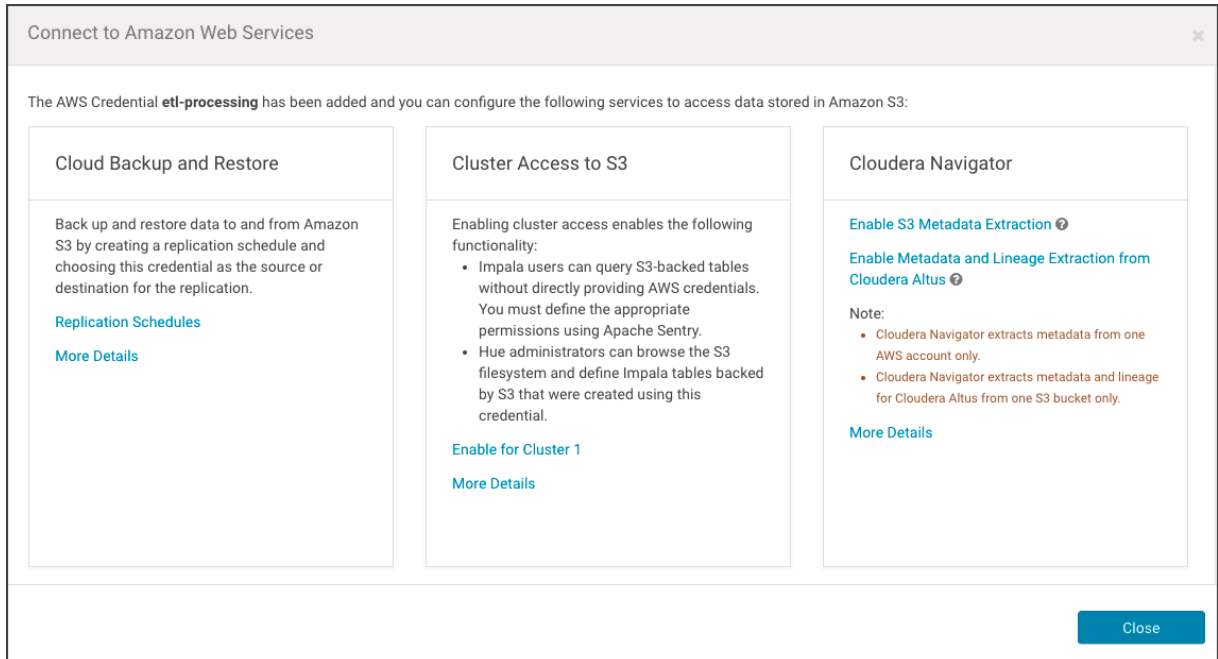
AWS Access Key ID *

AWS Secret Key *

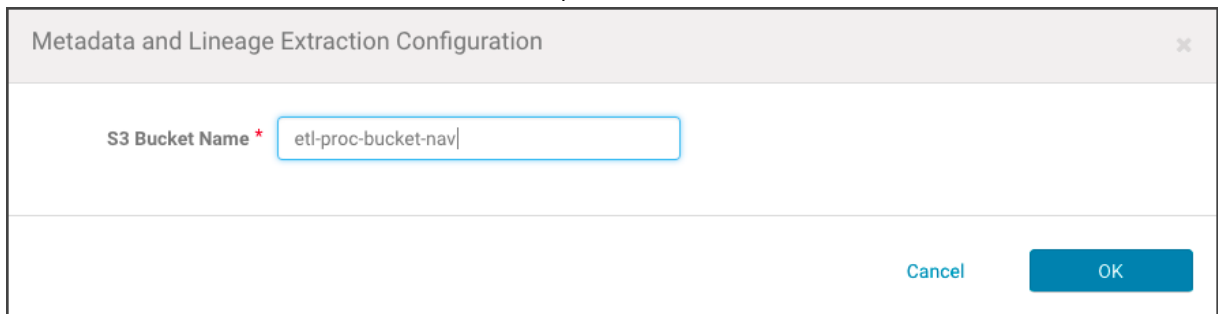
Cancel

4. Click **Add** to save the credentials. The S3Guard option page displays, reflecting the credential name (for example, **Edit S3Guard: etl-processing**). Disregard this option.

- Click **Save**. The Connect to Amazon Web Services page displays, showing the options available for this specific AWS credential.



- Click **Enable Metadata and Lineage Extraction from Cloudera Altus**. The Metadata and Lineage Extraction Configuration setting page displays a field for specifying the Amazon S3 bucket name.
- Enter the name of the Amazon S3 bucket. For example:



- Click **OK**. The AWS Credentials page re-displays, and the newly added AWS Credential is listed with any other AWS Credentials held by the Cloudera Manager Server.
- Restart the Cloudera Management Service.

When the service restarts, the AWS credentials will be used by Cloudera Navigator to authenticate to the AWS account and extract metadata and lineage stored on the specified S3 bucket.

Configuring Connectivity for AWS Credentials

If the AWS Credentials are available for the Amazon S3 bucket, you can configure them as follows:

- Log in to the Cloudera Manager Admin Console.
- Select **Administration > AWS Credentials**.
- Find the available AWS Credentials that provide access to the Amazon S3 bucket used to collect metadata and lineage from transient clusters.

AWS Credentials					
<p>AWS Credentials allow CDH services and Cloudera tools to securely query data, browse data, backup and restore data/metadata, search metadata and view data lineage of data in Amazon S3. More Details</p> <p> Add Access Key Credentials Add IAM Role-based Authentication </p>					
Name	Type	Connectivity	Creation	Last Modified	Actions
data-eng-s3-nav-bucket	Access Key Credentials		September 15, 2017 8:14 PM	September 15, 2017 8:14 PM	Actions
cust-data	Access Key Credentials		September 15, 2017 8:15 PM	September 15, 2017 8:15 PM	Actions
offside-hdfs-backup	Access Key Credentials	Cloudera Navigator	September 15, 2017 8:15 PM	September 15, 2017 8:15 PM	Actions

4. Click the **Actions** drop-down menu and select **Edit Connectivity**. The Connect to Amazon Web Services page displays the three sections of possible configurations.
5. In the Cloudera Navigator section, click the **Enable Metadata and Lineage Extraction from Cloudera Altus** link. The Metadata and Lineage Extraction Configuration page displays.
6. Enter the name of the Amazon S3 bucket in the **S3 Bucket Name** field.
7. Click **OK**.
8. Restart the Cloudera Management Service.

This completes the setup process. After the restart, metadata and lineage for transient clusters deployed using Cloudera Altus should be available in the Cloudera Navigator console.

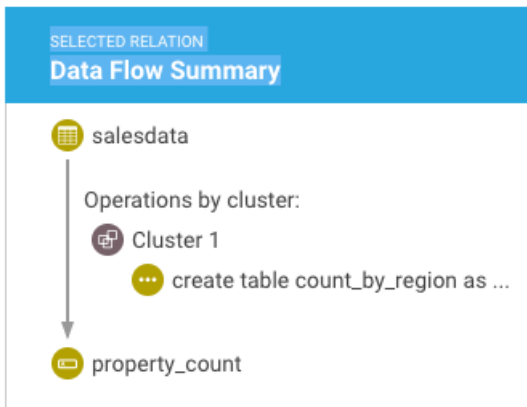


Note: See Troubleshooting to identify possible issues if metadata and lineage do not display in the Cloudera Navigator console after completing the configuration and restarting the system.

Technical metadata specific to clusters deployed using Altus include the following property names and types:

- Cluster (Source Type)
- Cluster-*name* (Cluster Group)
- Transient (Deployment Type)
- Cluster Template, Cluster Instance (Classname)

For example:



See [Search: Syntax and Properties Reference](#) on page 96 for more information.

Configuring Extraction for Amazon S3

Depending on the specifics of the Amazon S3 bucket targeted for extraction by Cloudera Navigator, the configuration process follows one of two alternative paths:

- **Default configuration**—The default configuration is available for Amazon S3 buckets that have no existing Amazon SQS or Amazon SNS services configured. During configuration, Cloudera Navigator accesses the configured AWS

account, performs an initial bulk extract from the Amazon S3 bucket, sets up Amazon SQS queues in each region with buckets, and sets up event notifications for each bucket for subsequent incremental extracts—all transparently to the Cloudera Manager administrator handling the configuration process.

- **Custom configuration**—Custom configuration is required for any Amazon S3 bucket that is currently using Amazon SQS (has queues set up for other applications, for example) or is setup for notifications using Amazon SNS. In these cases you must manually configure a new queue—**bring your own queue**—and in some cases, additionally configure Amazon SNS for *fanout*.

Continue reading:

- [AWS Credentials Requirements](#) on page 21
- [Default Configuration](#) on page 21
- [Custom Configurations](#) on page 23
 - [Configuring Your Own Queues](#) on page 23
 - [Configure the Queue for Cloudera Navigator](#) on page 24
 - [Configure Event Notification for the Queues](#) on page 25
 - [Configuring Amazon SNS Fan-out](#) on page 25
 - [Defining and Attaching Policies](#) on page 26
 - [Event Notification Policy for Custom Queues](#) on page 26
 - [Extraction Policies for Custom Queues](#) on page 26
 - [Extraction Policies JSON Reference](#) on page 27
 - [Setting Properties with Advanced Configuration Snippets](#) on page 27
 - [Cloudera Navigator Properties for Amazon S3](#) on page 28

AWS Credentials Requirements

Cloudera Manager can have multiple AWS credentials configured for various purposes at any given time. These are listed by name on the AWS Credentials page which is accessible from the Cloudera Manager Admin Console, under the Administration menu. However, there are specific constraints on AWS credentials for Cloudera Navigator as follows:

- Navigator supports a single key for authentication; only one AWS credential can be used for a given Navigator instance. Navigator can extract metadata from any number of S3 buckets, assuming the buckets can be accessed with the configured credential.
- An AWS credential configured for connectivity from one Cloudera Navigator instance cannot be used by another Cloudera Navigator instance. Configuring the same AWS credentials for use with different Cloudera Navigator instances can result in unpredictable behavior.
- Cloudera Navigator requires an AWS credential associated with an IAM *user* identity rather than an IAM *role*.
- Any changes to the AWS credentials (for example, if you rotate credentials on a regular basis) must be for the same AWS account (IAM user). Changing the AWS credentials to those of a different IAM user results in errors from the Amazon Simple Queue Service (used transparently by Cloudera Navigator). If a new key is provided to Cloudera Navigator, the key must belong to the same AWS account as the prior key.
- For the default configuration, the account for this AWS credential must have administrator privileges for:
 - [Amazon S3](#)
 - [Amazon Simple Queue Service](#) (SQS)
- For the custom configurations, the account needs privileges for Amazon S3, Amazon SQS, and for [Amazon Simple Notification Service](#) (SNS).

Default Configuration

The steps below assume that you have the required AWS credentials for the IAM user with the Amazon S3 bucket. Amazon Web Services (AWS) account (an IAM user account) and that you can use the [AWS Management Console](#).

The AWS credentials for the IAM user are configured for Cloudera Navigator using the Cloudera Manager Admin Console during the configuration process below.



Important: If the Amazon S3 bucket is already configured for queuing or notification, do not follow the steps in this section. See [Custom Configurations](#) on page 23 instead.

Configuring Cloudera Navigator for Amazon S3

At the end of the configuration process detailed below, Cloudera Navigator authenticates to AWS using the credentials and performs an initial bulk extract of entities from the Amazon S3 bucket. It also sets up the necessary Amazon SQS queue (or queues, one for each region) to use for subsequent incremental extracts.

The steps below assume you have the [required AWS credentials](#) available.

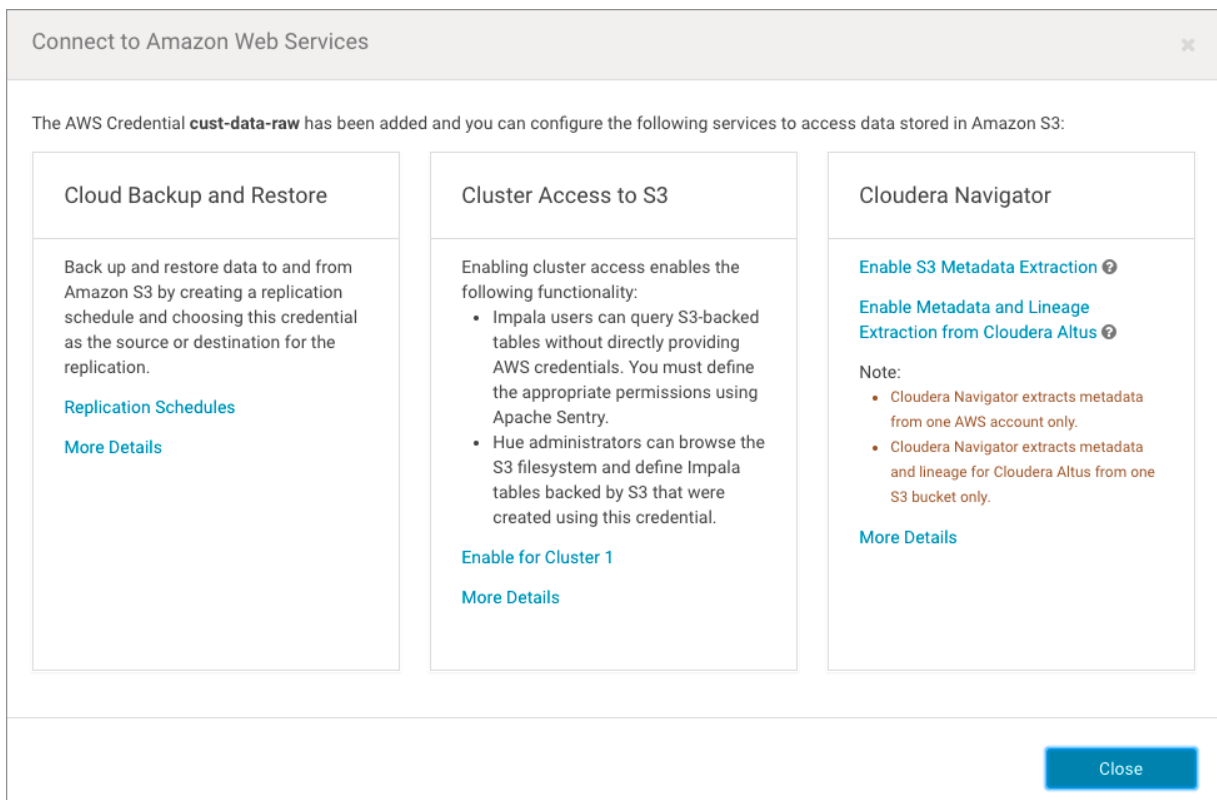
1. Log in to the Cloudera Manager Admin Console.
2. Click **Administration > AWS Credentials**. The AWS Credentials page displays, listing any existing credentials that have been setup for the Cloudera Manager cluster.
3. Click the **Add Access Key Credentials** button. In the Add Access Key Credentials page:
 - a. Enter a **Name** for the credentials. The name can contain alphanumeric characters and can include hyphens, underscores, and spaces but should be meaningful in the context of your production environment. Use the name of the Amazon S3 bucket or its functionality, for example, *cust-data-raw* or *post-proc-results*.
 - b. Enter the **AWS Access Key ID**.
 - c. Enter the **AWS Secret Key**.

4. Click **Add**. The **Edit S3Guard:aws-cred-name** page displays, giving you the option to enable S3Guard for the S3 bucket.
 - Click the Enable S3Guard box only if the AWS credential has privileges on Amazon DynamoDB and if the preliminary S3Guard configuration is complete. See [Cloudera Administration](#) for details about [Configuring and Managing S3Guard](#).
5. Click **Save**.



Note: Cloudera Manager stores the AWS credential securely in a non-world readable location. The access key ID and secret values are masked in the Cloudera Manager Admin Console, encrypted before being passed to other processes, and [redacted](#) in the logfiles.

The Connect to Amazon Web Services page displays the credential name and services available for its use:



- Click the **Enable S3 Metadata Extraction** link in the Cloudera Navigator section of the page. A Confirm prompt displays, notifying you that Cloudera Navigator must be manually restarted after this change.
- Click **OK** to enable the connection.
- At the top of the Cloudera Manager Admin Console, click the **Stale Configuration** restart button when you are ready to restart Cloudera Navigator.

Metadata and lineage for Amazon S3 buckets will be available in the Cloudera Navigator console along with other sources, such as HDFS, Hive, and so on. It may take several minutes to complete the initial extraction depending on the number of objects stored on the Amazon S3 bucket.

Custom Configurations

Follow these steps for Amazon S3 buckets that are already configured with queues or event notifications. Custom configurations include configuring your own queue (BYOQ) and BYOQ with Fan-out, as detailed below.

Configuring Your Own Queues

Sometimes referred to as Bring Your Own Queue (BYOQ), configuring your own queue is required if the Amazon S3 bucket being targeted for extraction by Cloudera Navigator already has existing queues or is configured for notifications. The process involves stopping Cloudera Navigator and then using the AWS Management Console for the following tasks:

- Creating and configuring an Amazon Simple Queue Service (SQS) queue for Cloudera Navigator for each region in which the AWS (IAM user) account has Amazon S3 buckets.
- Configuring Amazon Simple Notification Service (SNS) on each bucket to send Create, Rename, Update, Delete (CRUD) events to the Cloudera Navigator queue.
- Configuring the bucket for Notification Fan-Out if needed to support existing notifications configured for other applications.
- Adding a Policy for the appropriate extraction process (Bulk + Incremental, Bulk Only) to the IAM user account.
- Adding the Policy for event notifications to the IAM user account.



Important: Always make sure any newly created Amazon S3 buckets are configured for event notifications before adding data so the queues are properly updated.

Configure the Queue for Cloudera Navigator

This manual configuration process requires stopping Cloudera Navigator. You must create a queue for each region that has S3 buckets.

1. Log in to Cloudera Manager Admin Console and stop Cloudera Navigator:
 - Select **Clusters > Cloudera Management Service**
 - Click the **Instances** tab.
 - Click the checkbox next to **Navigator Audit Server** and **Navigator Metadata Server** in the Role Type list to select these roles.
 - From the **Actions for Selected (2)** menu button, select **Stop**
2. Log in to the [AWS Management Console](#) with AWS account (IAM user) and open the **Simple Queue Service** setup page (select **Services > Messaging > Simple Queue Service**. Click **Create New Queue** or **Get Started Now** if region has no configured queues.)
3. For each region that has Amazon S3 buckets, create a queue as follows:

- a. Click the **Create New Queue** button. Enter a Queue Name, click the Standard Queue (not FIFO), and then click **Configure Queue**. Configure the queue using the following settings:

Default Visibility Timeout	10 minutes
Message Retention Period	14 days
Delivery Delay	0 seconds
Receive Message Wait Time	0 seconds

- b. Select the queue you created, click the **Permissions** tab, click **Add a Permission**, and configure the following in the **Add a Permission to...** dialog box:

Effect	Allow
Principal	Everybody
Actions	SendMessage

- c. Click the **Add Conditions (optional)** link open the condition fields and enter the following values:

Qualifier	None
Condition	ArnLike
Key	aws:SourceArn
Value	arn:aws:s3::*:*

- d. Click **Add Condition** to save the settings.
- e. Click **Add Permission** to save all settings for the queue.

1 SQS Queue selected

Details | Permissions | Redrive Policy | Monitoring | Encryption

Name: NavBYOQ_Eastern
URL: https://sqs.us-east-1.amazonaws.com/141229114088/NavBYOQ_Eastern
ARN: arn:aws:sqs:us-east-1:141229114088:NavBYOQ_Eastern
Created: 2017-09-19 20:24:44 GMT-07:00
Last Updated: 2017-09-19 20:24:44 GMT-07:00
Delivery Delay: 0 seconds
Queue Type: Standard
Content-Based Deduplication: N/A

Default Visibility Timeout: 30 seconds
Message Retention Period: 4 days
Maximum Message Size: 256 KB
Receive Message Wait Time: 0 seconds
Messages Available (Visible): 0
Messages in Flight (Not Visible): 0
Messages Delayed: 0

Repeat this process for each region that has Amazon S3 buckets.

Configure Event Notification for the Queues

After creating queues for all regions with Amazon S3 buckets, you must configure [event notification](#) for each Amazon S3 bucket. Assuming you are still logged into the [AWS Management Console](#):

1. Navigate to the Amazon S3 bucket for the region (**Services > Storage > S3**).
2. Select the bucket.
3. Click the **Properties** tab.
4. Click the **Events** settings box.
5. Click **Add notification**.
6. Configure event notification for the bucket as follows:

Name	nav-send-metadata-on-change
Events	<ul style="list-style-type: none"> • ObjectCreated(All) • ObjectRemoved(All)
Send to	SQS Queue
SQS queue	Enter the name of your queue



Important: Cloudera Navigator extracts metadata from one queue only for each region.

Configuring Amazon SNS Fan-out

Configure SNS fanout if you have existing S3 event notification. For more information about SNS fanout, see Amazon documentation for [Common SNS Scenarios](#)

1. Add the following to the **Navigator Metadata Server Advanced Configuration Snippet (Safety Valve) for cloudera-navigator.properties**. See [Setting Properties with Advanced Configuration Snippets](#) on page 27 for details about using Cloudera Manager Admin Console if necessary.

```
nav.s3.extractor.incremental.enable=true
nav.s3.extractor.incremental.auto_setup.enable=false
nav.s3.extractor.incremental.queues=queue_json
```

Specify the queue properties using the following JSON template (without any spaces). Escape commas (,) by preceding them with two backslashes (\,), as shown in the template:

```
[{"region":"us-west-1", "queueUrl":"https://sqs.aws_region.amazonaws.com/account_num/queue_name"}, {queue_2},
...
{queue_n}]
```

2. Restart Cloudera Navigator.

Defining and Attaching Policies

Event Notification Policy for Custom Queues

To enable event notification for an custom queue, create the following policy by copying the policy text and pasting it in the [policy editor](#), and then attaching it to the Cloudera Navigator user in AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1481678612000",
      "Effect": "Allow",
      "Action": [
        "sqs:DeleteMessage",
        "sqs:DeleteMessageBatch",
        "sqs:GetQueueAttributes",
        "sqs:ReceiveMessage"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Stmt1481678744000",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetBucketNotification",
        "s3:PutBucketNotification"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ]
    }
  ]
}
```

Extraction Policies for Custom Queues

Custom configurations require a valid extraction policy be defined and attached to the AWS user account associated with the Amazon S3 bucket. The policy is a JSON document that specifies the type of extraction. As mentioned in [Overview of Amazon S3 Extraction Processes](#) on page 32, the two types of extraction are as follows:

- Bulk + Incremental—This is the recommended approach for both cost and performance reasons and is used by the [Default Configuration](#) process automatically.
- Bulk Only—This approach is recommended for proof-of-concept deployments. It is required for the BYOQ with Fan-out configuration. In addition to applying the policy as detailed below, this approach also requires setting the `nav.s3.extractor.incremental.enable` property to `false`. See [Setting Properties with Advanced Configuration Snippets](#) on page 27 and the [Cloudera Navigator Properties for Amazon S3](#) on page 28 for details.

To configure the policy:

- Log in to the AWS Management Console using the IAM user account associated with the target Amazon S3 bucket.
- Copy the appropriate JSON text from the table below [Extraction Policies JSON Reference](#) on page 27 and paste into the AWS Management Console [policy editor](#) for the Navigator user account on (the IAM user) through the AWS Management Console.

Extraction Policies JSON Reference

Bulk + Incremental (Recommended)	Bulk Only
<ul style="list-style-type: none"> Initial bulk process extracts all metadata. Subsequent incremental process extracts changes only (CRUD). Cannot be used with Amazon S3 buckets that use event notification. 	<ul style="list-style-type: none"> Must be used for Amazon S3 buckets configured for event notifications.
<pre> { "Version": "2012-10-17", "Statement": [{ "Sid": "Stmnt1481678612000", "Effect": "Allow", "Action": ["sqs:CreateQueue", "sqs:DeleteMessage", "sqs:DeleteMessageBatch", "sqs:GetQueueAttributes", "sqs:GetQueueUrl", "sqs:ReceiveMessage", "sqs:SetQueueAttributes"], "Resource": "*" }, { "Sid": "Stmnt1481678744000", "Effect": "Allow", "Action": ["s3:GetBucketLocation", "s3:ListAllMyBuckets", "s3:ListBucket", "s3:GetObject", "s3:GetObjectAcl", "s3:GetBucketNotification", "s3:PutBucketNotification"], "Resource": ["arn:aws:s3:::*"] }] } </pre>	<pre> { "Version": "2012-10-17", "Statement": [{ "Sid": "Stmnt1481676614000", "Effect": "Allow", "Action": ["s3:GetBucketLocation", "s3:ListAllMyBuckets", "s3:ListBucket", "s3:GetObject", "s3:GetObjectAcl"], "Resource": ["arn:aws:s3:::*"] }] } </pre>

Setting Properties with Advanced Configuration Snippets

Certain features require additional settings or changes to the Cloudera Navigator configuration. For example, configuring BYOQ queues to use bulk-only extraction requires not only [creating and attaching the extraction policy](#) but also adding the following snippet to the **Navigator Metadata Server Advanced Configuration Snippet (Safety Valve) for cloudera-navigator.properties** setting:

```
nav.s3.extractor.incremental.enable=false
```

To change property values by adding an advanced configuration snippet:

- Log in to the Cloudera Manager Admin Console.

- Select **Clusters > Cloudera Management Service**.
- Click **Configuration**.
- Click **Navigator Metadata Server** under the Scope filter, and click **Advanced** under the Category filter.
- Enter **Navigator Metadata Server Advanced Configuration Snippet (Safety Valve) for cloudera-navigator.properties** in the Search field to find the property.
- Enter the property and its setting as a key-value pair, for example:

```
property=your_setting
```


in:

- Click **Save Changes**.
- Restart the Navigator Metadata Server instance.

Cloudera Navigator Properties for Amazon S3

The table below lists the Navigator Metadata Server properties (`cloudera-navigator.properties`) that control extraction and other features related to Amazon S3. These properties can be set using the Cloudera Manager Admin Console to set properties in the advanced configuration settings.

Changing any of the values in the table requires a restart of Cloudera Navigator.

Option	Description
<code>nav.aws.api.limit</code>	Default is 5,000,000. Maximum number of Amazon Web Services (AWS) API calls that Cloudera Navigator can make per month.
<code>nav.sqs.max_receive_count</code>	Default is 10. Number of retries for inconsistent SQS messages (inconsistent due to eventual consistency).
<code>nav.s3.extractor.enable</code>	Default is <code>true</code> when an AWS credential has been configured to extract metadata from Amazon S3.
<code>nav.s3.extractor.incremental.auto_setup.enable</code>	Default is <code>true</code> . Enables Cloudera Navigator to set up Amazon SQS queues to receive notifications from Amazon S3 events. Set to <code>false</code> to disable the automatic setup and custom configure your own queue (BYOQ with Fan-out).
<code>nav.s3.extractor.incremental.batch_size</code>	Default is 1000. Number of messages held in memory during the extraction process.
<code>nav.s3.extractor.incremental.enable</code>	Default is <code>true</code> . Enables incremental extraction. Setting to <code>false</code> disables incremental extraction and effectively enables bulk-only extraction.
<code>nav.s3.extractor.incremental.event.override</code>	<p>Default is <code>false</code>. Prevents any existing event notifications from being overwritten by Cloudera Navigator auto-generated queues created during default configuration process.</p> <div style="border: 1px solid yellow; padding: 10px; margin-top: 10px;"> <p> Important: Do not set to <code>true</code> unless you fully understand the impact of overwriting event notifications. Setting to <code>true</code> may overwrite critical existing business logic.</p> </div>
<code>nav.s3.extractor.incremental.queues</code>	No default queue. Used by the custom configuration only. Specify a list of queues for the custom configuration using the JSON template . The list of queues should include the existing queues already in use and the newly configured queue that Cloudera Navigator will use for incremental extracts.
<code>nav.s3.extractor.max_threads</code>	Default is 3. The number of extractors (worker processes) to run in parallel.

Option	Description
<code>nav.s3.home_region</code>	Default is <code>us-west-1</code> . AWS region closest to the cluster and the Cloudera Navigator instance. Select the same AWS region (or the nearest one geographically) to minimize latency for API requests.
<code>nav.s3.implicit.batch_size</code>	Default is <code>1000</code> . Number of Solr documents held in memory when updating the state of implicit directories.

Using Cloudera Navigator with Amazon S3

[Amazon Simple Storage Service \(S3\)](#) is a storage solution offered by Amazon Web Services (AWS) that provides highly available storage in the cloud. Clusters deployed not only in the AWS cloud but also on-premises are using Amazon S3 as persistent storage. Common use cases include BDR (backup and disaster recovery) and persistent storage for transient clusters deployed to the cloud, such as storage for ETL workload input and output.

As with data stored on HDFS and processed using compute engines like Hive and Impala, Cloudera Navigator can obtain metadata and lineage from Amazon S3 storage. There are some limitations and constraints as discussed below, and some setup is required to enable this capability (see [Configuring Extraction for Amazon S3](#) on page 20).



Note: Cloudera Navigator does not audit Amazon S3 storage buckets. Only Cloudera Navigator metadata and lineage are supported.

This section provides conceptual information about Amazon S3 storage and shows you how to configure Cloudera Navigator to extract metadata and lineage from an Amazon S3 bucket.

Continue reading:

- [Amazon S3 Storage Characteristics](#) on page 29
- [Cloudera Navigator and Amazon S3](#) on page 30
 - [Object Lifecycle Rules Constraints](#) on page 30
 - [Amazon SQS and Amazon SNS Constraints](#) on page 30
 - [Object Storage and Implicit Folder Limitation](#) on page 31
- [Overview of Amazon S3 Extraction Processes](#) on page 32
 - [API Usage and Setting Limits](#) on page 32

Amazon S3 Storage Characteristics

Amazon S3 is an object store rather than a file store or block store. It does not have the hierarchy found in typical filesystems. Amazon S3 uses the construct of a [bucket](#) as a container for [objects](#). An object can be any kind of file—text file, image, photo, graphic, video, an ETL bundle to be ingested into a cluster, and so on.

Files can be added to Amazon S3 through the AWS Management Console, by using the AWS CLI, or by using scripts that invoke the CLI.

Amazon S3 storage is highly available because Amazon replicates data across multiple servers within its data centers and uses an [eventual consistency](#) model—not all accesses of an object on Amazon S3 may be reflected concurrently or instantaneously. However, eventually, all updates to data across servers are synchronized. The eventual consistency model can result in a short delay between the time objects are uploaded to Amazon S3 and the time their metadata is available in Cloudera Navigator. This is expected behavior and simply how eventual consistency works.



Note: The eventual consistency model of Amazon S3 can be augmented by using S3Guard, which leverages Amazon DynamoDB to provide support for transactions. See [Configuring and Managing S3Guard](#) in the Cloudera Administration guide for details.

For more information about Amazon S3, see [Amazon S3 documentation](#).

Cloudera Navigator and Amazon S3

Despite the implementation details of Amazon S3 storage, Cloudera Navigator collects metadata for Amazon S3 entities in much the same way as for HDFS entities, with some exceptions shown in the table below.



Note: In addition to metadata, Cloudera Navigator extracts lineage from Hive, Impala, and MapReduce (except for MapReduce Glob paths) on Amazon S3.

The following table lists some differences between object types and supported features offered by Amazon S3 and how those are supported by Cloudera Navigator:

Feature	Amazon S3	Cloudera Navigator
User-defined metadata consists of custom key-value pairs (in which each key is prefixed with <code>x-amz-meta-</code>) that can be used to tag objects on Amazon S3.	✓	⊘
System-defined metadata includes properties such as <code>Date</code> , <code>Content-Length</code> , <code>Last-Modified</code> . Some system-defined properties comprise the Technical Metadata for the object in Cloudera Navigator.	✓	✓
Tags for buckets and objects	✓	⊘
Versioning is not supported. Cloudera Navigator extracts metadata and lineage from the latest version only.	✓	⊘
Unnamed directories	✓	⊘
Object lifecycle rules . See Object Lifecycle Rules Constraints on page 30 for more information.	✓	✓
Amazon Simple Queue Service (SQS) . See Amazon SQS and Amazon SNS Constraints for usage limitations and requirements.	✓	✓
Amazon Simple Notification Service (SNS) . See Amazon SQS and Amazon SNS Constraints for usage limitations and requirements.	✓	✓

Object Lifecycle Rules Constraints

Cloudera Navigator does not support lifecycle rules that remove objects from Amazon S3. For example, an object lifecycle rule that removes objects older than n days deletes the object from Amazon S3 but the event is not tracked by Cloudera Navigator. This limitation applies to removing objects only. Using lifecycle rules requires using bulk-only extraction. See [Custom Configurations](#) on page 23 for details about configuring the necessary AWS Policy and applying it to the Amazon S3 bucket for use by Cloudera Navigator.

Amazon SQS and Amazon SNS Constraints

[Amazon Simple Queue Service \(SQS\)](#) is a distributed, highly scalable hosted queue for storing messages. [Amazon Simple Notification Service \(SNS\)](#) is publish-subscribe notification service that coordinates message delivery. Both services can be configured for use with Amazon S3 storage buckets. For example, Amazon S3 storage buckets can send notification messages to one or more queues or to email addresses whenever specified events occur, such as creating, renaming, updating, or deleting data on the Amazon S3 bucket.

During the [default configuration](#) process, Cloudera Navigator transparently sets up an Amazon SQS queue and configures Amazon S3 event notification for each bucket. The queue is used to hold event messages that are subsequently collected by the [Cloudera Navigator S3 extractor](#) process, for incremental extracts. Use the default configuration process only for Amazon S3 buckets that do not have existing queues or notifications configured.

For Amazon S3 buckets that are already configured for queues, use the [custom configuration](#) process—sometimes referred to as "Bring Your Own Queue" (BYOQ)—to manually configure queues for Cloudera Navigator. For Amazon S3 buckets that are already configured for notifications, use the BYOQ custom configuration in conjunction with Amazon SNS in a fan-out configuration. In a fan-out scenario, an Amazon SNS message is sent to a topic and then replicated

and pushed to multiple Amazon SQS queues, HTTP endpoints, or email addresses. See [Common Amazon SNS Scenarios](#) for more information about fan-out configuration, and see [Custom Configurations](#) on page 23 for details about configuring Cloudera Navigator when the Amazon S3 bucket is already set up for either Amazon SQS or Amazon SNS.

Object Storage and Implicit Folder Limitation

Amazon S3 storage does not use a directory structure or other hierarchy as found in a traditional file system. Each object has an **object key name** that identifies the object by its S3Uri location—the path to the object. This path includes the **object**, **prefix** if any, and **bucket** name. Including the S3 protocol specifier, the pattern is as follows:

```
s3://bucketname/prefix/objectkey
```

There can be more than one prefix in an object key name. Prefixes are separated by the forward slash character (/). Although Amazon S3 provides a **folder** metaphor for organizing objects in an S3 bucket, the folder does not provide actual containment or structure: it is the object key name and its S3Uri location that identifies the object.

Cloudera Navigator mimics file system behavior by mapping elements of the object key name to implicit folders. For example, for an Amazon S3 file with the object key name **2017/08_AUG/data-raw.csv**, Cloudera Navigator creates an entity with the path **2017/08_AUG/data-raw.csv** and also creates two directories: **2017** and **2017/08_AUG**.

Cloudera Navigator	Amazon S3																						
<div> <p>▼ Technical Metadata</p> <table> <tr><td>Source Type</td><td>S3</td></tr> <tr><td>Path</td><td>2017/08_AUG/data-raw.csv</td></tr> <tr><td>Region</td><td>us-east-1</td></tr> <tr><td>Size</td><td>341 B</td></tr> <tr><td>Owner</td><td>cluster-lab</td></tr> <tr><td>Last Modified</td><td>Sep 19, 2017 6:22 PM</td></tr> <tr><td>S3 Storage Class</td><td>STANDARD</td></tr> <tr><td>S3 Etag</td><td>4e928ee8b825e1c0e17035eb9306bf4c</td></tr> <tr><td>Source</td><td>S3</td></tr> <tr><td>Classname</td><td>S3 Object</td></tr> <tr><td>Package Name</td><td>nav</td></tr> </table> </div>	Source Type	S3	Path	2017/08_AUG/data-raw.csv	Region	us-east-1	Size	341 B	Owner	cluster-lab	Last Modified	Sep 19, 2017 6:22 PM	S3 Storage Class	STANDARD	S3 Etag	4e928ee8b825e1c0e17035eb9306bf4c	Source	S3	Classname	S3 Object	Package Name	nav	<div> <p>Amazon S3 > etl-source-data / 2017 / 08_AUG</p> <p>data-raw.csv Latest version ▼</p> <p>Overview Properties Permissions</p> <p>Open Download Download as Make public Copy path</p> <p>Owner cluster-lab</p> <p>Last modified Sep 19, 2017 6:22:44 PM</p> <p>Etag 4e928ee8b825e1c0e17035eb9306bf4c</p> <p>Storage class Standard</p> <p>Server side encryption None</p> <p>Size 341</p> <p>Link https://s3.amazonaws.com/etl-source-data/2017/08_AUG/data-raw.csv</p> </div>
Source Type	S3																						
Path	2017/08_AUG/data-raw.csv																						
Region	us-east-1																						
Size	341 B																						
Owner	cluster-lab																						
Last Modified	Sep 19, 2017 6:22 PM																						
S3 Storage Class	STANDARD																						
S3 Etag	4e928ee8b825e1c0e17035eb9306bf4c																						
Source	S3																						
Classname	S3 Object																						
Package Name	nav																						

Cloudera Navigator console Lineage tab for the file with object key `2017/08_AUG/data-raw.csv` shows it in the context of implicit folders:

data-raw.csv
2017/08_AUG/data-raw.csv

etl-source-data

- 2017
 - 08_AUG
 - data-raw.csv

Cloudera Navigator has some limitations specifically for deleted objects and implicit folders as follows:

- Cloudera Navigator does not mark an implicit folder as deleted even after all its child objects have been deleted.
- Cloudera Navigator does not mark as deleted any objects and folders deleted using Amazon S3 tools, such as the AWS CLI (`aws s3` commands) or the AWS Management Console.



Note: To filter out implicit folders from the S3 entities displayed, enter `implicit:false` in the Search field. Conversely, to find implicit entities enter `implicit:true` in the Search field.

For more details about the properties shown by Cloudera Navigator, see [S3 Properties](#) on page 101.

Despite the differences between an object store and a hierarchical store, data engineers can work with Amazon S3 using the Cloudera Navigator in much the same way as for HDFS and other entities.

Overview of Amazon S3 Extraction Processes

By default, Cloudera Navigator uses combined bulk and incremental extraction processes. An initial bulk process extracts all metadata from an Amazon S3 bucket during the configuration process. Subsequent extracts are incremental. Changes are collected from an Amazon SQS queue created by Cloudera Navigator during the [default configuration](#) process.

This bulk-plus-incremental extraction combination provides the optimal performance for production systems and is also the most cost-effective in terms of Amazon API usage:

- For the bulk extract, Cloudera Navigator invokes the Amazon S3 API.
- For the incremental extract, Cloudera Navigator invokes the Amazon SQS API.

Amazon meters usage and charges differently for each of these APIs.

API Usage and Setting Limits

Amazon bills on a monthly basis and resets the billing cycle each month. To help manage the monthly cost of using these APIs, Cloudera Navigator provides a [safety valve property](#) that can be set to limit its use of the AWS APIs. If you decide to configure this property to set a limit on API usage, keep the following in mind:

- If the limit is reached in any given 30-day interval, Cloudera Navigator suspends extraction from the configured Amazon S3 buckets until the next 30-day interval begins.
- When the new 30-day interval begins, Cloudera Navigator extracts any data that was not extracted while extraction was suspended.

To set a limit on the AWS API usage:

- Use Cloudera Manager Admin Console to access the **Navigator Metadata Server Advanced Configuration Snippet (Safety Valve) for cloudera-navigator.properties**.
- Set the value of `any_int` to your chosen limit.

```
nav.aws.api.limit=any_int
```

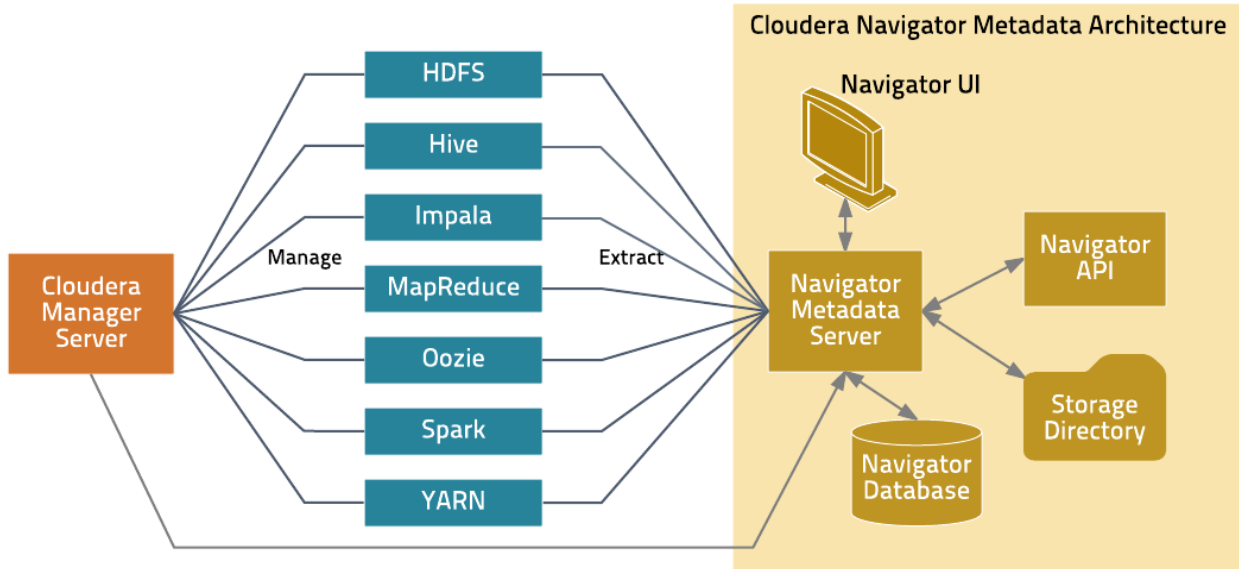
See [Setting Properties with Advanced Configuration Snippets](#) on page 27 for details about using Cloudera Manager Admin Console for this task.



Note: Cloudera Navigator does not notify you if extraction is ever suspended due to the limit you have set in the safety valve. Cloudera recommends setting a [billing alarm](#) in the AWS Management Console to get notified by Amazon when your monthly use of the APIs exceeds the limit you have set in the safety valve.

Cloudera Navigator Metadata Architecture

Cloudera Navigator metadata supports data discovery and data lineage functions. The following figure depicts the Cloudera Navigator metadata architecture.



The Navigator Metadata Server performs the following functions:

- Obtains connection information about CDH services from the Cloudera Manager Server
- Extracts metadata periodically for the entities managed by those services
- Manages and applies metadata extraction policies during metadata extraction
- Indexes and stores entity metadata
- Manages authorization data for Cloudera Navigator users
- Manages audit report metadata
- Generates metadata and audit analytics
- Exposes the Cloudera Navigator APIs
- Hosts the web server that provides the Cloudera Navigator console

The Navigator database stores policies, user authorization and audit report metadata, and analytic data. Extracted metadata and the state of extractor processes is kept in the storage directory.

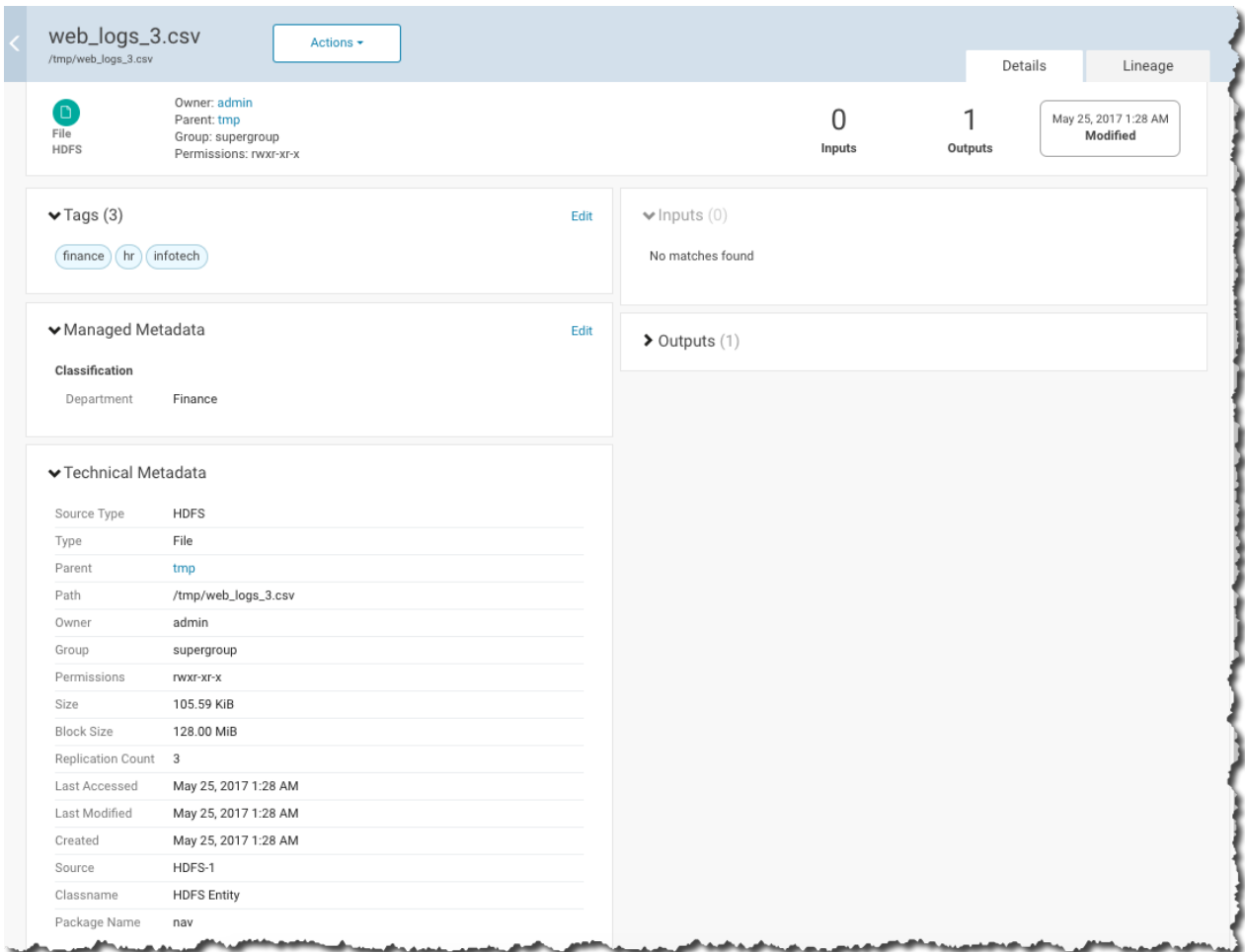
Three Different Classes of Metadata

The Cloudera Navigator Metadata Server manages metadata about the entities in a CDH cluster and relations between the entities. Any given entity can be identified by one or more of the three different classes of metadata listed in the table:

Category	Description	Usage Note
Technical Metadata	Characteristics inherent to the entity that are obtained when extracted.	Not modifiable.
Managed Metadata	Descriptions, tags, and key-value pairs that can be added to entities after extraction. Keys are defined within namespaces, and values can be constrained by type (Text, Number, Boolean, Date, Enumeration, for example).	Add to entities or modify after extraction only.

Category	Description	Usage Note
Custom Metadata	Key-value pairs that can be added to entities before or after extraction. Displayed in the Tags area of the Details page for a given entity.	Add to entities before or after extraction.

For example, the screenshot below shows the details of a web log saved to HDFS as a comma-separated value (CSV) file. This particular entity has all three types of metadata associated with it. Its Technical Metadata is applied by the source system, in this case, HDFS. The self-service data discovery aspect of this file—the ability for business users to find it—has been augmented thanks to the addition of the Finance department metadata (from the Classification namespace), shown in the Managed Metadata section of the file details. Finally, the Tags area of file details shows that Custom Metadata has also been applied to this file.



The Technical Metadata is obtained from the source entity and cannot be modified. Common examples of Technical Metadata include an entity's name, type (directory or file, for example), path, creation date and time, and access permissions. For entities created or managed by cluster services, Technical Metadata may include the name of the service that manages or uses that entity and relations—parent-child, data flow, and instance of—between entities.

As another example, Technical Metadata for an Amazon S3 bucket includes Bucket name, Region (AWS Region, such as us-west-1), S3 Encryption, S3 Storage Class, S3 Etag, Source (S3), and so on. In short, Technical Metadata is simply whatever metadata is provided for the entity by the system that created the entity. For example, for Hive entities, Cloudera Navigator extracts the extended attributes added by Hive clients to the entity.

Defining Properties for Managed Metadata

Required Role: [Metadata Administrator](#) (or **Full Administrator**)

You can use managed metadata to add typed metadata to classes of entities. You can add namespaces and properties.

A **namespace** is a container for properties. Four namespaces are reserved:

- `nav` for Navigator [metadata classes](#) (for example, `fselement` and user-defined custom fields)
- `up` ([custom metadata](#))
- `tp` (technical properties)
- `xt` (partner applications)

The combination of namespace and property name must be unique.



Note: The Cloudera Navigator console cannot delete namespaces. Empty namespaces can be deleted using the Cloudera Navigator APIs.

A property can be one of the following types:


- Boolean
- date
- integer
- long
- float
- double
- text (with optional maximum length and regular expression validation criteria)
- enum (of string)

A property can be single-valued or assume multiple values.

Once you have created properties and assigned property values to specific entities, you can create [search filters](#) for property values.

Creating Custom Properties with the Cloudera Navigator Console

To create custom properties:

1. Log in to the Cloudera Navigator console using an account that has one of the following user roles:
 - Cloudera Manager Full Administrator
 - Cloudera Manager Navigator Administrator
2. Click the **Administration** link in the upper right. The **Managed Metadata** tab displays the list of namespaces and the properties defined in the namespaces.
3. Click the **New Property...** button.
4. In the Classes field, click the  or type the beginning of a [Navigator entity classname](#).
5. Select the class of entities to which the property applies. To clear the field, hover over the field and click the delete icon (x) that displays at the right of the field.
6. Click the Namespace field and select a namespace. If the Namespace drop-down list is empty, click **Create Namespace...**
 - a. Specify a namespace name and optional description.
 - b. Click **Continue**.
7. Add the name for the property.

The name can contain letters, numbers, underscores, and hyphens and can be up to 50 characters long.

8. Specify an optional description.
9. Select the **Multivalued Enable** checkbox if the property can have more than one value. For example, an `emailFrom` property should accept only one value, but an `emailTo` property could accept more than one value.
10. In the **Type** drop-down list, select the property type and specify constraints on the value.
 - **Boolean** - Boolean: true or false.
 - **Date** - Date and time.
 - **Enumeration** - A set of values. In the **Enumeration** field, type valid enumeration values and press **Enter** or **Tab**.
 - **Number** - A number. In the **Number Type** field, select the type of the number: **Integer**, **Long**, **Float**, **Double**.
 - **Text** - A string.
 - **Maximum Length** - The maximum length of the string.
 - **Regular Expression** - A regular expression that determines whether a string is a valid value. You can test the expression by clicking **Show regex tester**, entering input that you expect to match the expression, and clicking **Execute test**. In the following example, the expression tester indicates that `test@example.com` matches the defined expression.

Regular Expression

[Hide regex tester](#)

Regular expression tester

[Execute test](#)

✔ The input matches the regular expression.

11. Click **Continue to Review**. The Review screen displays.
12. Click **Create** to create the property, **Cancel** to return to the Properties page, or **Back to Edit Property** to continue editing the property.

Example Properties

The following figure shows two properties in the namespace `MailAnnotation` that apply to entities of the `HDFS Entity` class (HDFS files and directories). The `emailFrom` property is of type `TEXT` and can be assigned a single value. The `MailTo` property is also of type `TEXT` but can have multiple values.

Properties

See the [documentation](#) before creating or modifying properties.

[Purge Deleted Properties](#)
[New Property](#)

Name	Status	Type	Classes	Multivalued	Actions
MailAnnotation					
EmailFrom	Active	Text	HDFS Entity	No	Actions ▾
EmailTo	Active	Text	HDFS Entity	Yes	Actions ▾

Using Cloudera Navigator Console to Manage Properties

You can view managed metadata property summary details by clicking property name in the Properties table, or by clicking the **Actions** box in the property row and then clicking **View** in the dropdown.

You can also edit some aspects of a property, delete and restore a property, and purge a deleted property.

Editing a Property

After a property is created, you can edit property data in the following ways:

- Add [classes](#) to which the property applies
- Add and remove enumeration values
- Change the description
- Change the maximum length
- Change the regex pattern

1. Log in to the Cloudera Navigator console using administrator credentials roles:

- Cloudera Manager Full Administrator
- Cloudera Manager Navigator Administrator
- Cloudera Navigator Full Administrator
- Cloudera Navigator Metadata Administrator

2. Click the **Administration** link in the upper right. The **Managed Metadata** tab displays the list of namespaces and the properties defined in the namespaces.

3. Open the property Edit page by clicking the **Actions** box in the property row and then clicking **Edit** in the dropdown.

4. In the Additional Class field, click the



or type the beginning of a [Navigator entity classname](#).

5. Select the class of entities to which the property applies. To clear the field, hover over the field and click the delete icon (x) that displays at the right of the field.

6. In the Description field, add a description or edit an existing description.

7. If the property is of the Enumeration type, you can add or remove values in the Enumeration field.



Note: If you delete an enumeration value, it cannot be used in new Managed Metadata assignments, but it continues to exist in entities or policies that are already using them.

8. For Text properties:

- In the Maximum Length field, add or change the value for the maximum length.
- In the Regular Expression field, edit the expression. Click [Show regex tester](#) to test input against any changes you make.

9. Click **Continue to Review**. The Review screen displays.

10. Click **Update** to commit the change or **Back to Edit Property** to continue editing the property, or **Cancel** to return to the Properties page.

Deleting, Restoring, and Purging Managed Metadata Properties

After a property is deleted, it cannot be used in a filter or in value assignments for entities. The deleted property is visible only from the Admin interface; it is labeled as Deleted in the Status column of the Properties table. In the following example, the `EmailFrom` property has been deleted.

Purge Deleted Properties
New Property

Name	Status	Type	Classes	Multivalued	Actions
MailAnnotation					
EmailFrom	Deleted	Text	HDFS Entity	No	Actions ▾
EmailTo	Active	Text	HDFS Entity	Yes	Actions ▾

For non-Admin users, a deleted property is hidden. However, the property still exists, and the values assigned to entities using this property are not affected until the deleted property is purged.

The Cloudera Navigator purge process permanently removes properties and any values from all entities. (Policies that assign metadata using a property that has been purged will fail the next time they are run.)

Because deleted properties are not removed from the system until they have been purged, the name of any deleted property cannot be re-used until after purging the system.

Deleting a Property

1. In the Properties table, for the property that you are deleting, click the **Actions** button, and then click **Delete** in the drop-down menu.
2. In the Delete Property dialog box, review the property deletion information. If any entities are affected, you see a **View affected entities** link; click to see all entities that use the property. Click **Confirm Delete** to delete the property, or click **Cancel**.

Restoring a Property

If you have not yet purged a deleted property, you can restore it.

- In the Properties table, for the property that you are restoring, click the **Actions** button, and then click **Restore** in the drop-down menu.

Purging a Property

You can permanently remove deleted properties by purging them. All values assigned to the deleted properties are lost; however, the affected entities are not deleted. Purging permanently removes all properties marked as Deleted in the Status column.



Note: Purging is a long-running task and may take some time. Navigator is unavailable to all users until the purge is completed.

1. In the Properties table, click **Purge Deleted Properties**. The Purge all Deleted Properties dialog box opens, describing the effects of the purge and reporting the number of entities that use the property.
2. In the Purge all Deleted Properties dialog box, click **Confirm Purge** to permanently remove all deleted properties, or click **Cancel** to return to the Properties page.

Navigator Built-in Classes

Class	Description
HDFS Dataset	Logical dataset backed by a path in HDFS.
HDFS Dataset Field	Field in an HDFS dataset.
HDFS Entity	HDFS file or directory.
Hive Column	Column in a Hive table.
Hive Database	Hive database.

Class	Description
Hive Partition	Partition of a Hive table.
Hive Query	Hive query template.
Hive Query Execution	Instance of a Hive query.
Hive Query Part	Component of a Hive query that maps specific input columns to output columns.
Hive Table	A Hive table.
Hive View	View on one or more Hive tables.
Impala Query	Impala query template.
Impala Query Execution	Instance of an Impala query.
Impala Query Part	Component of an Impala query that maps specific input columns to output columns.
Job Instance	Instance of a MapReduce, YARN, or Spark job.
Job Template	Template for a MapReduce, YARN, or Spark job.
Oozie Workflow	Template for an Oozie workflow.
Oozie Workflow Instance	Instance of an Oozie workflow.
Pig Field	Field for a relation in Pig; similar to a column in a Hive table.
Pig Operation	Template for a Pig transformation.
Pig Operation Execution	Instance of a Pig transformation.
Pig Relation	Pig relation; similar to a Hive table.
S3 Bucket	A bucket in S3.
S3 Object	A file or directory in an S3 bucket.
Sqoop Export Sub-operation	Sqoop export component that connects specific columns.
Sqoop Import Query	Sqoop import job with query options.
Sqoop Import Sub-operation	Sqoop import component that connects specific columns.
Sqoop Operation Execution	Instance of a Sqoop job.
Sqoop Table Export	Sqoop table export operation template.
Sqoop Table Import	Sqoop table import operation template.
User Sub-operation	User-specified sub-operation of a MapReduce or YARN job; used for specifying custom column-level lineage.

Defining Metadata with the Navigator API and Navigator SDK

In addition to defining metadata using features provided by the Cloudera Navigator console, you can also define metadata using the Cloudera Navigator API and Navigator SDK.

For information on the Navigator API, see [Cloudera Navigator APIs](#).

For information on the SDK, see the [Navigator SDK documentation](#).


Adding and Editing Metadata


Required Role: [Metadata Administrator](#) (or **Full Administrator**)

Cloudera Navigator supports adding metadata to extracted entities. You can add and edit two types of metadata:

- Custom metadata - Display name, description, tags, and key-value pairs. You can add and edit custom metadata using the Navigator UI, MapReduce service and job properties, HDFS metadata files, and the [Cloudera Navigator APIs](#). Custom metadata is typically implemented by end users who want to be able to classify and organize information for their own uses or to collaborate with other users.
- [Managed metadata](#). You can add and edit managed metadata using the Cloudera Navigator console and the API. Managed metadata is typically implemented for centralized curation of data sets.

Adding and Editing Metadata Using the Cloudera Navigator Console

1. Run a [search](#) in the Navigator UI.
2. Click an entity link returned in the search. The Details tab displays.
3. To the left of the Details tab, click **Actions > Edit Metadata....** The Edit Metadata dialog box drops down.
4. Add metadata fields:
 - In the Name field, type a new display name.
 - In the Description field, type a description.
 - **Managed Metadata**
 1. Click the  and select a property.
 2. Click the value field after the : to display type-specific selection controls such as integer spinners and date selection controls. Either type the value or use the controls to select a value.

Click the plus icon (+) to add another managed property key-value pair or another value for a given key.
 - **Custom Metadata:** In the Tags field, type a tag and press **Enter** or **Tab** to create new tag entries.
 - **Key-Value Pairs**
 1. Click  to add a key-value pair.
 2. Type a key and a value. You can specify special characters (for example, ".", " ") in the name, but it makes searching for the entity more difficult because some characters collide with special characters in the [search syntax](#).



Note: You cannot assign managed metadata in the Key-Value Pairs field because you cannot specify the namespace. All properties specified in the Key-Value Pairs field are treated as custom metadata.

5. Click **Save**. The new metadata appears in the Managed Metadata or Custom Metadata pane.

Custom Metadata Example

In the following example, the tag `archive_personal` and the property `year` with value `2015` have been added to the file `2015_11_20`:

Custom Metadata

Tags

archive_personal x

Key-Value Pairs

year : 2015 - +

Cancel Save

After you save, the metadata appears in the Tags and Custom Metadata panes:

2015_11_20
/user/admin/2015_11_20 Actions

Directory HDFS
Owner: admin
Parent: admin
Group: admin
Permissions: rwxrwxrwx

Tags (1) Edit
archive_personal

Custom Metadata Edit
year 2015

Managed Metadata Example

The following example shows the Department and RetainUntil managed properties for the customers entity:

✕
Edit Metadata

Name

Description

Managed Metadata

Department

:

Finance

-
+

RetainUntil

:

12/31/2017, 12:00 AM

-
+

After you specify the values and save, the properties display in the Managed Metadata pane:

account
Actions ▾

hdfs://Enchilada/data/gfcharan/work/hive/l4_amlharan/acco...

Details
Lineage

📅
 Table
Hive

Owner:
gfcharan
Parent:
l4_amlharan

43

21

16

0

Columns
Partitions
Inputs
Outputs

▼ Managed Metadata Edit

realestate

Department	Finance
AccessAttempts	10

▼ Columns (43) 🔍

- 📄 accommodation_account_flag string
- 📄 account_booking_country string
- 📄 account_closing_date string
- 📄 account_currency string

Editing MapReduce Custom Metadata

You can associate custom metadata with arbitrary configuration parameters to MapReduce jobs and job executions. The configuration parameters to be extracted by Navigator can be specified statically or dynamically.

To specify configuration parameters statically for all MapReduce jobs and job executions, do the following:

1. Do one of the following:
 - Select **Clusters > Cloudera Management Service**.
 - On the **Home > Status** tab, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
2. Click the **Configuration** tab.
3. Select **Scope > Navigator Metadata Server**.
4. Select **Category > Advanced**.
5. Click **Navigator Metadata Server Advanced Configuration Snippet for cloudera-navigator.properties**.
6. Specify values for the following properties:
 - `nav.user_defined_properties` - A comma-separated list of user-defined property names.

- `nav.tags` - A comma-separated list of property names that serve as tags. The property `nav.tags` can point to multiple property names that serve as tags, but each of those property names can only specify a *single* tag.

7. Click **Save Changes** to commit the changes.

8. Click the **Instances** tab.

9. Restart the role.

10. In the MapReduce job configuration, set the value of the property names you specified in step 6.

To specify configuration parameters dynamically:

1. Specify one or more of the following properties in a job configuration:

- Job properties (`type:OPERATION`)
 - `nav.job.user_defined_properties` - A comma-separated list of user-defined property names
 - `nav.job.tags` - A comma-separated list of property names that serve as tags
- Job execution properties (`type:OPERATION_EXECUTION`)
 - `nav.jobexec.user_defined_properties` - A comma-separated list of user-defined property names
 - `nav.jobexec.tags` - A comma-separated list of property names that serve as tags

The properties `nav.job.tags` and `nav.jobexec.tags` can point to multiple property names that serve as tags, but each of those property names can only specify a *single* tag.

2. In the MapReduce job configuration, set the value of the property names you specified in step 1.

Example: Setting Properties Dynamically

Add the tags `onetag` and `twotag` to a job:

1. Dynamically add the `job_tag1` and `job_tag2` properties:

```
conf.set("nav.job.tags", "job_tag1, job_tag2");
```

2. Set the `job_tag1` property to `onetag`:

```
conf.set("job_tag1", "onetag");
```

3. Set the `job_tag2` property to `twotag`:

```
conf.set("job_tag2", "twotag");
```

Add the tag `atag` to a job execution:

1. Dynamically add the `job_tag` property:

```
conf.set("nav.jobexec.tags", "job_exec_tag");
```

2. Set the `job_exec_tag` property to `atag`:

```
conf.set("job_exec_tag", "atag");
```

Add the user-defined key `key` with the value `value`:

1. Dynamically add the user-defined key `bar`:

```
conf.set("nav.job.user_defined_properties", "key");
```

2. Set the value of the user-defined key `key` to `value`:

```
conf.set("key", "value")
```

Editing HDFS Custom Metadata Using Metadata Files

You can add tags and properties to HDFS entities using metadata files. With metadata files, you can assign metadata to entities in bulk and create metadata before it is extracted. A metadata file is a JSON file with the following structure:

```
{
  "name" : "aName",
  "description" : "a description",
  "properties" : {
    "prop1" : "value1", "prop2" : "value2"
  },
  "tags" : [ "tag1" ]
}
```

To add metadata files to files and directories, create a metadata file with the extension `.navigator`, naming the files as follows:

- **File** - The path of the metadata file must be `.filename.navigator`. For example, to apply properties to the file `/user/test/file1.txt`, the metadata file path is `/user/test/.file1.txt.navigator`.
- **Directory** - The path of the metadata file must be `dirpath/.navigator`. For example, to apply properties to the directory `/user`, the metadata path must be `/user/.navigator`.

The metadata file is applied to the entity metadata when the extractor runs.

Editing HDFS and Hive Metadata Using the Navigator Metadata API

You can use the [Cloudera Navigator APIs](#) to modify the custom metadata of HDFS or Hive entities, whether the entities have been extracted or not. If an entity has been extracted when the API is called, the metadata is applied immediately. If the entity has not been extracted, you can preregister metadata, which is then applied once the entity is extracted. Metadata is saved regardless of whether or not a matching entity is extracted, and Navigator does not perform any cleanup of unused metadata.

If you call the API before the entity is extracted, the custom metadata is stored with the entity's:

- Identity
- Source ID
- Metadata fields (name, description, tags, properties)
- Fields relevant to the identifier

The rest of the entity fields (such as type) are not present. To view all stored metadata, use the API to search for entities without an internal type:

```
curl http://fqdn-n.example.com:port/api/APIversion/entities/?query=-internalType:* -u
username:password -X GET
```

Custom metadata provided through the API overwrites existing metadata. For example, if you call the API with an empty name and description, empty array for tags, and empty dictionary for properties, the call removes this metadata. If you omit the tags or properties fields, the existing values remain unchanged.

Modifying custom metadata using HDFS metadata files and the metadata API at the same time *is not* supported. You must use one or the other, because the two methods work differently. Metadata specified in files is merged with existing metadata, whereas the API overwrites metadata. Also, the updates provided by metadata files wait in a queue before being merged, but API changes are committed immediately. Some inconsistency can occur if a metadata file is merged when the API is in use.

You modify metadata using either the `PUT` or `POST` method. Use the `PUT` method if the entity has been extracted, and the `POST` method to preregister metadata. Use the following syntax:

- PUT

```
curl http://fqdn-n.example.com:port/api/APIversion/entities/identity -u username:password
-X PUT -H\
"Content-Type: application/json" -d '{properties}'
```

where *identity* is an entity ID and *properties* are:

- name - Name metadata.
- description - Description metadata.
- tags - Tag metadata.
- properties - Custom metadata properties. The format is {key: value}.
- customProperties - Managed metadata properties. The format is {namespace: {key: value}}. If a property is assigned a value that does not conform to type constraints, an error is returned.

All existing naming rules apply, and if any value is invalid, the entire request is denied.

- POST

```
curl http://fqdn-n.example.com:port/api/APIversion/entities/ -u username:password -X
POST -H\
"Content-Type: application/json" -d '{properties}'
```

where *properties* are:

- [sourceId](#) (required) - An existing source ID. After the first extraction, you can retrieve source IDs using the call:

```
curl http://fqdn-n.example.com:port/api/APIversion/entities/?query=type:SOURCE -u
username:password -X GET
```

For example:

```
[ ...
{
  {
    "identity": "61cfefd303d4284b7f5014b701f2c76d",
    "originalName": "source.listing",
    "originalDescription": null,
    "sourceId": "012437f9eeb3c23dc69e679ac94a7fa2",
    "firstClassParentId": null,
    "parentPath": "/user/hdfs/.cm/distcp/2016-02-03_487",
    ...
    "properties": {
      "__cloudera_internal_hueLink":
"http://fqdn-2.example.com:8888/filebrowser/#/user/hdfs/.cm/distcp/2016-02-03_487/source.listing"
    },
    "technicalProperties": null,
    "filePath": "/user/hdfs/.cm/distcp/2016-02-03_487/source.listing",
    "type": "FILE",
    "size": 92682,
    "created": "2016-02-03T21:12:16.587Z",
    "lastModified": "2016-02-03T21:12:16.587Z",
    "lastAccessed": "2016-02-03T21:12:16.587Z",
    "permissions": "rw-r--r--",
    "owner": "hdfs",
    "group": "supergroup",
    "blockSize": 134217728,
    "mimeType": "application/octet-stream",
    "replication": 3,
    "userEntity": false,
    "deleted": false,
    "sourceType": "HDFS",
    "metaClassName": "fselement",
    "packageName": "nav",
    "internalType": "fselement"
  }, ...
}
```

If you have multiple services of a given type, you must specify the source ID that contains the entity you expect it to match.

- `parentPath` - The path of the parent entity, defined as:
 - HDFS file or directory - `filePath` of the parent directory. (Do not provide this field if the entity affected is the root directory.) Example `parentPath` for `/user/admin/input_dir`: `/user/admin`. If you add metadata to a directory, the metadata does not propagate to any files or folders in that directory.
 - Hive database - If you are updating database metadata, do not specify this field.
 - Hive table or view - The name of database containing the table or view. Example for a table in the default database: `default`.
 - Hive column - `database name/table name/view name`. Example for a column in the `sample_07` table: `default/sample_07`.
- `originalName` (required) - The name as defined by the source system.
 - HDFS file or directory- Name of file or directory (`ROOT` if the entity is the root directory). Example `originalName` for `/user/admin/input_dir`: `input_dir`.
 - Hive database, table, view, or column - The name of the database, table, view, or column.
 - Example for default database: `default`
 - Example for `sample_07` table: `sample_07`
- `name` - Name metadata.
- `description` - Description metadata.
- `tags` - Tag metadata.
- `properties` - Custom metadata properties. The format is `{key: value}`.

All existing naming rules apply, and if any value is invalid, the entire request is denied.

HDFS PUT Custom Metadata Example for `/user/admin/input_dir` Directory

```
curl
http://fqdn-n.example.com:port/api/APIversion/entities/e461de8de38511a3ac6740dd7d51b8d0
-u username:password -X PUT -H "Content-Type: application/json" \
-d '{"name":"my_name","description":"My description",
"tags":["tag1","tag2"],"properties":{"property1":"value1","property2":"value2"}}'
```

HDFS POST Custom Metadata Example for `/user/admin/input_dir` Directory

```
curl http://fqdn-n.example.com:port/api/APIversion/entities/ -u username:password -X
POST -H "Content-Type: application/json" \
-d '{"sourceId":"a09b0233cc58ff7d601eaa68673a20c6",
"parentPath":"/user/admin","originalName":"input_dir","name":"my_name","description":"My
description",\
"tags":["tag1","tag2"],"properties":{"property1":"value1","property2":"value2"}}'
```

Hive POST Custom Metadata Example for `total_emp` Column

```
curl http://fqdn-n.example.com:port/api/APIversion/entities/ -u username:password -X
POST -H "Content-Type: application/json" \
-d '{"sourceId":"4fbdadc6899638782fc8cb626176dc7b",
"parentPath":"default/sample_07","originalName":"total_emp",\
"name":"my_name","description":"My description",
"tags":["tag1","tag2"],"properties":{"property1":"value1","property2":"value2"}}'
```

HDFS PUT Managed Metadata Example

The following example demonstrates how to set two properties in the MailAnnotation namespace: a multivalued property emailTo and a single-valued property emailFrom:

```
curl
http://fqdn-n.example.com:port/api/APIVersion/entities/87afcb92d5de856c7e8292e2e12cf1ea
-u admin:admin -X PUT -H "Content-Type: application/json"
-d
'{"customProperties":{"MailAnnotation":{"emailTo":["lee@example.com","dana@example.com"],"emailFrom":"terry@email.com"}}}'
```

The response is:

```
{
  "identity" : "87afcb92d5de856c7e8292e2e12cf1ea",
  "originalName" : "years",
  "originalDescription" : null,
  "sourceId" : "012437f9eeb3c23dc69e679ac94a7fa2",
  "firstClassParentId" : null,
  "parentPath" : "/user/admin",
  "extractorRunId" : "012437f9eeb3c23dc69e679ac94a7fa2##1",
  "customProperties" : {
    "MailAnnotation" : {
      "emailTo" : [ "lee@example.com", "dana@example.com" ],
      "emailFrom" : "terry@email.com"
    }
  },
  "name" : null,
  "description" : null,
  "tags" : null,
  "properties" : {
    "__cloudera_internal_hueLink" : "Hue_Server_host:8888/filebrowser/#/user/admin/years"
  },
  "technicalProperties" : null,
  "filePath" : "/user/admin/years",
  "type" : "DIRECTORY",
  "size" : null,
  "created" : "2016-03-22T17:55:31.902Z",
  "lastModified" : "2016-03-22T17:59:14.065Z",
  "lastAccessed" : null,
  "permissions" : "rwxr-xr-x",
  "owner" : "hdfs",
  "group" : "admin",
  "blockSize" : null,
  "mimeType" : null,
  "replication" : null,
  "sourceType" : "HDFS",
  "metaClassName" : "fselement",
  "userEntity" : false,
  "deleted" : false,
  "packageName" : "nav",
  "internalType" : "fselement"
}
```

Accessing and Editing Metadata with the Cloudera Navigator SDK

To facilitate editing metadata with the Cloudera Navigator APIs, Cloudera provides a Cloudera Navigator SDK at GitHub. Cloudera Navigator SDK is a client library that provides functionality for extracting and enriching metadata with custom models, entities, and relationships. See GitHub [cloudera/navicator-sdk](#) for details.

Metadata Extraction and Indexing

The Navigator Metadata Server extracts metadata for the resource types listed in the table.

Table 1: Metadata Extraction by Resource Type (Service, Role)

Resource Type	Metadata Extracted
HDFS	HDFS metadata at the next scheduled extraction run after an HDFS checkpoint. If high availability is enabled, metadata is extracted as soon as it is written to the JournalNodes.
Hive	Database, table, and query metadata from Hive lineage logs. See Managing Hive and Impala Lineage Properties . Hive entities include tables that result from Impala queries and Sqoop jobs.
Impala	Database, table, and query metadata from the Impala Daemon lineage logs. See Managing Hive and Impala Lineage Properties .
MapReduce	Job metadata from the JobTracker. The default setting in Cloudera Manager retains a maximum of five jobs; if you run more than five jobs between Navigator extractions, the Navigator Metadata Server extracts the five most recent jobs.
Oozie	Oozie workflows from the Oozie Server.
Pig	Pig script runs from the JobTracker or Job History Server.
S3	Bucket and object metadata.
Spark	Spark job metadata from YARN logs. (Unsupported and disabled by default. To enable, see Enabling Spark Metadata Extraction .)
Sqoop 1	Database and table metadata from Hive lineage logs; job runs from the JobTracker or Job History Server.
YARN	Job metadata from the ResourceManager.

An entity created at system time t_0 is extracted and linked by Cloudera Navigator after the 10-minute (default) extraction poll period and the appropriate service-specific interval, as follows:

- **HDFS:** $t_0 + (\text{extraction poll period}) + (\text{HDFS checkpoint interval (1 hour by default)})$
- **HDFS + HA:** $t_0 + (\text{extraction poll period})$
- **Hive:** $t_0 + (\text{extraction poll period}) + (\text{Hive maximum wait time (60 minutes by default)})$
- **Impala:** $t_0 + (\text{extraction poll period})$

Metadata Indexing

After metadata is extracted, it is indexed and made available for [searching](#) by the embedded [Solr](#) engine. The Solr instance indexes entity properties and the relationships between entities.

Use the [Cloudera Navigator console](#) to [search](#) entity metadata. Relationship metadata is implicitly visible in [lineage diagrams](#) and explicitly available by downloading the lineage using the [Cloudera Navigator APIs](#).

Searching Metadata

Required Role: [Metadata Viewer](#) (or **Metadata Administrator, Full Administrator**)

You can access metadata through the Cloudera Navigator console or by using the Navigator API.

Using the Cloudera Navigator Console

Searching Metadata

1. Open your browser.
2. Navigate to the host within the cluster running the Cloudera Navigator Metadata Server role.

```
http://fqdn-1.example.com:7187/login.html
```


In this example, node 1 of the cluster is running the Navigator Metadata Server role, hosted on the default port 7187. The login page displays.

3. Log in to the Cloudera Navigator console using the [credentials](#) assigned by your administrator. The Cloudera Navigator console opens to the Search tab.
4. To display all entities, click **Clear all filters**.
5. Filter the search results by specifying filters or typing search strings in the Search box.

Filter Example

The following filter example demonstrates how to narrow search results by selecting a built-in Source Type filter set to **HDFS** and the managed property **BLOCK SIZE (MIB)** with the value **>=1024**. This example also shows the full query, which you can see by clicking **Show full query** in the results pane.

The screenshot shows the Cloudera Navigator Search interface. On the left, there are filter sections for 'BLOCK SIZE (MIB)' and 'SOURCE TYPE'. The 'BLOCK SIZE (MIB)' filter is set to '>= 1024'. The 'SOURCE TYPE' filter is set to 'HDFS'. The search results pane shows 3 results, each with a 'View in Hue' link. The full query is displayed as '+blockSize:[1073741824 TO *] +sourceType:hdfs'.

Search [Search Box] [Actions]

Filters Add Filters Clear All Filters

Add Filter...

▼ BLOCK SIZE (MIB) x

- 0 - < 256
- 256 - < 512
- 512 - < 1024
- >= 1024
- Custom

▼ SOURCE TYPE

- HDFS 3
- > SIZE (MIB)
- > CREATED
- > GROUP

Full Query [Close]

+blockSize:[1073741824 TO *] +sourceType:hdfs

3 results

Block Size >= 1024 MIB Source Type = HDFS

	HDFS 5179ab77-80c2-4513-be69-3b7aa4f83d04	Type File Path /accumulo/wal/msd-22.gce.cloudera.com+10011/5179ab77-80c2-4513-be69-3b7aa4f83d04	Owner accumulo Group accumulo Permissions rwxr-xr-x Size 1.75 KiB Block Size 1.10 GiB	View in Hue
	HDFS dba466fa-fbea-4e80-b958-85dc8822c0bb	Type File Path /accumulo/wal/msd-22.gce.cloudera.com+10011/dba466fa-fbea-4e80-b958-85dc8822c0bb	Owner accumulo Group accumulo Permissions rwxr-xr-x Size 212 B Block Size 1.10 GiB	View in Hue
	HDFS fc96462e-c480-49f5-8fdd-8637caf1351f	Type File Path /accumulo/wal/msd-24+10011/fc96462e-c480-49f5-8fdd-8637caf1351f	Owner accumulo Group accumulo Permissions rwxr-xr-x Size 0 B Block Size 1.10 GiB	View in Hue

Viewing Search Results

The Search Results pane displays the number of matching entries in pages listing 25 entities per page. You can view the pages using the page control at the bottom of each page.

Each entry in the result list contains:

- Source type
- Name - A link to a page that displays the entity details and [lineage diagram](#)
- Properties
- If Hue is running, a link at the far right labeled **View in Hue** that opens the Hue browser for the entity. For example:

The screenshot shows a search result entry for 'Hive web_logs'. It includes a 'View in Hue' link and the following details:

Hive web_logs
 Type **Table** Parent Path /default
 Path hdfs://eng-nav-1.gce.cloudera.com:8020/user/hive/warehouse/web_logs Owner admin
 Created Dec 16, 2016 2:55 AM Source HIVE-1

Displaying Entity Details

The entity Details page displays all three types of metadata for an entity—technical, managed, and custom—and entity type-specific information:

- HDFS directories - Directory contents
- HDFS datasets and fields - Schema
- Hive and Impala databases - Tables and views

- Hive tables - Extended attributes, table schema, partitions
- Impala tables - Table schema
- MapReduce, YARN, and Spark operations - Instances
- Pig operation executions - Tables
- S3 buckets and folders - Folder contents

All entity types can display inputs and outputs. See [Configuring Display of Inputs and Outputs](#).

If managed properties have been defined for a particular entity type, the **Show All** checkbox in the Managed Metadata pane displays all properties that can be assigned values for the selected entity. To display only those properties that have values, clear the checkbox. If all properties have values, the checkbox has no effect.

To display entity details:

1. Perform a search.
2. In the search results, click an entity name link. The Details tab displays.

Hive Table Entity Details

For example, if you click the Hive table `web_logs` link in the search result displayed in preceding example, you see the following details:

The screenshot displays the details for a Hive table named `web_logs`. The interface includes a header with the table name, a path, and an 'Actions' dropdown. Below the header, there are several key metrics: Owner (admin), Parent (default), 29 Columns, 4 Partitions, 0 Inputs, and 0 Outputs. A 'Created' timestamp of Dec 16, 2016 2:55 AM is also shown. The main content area is divided into two columns. The left column contains 'Technical Metadata', 'Description', 'Tags', and 'Custom Metadata', each with an 'Add' button. The right column contains 'Columns (29)', 'Inputs (0)', and 'Outputs (0)', each with a caret icon to expand the view.

The caret > indicates fields with content; click to expand them. In addition to the technical metadata, you can view the names of the columns and the inputs and outputs.

Managed Metadata Example

The following `account` table entity has two managed properties in the `realestate` namespace: `Department` and `AccessAttempts`.

The screenshot displays the Cloudera Navigator Metadata Architecture interface for an account named 'account'. The breadcrumb path is 'hdfs://Enchilada/data/gfcharan/work/hive/I4_amlharan/acco...'. The interface includes an 'Actions' dropdown menu and two tabs: 'Details' (selected) and 'Lineage'. Below the header, a summary row shows:

- Table Hive icon
- Owner: gfcharan, Parent: I4_amlharan
- 43 Columns
- 21 Partitions
- 16 Inputs
- 0 Outputs

 The main content area is divided into two panels:


- Managed Metadata:** Shows a table with columns 'Department' and 'AccessAttempts'. The 'Department' value is 'Finance' and 'AccessAttempts' is '10'. An 'Edit' link is present.
- Columns (43):** A scrollable list of columns with their data types:
 - accommodation_account_flag string
 - account_booking_country string
 - account_closing_date string
 - account_currency string

Filtering Search Results

To filter search results, specify filters in the Filters pane or type [search strings](#) in the **Search** box.

The Filters pane lists default properties (source type, type, owner, cluster, and tags) and property values. Add a filter by clicking **Add Filters...** and scrolling, or by typing in the filter combo box to search for it. To remove non-default filter properties, click the x in the filter.

Specify a property value as follows:

- **Boolean** - Select the option to respectively not display, or display only those entries, with the value set to true: **Do not show XXX** (the default) or **Show XXX only**, where XXX is the Boolean property.
- **Enumerated or freeform string**
 - Select the checkbox next to a value or click a value link.
 - If a property has no values, click **add a new value**, click the text box, and select from the populated values in the drop-down list or type a value.
- **Timestamp** - Timestamps are used for started, ended, created, last accessed, and last modified properties. The server stores the timestamp in UTC, and the UI displays the timestamp converted to the local timezone. Select one of the timestamp options:
 - A **Last XXX day(s)** link.
 - The **Last** checkbox, type or specify the value using the spinner control  and select the unit minutes, hours, or days.
 - The **Custom period** checkbox and specify the start and end date.
 - Date - Click the down arrow to display a calendar and select a date, or click a field and click the spinner arrows or press the up and down arrow keys.
 - Time - Click the hour, minute, and AM/PM fields and click the spinner arrows or press the up and down arrow keys to specify the value.
 - Move between fields by clicking fields or by using the right and left arrow keys.

To remove filter values, clear the checkbox.

When you select a specific source type value, additional properties that apply to that source type display. For example, HDFS has size, created, and group properties:

▼ SOURCE TYPE

● HDFS 415,271

▼ SIZE (MIB)

0 - < 256 412,676 512 - < 1024 0

256 - < 512 0 >= 1024 0

Custom

▼ CREATED

Last 30 days 264,154 Last 90 days 415,270

Last day 9,710 Last 365 days 415,270

Last ↕

Custom period

▼ GROUP

spark 407,266

hadoop 3,159

cmjobuser 2,131

hbase 1,150

oozie 712

[Add New Value](#)

The number to the right a property value is the number of extracted entities that have that property value:

▼ SOURCE TYPE

● HDFS 415,301

> SIZE (MIB)

> CREATED

> GROUP

● YARN 1,084

> STARTED

> ENDED

● Hive 239

> STARTED

> ENDED

● Pig 18

● Impala 5

Facet values with the count 0 are not displayed.

When you type values, the value is enclosed in quotes; the value inside the quotes must match the metadata exactly. For example:

- Typing "sample_*" in the `originalName` property returns only entities whose names match that exact string.
- To perform a wildcard search, type the wildcard string in the Search box. For example, typing the string "sample_*" in the Search box returns all entities with "sample_" at the beginning of their original name.

When you construct search strings with filters, use parentheses () to specify multiple values of a property. Add multiple properties by using the + operator. For example, entities of type HDFS or Hive that are of type file or directory:

```
+(sourceType:hdfs sourceType:hive) +(type:file type:directory)
```

and:

```
((+sourceType:hdfs +created:[NOW/DAY-30DAYS TO NOW/DAY+1DAY]) sourceType:hive)
```

Saving Searches

1. Specify a search string or set of filters.
2. To the right of the Search box, select **Actions > Save**, **Actions > Save Search_name**, or **Actions > Save As....**
3. If you have not previously saved the search, specify a name and click **Save**.

Reusing a Saved Search

1. To the right of the Search box, select **Actions > View saved searches....** A label with the saved search name is added under the Search box.
2. Click the saved search name. The breadcrumbs and full query if displayed are updated to reflect the saved search, and the search results are refreshed immediately.

Performing Actions on Entities

Required Role: [Policy Administrator](#) (or **Full Administrator**)

Moving an HDFS Entity and Moving an HDFS Entity to Trash

You can move an HDFS entity to another HDFS location, and to [HDFS trash](#). To perform such actions, you must be a member of a user group that has the appropriate access to HDFS files.



Note: To move HDFS entities to another file system such as S3, consider designating an HDFS directory as an archive location, define a Navigator policy to move the HDFS entities to that directory, and define a CRON job or other process to regularly move the content of the archive directory to the non-HDFS location.

You can also schedule a move or move to trash in a [policy](#).

1. Open your browser.
2. Navigate to the host within the cluster running the Cloudera Navigator Metadata Server role using the correct host name and port for your instance:

```
http://fqdn-1.example.com:7187/login.html
```

The login prompt displays.

3. Log in to the Cloudera Navigator console using the [credentials](#) assigned by your administrator.
4. Run a [search](#) in the Cloudera Navigator console.
5. Click an HDFS entity link returned in the search. The entity Details tab displays.
6. To the left of the Details tab, select **Actions > Move...** or **Actions > Move to Trash....**
7. For a move, specify the target path.
8. Click **Run Action**. When you delete a file, after a short delay the file displays a Deleted badge.

Viewing Command Action Status

1. Access the Cloudera Navigator console using the correct host name and port for your instance:

```
http://fqdn-1.example.com:7187/login.html
```

The login prompt displays.

2. Log in to the Cloudera Navigator console using the [credentials](#) assigned by your administrator.
3. In the top right, click **username** > **Command Actions**. The Command Actions status page displays with a list of actions performed and the policy that caused the action, if applicable.
4. If an action failed, a View Log button displays, which you can click to view the error message associated with the failure.

Viewing an Entity in Hue

If the [Hue](#) component is running on the cluster view some entities using Hue and related tools, as follows:

File Browser	HDFS directories and files
Hive Metastore Manager (HMS)	Hive database and tables
Job Browser	YARN, Oozie

1. Access the Cloudera Navigator console using the correct host name and port for your instance:

```
http://fqdn-1.example.com:7187/login.html
```

The login prompt displays.

2. Log in to the Cloudera Navigator console using the [credentials](#) assigned by your administrator.
3. Run a [search](#) in the Cloudera Navigator console.
4. Do one of the following:
 - Search results
 1. Click the **View in Hue** link in a search result entry.
 - Entity details
 1. Click an entity link returned in the search. The entity Details tab displays.
 2. To the left of the Details tab, select **Actions** > **View in Hue**.

The entity displays in the supported Hue application.

Metadata Policies

Cloudera Navigator lets you automate the application of metadata to specified classes of entities using the policies you define. The policy specifies the actions to be performed by Navigator Metadata Server and the conditions under which to apply them. For data stewards who want to facilitate self-service discovery in their organizations, Cloudera Navigator's metadata policy feature provides a robust mechanism for streamlining the metadata management tasks. For example, you can define policies that:

- Add managed metadata, tags, and custom metadata to entities as they are ingested by the cluster
- Move entities of specific class to a specific target path, or to the trash
- Send messages to a JMS message queue for notifications. This requires configuring the JMS server on the Cloudera Management Service. See [Configuring a JMS Server for Policy Messages](#) for details.

Messages sent to JMS queues are formatted as JSON and contain the metadata of the entity to which the policy should apply and the policy's specified message text. For example:

```
{
  "entity":entity_property,
  "userMessage":"some message text"
}
```

A policy is run as the user who created the policy, in the home directory of the user who created the policy. To change a policy's ownership, log in to Cloudera Navigator console as the user account to which you want to transfer ownership of the policy, clone the policy, and then delete or disable the old policy.

Policies execute in the home directory of the user account that creates, and can only take actions for which the user account has privileges. The privileges needed are for file (or directory) access and any command executed. A policy will fail at runtime if the associated user account does not have privileges to perform all the actions defined in the policy.



Note: To ensure that command actions do not fail, policies containing command actions should be created by data stewards who are members of a user group that has the appropriate access to HDFS files.

Certain actions can be specified using Java expressions. See [Metadata Policy Expressions](#) for details.

Creating Policies

Required Role: [Policy Administrator](#) (or **Full Administrator**)

These steps begin from the Cloudera Navigator console.

1. Select **Clusters > Cloudera Management Service**.
2. Click the **Policies** tab.
3. In the Status field, check the **Enable** checkbox.
4. Enter a name for the policy.
5. Specify the [search query](#) that defines the class of entities to which the policy applies. If you arrive at the Policies page by clicking a search result, the query property is populated with the query that generated the result. To display a list of entities that satisfy a search query, click the **Test Query** link.
6. Specify an optional description for the policy.
7. If you use policy expressions in properties that support expressions, specify required imports in the **Import Statements** field. See [Metadata Policy Expression Examples](#) on page 58.
8. Choose the schedule for applying the policy:
 - **On Change** - When the entities matching the search string change.
 - **Immediate** - When the policy is created.
 - **Once** - At the time specified in the Start Time field.
 - **Recurring** - At recurring times specified by the Start and End Time fields at the interval specified in the Interval field.

For the Once and Recurring fields, specify dates and times:

- **Date** - Click the down arrow to display a calendar and select a date, or click a field and click the spinner arrows or press the up and down arrow keys.
 - **Time** - Click the hour, minute, and AM/PM fields and click the spinner arrows or press the up and down arrow keys to specify the value.
 - **Move between fields** by clicking fields or by using the right and left arrow keys.
9. Follow the appropriate procedure for the actions performed by the policy:

- **Metadata Assignments:** Specify the custom [metadata or managed metadata](#) to be assigned. Optionally, you can specify a Java [policy expression](#) for fields that support expressions by checking the **Expression** checkbox. The following fields support expressions:

- Name
- Description
- Managed Metadata
- Key-Value Pairs

- **Command Actions:** Select **Add Action > Move to Trash** or **Add Action > Move**. For a move, specify the location to move the entity to in the Target Path field. If you specify multiple actions, they are run in the order in which they are specified.

Command actions are supported only for HDFS entities. If you configure a command action for unsupported entities, a runtime error is logged when the policy runs.

See [Viewing Command Action Status](#) on page 54.

- **JMS Notifications:** If not already configured, [configure a JMS server and queue](#). Specify the queue name and message. Optionally, check the **Expression** checkbox and specify a policy expression for the message.

10 Click **Save**.

Viewing Policies

Required Role: [Policy Viewer](#) (or **Policy Administrator**, or **Full Administrator**)

1. [Accessing the Cloudera Navigator console](#).
2. Click the **Policies** tab.
3. In a policy row, click a policy name link or select **Actions > View**. The policy detail page is displayed.

You can also [edit](#), [copy](#), or [delete](#) a policy from the policy details page by clicking the **Actions** button.



The screenshot shows the 'Policies' page in Cloudera Navigator. The title 'Policies' is at the top left. Below it, the policy name 'hdfsImmediatePolicy' is displayed. To the right of the name is an 'Actions' dropdown menu. The policy details are as follows:

Status:	✓ Enabled
Search Query:	filePath:"/tmp/policy_hdfs_data/testfile1" AND sourceType:hdfs AND deleted:false
Policy Description:	
Last Run On:	Friday, December 9th 2016, 6:47 am

The 'Actions' dropdown menu is open, showing three options: 'Edit', 'Copy', and 'Delete'.

Enabling and Disabling Policies

As a policy administrator, you can manage access to policies by enabling and disabling them.

Required Role: [Policy Administrator](#) (or **Full Administrator**)

1. [Accessing the Cloudera Navigator console](#).
2. Click the **Policies** tab.
3. In a policy row, click a policy name link or select **Actions > Enable** or **Actions > Disable**.

Copying and Editing a Policy

If you have an existing policy that you want to use as a template for another similar property, you can copy it and then make any required adjustments. You can also edit existing policies if you need to make changes to it.

Required Role: [Policy Administrator](#) (or **Full Administrator**)

1. [Accessing the Cloudera Navigator console](#).
2. Click the **Policies** tab.
3. In a policy row, select **Actions > Copy** or **Actions > Edit**. You can also click the policy row and then on the policy details page, select **Actions > Copy** or **Actions > Edit**.
4. Edit the policy name, search query, or policy actions.
5. Click **Save**.

Deleting Policies

Required Role: [Policy Administrator](#) (or **Full Administrator**)

1. [Accessing the Cloudera Navigator console](#).
2. Click the **Policies** tab.
3. In a policy row, select **Actions > Delete** and **OK** to confirm.

Metadata Policy Expressions

A **metadata policy expression** allows you to specify certain [metadata extraction policy](#) properties using Java expressions instead of string literals. The supported properties are entity name and description, managed metadata, key-value pairs, and JMS notification message.

You must declare classes accessed in the expression in the policy's **Import Statements** field. A metadata policy expression must evaluate to a string.

In the Cloudera Navigator console, you see an Expression check box under or next to elements for which you can define an expression, as well as a pop-up that you can open to see an expression example:



You can define expressions for the following when you create a policy:

- Metadata Assignments
 - Name
 - Description
 - Managed Metadata
 - Key-Value Pairs
- JMS Notification Messages



Note: Using Java expressions to define policies is not enabled by default. See [Managing Metadata Policies](#) for details about configuring the Navigator Metadata Server role instance to support this capability.

Including Entity Properties in Policy Expressions

To include entity properties in property expressions, use the `entity.get` method, which takes a property and a return type:

```
entity.get(XXProperties.Property, return_type)
```

`XXProperties.Property` is the Java enumerated value representing an entity property, where

- `XX` is [FSEntity](#), [HiveColumn](#), [HiveDatabase](#), [HivePartition](#), [HiveQueryExecution](#), [HiveQueryPart](#), [HiveQuery](#), [HiveTable](#), [HiveView](#), [JobExecution](#), [Job](#), [WorkflowInstance](#), [Workflow](#), [PigField](#), [PigOperationExecution](#), [PigOperation](#), [PigRelation](#), [SqoopExportSubOperation](#), [SqoopImportSubOperation](#), [SqoopOperationExecution](#), [SqoopQueryOperation](#), [SqoopTableExportOperation](#), or [SqoopTableImportOperation](#).
- `Property` is one of the properties listed in [Entity Property Enum Reference](#) on page 58.

If you do not need to specify a return type, use `Object.class` as the return type. However, if you want to do type-specific operations with the result, set the return type to the type in the comment in the enum property reference. For example, in `FSEntityProperties`, the return type of the `ORIGINAL_NAME` property is `java.lang.String`. If you use `String.class` as the return type, you can use the `String` method `toLowerCase()` to modify the returned value: `entity.get(FSEntityProperties.ORIGINAL_NAME, String.class).toLowerCase()`.

Metadata Policy Expression Examples

- Set a filesystem entity name to the original name concatenated with the entity type:

```
entity.get(FSEntityProperties.ORIGINAL_NAME, Object.class) + " " +
entity.get(FSEntityProperties.TYPE, Object.class)
```

Import Statements:

```
import com.cloudera.nav.hdfs.model.FSEntityProperties;
```

- Add the entity's creation date to the entity name:

```
entity.get(FSEntityProperties.ORIGINAL_NAME, Object.class) + " - "
+ new SimpleDateFormat("yyyy-MM-dd").format(entity.get(FSEntityProperties.CREATED,
Instant.class).toDate())
```

Import Statements:

```
import com.cloudera.nav.hdfs.model.FSEntityProperties; import java.text.SimpleDateFormat;
import org.joda.time.Instant;
```

- Set the key-value pair: retain_util-seven years from today's local time:

```
new DateTime().plusYears(7).toLocalDateTime().toString("MMM dd yyyy", Locale.US)
```

Import statements:

```
import org.joda.time.DateTime; import java.util.Locale;
```

Entity Property Enum Reference

The following reference lists the Java enumerated values for retrieving properties of each entity type.

```
com.cloudera.nav.hdfs.model.FSEntityProperties
public enum FSEntityProperties implements PropertyEnum {
  PERMISSIONS, // Return type: java.lang.String
  TYPE, // Return type: java.lang.String
  SIZE, // Return type: java.lang.Long
  OWNER, // Return type: java.lang.String
  LAST_MODIFIED, // Return type: org.joda.time.Instant
  SOURCE_TYPE, // Return type: java.lang.String
  DELETED, // Return type: java.lang.Boolean
  FILE_SYSTEM_PATH, // Return type: java.lang.String
  CREATED, // Return type: org.joda.time.Instant
  LAST_ACCESSED, // Return type: org.joda.time.Instant
  GROUP, // Return type: java.lang.String
  MIME_TYPE, // Return type: java.lang.String
  DELETE_TIME, // Return type: java.lang.Long
```

```

NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HiveColumnProperties
public enum HiveColumnProperties implements PropertyEnum {
TYPE, // Return type: java.lang.String
SOURCE_TYPE, // Return type: java.lang.String
DELETED, // Return type: java.lang.Boolean
DATA_TYPE, // Return type: java.lang.String
ORIGINAL_DESCRIPTION, // Return type: java.lang.String
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HiveDatabaseProperties
public enum HiveDatabaseProperties implements PropertyEnum {
TYPE, // Return type: java.lang.String
ORIGINAL_DESCRIPTION, // Return type: java.lang.String
SOURCE_TYPE, // Return type: java.lang.String
DELETED, // Return type: java.lang.Boolean
FILE_SYSTEM_PATH, // Return type: java.lang.String
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HivePartitionProperties
public enum HivePartitionProperties implements PropertyEnum {
TYPE, // Return type: java.lang.String
SOURCE_TYPE, // Return type: java.lang.String
DELETED, // Return type: java.lang.Boolean
FILE_SYSTEM_PATH, // Return type: java.lang.String
CREATED, // Return type: org.joda.time.Instant
LAST_ACCESSED, // Return type: org.joda.time.Instant
COL_VALUES, // Return type: java.util.List
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HiveQueryExecutionProperties
public enum HiveQueryExecutionProperties implements PropertyEnum {
SOURCE_TYPE, // Return type: java.lang.String
TYPE, // Return type: java.lang.String
ENDED, // Return type: org.joda.time.Instant
INPUTS, // Return type: java.util.Collection
OUTPUTS, // Return type: java.util.Collection
STARTED, // Return type: org.joda.time.Instant
PRINCIPAL, // Return type: java.lang.String
WF_INST_ID, // Return type: java.lang.String
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
}

```

```

EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HiveQueryPartProperties
public enum HiveQueryPartProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HiveQueryProperties
public enum HiveQueryProperties implements PropertyEnum {
    SOURCE_TYPE, // Return type: java.lang.String
    INPUTS, // Return type: java.util.Collection
    OUTPUTS, // Return type: java.util.Collection
    QUERY_TEXT, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    WF_IDS, // Return type: java.util.Collection
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HiveTableProperties
public enum HiveTableProperties implements PropertyEnum {
    OWNER, // Return type: java.lang.String
    INPUT_FORMAT, // Return type: java.lang.String
    OUTPUT_FORMAT, // Return type: java.lang.String
    DELETED, // Return type: java.lang.Boolean
    FILE_SYSTEM_PATH, // Return type: java.lang.String
    COMPRESSED, // Return type: java.lang.Boolean
    PARTITION_COL_NAMES, // Return type: java.util.List
    CLUSTERED_BY_COL_NAMES, // Return type: java.util.List
    SORT_BY_COL_NAMES, // Return type: java.util.List
    SER_DE_NAME, // Return type: java.lang.String
    SER_DE_LIB_NAME, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    CREATED, // Return type: org.joda.time.Instant
    LAST_ACCESSED, // Return type: org.joda.time.Instant
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HiveViewProperties
public enum HiveViewProperties implements PropertyEnum {
    DELETED, // Return type: java.lang.Boolean
    QUERY_TEXT, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    CREATED, // Return type: org.joda.time.Instant
    LAST_ACCESSED, // Return type: org.joda.time.Instant
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
}

```

```

EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.mapreduce.model.JobExecutionProperties
public enum JobExecutionProperties implements PropertyEnum {
SOURCE_TYPE, // Return type: java.lang.String
JOB_ID, // Return type: java.lang.String
ENDED, // Return type: org.joda.time.Instant
INPUT_RECURSIVE, // Return type: boolean
TYPE, // Return type: java.lang.String
INPUTS, // Return type: java.util.Collection
OUTPUTS, // Return type: java.util.Collection
STARTED, // Return type: org.joda.time.Instant
PRINCIPAL, // Return type: java.lang.String
WF_INST_ID, // Return type: java.lang.String
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.mapreduce.model.JobProperties
public enum JobProperties implements PropertyEnum {
ORIGINAL_NAME, // Return type: java.lang.String
INPUT_FORMAT, // Return type: java.lang.String
OUTPUT_FORMAT, // Return type: java.lang.String
OUTPUT_KEY, // Return type: java.lang.String
OUTPUT_VALUE, // Return type: java.lang.String
MAPPER, // Return type: java.lang.String
REDUCER, // Return type: java.lang.String
SOURCE_TYPE, // Return type: java.lang.String
TYPE, // Return type: java.lang.String
WF_IDS, // Return type: java.util.Collection
NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.oozie.model.WorkflowInstanceProperties
public enum WorkflowInstanceProperties implements PropertyEnum {
TYPE, // Return type: java.lang.String
SOURCE_TYPE, // Return type: java.lang.String
CREATED, // Return type: org.joda.time.Instant
JOB_ID, // Return type: java.lang.String
STATUS, // Return type: java.lang.String
ENDED, // Return type: org.joda.time.Instant
INPUTS, // Return type: java.util.Collection
OUTPUTS, // Return type: java.util.Collection
STARTED, // Return type: org.joda.time.Instant
PRINCIPAL, // Return type: java.lang.String
WF_INST_ID, // Return type: java.lang.String
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.oozie.model.WorkflowProperties
public enum WorkflowProperties implements PropertyEnum {
TYPE, // Return type: java.lang.String
SOURCE_TYPE, // Return type: java.lang.String
WF_IDS, // Return type: java.util.Collection
}

```

```

NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.pig.model.PigFieldProperties
public enum PigFieldProperties implements PropertyEnum {
TYPE, // Return type: java.lang.String
INDEX, // Return type: int
SOURCE_TYPE, // Return type: java.lang.String
DATA_TYPE, // Return type: java.lang.String
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.pig.model.PigOperationExecutionProperties
public enum PigOperationExecutionProperties implements PropertyEnum {
SOURCE_TYPE, // Return type: java.lang.String
TYPE, // Return type: java.lang.String
ENDED, // Return type: org.joda.time.Instant
INPUTS, // Return type: java.util.Collection
OUTPUTS, // Return type: java.util.Collection
STARTED, // Return type: org.joda.time.Instant
PRINCIPAL, // Return type: java.lang.String
WF_INST_ID, // Return type: java.lang.String
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.pig.model.PigOperationProperties
public enum PigOperationProperties implements PropertyEnum {
SOURCE_TYPE, // Return type: java.lang.String
OPERATION_TYPE, // Return type: java.lang.String
SCRIPT_ID, // Return type: java.lang.String
TYPE, // Return type: java.lang.String
WF_IDS, // Return type: java.util.Collection
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.pig.model.PigRelationProperties
public enum PigRelationProperties implements PropertyEnum {
TYPE, // Return type: java.lang.String
SOURCE_TYPE, // Return type: java.lang.String
FILE_SYSTEM_PATH, // Return type: java.lang.String
SCRIPT_ID, // Return type: java.lang.String
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
}

```

```

    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.sqoop.model.SqoopExportSubOperationProperties
public enum SqoopExportSubOperationProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    INPUTS, // Return type: java.util.Collection
    FIELD_INDEX, // Return type: int
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.sqoop.model.SqoopImportSubOperationProperties
public enum SqoopImportSubOperationProperties implements PropertyEnum {
    DB_COLUMN_EXPRESSION, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    INPUTS, // Return type: java.util.Collection
    FIELD_INDEX, // Return type: int
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.sqoop.model.SqoopOperationExecutionProperties
public enum SqoopOperationExecutionProperties implements PropertyEnum {
    SOURCE_TYPE, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    ENDED, // Return type: org.joda.time.Instant
    INPUTS, // Return type: java.util.Collection
    OUTPUTS, // Return type: java.util.Collection
    STARTED, // Return type: org.joda.time.Instant
    PRINCIPAL, // Return type: java.lang.String
    WF_INST_ID, // Return type: java.lang.String
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.sqoop.model.SqoopQueryOperationProperties
public enum SqoopQueryOperationProperties implements PropertyEnum {
    SOURCE_TYPE, // Return type: java.lang.String
    INPUTS, // Return type: java.util.Collection
    QUERY_TEXT, // Return type: java.lang.String
    DB_USER, // Return type: java.lang.String
    DB_URL, // Return type: java.lang.String
    OPERATION_TYPE, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    WF_IDS, // Return type: java.util.Collection
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
}

```

```
    PARENT_PATH; // Return type: java.lang.String  
}
```

```
com.cloudera.nav.sqoop.model.SqoopTableExportOperationProperties  
public enum SqoopTableExportOperationProperties implements PropertyEnum {  
    DB_TABLE, // Return type: java.lang.String  
    SOURCE_TYPE, // Return type: java.lang.String  
    DB_USER, // Return type: java.lang.String  
    DB_URL, // Return type: java.lang.String  
    OPERATION_TYPE, // Return type: java.lang.String  
    TYPE, // Return type: java.lang.String  
    WF_IDS, // Return type: java.util.Collection  
    NAME, // Return type: java.lang.String  
    ORIGINAL_NAME, // Return type: java.lang.String  
    USER_ENTITY, // Return type: boolean  
    SOURCE_ID, // Return type: java.lang.String  
    EXTRACTOR_RUN_ID, // Return type: java.lang.String  
    PARENT_PATH; // Return type: java.lang.String  
}
```

```
com.cloudera.nav.sqoop.model.SqoopTableImportOperationProperties  
public enum SqoopTableImportOperationProperties implements PropertyEnum {  
  
    DB_TABLE, // Return type: java.lang.String  
    DB_WHERE, // Return type: java.lang.String  
    SOURCE_TYPE, // Return type: java.lang.String  
    DB_USER, // Return type: java.lang.String  
    DB_URL, // Return type: java.lang.String  
    OPERATION_TYPE, // Return type: java.lang.String  
    TYPE, // Return type: java.lang.String  
    WF_IDS, // Return type: java.util.Collection  
    NAME, // Return type: java.lang.String  
    ORIGINAL_NAME, // Return type: java.lang.String  
    USER_ENTITY, // Return type: boolean  
    SOURCE_ID, // Return type: java.lang.String  
    EXTRACTOR_RUN_ID, // Return type: java.lang.String  
    PARENT_PATH; // Return type: java.lang.String  
}
```


Cloudera Navigator Auditing Architecture

In addition to metadata management, another primary capability provided by Cloudera Navigator with governance and security teams in mind is its auditing function.

Events are the actions that occur throughout the cluster during regular operations, often accompanied by internal system messages that convey information about success or failure and include other details about the internal process. In general, events can be captured to log files or recorded in various ways.

Cloudera Manager records lifecycle events at the cluster, host, role, service, and user level, and records actions that involve licenses and parcels. Downloading a parcel is one example of a lifecycle event captured by Cloudera Manager; starting up the cluster is another. In addition to lifecycle events, Cloudera Manager also captures security-related events, such as adding users, deleting users, login failures, and login successes. See [Lifecycle and Security Auditing](#) for more information about Cloudera Manager's inherent auditing capabilities.

Cloudera Navigator generates some of its own events and also coalesces events generated by the services running on the cluster.

The Cloudera Navigator console lets you view audit events. Several pre-configured reports are available but you can use the filters and quickly create your own reports, export as CSV or JSON, or simply view in the console. For example, here is a partial export:

Recent Denied Accesses

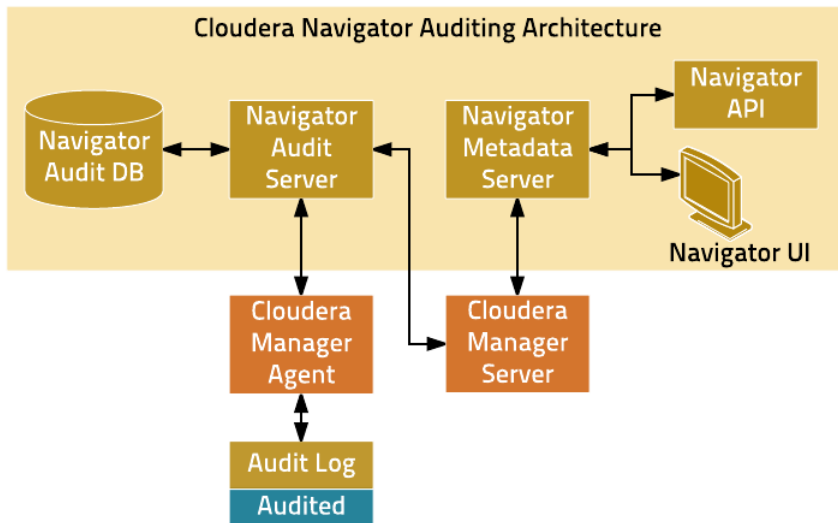
> FILTERS

Allowed = false

> Timestamp	Username	IP Address	Service Name	Operation
... Jun 23, 2017 8:57:01.127 AM	admin	172.21.2.218	Navigator	authentication
... Jun 23, 2017 8:56:43.293 AM	rogue-user	172.21.2.218	Navigator	authentication
... Jun 23, 2017 8:56:23.374 AM	nav-admin	172.21.2.218	Navigator	authentication
... Jun 23, 2017 8:56:11.233 AM	kelli	172.21.2.218	Navigator	authentication

Cloudera Navigator Auditing Architecture

The figure below shows a high level view of the Cloudera Navigator auditing architecture:



During system setup, plug-ins for the various services—HDFS, HBase, and Hive (HiveServer2, Beeswax servers) services, for example—are enabled. These plug-ins work with the service to collect and filter events emitted by the respective service, writing the events to an audit log on the local filesystem. Impala, Sentry, and the Cloudera Navigator Metadata Server also collect and filter events and write them to their respective audit log files.



Note: When upgrading Cloudera Navigator, always check the [requirements](#) for any preliminary tasks or constraints due to the service plug-ins or other factors.

Auditing Architecture In More Detail: How It Works

Here is some more detail about the auditing architecture and interaction among Cloudera Manager Agent, local log file, and Navigator Audit Server.

The Cloudera Manager Agent process on each host in the cluster:

- Monitors local audit log files
- Sends events captured in the logs to the Navigator Audit Server
- Retries sending any event that fails to transmit successfully
- Keeps track of successfully transmitted events from the logfile (offset position in the file) to prevent re-sending any already processed events after a system failure and restart
- Purges old audit logs after successful transmission to the Navigator Audit Server

Once any event is written to the audit log file (and assuming space available on the filesystem), its delivery is guaranteed. In other words, transient (in-memory) buffer handling is not involved in this part of the process. Audit logs are rotated and the Cloudera Manager Agent follows the rotation of the log.

The plug-in for each of the various services effectively writes the events to the audit log file. Policies for queue A plug-in that fails to write an event to the audit log file can either drop the event or can shut down the process in which it is running depending on the configured queue policy.

The Navigator Audit Server performs the following functions:

- Tracks and coalesces events obtained from Cloudera Manager
- Stores events to the Navigator Audit database

Exploring Audit Data Using the Cloudera Navigator Console

Required Role: [Auditing Viewer](#) (or **Full Administrator**)



Note: All steps below start from the Cloudera Navigator console.

Logging in to the Cloudera Navigator Console

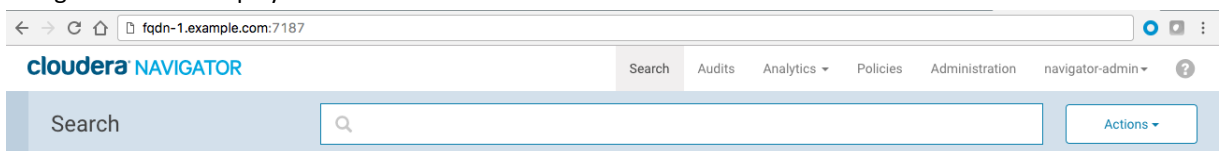
To access the Cloudera Navigator console:

1. Open your browser.
2. Navigate to the host within the cluster running the Cloudera Navigator Metadata Server role. Replace the example host name with the appropriate host name in your cluster, and replace the default port (7187) if necessary.

```
http://fqdn-1.example.com:7187/login.html
```

The login page displays.

3. Log in to the Cloudera Navigator console using the [credentials](#) assigned by your administrator. The Cloudera Navigator console displays:



By default Cloudera Navigator console opens to the Search menu as shown above.

- See [Getting Started with Cloudera Navigator](#) for a general overview of the Cloudera Navigator console.
- See [Accessing the Cloudera Navigator Console](#) for details about the alternative path to the Cloudera Navigator console, that is, from the Cloudera Manager Admin Console. That access path requires having Cloudera Manager Full Administrator or the Cloudera Manager [Navigator Administrator](#) role.

Viewing Audit Events

1. Click the **Audits** tab. By default, the Audit Events report opens, listing all events that occurred within the last hour, with the most recent at the top:

Timestamp	Username	IP Address	Service Name	Operation	Resource
Jun 25, 2017 8:21:48.013 AM	admin	172.31.114.11			
Jun 25, 2017 8:21:44.845 AM	navigator-admin	172.18.18.121	Navigator	auditReport	
Jun 25, 2017 8:21:40.833 AM	navigator-admin	172.18.18.121	Navigator	savedSearch	
Jun 25, 2017 8:21:40.358 AM	navigator-admin	172.18.18.121	Navigator	authentication	
Jun 25, 2017 8:21:40.339 AM	navigator-admin	172.31.113.66			
Jun 25, 2017 8:21:24.260 AM	admin	172.18.18.121	Navigator	savedSearch	
Jun 25, 2017 8:21:22.487 AM	oozie/nightly51...	172.31.112.215	HDFS-1	listStatus	/user/oozie/share/lib
Jun 25, 2017 8:21:10.117 AM	admin	172.18.18.121			

The Audit Events and Recent Denied Accesses reports are available by default. You create your own reports and apply a variety of filters as detailed in the next section.


Filtering Audit Events

You filter audit events by specifying a time range or adding one or more filters containing an audit event field, operator, and value.


Specifying a Time Range

1. Click the date-time range at the top right of the Audits tab.
2. Do one of the following:
 - Click a **Last *n* hours** link.
 - Specify a custom range:
 1. Click **Custom range**.
 2. In the Selected Range endpoints, click each endpoint and specify a date and time in the date control fields.
 - Date - Click the down arrow to display a calendar and select a date, or click a field and click the spinner arrows or press the up and down arrow keys.
 - Time - Click the hour, minute, and AM/PM fields and click the spinner arrows or press the up and down arrow keys to specify the value.
 - Move between fields by clicking fields or by using the right and left arrow keys.
3. Click **Apply**.

Adding a Filter

1. Do one of the following:
 - Click the  icon that displays next to a field when you hover in one of the event entries.
 - Click the **Filters** link. The Filters pane displays.
 1. Click **Add New Filter** to add a filter.
 2. Choose a field in the **Select Property...** drop-down list. You can search by fields such as username, service name, or operation. The fields vary depending on the service or role. The service name of the Navigator Metadata Server is Navigator.
 3. Choose an operator in the operator drop-down list.
 4. Type a field value in the value text field. To match a substring, use the `like` operator. For example, to see all the audit events for files created in the folder `/user/joe/out`, specify `Source like /user/joe/out`.
- A filter control with field, operation, and value fields is added to the list of filters.
2. Click **Apply**. A field, operation, and value breadcrumb is added above the list of audit events and the list of events displays all events that match the filter criteria.

Removing a Filter

1. Do one of the following:
 - Click the **x** next to the filter above the list of events. The list of events displays all events that match the filter criteria.
 - Click the **Filters** link. The Filters pane displays.
 1. Click the  at the right of the filter.
 2. Click **Apply**. The filter is removed from above the list of audit event and the list of events displays all events that match the filter criteria.

Monitoring Navigator Audit Service Health

Cloudera recommends that administrators monitor the Navigator Audit Service to ensure that it is always running. This is especially important when complete and immutable audit records may be needed for corporate governance, legal, and other purposes associated with compliance.

The Navigator Audit Service has a self-check—the Audit Pipeline Health Check—that administrators can enable to generate warning messages when the system slows down or fails. The health check keeps track of bytes—for audits processed, audits remaining to be processed, and count of errors when sending audit data from Cloudera Manager Agent process to Cloudera Manager.

Cloudera Manager generates a warning if:

- Audit bytes process = 0
- Audit bytes unprocessed != 0
- Send errors > 0 and retries unsuccessful

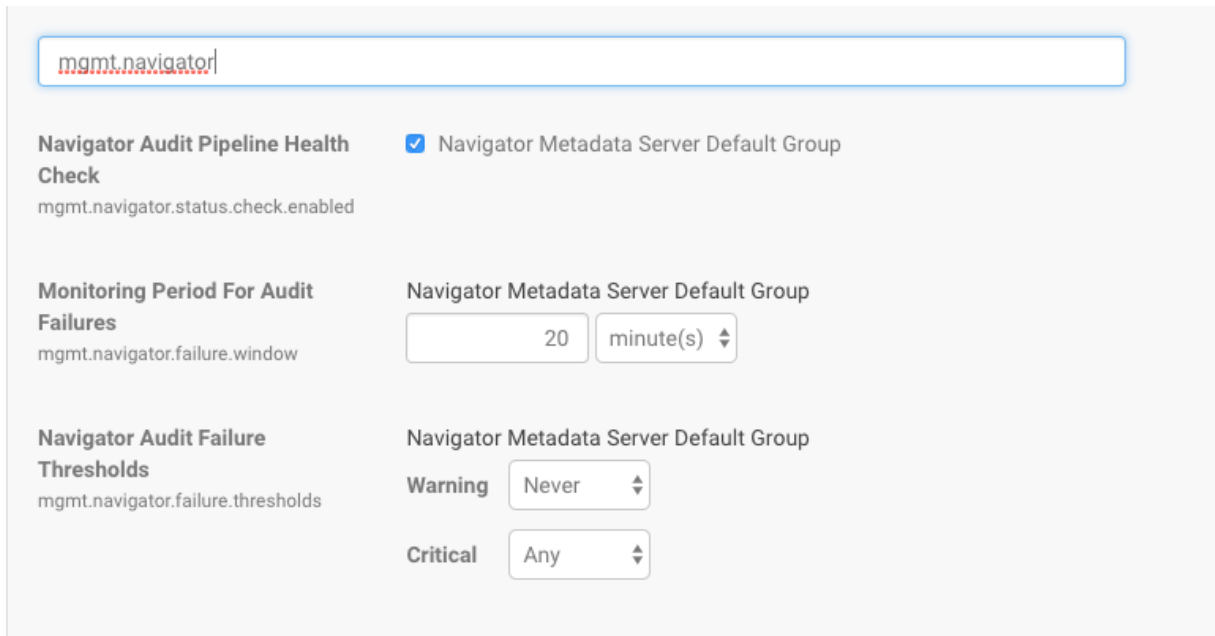
The health check is run for each service role (daemon) that generates events.

Configuring the Audit Pipeline Health Check

Cloudera Manager Required Role: [Navigator Administrator](#) (or **Full Administrator**)

Log in to the Cloudera Manager Admin Console.

1. Select **Clusters > Cloudera Management Service**.
2. Click the **Configuration** tab.
3. In the Search field, type "mgmt.navigator" to find the configuration properties, as shown below:



4. Modify the settings:

Property	Description
Navigator Audit Pipeline Health Check	Check the box to enable the healthcheck. Health check can be enabled for specific groups. By default, the health check is enabled for all groups.
Monitoring Period for Audit Failures	Amount of time within which the counts of audits processed and other metrics are evaluated before generating warnings. The default is 20 minutes.
Navigator Audit Failure Thresholds	Size (in bytes) of audit failure that generates messages. Two different thresholds are available: Warning, and Critical. Set Warning to the number of bytes of unsent audit data at which you want a warning triggered. Critical messages are sent for any failure regardless of size.

5. Click **Save Changes**.

For example, as shown in the Cloudera Manager Admin Console, the pipeline health check is enabled for all groups in the service. The failure period is set to 15 minutes, and the health check sends a warning for failures of any size and a critical error when more than 2 KiB of audit events have not been sent.

The screenshot shows the configuration for the 'mgmt.navigator' service. It is divided into three sections:

- Navigator Audit Pipeline Health Check:** The 'Health Check' is enabled (checked). The configuration key is 'mgmt.navigator.status.check.enabled'. There is a link to 'Edit Individual Values'.
- Monitoring Period For Audit Failures:** The 'Monitoring Period For Audit Failures' is set to 15 minute(s). The configuration key is 'mgmt.navigator.failure.window'. There is a link to 'Edit Individual Values'.
- Navigator Audit Failure Thresholds:** The 'Warning' threshold is set to 'Any' and the 'Critical' threshold is set to '2 KiB'. The configuration key is 'mgmt.navigator.failure.thresholds'. There is a link to 'Edit Individual Values'.

Service Audit Events

Service audit events are the events generated by a given service running on the cluster. Users with the appropriate permissions (Auditing Viewer, Full Administrator) can view audit events in the Cloudera Navigator console or by using the APIs. Audit events can include the fields listed in the tables below. The field names differ between the Navigator API and the events as they appear when [streaming](#) to Kafka or syslog.

The Cloudera Navigator console Audits includes events collected by Cloudera Manager: service **lifecycle events** (activate, create, delete, deploy, download, install, start, stop, update, upgrade, and so on) and user **security-related events** (add and delete user, login failed and succeeded). See [Lifecycle and Security Auditing](#) for more details on Cloudera Manager audit events.

Display Name	Field in API	Field in Streaming	Description
Additional Info	additional_info	additionalInfo	JSON text that contains more details about an operation performed on entities in Navigator Metadata Server.
Allowed	allowed	allowed	Indicates whether the request to perform an operation failed or succeeded. A failure occurs if the user is not authorized to perform the action.
Collection Name	collection_name	collectionName	The name of the affected Solr collection.
Database Name	database_name	db databaseName (Sentry)	For Sentry, Hive, and Impala, the name of the database on which the operation was performed.
Delegation Token ID	delegation_token_id	delegationTokenId	Delegation token identifier generated by HDFS NameNode that is then used by clients when submitting a job to JobTracker.

Display Name	Field in API	Field in Streaming	Description
Destination	dest	dst	Path of the final location of an HDFS file in a rename or move operation.
Entity ID	entity_id	entityId	Identifier of a Navigator Metadata Server entity. The ID can be retrieved using the Navigator Metadata Server API.
Event Time	timestamp	time	Date and time an action was performed. The Navigator Audit Server stores the timestamp in the timezone of the Navigator Audit Server. The Cloudera Navigator console displays the timestamp converted to the local timezone. Exported audit events contain the stored timestamp.
Family	family	family	HBase column family.
Impersonator	impersonator	impersonator	Name of user (service) that invokes an action on behalf of another user (service). Impersonator field always displays values when Sentry is not used with the cluster. For clusters that use Sentry, the Impersonator field displays values for all services other than Hive.
IP Address	ipAddress	ip	The IP address of the host where an action occurred.
Object Type	object_type	objType objectType (Sentry)	For Sentry, Hive, and Impala, the type of the object (TABLE, VIEW, DATABASE) on which operation was performed.
Operation	command	op	Commands executed by component. See Operations by Component on page 72 for details. For Cloudera Navigator operations, see Navigator Metadata Server Sub Operations on page 73.
Operation Params	operation_params	operationParams	Solr query or update parameters used when performing the action.
Operation Text	operation_text	opText operationText (Sentry)	For Sentry, Hive, and Impala, the SQL query that was executed by user. For Hue, the user or group that was added, edited, or deleted.
Permissions	permissions	perms	HDFS permission of the file or directory on which the HDFS operation was performed.
Privilege	privilege	privilege	Privilege needed to perform an Impala operation.
Qualifier	qualifier	qualifier	HBase column qualifier.
Query ID	query_id	—	The query ID for an Impala operation. (Internal use only)
Resource	resource	—	A service-dependent combination of multiple fields generated during fetch. This field is not supported for filtering as it is not persisted.
Resource Path	resource_path	path resourcePath (Sentry)	HDFS URL of Hive objects (TABLE, VIEW, DATABASE, and so on). Used for HDFS, Sentry.
Service Name	service	service	The name of the service that performed the action.
Session ID	session_id	—	Impala session ID. (Internal use only)
Solr Version	solr_version	solrVersion	Solr version number.

Display Name	Field in API	Field in Streaming	Description
Source	src	src	Path of the HDFS file or directory present in an HDFS operation.
Status	status	status	Status of an Impala operation providing more information on success or failure.
Stored Object Name	stored_object_name	name	Name of a policy, saved search, or audit report in Navigator Metadata Server.
Sub Operation	sub_operation	subOperation	Operations performed by Navigator Metadata Server are identified by subsystem (authorization, auditing, for example) and by sub-operation within that subsystem. See Navigator Metadata Server Sub Operations on page 73 for details.
Table Name	table_name	table tableName (Sentry)	For Sentry, HBase, Hive, and Impala, the name of the table on which action was performed.
Url		url	Hue only. The URL for the Hue page or API that triggered the event.
Usage Type		objUsageType	Hive only.
Username	username	user	The name of the user that performed the action.

Operations by Component

The Operation field of an audit event includes the actions taken (commands run) on the component. Operations for Cloudera Navigator (and sub-operations) are listed [Navigator Metadata Server Sub Operations](#) on page 73.



Note:

Cloudera Navigator does not capture audit events for queries that are run on HiveServer1/Hive CLI. If you want to use Cloudera Navigator to capture auditing for Hive operations, upgrade to HiveServer2 if you have not done so already.

Component	Action taken
HBase	addColumn, append, assign, balance, balanceSwitch, checkAndDelete, checkAndPut, compact, compactSelection, createTable, delete, deleteColumn, deleteTable, disableTable, enableTable, exists, flush, get, getClosestRowBefore, grant, increment, incrementColumnValue, modifyColumn, modifyTable, move, put, revoke, scannerOpen, shutdown, split, stopMaster, unassign
HDFS	append, concat, create, createSymlink, delete, fsck, getfileinfo, listSnapshottableDirectory, listStatus, mkdirs, open, rename, setOwner, setPermission, setReplication, setTimes
HiveServer2 /Beeline	ALTER_PARTITION_MERGE, ALTER_TABLE_MERGE, ALTERDATABASE, ALTERINDEX_PROPS, ALTERINDEX_REBUILD, ALTERPARTITION_FILEFORMAT, ALTERPARTITION_LOCATION, ALTERPARTITION_PROTECTMODE, ALTERPARTITION_SERDEPROPERTIES, ALTERPARTITION_SERIALIZER, ALTABLE_ADDCOLS, ALTABLE_ADDPARTS, ALTABLE_ARCHIVE, ALTABLE_CLUSTER_SORT, ALTABLE_DROPPARTS, ALTABLE_FILEFORMAT, ALTABLE_LOCATION, ALTABLE_PROPERTIES, ALTABLE_PROTECTMODE, ALTABLE_RENAME, ALTABLE_RENAMECOL, ALTABLE_RENAMEPART, ALTABLE_REPLACECOLS, ALTABLE_SERDEPROPERTIES, ALTABLE_SERIALIZER, ALTABLE_TOUCH, ALTABLE_UNARCHIVE, ALTABLEVIEW_PROPERTIES, CREATEDATABASE, CREATEFUNCTION, CREATEINDEX, CREATEROLE, CREATETABLE_AS_SELECT, CREATETABLE, CREATEVIEW, DESCDATABASE,

Component	Action taken
	DESCFUNCTION, DESCTABLE, DROPDATABASE, DROPFUNCTION, DROPINDEX, DROPROLE, DROPTABLE, DROPVIEW, EXPLAIN, EXPORT, GRANT_PRIVILEGE, GRANT_ROLE, IMPORT, LOAD, LOCKTABLE, MSCK, QUERY, REVOKE_PRIVILEGE, REVOKE_ROLE, SHOW_GRANT, SHOW_ROLE_GRANT, SHOW_TABLESTATUS, SHOW_TBLPROPERTIES, SHOWDATABASES, SHOWFUNCTIONS, SHOWINDEXES, SHOWLOCKS, SHOWPARTITIONS, SHOWTABLES, SWITCHDATABASE, UNLOCKTABLE
Hue	ADD_LDAP_GROUPS, ADD_LDAP_USERS, CREATE_GROUP, CREATE_USER, DELETE_GROUP, DELETE_USER, DOWNLOAD, EDIT_GROUP, EDIT_PERMISSION, EDIT_USER, EXPORT, NAVIGATOR_ADD_TAG, NAVIGATOR_DELETE_TAG, SYNC_LDAP_USERS_GROUPS, USER_LOGIN, USER_LOGOUT
Impala	CREATE ROLE, DELETE, DROP ROLE, GRANT <i>privilege</i> , GRANT ROLE, INSERT, Query, REVOKE <i>privilege</i> , REVOKE ROLE, SHOW GRANT ROLE, SHOW ROLE GRANT, UPDATE, Data Manipulation Language statements
Sentry	ADD_ROLE_TO_GROUP, CREATE_ROLE, DELETE_ROLE_FROM_GROUP, DROP_ROLE, GRANT_PRIVILEGE, REVOKE_PRIVILEGE
Solr	add, commit, CREATE, CREATEALIAS, CREATESHARD, DELETE, DELETEALIAS, deleteById, deleteByQuery, DELETESHARD, finish, LIST, LOAD_ON_STARTUP, LOAD, MERGEINDEXES, PERSIST, PREPRECOVERY, query, RELOAD, RENAME, REQUESTAPPLYUPDATES, REQUESTRECOVERY, REQUESTSYNCSHARD, rollback, SPLIT, SPLITSHARD, STATUS, SWAP, SYNCSHARD, TRANSIENT, UNLOAD

Navigator Metadata Server Sub Operations

Operation	Sub Operation
auditReport	createAuditReport, deleteAuditReport, fetchAllReports, updateAuditReport
authorization	deleteGroup, fetchGroup, fetchRoles, searchGroup, updateRoles
metadata	fetchAllMetadata, fetchMetadata, updateMetadata
policy	createPolicy, deletePolicy, deletePolicySchedule, fetchAllPolicies, fetchPolicySchedule, updatePolicy, updatePolicySchedule
savedSearch	createSavedSearch, deleteSavedSearch, fetchAllSavedSearches, fetchSavedSearch, updateSavedSearch

Cloudera Navigator Audit Event Reports

Required Role: [Auditing Viewer](#) (or **Full Administrator**)

Cloudera Navigator provides two default reports for [audit events](#) (Recent Denied Accesses, for example) but you can create new reports, apply various filters to fine tune the results displayed, save the filtered report specification for future use, and export (download) any report (CSV, JSON file format). Metadata about the audit reports you create and save is recorded in the [Cloudera Navigator Metadata Server](#).



Note: All steps below start from the Cloudera Navigator console.

Creating Audit Event Reports

Selecting the Audit menu in the Cloudera Navigator console displays the Audit Events report. This report displays all audit events captured in the last 1 hour. You can modify the filters configured for this report and save it, giving it a new name, as follows.

1. To save a filtered version of the Audit Events report:
 - a. Optionally specify [filters](#).
 - b. Click **Save As Report**.
- Create a new report by clicking **New Report**.

The screenshot shows the Cloudera Navigator interface. On the left, there is a 'Reports' pane with a search box 'Find a report...' and a list of reports: 'Audit Events' and 'Recent Denied Accesses'. On the right, there is a 'New Report' dialog box. It contains the following fields and controls:

- Name ***: A text input field.
- Default time range**: A dropdown menu currently set to 'Last hour'.
- Filters**: A section containing a dropdown menu 'Select Property...', an equals sign (=), a text input field, and two buttons: a minus sign (-) and a plus sign (+).

2. Enter a report name.
3. In the **Default time range** field, specify a relative time range. If you had specified a custom absolute time range before selecting **Save As Report**, the *custom absolute time range is discarded*.
4. Optionally add [filters](#).
5. Click **Save**.

Editing Audit Event Reports

1. In the left pane, click a report name.
2. Click **Edit Report**.
3. In the **Default time range** field, specify a relative time range. If you had specified a custom absolute time range before selecting **Save As Report**, the *custom absolute time range is discarded*.
4. Optionally add [filters](#).
5. Click **Save**.

Downloading Audit Event Reports

You can download audit event reports in the Cloudera Navigator console or using in CSV and JSON formats. An audit event contains the following fields:

- timestamp
- service
- username
- ipAddress
- command
- resource
- allowed
- [operationText]
- serviceValues

The values for `resource` and `serviceValues` fields depend on the type of the service. In addition, Hive, Hue, Impala, and Sentry events have the `operationText` field, which contains the operation string. See [Service Audit Events](#) on page 70.

In addition to downloading audit events, you can configure the Navigator Audit Server to publish audit events to a Kafka topic or syslog. See [Publishing Audit Events](#).

Downloading Audit Event Reports from

1. Do one of the following:

- Add [filters](#).
- In the left pane, click a report name.

2. Select **Export** > *format*, where *format* is CSV or JSON.

Downloading Audit Events Using the Audit API

You can filter and download audit events using the [Cloudera Navigator APIs](#).

Hive Audit Events Using the Audit API

To use the API to download the audits events for a service named `hive`, use the `audits` endpoint with the `GET` method:

```
http://fqdn-n.example.com:port/api/APIversion/audits/?parameters
```

where `fqdn-n.example.com` is the host running the Navigator Metadata Server role instance listening for HTTP connections at the specified `port` number (7187 is the default port number). `APIversion` is the running version of the API as indicated in the footer of the API documentation (available from the Help menu in the Navigator console) or by calling `http://fqdn-n.example.com:port/api/version`.

For example:

```
curl http://node1.example.com:7187/api/v12/audits/?query=service%3D%3Dhive\
&startTime=1431025200000&endTime=1431032400000&limit=5&offset=0&format=JSON&attachment=false\
-X GET -u username:password
```

The `startTime` and `endTime` parameters are required and must be specified in [epoch time](#) in milliseconds.

The request could return the following JSON items:

```
[ {
  "timestamp" : "2015-05-07T20:34:39.923Z",
  "service" : "hive",
  "username" : "hdfs",
  "ipAddress" : "12.20.199.170",
  "command" : "QUERY",
  "resource" : "default:sample_08",
  "operationText" : "INSERT OVERWRITE \n TABLE sample_09 \nSELECT \n
sample_07.code,sample_08.description \n FROM sample_07 \n JOIN sample_08 \n WHERE
sample_08.code = sample_07.code",
  "allowed" : true,
  "serviceValues" : {
    "object_type" : "TABLE",
    "database_name" : "default",
    "operation_text" : "INSERT OVERWRITE \n TABLE sample_09 \nSELECT \n
sample_07.code,sample_08.description \n FROM sample_07 \n JOIN sample_08 \n WHERE
sample_08.code = sample_07.code",
    "resource_path" : "/user/hive/warehouse/sample_08",
    "table_name" : "sample_08"
  }
}, {
  "timestamp" : "2015-05-07T20:33:50.287Z",
  "service" : "hive",
  "username" : "hdfs",
  "ipAddress" : "12.20.199.170",
  "command" : "SWITCHDATABASE",
  "resource" : "default:",
  "operationText" : "USE default",
  "allowed" : true,
  "serviceValues" : {
    "object_type" : "DATABASE",
    "database_name" : "default",
    "operation_text" : "USE default",
    "resource_path" : "/user/hive/warehouse",
    "table_name" : ""
  }
}, {
  "timestamp" : "2015-05-07T20:33:23.792Z",
  "service" : "hive",
```

```

"username" : "hdfs",
"ipAddress" : "12.20.199.170",
"command" : "CREATETABLE",
"resource" : "default:",
"operationText" : "CREATE TABLE sample_09 (code string,description string) ROW FORMAT
DELIMITED FIELDS TERMINATED BY '\\t' STORED AS TextFile",
"allowed" : true,
"serviceValues" : {
  "object_type" : "DATABASE",
  "database_name" : "default",
  "operation_text" : "CREATE TABLE sample_09 (code string,description string) ROW
FORMAT DELIMITED FIELDS TERMINATED BY '\\t' STORED AS TextFile",
  "resource_path" : "/user/hive/warehouse",
  "table_name" : ""
}
}
]

```

Downloading HDFS Directory Access Permission Reports

Minimum Required Role: [Cluster Administrator](#) (also provided by **Full Administrator**)

For each HDFS service, you can download a report that details the HDFS directories a group has permission to access.

1. In the Cloudera Manager Admin Console, click **Clusters** > **ClusterName** > **Reports**.
2. In the Directory Access by Group row, click **CSV** or **XLS**. The Download User Access Report pop-up displays.
 - a. In the pop-up, type a group and directory.
 - b. Click **Download**. A report of the selected type will be generated containing the following information – path, owner, permissions, and size – for each directory contained in the specified directory that the specified group has access to.

Cloudera Navigator Auditing Use Cases

The Navigator Audit Server tracks the actions performed on the data in a Hadoop cluster. By applying filters on these actions, you can use Cloudera Navigator auditing to view specific information and answer a variety of questions about data and user actions; for example:

- What was a specific user doing on a specific day?
- Who deleted a particular directory?
- What happened to data in a production database, and why is it no longer available?

To answer these questions using Navigator auditing, you begin by [logging into the Cloudera Navigator data management UI](#) and clicking the **Audits** tab. Cloudera Navigator displays a list of all audit events for the last hour. The following use cases describe how Navigator can answer some specific questions about data and users.

What Did a User Do on a Specific Day?

In some cases, you may want to identify actions that a specific user performed during a period of time. To determine a user's actions for a time period, you use filters to first specify the user and then define the time period.

The following example identifies the actions of the user named **navigator_user** on June 9, 2016:

1. Filter the list of events for a specific user:
 - a. Click **Filters**.
 - b. Select **Select Property...** > **Username**.
 - c. In the field to the right of =, type the username and click **Apply**. The username filter is added to the list of filters, and the list of events is filtered and reloaded. This filter specifies the user **cmjobuser**.

2. Filter the list of events for a specific date and time:

- a. Click the date-time field at the top right of the Audit Events page. A set of links display with relative time periods (**Last hour**, **Last 2 hours**, and so on) and a **Custom Range** link that you can use to specify an absolute time range. The Selected Range field displays the currently selected range, which by default is the last hour of the current day.
- b. To choose a specific day, click **Custom Range**. The Selected Range field is enabled for input.
- c. Use the field controls to choose specific dates and times. The following figure shows the selections for November 3, 2016, 3:00 PM to November 3, 2016, 4:00 PM.

- d. Click **Apply**.

The following figure shows the first page of the filter results: audit events for the user **cmjobuser** during the 24 hour period from June 9, 2016 12:00 a.m. to June 10, 2016 12:00 a.m.

Audit Events Actions ▾

> FILTERS > NOV 3, 2016 3:00 PM - NOV 3, 2016 4:00 PM

> Timestamp	Username	IP Address	Service N...	Operation	Resource
> Nov 3, 2016 3:06:54.595 PM	cmjobuser	172.31.8.56	HDFS-1	delete	/user/cmjobuser/.staging/job_1474...
> Nov 3, 2016 3:06:53.565 PM	cmjobuser	172.31.8.56	HDFS-1	rename	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.563 PM	cmjobuser	172.31.8.56	HDFS-1	rename	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.561 PM	cmjobuser	172.31.8.56	HDFS-1	rename	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.559 PM	cmjobuser	172.31.8.56	HDFS-1	setPermi..	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.539 PM	cmjobuser	172.31.8.56	HDFS-1	create	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.536 PM	cmjobuser	172.31.8.56	HDFS-1	open	/user/cmjobuser/.staging/job_1474...
> Nov 3, 2016 3:06:53.534 PM	cmjobuser	172.31.8.56	HDFS-1	getFileinfo	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.532 PM	cmjobuser	172.31.8.56	HDFS-1	getFileinfo	/user/cmjobuser/.staging/job_1474...

Who Deleted Files from Hive Warehouse Directory?

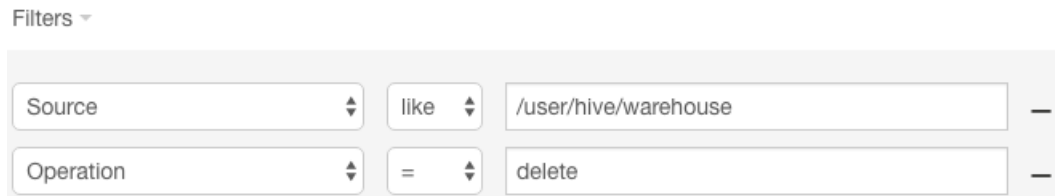
The Hive warehouse directory is usually set to `/user/hive/warehouse`. In this example, files have been deleted from the directory and you want to identify who removed them.

To determine who deleted files from this directory, use filters in Cloudera Navigator to do the following:

1. Filter the list of events for the source `/user/hive/warehouse`:
 - a. Click **Filters**.
 - b. Select **Select Property... > Source**.
 - c. In the operator field, select **like**.

- d. In the empty field to the right of **like**, type `/user/hive/warehouse` and click **Apply**. The source filter is added to the list of filters and the list of events is filtered and reloaded.
2. Filter the list of events for the delete operation:
 - a. Click **Add New Filter**.
 - b. Select **Select Property... > Operation**.
 - c. In the operator field, select `=`.
 - d. In the empty field to the right of `=`, type `delete` and click **Apply**. The operation filter is added to the list of filters and the list of events is filtered and reloaded.

The following figure shows the resulting filters.



The following figure shows the results of the filters: **navigator_user** deleted or attempted to delete (indicated by the red text) the displayed resources from the Hive warehouse directory during the 30-day period from May 28, 2016 to June 27, 2016.

Audit Events						Save As Report
Filters		Source like /user/hive/warehouse x		Operation = delete x		May 28 2016 12:18 PM - Jun 27 2016 12:18 PM
Export						1 - 5
Timestamp	Username	IP Address	Service Name	Operation	Resource	
Jun 9 2016 7:46 PM	navigator_user	10.17.207.26	hdfs	delete	/user/hive/warehouse/sample_09/hive-staging_hive_2016-06-09	
Jun 9 2016 1:39 PM	navigator_user	10.17.207.26	hdfs	delete	/user/hive/warehouse/sample_09/hive-staging_hive_2016-06-09	
Jun 9 2016 1:39 PM	navigator_user	10.17.207.26	hdfs	delete	/user/hive/warehouse/sample_09/hive-staging_hive_2016-06-09	
Jun 9 2016 1:19 PM	navigator_user	10.17.207.26	hdfs	delete	/user/hive/warehouse/sample_09/000000_0	
Jun 9 2016 1:19 PM	navigator_user	10.17.207.26	hdfs	delete	/user/hive/warehouse/sample_09/hive-staging_hive_2016-06-09	

What Happened to Data in the Database?

Typically, data in the database is partitioned into folders or files labeled by date. In this example, data from 2015 is missing from the production database, and you want to find out what happened to it. You can use Cloudera Navigator to determine what happened to data that was created during this period of time.

Data created in 2015 has the string "2015" in the filename. To determine what happened to the data stored in folders and files in the year 2015, do the following:

1. Filter the list of events for sources containing the string "2015":
 - a. Click **Filters**.
 - b. Select **Select Property... > Source** to specify the path of an HDFS file or directory.
 - c. In the operator field, select **like**.
 - d. In the empty field to the right of **like**, type `2015` and click **Apply**. The source filter is added to the list of filters, and the list of events is filtered and reloaded.
2. Filter the list of events for the delete operation:
 - a. Click **Add New Filter**.
 - b. Select **Select Property... > Operation**.
 - c. In the operator field, select `=`.

d. In the empty field to the right of =, type `delete` and click **Apply**. The operation filter is added to the list of filters and the list of events is filtered and reloaded.

3. Set the date range to one year:

- a. Click the date-time field at the top right of the Audit Events page.
- b. To set the range to be the last year, click **Custom Range**. The Selected Range field is enabled for input.
- c. In the left date field, use the field controls to specify a date one year ago.
- d. Click **Apply**.

The following figure shows the resulting filters.

Filters ▾

The screenshot shows a filter configuration interface with two filter rows. The first row has a dropdown menu set to 'Source', a relationship dropdown set to 'like', and a text input field containing '2015'. The second row has a dropdown menu set to 'Operation', a relationship dropdown set to '=', and a text input field containing 'delete'. Each row has a minus sign on the right side to remove the filter.

The following figure shows the results of the filter application. During the last year, the user **hdfs** deleted the directories with names that contain "2015":

Audit Events

[Save As Report](#)

Filters ▾ Source like 2015 ✕ Operation = delete ✕

Jun 13 2015 12:36 PM - Jun 13 2016 12:36 PM ▾

Export ▾ < 1 - 4 >

Timestamp	Username	IP Address	Service Name	Operation	Resource
Jun 13 2016 11:45 AM	hdfs	10.17.207.26	hdfs	delete	/user/navigator_user/2015_11_21
Jun 13 2016 11:45 AM	hdfs	10.17.207.26	hdfs	delete	/user/navigator_user/2015_11_20
Jun 13 2016 11:45 AM	hdfs	10.17.207.26	hdfs	delete	/user/navigator_user/2015_11_19
Jun 13 2016 11:45 AM	hdfs ▾	10.17.207.26 ▾	hdfs ▾	delete ▾	/user/navigator_user/2015_11_18

Cloudera Navigator Provenance Use Case

A number of business decisions and transactions rely on the verifiability of the data used in those decisions and transactions. Data-verification questions might include:

- How was this mortgage credit score computed?
- How can I prove that this number on a sales report is correct?
- What data sources were used in this calculation?

You can use Cloudera Navigator to answer these and other questions about your data. Using metadata and lineage, you can get track the life of the data to verify its **provenance**—that is, determine its origin.

How Can I Verify a Value in a Table?

A number of business transactions require you to verify that information is correct and that it is derived from a reliable source. For example, if you work in a sales organization, you might verify that information in sales reports is accurate, that you can trust the contents, and that you can identify the origin of the information.

The following example shows how you can verify information in a field named **s_neighbor** by tracing it to its source. You will replace the fields and other information in this example with the actual information that you want to verify.

1. [Log into the Cloudera Navigator data management UI](#) and click the **Search** tab.
2. Type **s_neighbor** in the search box.

The screenshot shows the Cloudera Navigator Search interface. At the top, there is a search bar with the text 's_neighbor' and a search icon. To the right of the search bar is an 'Actions' dropdown menu. Below the search bar, the interface is divided into two main sections: 'Filters' on the left and '4 results' on the right. The 'Filters' section includes three expandable categories: 'SOURCE TYPE' with 'Hive' selected (4 results), 'TYPE' with 'Field' selected (4 results), and 'OWNER' with an 'Add New Value' link. The '4 results' section displays a list of search results for the field 's_neighbor'. The first result is '/default/salesdata', the second is '/default/top_10' (highlighted in yellow), the third is '/default/sales_by_region', and the fourth is '/default/count_by_region'. Each result has a small icon to its left.

You see four instances of the **s_neighbor** field.

3. View details of the field in the **top_10** table by clicking **s_neighbor** in the entry with the Parent Path `/default/top10`.

s_neighbor Actions ▾

Field
Hive

Parent: [top_10](#)
Data Type: string

1 Inputs
0 Outputs

Tags (1) Edit

sensitive

Inputs (1) 🔍

salesdata

Outputs (0)

Technical Metadata

Source Type	Hive
Type	Field
Parent	top_10
Data Type	string
Parent Path	/default/top_10
Source	HIVE-1
Classname	Hive Column
Package Name	nav

You see that the parent table is **top_10**, and the input or upstream source of the data is the **salesdata** database.

Where did **salesdata** come from originally? It was imported using sqoop, with syntax similar to the following; actual arguments vary:

```
> sqoop import-all-tables
  -m {{cluster_data.worker_node_hostname.length}} \
  --connect jdbc:mysql://{{cluster_data.manager_node_hostname}}:3306/retail_db \
  --username=admin \
  --password=password \
  --compression-codec=snappy \
  --as-parquetfile \
  --warehouse-dir=/user/hive/warehouse \
  --hive-import
```

4. To see a graphical representation of the relationships among the entities:

a. Click the **Lineage** tab.

b. In Lineage Options, select **Operations** and clear any other check boxes.

s_neighbor Actions ▾

Details Lineage

Lineage Options

- Operations
- Control Flow Relations
- Only Upstream Downstream
- Latest Partition and Operation Execution
- Deleted Entities

Search

Click an entity or link to view more information.

Lineage Graph:

```

graph LR
    salesdata[salesdata] --> create_table[create table top_10 as select s_neighbor, avg(s_pri...)]
    create_table --> top_10[top_10]
    create_table --> s_neighbor[s_neighbor]
  
```

See that **s_neighbor** can be traced back to the original table **salesdata**.

- 5. Click the operation entity in the center of the lineage diagram, and see details about it on the lower right side of the lineage window.

SELECTED ENTITY
**create table top_10 as select s_neighbor,
avg(s_pri...**

[View Details](#) [View Lineage](#)

Source Type	Impala
Type	Operation
Query Text	create table top_10 ... ⓘ
Source	IMPALA-1
Classname	Impala Query
Package Name	nav

Information about the selected entity indicates that the operation is an Impala query. Click the information icon on the Query Text line to see the entire query. This query was used to derive **top_10** from the original table.

Cloudera Navigator Lineage Diagram Reference


















Required Role: [Lineage Administrator](#) (or [Metadata Administrator](#), [Full Administrator](#))


Cloudera Navigator provides an automatic collection and easy visualization of upstream and downstream data lineage to verify reliability. For each data source, it shows, down to the column level within that data source, what the precise upstream data sources were, the transforms performed to produce it, and the impact that data has on downstream artifacts.

A **lineage diagram** is a directed graph that depicts an extracted entity and its relations with other entities. A lineage diagram is limited to 400 entities. Once that limit is reached, certain entities display as a "hidden" icon.

Entities

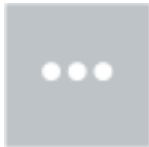
In a lineage diagram, entity types are represented by icons:

HDFS		Pig	
<ul style="list-style-type: none"> • File • Directory 	<ul style="list-style-type: none"> •  •  	<ul style="list-style-type: none"> • Table • Pig field • Pig operation, operation execution 	<ul style="list-style-type: none"> •  •  • 
Hive and Impala		Spark (Supported in CDH 5.11 and higher.) Spark Lineage information is produced only for data that is read/written and processed using the Dataframe and SparkSQL APIs. Lineage is not available for data that is read/written or processed using Spark's RDD APIs. To turn metadata extraction off or on, see Enabling and Disabling Metadata Extraction.)	
<ul style="list-style-type: none"> • Table • Field • Operation, suboperation, execution • Impala operation, suboperation, execution 	<ul style="list-style-type: none"> •  •  •  •  	<ul style="list-style-type: none"> • Operation, operation execution. (Spark RDDs and aggregation operations are not included in the diagrams.) 	<ul style="list-style-type: none"> • 
MapReduce and YARN		Sqoop	
<ul style="list-style-type: none"> • MapReduce operation and operation execution • YARN operation and operation execution 	<ul style="list-style-type: none"> •  •  	<ul style="list-style-type: none"> • Operation, suboperation, execution 	<ul style="list-style-type: none"> • 
Oozie		S3	
<ul style="list-style-type: none"> • Operation, operation execution 	<ul style="list-style-type: none"> •  	<ul style="list-style-type: none"> • Directory • File • S3 Bucket 	<ul style="list-style-type: none"> •  •  • 

Hidden			
<ul style="list-style-type: none">  	See Viewing the Lineage of Hidden Entities on page 86.		

Important: Hive entities include tables that result from Impala queries and Sqoop jobs.

In the following circumstances, the entity type icon appears as



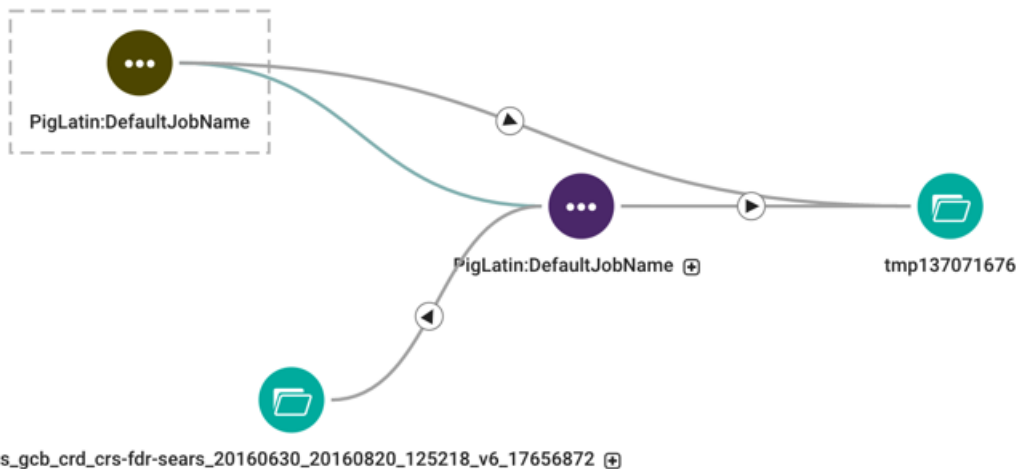
- The entity has not yet been extracted. In this case,



is eventually replaced with the correct entity icon after the entity is extracted and linked in Navigator. For information on how long it takes for newly created entities to be extracted, see [Metadata Extraction and Indexing](#).

- A Hive entity has been deleted from the system before it could be extracted.

The following lineage diagram illustrates the relations between the YARN operation `DefaultJobName` and Pig script `DefaultJobName` and the source file in the `ord_us_gcb_crd_crs-fdr-sears` folder and destination folder `tmp137071676`:







Relations

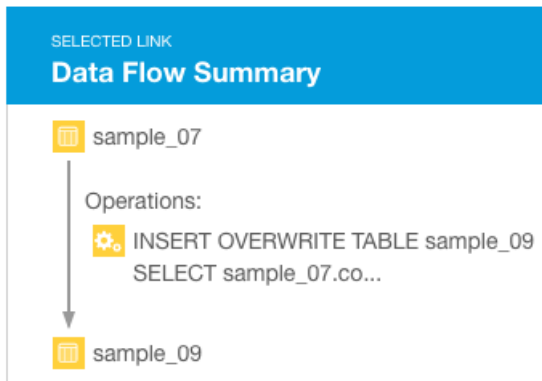
Relations between the entities are represented graphically by lines, with arrows indicating the direction of the data flow. Navigator supports the following types of relations:

Relation Type	Description
Data flow	Describes a relation between data and a processing activity; for example, between a file and a MapReduce job or vice versa.
Parent-child	Describes a parent-child relation. For example, between a directory and a file.

Relation Type	Description
Logical-physical	Describes the relation between a logical entity and its physical entity. For example, between a Hive query and a MapReduce job.
Instance of	Describes the relation between a template and its instance. For example, an operation execution is an instance of operation. Instance of relations are never visualized in the lineage, however you can navigate between template and instance lineage diagrams. See Displaying an Instance Lineage Diagram on page 92 and Displaying the Template Lineage Diagram for an Instance Lineage Diagram on page 92.
Control flow	Describes a relation where the source entity controls the data flow of the target entity. For example, between the columns used in an <code>insert</code> clause and the <code>where</code> clause of a Hive query.

Lineage diagrams contain the following line types:

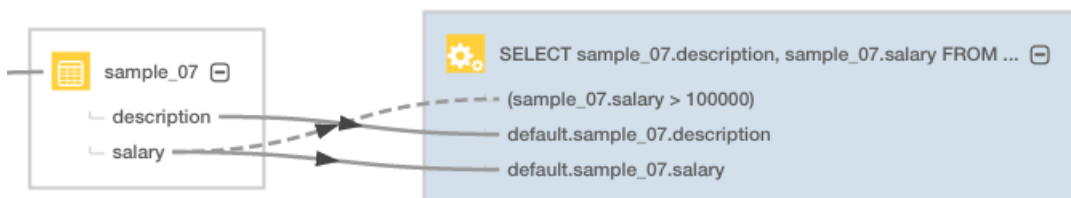
- Solid () represents a "data flow" relationship, indicating that the columns appear (possibly transformed) in the output (when directional with arrow) and "logical- physical" (when no arrow). For example, a solid line appears between the columns used in a `select` clause.
- Dashed () represents a "control flow" relationship, indicating that the columns determine which rows flow to the output. For example, a dashed line appears between the columns used in an `insert` or `select` clause and the `where` clause of a Hive query. Control flow lines are hidden by default. See [Filtering Lineage Diagrams](#) on page 87.
- Blue () represents a selected link.
- Green () represents a summary link that contains operations. When you click the link, the link turns blue (for selected) and the nested operations display in the selected link summary:



The following query:


```
SELECT sample_07.description, sample_07.salary FROM sample_07
WHERE ( sample_07.salary > 100000)
ORDER BY sample_07.salary DESC LIMIT 1000
```

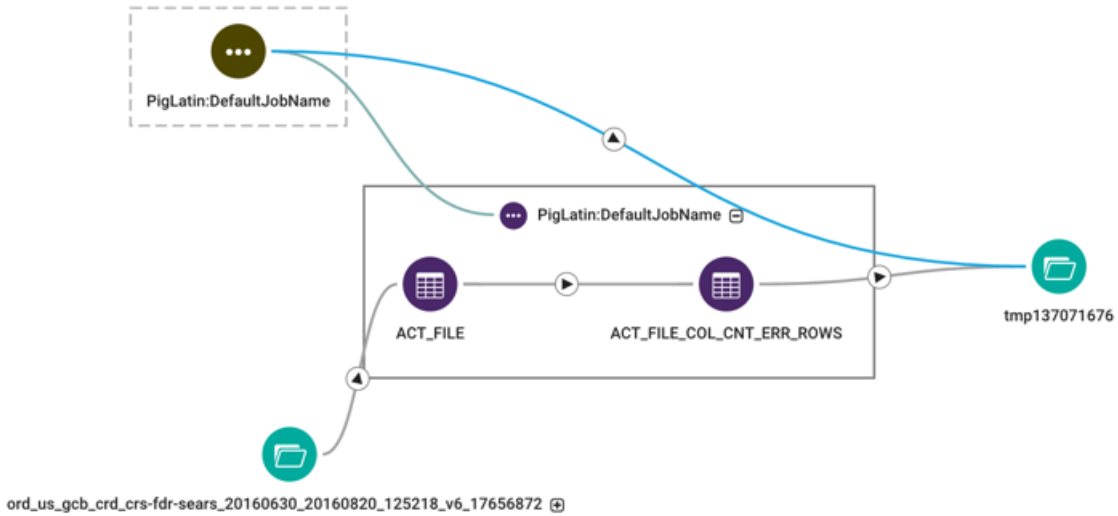
has solid, directed lines between the columns in the `select` clause and a dashed line between the columns in the `where` clause:



Manipulating Lineage Diagrams

Expanding Entities

You can click a  icon in a parent entity to display its child entities. For example, you can click an Oozie job to display its child Pig script and the Pig script to display its child tables:



Modifying Lineage Layout

- To improve the layout of a lineage diagram, you can drag entities (like tmp137071676) located outside a parent box.
- Use the mouse scroll wheel or the



control to zoom the lineage diagram in and out.

- You can move an entire lineage diagram in the lineage pane by pressing the mouse button and dragging it.

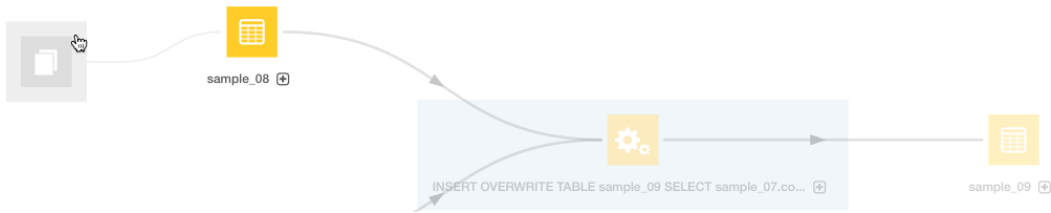
Viewing the Lineage of Hidden Entities

Lineage that is not fully traversed (that is, you do not see a subset of the actual lineage) is illustrated by the



icon. This icon displays when the lineage diagram has more than 400 entities. For example:

One or more links from sample_08 were not included in this lineage. To explore these links further, [view the lineage of sample_08](#)



To view the lineage of hidden entities, select the hidden entity and click **view the lineage** in the box on the right to display a new lineage centered around that entity. After clicking the link, you would see the following:



Filtering Lineage Diagrams

To reduce the time and resources required to render large lineage diagrams, you can filter out classes of entities and links by selecting checkboxes in the **Lineage Options** box on the right of the diagram. The following are the default selections:

▼ Lineage Options

- Operations
- Control Flow Relations
- Only Upstream Downstream
- Deleted Entities
- Latest Partition and Operation

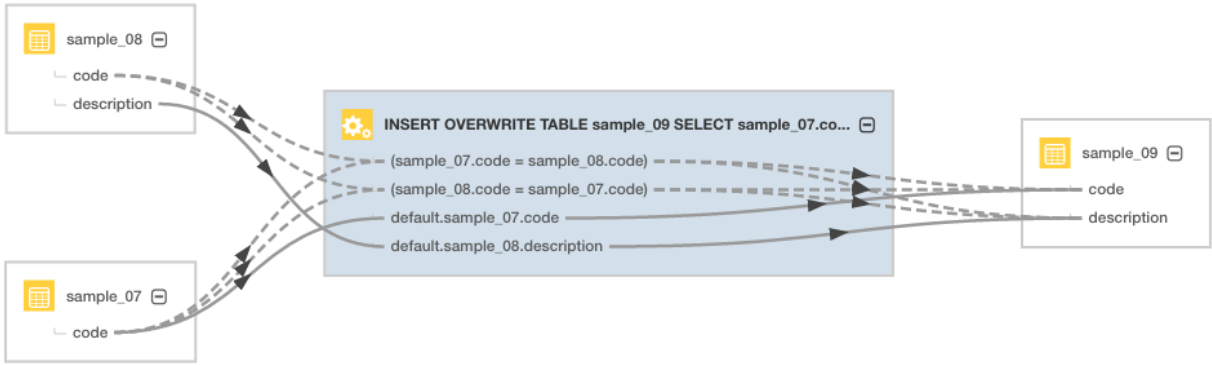
The **Only Upstream/Downstream** filter allows you to filter out entities and links that are input (upstream) to and output (downstream) from another entity.

Use the **Latest Partition and Operation** filter to reduce rendering time when you have similar partitions created and operations performed periodically. For example, if Hive partitions are created daily, the filter allows you to display only the latest partition.

Note: This filter applies only to metadata collected with Navigator Metadata Server installed with Cloudera Manager version 5.12 or later. Operations against multiple partitions that were collected by earlier versions will not collapse into a single partition.

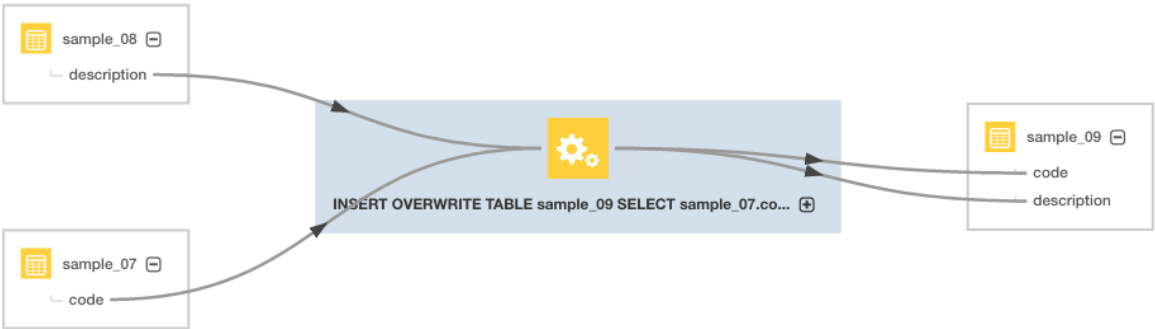
Filter Example

If you display the lineage of the `sample_09` table with no filtering options selected (other than hiding deleted items), the lineage appears as follows.

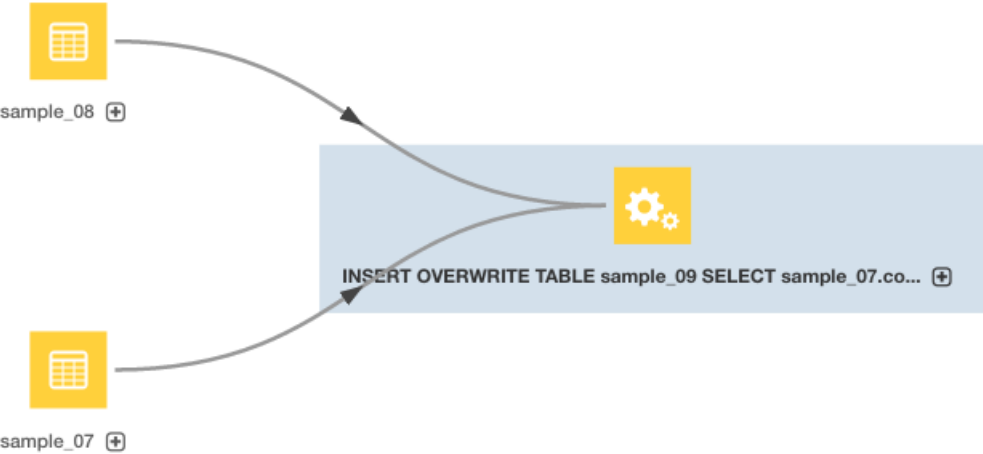


Subsequent diagrams show the result of using each supported filter type:

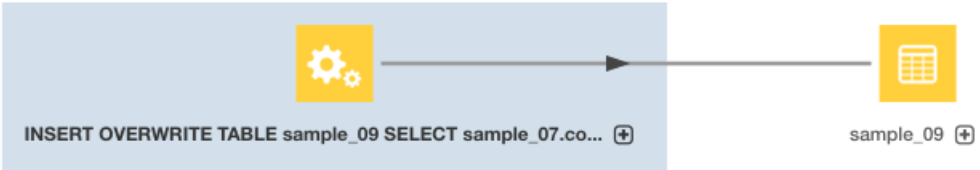
- **Control Flow Relations** - The operation is collapsed and control flow links are hidden.



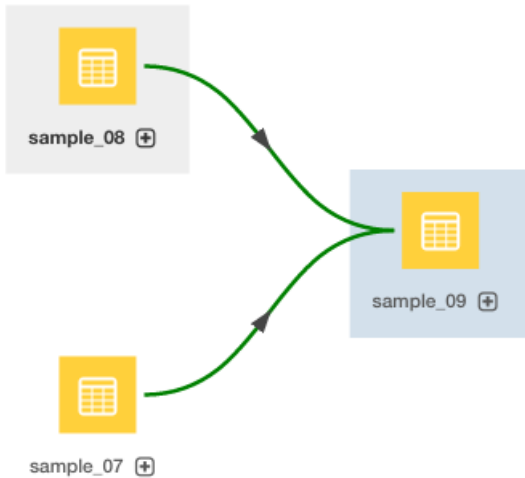
- **Show Upstream and Show Downstream** - The operation is collapsed and only upstream entities and links are shown. The output table is hidden.



Here, the operation is collapsed and only downstream entities and links are shown. The input tables are hidden.

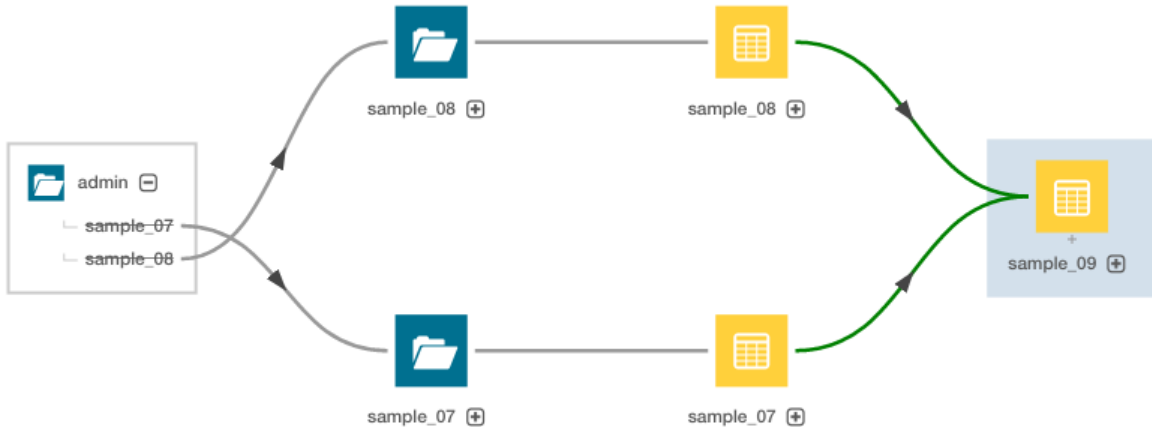


- **Operations** - In the diagram, the operation is hidden.



The green links indicate that one or more operations are collapsed into the links.

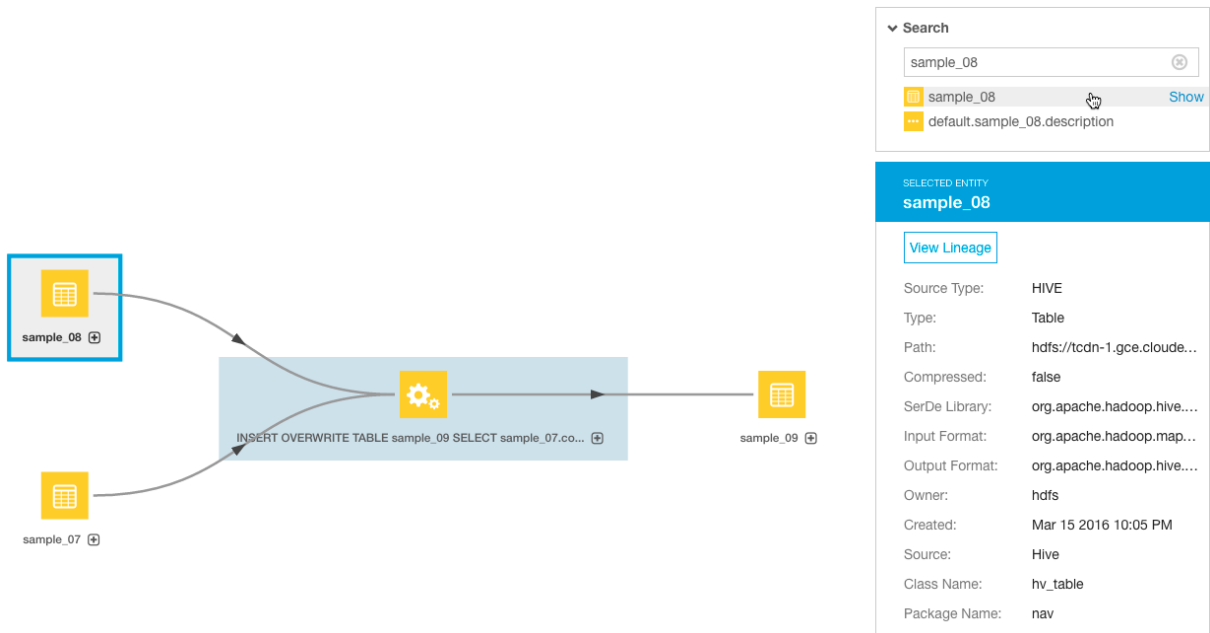
- **Deleted Entities** - Here, the operation is hidden but deleted entities are displayed.



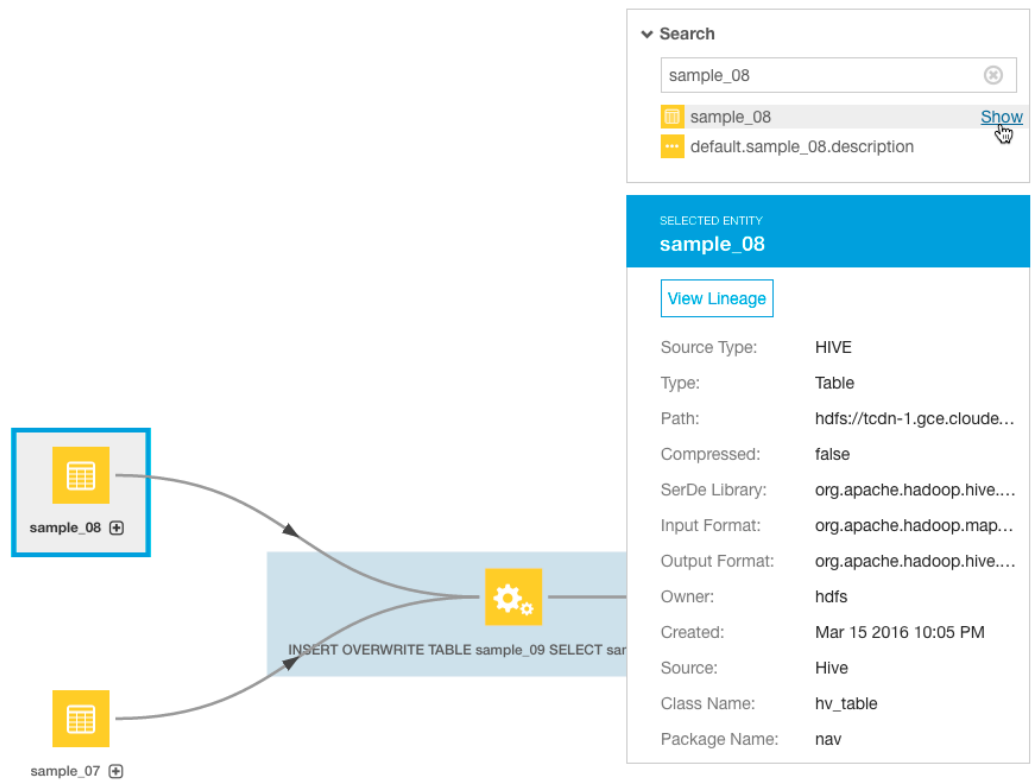
Searching a Diagram

You can search a lineage diagram for an entity by doing the following:

1. In the Search box at the right of the diagram, type an entity name. A list of matching entities displays below the box.
2. Click an entity in the list. A blue box is drawn around the entity and the entity details display in a box below the Search box.



3. Click the **Show** link next to the entity. The selected entity moves to the center of the diagram.



4. Optionally, click the **View Lineage** link in the entity details box to view the lineage of the selected entity.

Displaying a Template Lineage Diagram

A **template lineage diagram** contains template entities, such as jobs and queries, that can be instantiated, and the input and output entities to which they are related.

To display a template lineage diagram:

1. Perform a metadata [search](#).

2. In the list of results, click an entity. The entity Details page displays. For example, when you click the `sample_09` result entry:

 Hive `sample_09`
 Type: Table Parent Path: /default Path: hdfs://tcdn1-1.ent.cloudera.com:8020/user/hive/warehouse/sample_09 Owner: hdfs
 Created: Apr 8 2015 11:04 AM Source: Hive

the Search screen is replaced with a Details page that displays the entity property sheet:

`sample_09` Actions ▾ Details Lineage

Technical Metadata

Source Type: HIVE
 Type: Table
 Parent Path: /default
 Path: hdfs://nightly57-1.gce.cloudera.com:80...
 Compressed: false
 SerDe Library: org.apache.hadoop.hive.serde2.lazy.La...
 Input Format: org.apache.hadoop.mapred.TextInputF...
 Output Format: org.apache.hadoop.hive.qi.io.HiveIgnor...
 Owner: admin
 Created: Mar 18 2016 10:15 AM
 Source: HIVE-1
 Class Name: hv_table
 Package Name: nav

Managed Metadata

No metadata available

Schema

- code string
- description string

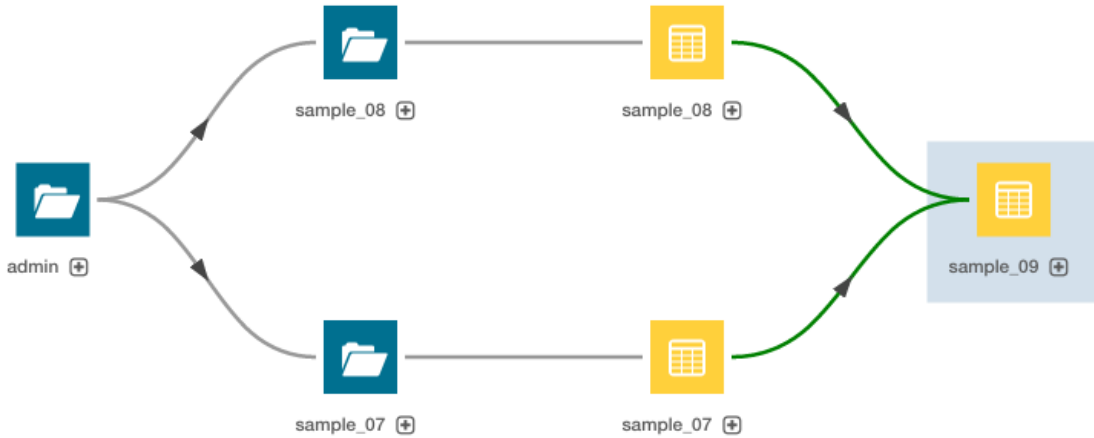
Custom Metadata

No metadata available

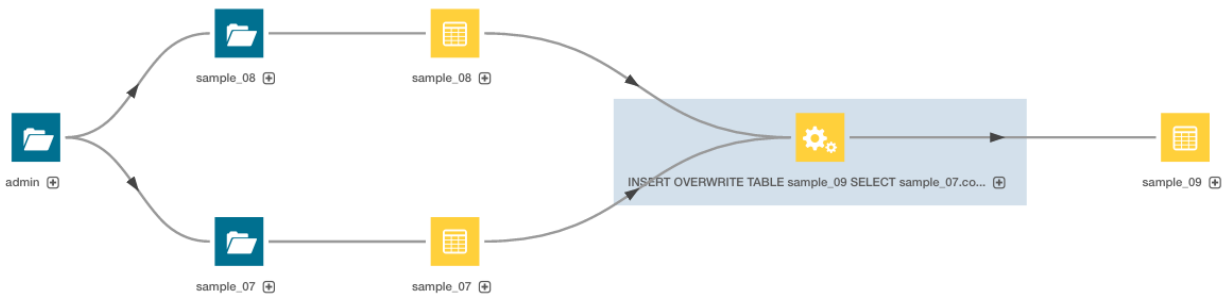
Inputs


- sample_07
- sample_08

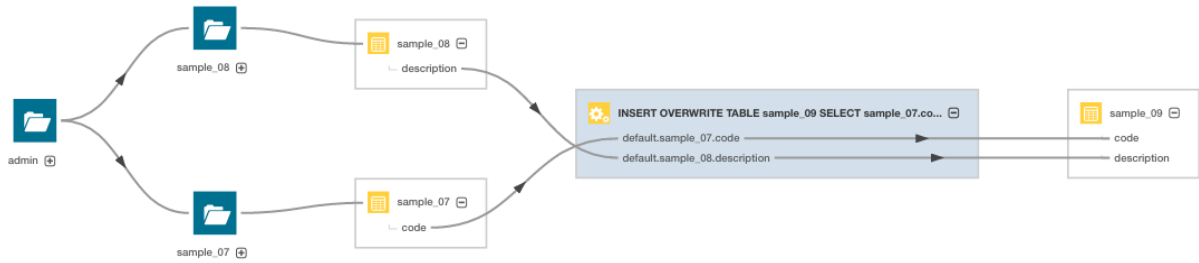
3. Click the **Lineage** tab. For example, clicking the Lineage tab for the `sample_09` table displays the following lineage diagram:



This example shows the relations between a Hive query execution entity and its source and destination tables:



When you click the  icon, columns and lines connecting the source and destination columns display:



Displaying an Instance Lineage Diagram

An **instance lineage diagram** displays instance entities, such as job and query executions, and the input and output entities to which they are related. To display an instance lineage diagram:

1. Perform a search and click a link of type Operation.
2. Click a link in the **Instances** box.
3. Click the **Lineage** tab.



Displaying the Template Lineage Diagram for an Instance Lineage Diagram

To browse from an instance diagram to its template:

1. Display an instance lineage diagram.
2. Click the **Details** tab.
3. Click the value of the **Template** property to go to the instance's template.



Using Lineage to Display Table Schema

Required Role: [Lineage Administrator](#) (or **Metadata Administrator**, **Full Administrator**)

A table schema contains information about the names and types of the columns of a table.

A Kite dataset ingested into HDFS contains information about the names and types of the fields in an HDFS Avro or Parquet file used to create the dataset.

Displaying Hive, Impala, and Sqoop Table Schema

1. Perform a metadata [search](#) for entities of source type **Hive** and type **Table**.
2. In the list of results, click a result entry. The table schema displays in the Details tab.

Displaying Pig Table Schema

1. Perform a metadata [search](#) for entities of source type **Pig**.
2. In the list of results, click a result entry of type **Table**. The table schema displays in the Details tab.

Displaying HDFS Dataset Schema

If you ingest a [Kite dataset](#) into HDFS, you can view the schema of the dataset. The schema is represented as an entity of type Dataset and is implemented as an HDFS directory.

For Avro datasets, primitive types such as null, string, int, and so on, are not separate entities. For example, if you have a record type with a field A that's a record type and a field B that's a string, the subfields of A become entities themselves, but B has no children. Another example would be if you had a union of null, string, map, array, and record types; the union has 3 children - the map, array, and record subtypes.

To display an HDFS dataset schema:

1. Perform a metadata [search](#) for entities of type **Dataset**.
2. Click a result entry. The dataset schema displays in the Details tab.

Stocks Schema

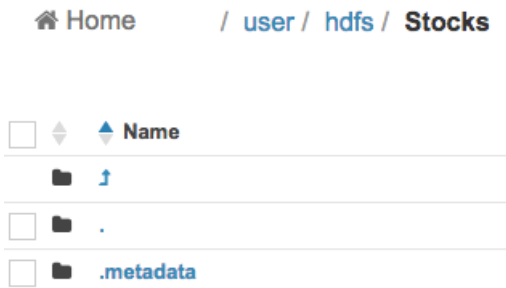
1. Use the Stocks Avro schema file:

```
{
  "type" : "record",
  "name" : "Stocks",
  "namespace" : "com.example.stocks",
  "doc" : "Schema generated by Kite",
  "fields" : [ {
    "name" : "Symbol",
    "type" : [ "null", "string" ],
    "doc" : "Type inferred from 'AAIT'"
  }, {
    "name" : "Date",
    "type" : [ "null", "string" ],
    "doc" : "Type inferred from '28-Oct-2014'"
  }, {
    "name" : "Open",
    "type" : [ "null", "double" ],
    "doc" : "Type inferred from '33.1'"
  }, {
    "name" : "High",
    "type" : [ "null", "double" ],
    "doc" : "Type inferred from '33.13'"
  }, {
    "name" : "Low",
    "type" : [ "null", "double" ],
    "doc" : "Type inferred from '33.1'"
  }, {
    "name" : "Close",
    "type" : [ "null", "double" ],
    "doc" : "Type inferred from '33.13'"
  }, {
    "name" : "Volume",
    "type" : [ "null", "long" ],
    "doc" : "Type inferred from '400'"
  } ]
}
```

and the `kite-dataset` command to create a Stocks dataset:

```
kite-dataset create dataset:hdfs:/user/hdfs/Stocks -s Stocks.avsc
```

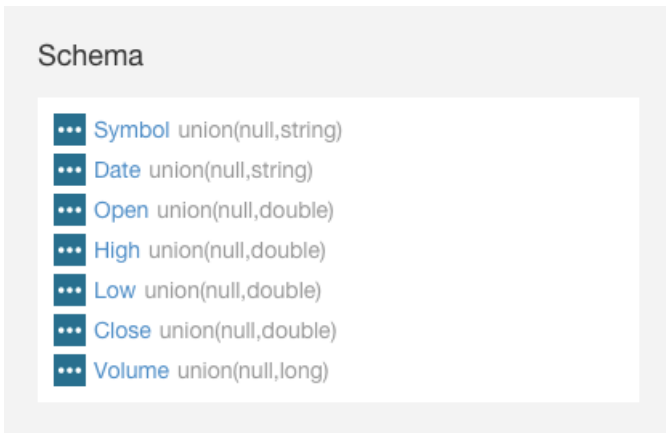
The following directory is created in HDFS:



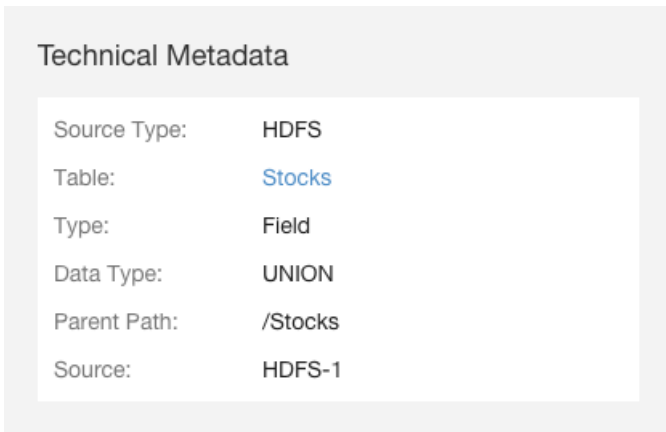
2. In search results, the Stocks dataset appears as follows:



3. Click the **Stocks** link. The schema displays at the right of the Details tab.



Each subfield of the Stocks record is an entity of type Field.



4. Then use the `kite-dataset csv-import` command to import structured data:

```
kite-dataset csv-import ./Stocks.csv dataset:hdfs:/user/hdfs/Stocks --no-header
```

where `Stocks.csv` is:

```
AAPL,20150206,120.02,120.25,118.45,118.93,43372000
AAPL,20150205,120.02,120.23,119.25,119.94,42246200
GOOG,20150304,571.87,577.11,568.01,573.37,1713800
GOOG,20150303,570.45,575.39,566.52,573.64,1694300
GOOG,20150302,560.53,572.15,558.75,571.34,2118400
GOOG,20150209,528,532,526.02,527.83,1264300
GOOG,20150206,527.64,537.2,526.41,531,1744600
GOOG,20150205,523.79,528.5,522.09,527.58,1844700
FB,20150304,79.3,81.15,78.85,80.9,28014500
```

```
FB,20150303,79.61,79.7,78.52,79.6,18567300
FB,20150302,79,79.86,78.52,79.75,21604400
FB,20150227,80.68,81.23,78.62,78.97,30635700
FB,20150226,79.88,81.37,79.72,80.41,31111900
TWTR,20150211,46.27,47.78,46.11,47.5,24747000
TWTR,20150210,47.35,47.39,45.57,46.26,32287800
TWTR,20150209,46.73,47.69,46.5,47.32,36177900
TWTR,20150206,46.12,48.5,45.8,48.01,102669800
TWTR,20150205,42.04,42.47,40.91,41.26,61997300
MSFT,20150304,43.01,43.21,42.88,43.06,25705800
MSFT,20150303,43.56,43.83,43.09,43.28,31748600
MSFT,20150302,43.67,44.19,43.55,43.88,31924000
MSFT,20150227,44.13,44.2,43.66,43.85,33807700
MSFT,20150226,43.99,44.23,43.89,44.06,28957300
ORCL,20150304,43.2,43.66,42.82,43.61,14663900
ORCL,20150303,43.83,43.88,43.17,43.38,10058700
ORCL,20150302,43.81,44.04,43.48,44.03,11091000
ORCL,20150227,43.77,44.11,43.68,43.82,9549500
ORCL,20150226,43.8,44.15,43.71,43.89,8519300
ORCL,20150225,43.83,44.09,43.38,43.73,11785400
```

Search: Syntax and Properties Reference

Cloudera Navigator metadata search uses an embedded Solr engine that follows the syntax specified for the [LuceneQParserPlugin](#).

Search Syntax

Search strings are constructed by specifying a [default property](#) value and one of the four types of key-value pairs as follows:

- **Technical metadata key-value pairs** - *key:value*
 - *key* is one of the properties listed in [Searchable Properties Reference](#) on page 97.
 - *value* is a single value or range of values specified as [*value1* TO *value2*]. In a value, * is a wildcard. In property values, you must escape special characters :, -, /, and * with the backslash character (\), or enclose the property value in quotes.

Technical metadata key-value pairs are read-only and cannot be modified.

- **Custom metadata key-value pairs** - *up_key:value*
 - *key* is a user-defined property.
 - *value* is a single value or range of values specified as [*value1* TO *value2*]. In a value, * is a wildcard. In property values, you must escape special characters :, -, /, and * with the backslash character (\), or enclose the property value in quotes.

Custom metadata key-value pairs can be modified.

- **Hive extended attribute key-value pairs** - *tp_key:value*
 - *key* is an extended attribute set on a Hive entity. The syntax of the attribute is specific to Hive.
 - *value* is a single value supported by the entity type.

Hive extended attribute key-value pairs are read-only and cannot be modified.

- **Managed metadata key-value pairs** - *namespace.key:value*
 - *namespace* is the namespace containing the property. See [Defining Properties for Managed Metadata](#) on page 35.
 - *key* is the name of a managed metadata property.
 - *value* is a single value, a range of values specified as [*value1* TO *value2*], or a set of values separated by spaces. In a value, * is a wildcard. In property values, you must escape special characters :, -, /, and * with the backslash character (\), or enclose the property value in quotes.

Only the values of managed metadata key-value pairs can be modified.

- **S3 key-value pairs** - *tp_key:value*
 - *key* is the name of [user-defined metadata](#).
 - *value* is a single value.
 - Only file metadata is extracted; bucket and folder metadata is not extracted.

Constructing Compound Search Strings

To construct compound search strings, you can join multiple property-value pairs using the [Lucene Query Parser Boolean operators](#):

- , +, -
- OR, AND, NOT

In both syntaxes, you use () to group multiple clauses into a single field and to form subqueries. When you [filter results](#) in the Navigator Metadata UI, the constructed search strings use the , +, - syntax.

Example Search Strings

- Entities in the path /user/hive that have not been deleted - +("/user/hive") +(-deleted:true)
- Descriptions that start with the string "Banking" - description:Banking*
- Entities of type MapReduce or entities of type Hive - sourceType:mapreduce sourceType:hive or sourceType:mapreduce OR sourceType:hive
- Entities of type HDFS with size equal to or greater than 1024 MiB or entities of type Impala - (+sourceType:hdfs +size:[1073741824 TO *]) sourceType:impala
- Directories owned by hdfs in the path /user/hdfs/input - +owner:hdfs +type:directory +filePath:"/user/hdfs/input" OR owner:hdfs AND type:directory AND filePath:"/user/hdfs/input"
- Job started between 20:00 to 21:00 UTC - started:[2013-10-21T20:00:00.000Z TO 2013-10-21T21:00:00.000Z]
- Custom key-value - project-customer1 - up_project:customer1
- Technical key-value - In Hive, specify table properties like this:

```
ALTER TABLE table_name SET TBLPROPERTIES ('key1'='value1');
```

To search for this property, specify tp_key1:value1.

- Managed key-value with multivalued property - MailAnnotation.emailTo:"dana@example.com" MailAnnotation.emailTo:"lee@example.com"



Note: When viewing MapReduce jobs in the Cloudera Manager Activities page, the string that appears in a job Name column equates to the originalName property. To specify a MapReduce job name in a search, use the string (sourceType:mapreduce) and (originalName:jobName), where jobName is the value in the job Name column.

Searchable Properties Reference

The following sections describe the categories (types) of properties that can be searched.

Default Properties

The following properties can be searched by specifying a property value: type, filePath, inputs, jobId, mapper, mimeType, name, originalName, outputs, owner, principal, reducer, and tags.

Common Properties

Name	Type	Description
description	text	Description of the entity.
group	caseInsensitiveText	The group to which the owner of the entity belongs.
name	ngrammedText	The overridden name of the entity. If the name has not been overridden, this value is empty. Names cannot contain spaces.
operationType	ngrammedText	The type of an operation: <ul style="list-style-type: none"> • Pig - SCRIPT • Sqoop - Table Export, Query Import
originalName	ngrammedText	The name of the entity when it was extracted.

Name	Type	Description
originalDescription	text	The description of the entity when it was extracted.
owner	caselnsensitiveText	The owner of the entity.
principal	caselnsensitiveText	For entities with type OPERATION_EXECUTION, the initiator of the entity.
properties	string	A set of key-value pairs that describe the entity.
tags	ngramedText	A set of tags that describe the entity.
type	tokenizedCaselnsensitiveText	The type of the entity. The available types depend on the entity's source type: <ul style="list-style-type: none"> • hdfs - DIRECTORY, FILE, DATASET, FIELD • hive - DATABASE, TABLE, FIELD, OPERATION, OPERATION_EXECUTION, SUB_OPERATION, PARTITION, RESOURCE, VIEW • impala - OPERATION, OPERATION_EXECUTION, SUB_OPERATION • mapreduce - OPERATION, OPERATION_EXECUTION • oozie - OPERATION, OPERATION_EXECUTION • pig - OPERATION, OPERATION_EXECUTION • spark - OPERATION, OPERATION_EXECUTION • sqoop - OPERATION, OPERATION_EXECUTION, SUB_OPERATION • yarn - OPERATION, OPERATION_EXECUTION, SUB_OPERATION
userEntity	Boolean	Indicates whether an entity was added using the Cloudera Navigator SDK .
Query		
queryText	string	The text of a Hive, Impala, or Sqoop query.
Source		
clusterName	string	The name of the cluster in which the source is managed.
sourceId	string	The ID of the source type.
sourceType	caselnsensitiveText	The source type of the entity: hdfs, hive, impala, mapreduce, oozie, pig, spark, sqoop, or yarn.
sourceUrl	string	The URL of web application for a resource.
Timestamps		
The available timestamp fields vary by the source type: <ul style="list-style-type: none"> • hdfs - created, lastAccessed, lastModified • hive - created, lastModified • impala, mapreduce, pig, spark, sqoop, and yarn - started, ended 	date	Timestamps in the Solr Date Format . For example: <ul style="list-style-type: none"> • lastAccessed: [* TO NOW] • created: [1976-03-06T23:59:59.999Z TO *] • started: [1995-12-31T23:59:59.999Z TO 2007-03-06T00:00:00Z] • ended: [NOW-1YEAR/DAY TO NOW/DAY+1DAY] • created: [1976-03-06T23:59:59.999Z TO 1976-03-06T23:59:59.999Z+1YEAR] • lastAccessed: [1976-03-06T23:59:59.999Z/YEAR TO 1976-03-06T23:59:59.999Z]

Dataset Properties

Name	Type	Description
compressionType	tokenizedCaseInsensitiveText	The type of compression of a dataset file.
dataType	string	The data type: record.
datasetType	tokenizedCaseInsensitiveText	The type of the dataset: Kite.
fileFormat	tokenizedCaseInsensitiveText	The format of a dataset file: Avro or Parquet.
fullDataType	string	The full data type: record.
partitionType	string	The type of the partition.
schemaName	string	The name of the dataset schema.
schemaNamespace	string	The namespace of the dataset schema.

HDFS Properties

Name	Type	Description
blockSize	long	The block size of an HDFS file.
deleted	Boolean	Indicates whether the entity has been moved to the Trash folder.
deleteTime	long	The time the entity was moved to the Trash folder.
filePath	path	The path to the entity.
mimeType	ngrammedText	The MIME type of an HDFS file.
parentPath	string	The path to the parent entity of a child entity. For example: <code>parent path: /default/sample_07</code> for the table <code>sample_07</code> from the Hive database <code>default</code> .
permissions	string	The UNIX access permissions of the entity.
replication	int	The number of copies of HDFS file blocks.
size	long	The exact size of the entity in bytes or a range of sizes. Range examples: <code>size:[1000 TO *]</code> , <code>size: [* TO 2000]</code> , and <code>size:[* TO *]</code> to find all fields with a size value.

Hive Properties

Name	Type	Description
Field		
dataType	ngrammedText	The type of data stored in a field (column).
Table		
compressed	Boolean	Indicates whether a table is compressed.
serDeLibName	string	The name of the library containing the SerDe class.
serDeName	string	The fully qualified name of the SerDe class.
Partition		
partitionColNames	string	The table columns that define the partition.
partitionColValues	string	The table column values that define the partition.

Name	Type	Description
technical_properties	string	Hive extended attributes.
clusteredByColNames	string	The column names that identify how table content is divided into buckets.
sortByColNames	string	The column names that identify how table content is sorted within a bucket.

MapReduce and YARN Properties

Name	Type	Description
inputRecursive	Boolean	Indicates whether files are searched recursively under the input directories, or only files directly under the input directories are considered.
jobId	ngramedText	The ID of the job. For a job spawned by Oozie, the workflow ID.
mapper	string	The fully qualified name of the mapper class.
outputKey	string	The fully qualified name of the class of the output key.
outputValue	string	The fully qualified name of the class of the output value.
reducer	string	The fully qualified name of the reducer class.

Operation Properties

Name	Type	Description
Operation		
inputFormat	string	The fully qualified name of the class of the input format.
outputFormat	string	The fully qualified name of the class of the output format.
Operation Execution		
inputs	string	The name of the entity input to an operation execution. For entities of resource type <code>mapreduce</code> , <code>yarn</code> , and <code>spark</code> , it is usually a directory. For entities of resource type <code>hive</code> , it is usually a table.
outputs	string	The name of the entity output from an operation execution. For entities of resource type <code>mapreduce</code> , <code>yarn</code> , and <code>spark</code> , it is usually a directory. For entities of resource type <code>hive</code> , it is usually a table.
engineType	string	The type of the engine used for an operation: MR or Spark.

Oozie Properties

Name	Type	Description
status	string	The status of the Oozie workflow: <code>RUNNING</code> , <code>SUCCEEDED</code> , or <code>FAILED</code> .

Pig Properties

Name	Type	Description
scriptId	string	The ID of the Pig script.

S3 Properties

Name	Type	Description
Object Properties		
region	string	The geographic region in which the bucket is stored
bucketName	string	The name of the bucket in which the object is stored
filePath	path	The key of the S3 object.
size	long	Object size in bytes.
lastModified	date	Object creation date or the last modified date, whichever is the latest.
etag	string	A hash of the object. The ETag reflects changes only to the contents of an object, not its metadata. The ETag may or may not be an MD5 digest of the object data.
storageClass	string	Storage class used for storing the object.
owner	string	Owner of the object.
sequencer	string	Latest S3 event notification sequencer. Used to order events.
parentPath	string	Parent of the S3 object.
technicalProperties	key-value pairs	Custom metadata for each S3 object.
Bucket Properties		
region	string	Region for the bucket.
created	date	Date the bucket was created.
owner	string	Owner of the bucket.

Sqoop Properties

Name	Type	Description
dbURL	string	The URL of the database from or to which the data was imported or exported.
dbTable	string	The table from or to which the data was imported or exported.
dbUser	string	The database user.
dbWhere	string	A where clause that identifies which rows were imported.
dbColumnExpression	string	An expression that identifies which columns were imported.

Cloudera Navigator Administration Tasks

The Administration tab of the Cloudera Navigator console is the starting point for several configuration and maintenance tasks.

Maintaining Metadata Store Using Purge

The volume of metadata maintained by Navigator Metadata Server can grow quickly and exceed the capacity of the Solr instance that processes the index and interfere with search speed and data lineage. For example, with stale metadata—properties no longer used to tag metadata in the system—lineage may take too long to display, or may show relationships that no longer exist.

For faster search and cleaner lineage tracing, Cloudera Navigator's Purge function prunes the system of metadata that has been deleted and that is aged beyond utility. Purge before upgrading Cloudera Navigator to a new release also can hasten the upgrade and guard against memory errors. See [Avoiding Out-of-Memory Errors During an Upgrade](#) for details.

The Purge function can be used in a few different ways:

- On an ad hoc basis after deleting metadata, as detailed in [Purging a Property](#).
- By using the Cloudera Navigator APIs, as detailed in [Using the Purge APIs for Metadata Maintenance Tasks](#).
- For Cloudera Navigator 2.11 (and higher) releases, by using the Cloudera Navigator console and [scheduling a regular weekly Purge](#), as detailed below.

Scheduling the Purge Process

Use the Cloudera Navigator console to configure a schedule for a regular weekly Purge of deleted and stale metadata from your Cloudera Navigator instance, specifically, the Navigator Metadata Server and its associated database.



Note: Cloudera recommends scheduling the Purge process for non-production hours because other users and processes will be unable to use Cloudera Navigator until the Purge process completes.

To configure Purge schedule:

1. Log in to the Cloudera Navigator console using an account with privileges as either Cloudera Manager Full Administrator or Navigator Administrator. The URL to access the Cloudera Navigator console directly (rather than from within Cloudera Manager) using the default port on the host running the Navigator Metadata Server role would be as follows:

```
http://fqdn-1.example.com:7187/login.html
```

2. Enter your administrator user account and password at the login page.
3. Click the **Purge Settings** tab. The current Metadata and Lineage purge schedule displays, along with lists of up to five upcoming scheduled purges and a list of up to five most recent completed purges.

cloudera NAVIGATOR

Search Audits Analytics Policies Administration admin ?

Administration

Managed Metadata Role Management Purge Settings

Metadata and Lineage purge

[Edit](#) Upcoming

Shows upcoming scheduled purges (up to 5)

Completed

Shows the most recent completed purges (up to 5)

The Metadata and Lineage purge is setup to run every Saturday at 12:00 AM.

Cloudera Navigator is unavailable to all users while the purge process runs.

- HDFS entities deleted more than 60 days ago will be purged
- Select operations older than 60 days will be purged

Start	End latest
Jul 8, 2017 12:00 AM	Jul 8, 2017 12:00 AM
Jul 15, 2017 12:00 AM	Jul 15, 2017 12:00 AM
Jul 22, 2017 12:00 AM	Jul 22, 2017 12:00 AM
Jul 29, 2017 12:00 AM	Jul 29, 2017 12:00 AM
Aug 5, 2017 12:00 AM	Aug 5, 2017 12:00 AM

Started	Ended	Status
---------	-------	--------

To change the existing schedule:

- Click the **Edit** button.
- Set the day, time, maximum purge duration, and time frame to hold on to deleted entities (Purge entities deleted more than*) settings best for your environment. See the descriptions and usage notes for these settings in the table below.

Property	Range of selectable values	Usage note
How often	Weekly	Not configurable. The Purge runs weekly per your specifications for Day and Time.
Day	Days of week, Sunday through Saturday	Select a day for the purge that will have minimal impact to your user community.
Time	Hourly time, from 12 Midnight through 11 PM	Select a time that will have minimal impact.
Maximum purge duration	10 minutes, 1 hour through 10 hours, 12 hours, 14 hours, 16 hours, 18 hours, 20 hours, 22 hours, 24 hours, 36 hours, 48 hours, 3 days through 7 days, inclusive	Set the amount of time you want to allow for the Purge process to run. The process will not run beyond your specified duration, whether it has completed the purge or not. All entities purged even if the process is cut short by this setting remain purged. During this timeframe, no other operations in Cloudera Navigator can occur.
Purge entities deleted more than*	1 day through 10 days, 20 days through 100 days, 150 days, 365 days	Enter the number of days after entity deletion that will pass before the purge process removes it. For example, a setting of 1 day means that entities deleted before yesterday are purged but entities deleted yesterday are retained.
Purge SELECT operations*	Enable	Select this option to enable Purge for Hive and Impala SELECT operations using the time period selected in the next setting (Only Purge SELECT operations older than*).

Property	Range of selectable values	Usage note
Only Purge SELECT operations older than*	10 days through 100 days (10-day increments), 150 days, 365 days	The purge will include only those SELECT operations deleted prior to older than this threshold.

If you running Hive and Impala queries on your system, you can have these purged from your system as well. Set appropriate thresholds for your use cases. Here is an example of a revised schedule:

The screenshot shows the Cloudera Navigator Administration interface. The top navigation bar includes 'Search', 'Audits', 'Analytics', 'Policies', 'Administration', and 'admin'. The 'Administration' section is active, with sub-tabs for 'Managed Metadata', 'Role Management', and 'Purge Settings'. The 'Purge Settings' tab is selected, displaying the 'Edit Purge schedule' form. The form includes the following fields:

- How often:** Weekly
- Day*:** Sunday
- Time*:** 2 AM
- Maximum purge duration* ?**: 6 hours
- Purge entities deleted more than* ?**: 30 days
- Purge SELECT operations* ?**: Enable
- Only Purge SELECT operations older than* ?**: 90 days

Below the form, a status message reads: "The next purge will start at Jul 9, 2017 2:00 AM and end at Jul 9, 2017 8:00 AM the latest. (PDT)". At the bottom right, there are 'Cancel' and 'Save' buttons.

Troubleshooting Navigator Data Management

This page contains troubleshooting tips and workarounds for various issues that can arise.

Cloudera Navigator-Cloudera Altus

No metadata or lineage collected from an Altus deployed cluster.

Symptom: No metadata collected from a Cloudera Altus instantiated cluster. Although both the Cloudera Altus environment has been setup to use a given Amazon S3 bucket and the Cloudera Navigator instance has been configured to read from that same S3 bucket, no metadata is collected.

Possible cause: Permissions on the Amazon S3 bucket have not been applied correctly.

Steps to resolve: To determine the root cause of this issue:

- Check that permissions have been set properly on the Amazon S3 bucket.
- Verify that the Amazon S3 bucket name has not been changed.

To determine if the cause of the issue is misconfiguration, look at the Telemetry Publisher logs. A failure message in the log such as the following means that the Amazon S3 bucket has either not been properly identified, or that the correct permissions have not been set:

```
...Metadata export failed com.amazonaws.services.s3.model.AmazonS3Exception: The specified
bucket does not exist
(Service: Amazon S3; Status Code: 404; Error Code: NoSuchBucket; Request ID:
D28543E2494BD521), S3 Extended Request ID:
...
```

The 404 status code and "NoSuchBucket" in the error message indicate the Telemetry Publisher was unable to write to Amazon S3 bucket because the bucket was misidentified or that permissions were configured incorrectly.

Navigator Audit Server

"No serializer found" error

Symptom: Selecting an Audits page results in error message.

Error message:

```
No serializer found
```

Possible cause: Communication failure somewhere between Navigator Metadata Server and Cloudera Manager Server and its communication to Navigator Audit Server. This is an existing Known Issue that needs to be resolved so that the error message can be properly mapped.

Workaround: This workaround does not resolve the issue, but provides additional information as to the actual source of the underlying error.

1. Log in to the Cloudera Navigator console.
2. Enable Navigator API logging.
3. Perform your action on the Audits page that raised the error message and see if the underlying issue is caught in the API log. If the Navigator API logging reveals no additional information, turn on API logging for Cloudera Manager:
 - Log in to the Cloudera Manager Admin Console.
 - Select **Administrator** > **Settings**

- Select **Advanced** for the **Category** filter.
- Click the **Enable Debugging of API** check box to select it. The server log will contain all API calls. Try your request again and see if the source error message displays in the response.

Processing a backlog of audit logs

Problem: You have logs that include audit events that were not processed by the Cloudera Navigator Audit Server, such as logs for events that took place before audit server was online or during a period when audit server was offline.

Solution: A backlog of logs can be processed by audit server as follows:

1. Backup audit files for all roles on all hosts.

Do this right away as there are retention periods configured for these files and they will gradually be deleted.

2. Determine what days were not processed. The audit log files have the UNIX epoch appended to their name:

```
hdfs-NAMENODE-341ad9b94435839ce45b8b22e7c805b3-HdfsAuditLogger-1499844328581
```

You want the dates from the oldest and newest files. You can do this by sorting the list and identifying the first and last files. For example, for HDFS audit files:

```
The oldest: $ ls ./hdfs* | head -1
```

```
The newest: $ ls ./hdfs* | tail -1
```

Depending on your shell, you can convert the epoch value to a date using `date -r epoch`.

3. Create partitions in the Navigator Audit Server database for days that have missing audits.

Create a partition from an existing template table using SQL commands. Use `SHOW TABLES` to list the template tables. For example, the template table for HDFS is `HDFS_AUDIT_EVENTS`.

Using MySQL as an example, if you have partitions up to March 31, 2017 and your audit logs include HDFS data for the first week of April, you would run the following SQL commands:

```
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_01 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_02 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_03 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_04 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_05 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_06 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_07 LIKE HDFS_AUDIT_EVENTS;
```

4. In the `PARTITION_INFO` table, add new rows for each of the tables you created in the previous step, including the "END_TS" in Unix epoch time, 24 hours after the previous entry.

The following example command inserts a row for Feb 14, 2019 into `PARTITION_INFO` table with correct `END_TS` value for multiple partition tables:

```
insert into PARTITION_INFO values (PARTITION_INFO_SEQUENCE.nextval,  
'HDFS_AUDIT_EVENTS_2019_02_03', 'HDFS_AUDIT_EVENTS', 1550145600000, null, 0);
```

5. Change the Navigator Audit expiration period to be longer than the time period of the logs you want to process.

Set the expiration period in Cloudera Manager. Search for the property "Navigator Audit Server Data Expiration Period".

6. Stop the Cloudera Manager agent on the host running the role for the service whose logs will be restored.

For example, if the logs are for HDFS, stop the agent on the namenode. For HiveServer2, stop the Cloudera Manager agent on the node where HiveServer2 is running. See [Starting, Stopping, and Restarting Cloudera Manager Agents](#).

If you aren't sure of which host is involved, the log files contain the UUID of the role which generated them. Go to the host defined for that role. For high-availability clusters, make sure that you stop the role on the active host.

7. On that host, copy the backed up audit logs to the audit directory.

The location is typically `/var/log/service/audit`. For example, `/var/log/hadoop-hdfs/audit`. The specific locations are configured in the Cloudera Manager `audit_event_log_dir` properties for each service.

8. Move the WAL file out of this directory. (Back it up in a safe location.)
9. Restart the Cloudera Manager agent.

On restart, Cloudera Manager agent will not find the WAL file and will create a new one. It will process the older audit logs one by one.

10. Verify that the audits appear in Navigator.

You can also check the contents of the WAL file to see which logs were processed.

Navigator Metadata Server

"Missing required field: id" (Solr exception)

Symptom: After removing the Navigator Metadata Server storage directory (the `datadir`), an error message for the `id` field displays.

Error message:

```
[qtp1676605578-63]: org.apache.solr.common.SolrException:
[doc=0002ac6b75cb17721ad8324700f473cb] missing required field: id
```

Possible cause: This error indicates that the Navigator Metadata Server database and storage directory are out-of-sync because of a change that was made in the Cloudera Navigator 2.9 (Cloudera Manager 5.10) release. When the Navigator Metadata Server role starts, it compares state information in the database with the same information in the storage directory. This error can appear if the storage directory has been deleted but the database has not.

Steps to resolve: If this error message occurs and the storage directory was recently deleted, you can remove the stale state information from the database. To do this:

1. Take note of the Navigator Metadata Server Storage Dir directory.
Log into Cloudera Manager and browse to **Cloudera Management Services > Configuration > Navigator Metadata Server > Navigator Metadata Server Storage Dir**. Note the storage directory location.
2. Stop the Navigator Metadata Server role by navigating to **Cloudera Management Service Instances > Navigator Metadata Server > Actions > Stop**.
3. Log in to the database instance configured for use with Navigator Metadata Server.
4. Run the following SQL commands:

```
delete from NAV_UPGRADE_ORDINAL;
insert into nav_upgrade_ordinal values (-1, -1);
```

5. Start Navigator Metadata Server role using **Cloudera Management Service Instances > Navigator Metadata Server > Actions > Start**.

The Navigator Metadata Server role should restart successfully.

Repairing metadata in the storage directory after upgrading

If you ran Cloudera Navigator on Cloudera Manager versions 5.10.0, 5.10.1, or 5.11.0, you should upgrade to a later release immediately to avoid a known software problem as reported in the knowledge base entry "Upgrade needed for Cloudera Navigator included with Cloudera Manager releases 5.10.0, 5.10.1, and 5.11.0 due to high risk of failure

Troubleshooting Navigator Data Management

through exhausting data directory resources." This document describes some additional steps to perform after upgrade to mitigate for duplicate and potentially broken relation metadata added to the Navigator storage directory because of the broken versions.

To fully recover the Navigator storage directory:

1. Upgrade Cloudera Manager to 5.10.2, 5.11.1, or later releases.

See [Upgrading Cloudera Manager](#).

2. Set properties to enable background repair tasks.

From Cloudera Manager, go to **Cloudera Management Service > Navigator Metadata Server > Advanced**.

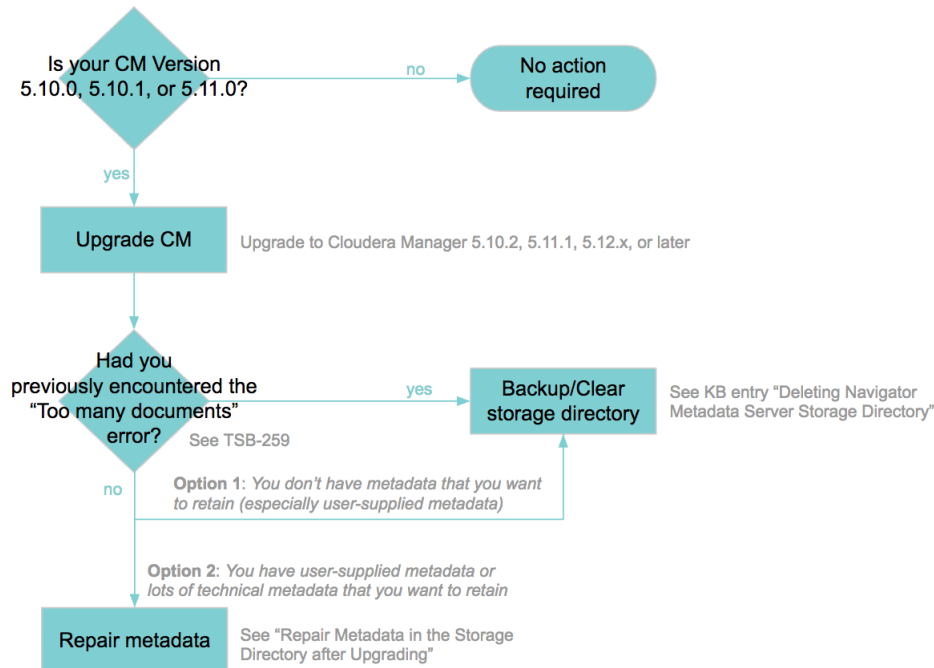
Enter the following configurations to the "Navigator Metadata Server Advanced Configuration Snippet (Safety Valve) for cloudera-navigator.properties" property.

```
nav.backend.task.TRIGGER_HIVE_TABLES_EXTRACTION.enabled=true
nav.backend.task.HDFS_PC_RELATIONS_BUILDER.enabled=true
nav.backend.task.HDFS_MR_RELATIONS_BUILDER.enabled=true
nav.backend.task.PIG_HDFS_RELATIONS_BUILDER.enabled=true
```

3. Restart the Navigator Metadata Server role.

The full cycle of cleaning up the relations may take days or weeks depending on the number of relations affected.

These instructions correspond to "Repair metadata" in the following decision tree:



Appendix: Apache License, Version 2.0

SPDX short identifier: Apache-2.0

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims

licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```