Cloudera Runtime 7.0.1

Atlas Use Cases

Date published: 2019-09-23
Date modified:

https://docs.cloudera.com/

Legal Notice

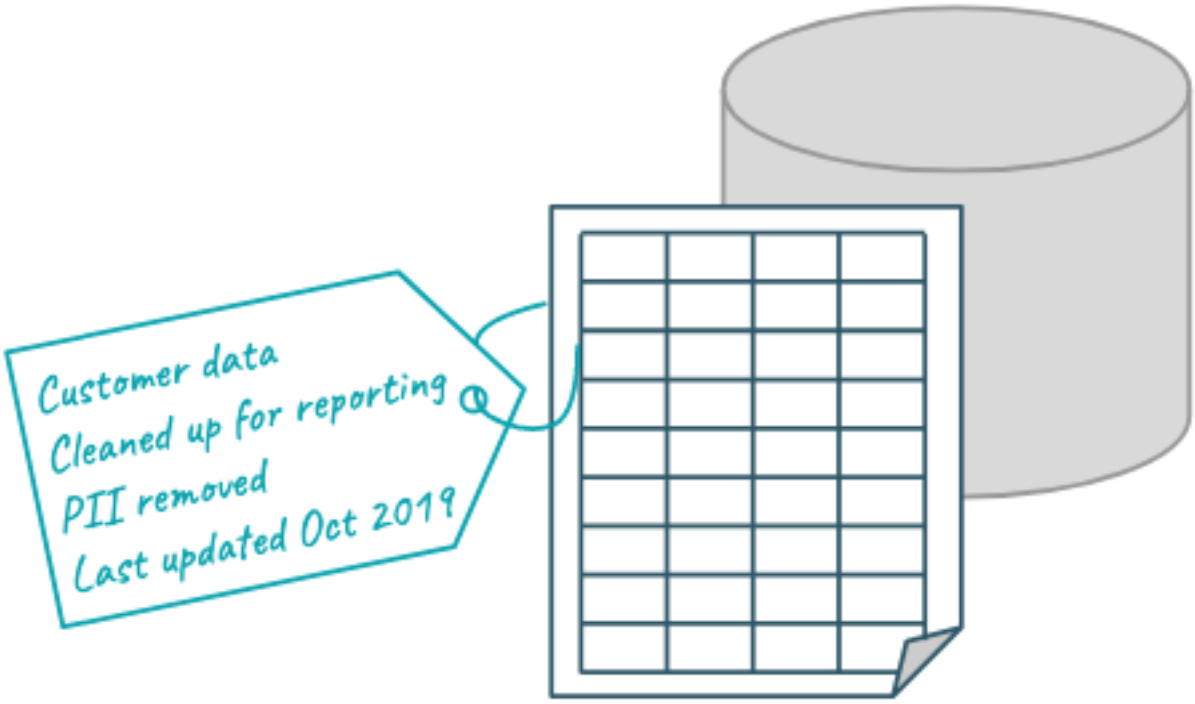
Contents

Governance overview

Concepts for collecting, creating, and using metadata.

What is Apache Atlas?

Atlas is a metadata management and governance system designed to help you find, organize, and manage data assets. Atlas creates “entities” or metadata representations of objects and operations in your data lake. You can add business labels to these entities so you can use business vocabulary to make it easier to search for specific data.



Apache Atlas uses metadata to create lineage relationships

Atlas reads the content of the metadata it collects to build relationships among data assets. When Atlas receives query information, it notes the input and output of the query and generates a lineage map that traces how data is used and transformed over time. This visualization of data transformations allows governance teams to quickly identify the source of data and to understand the impact of data changes.

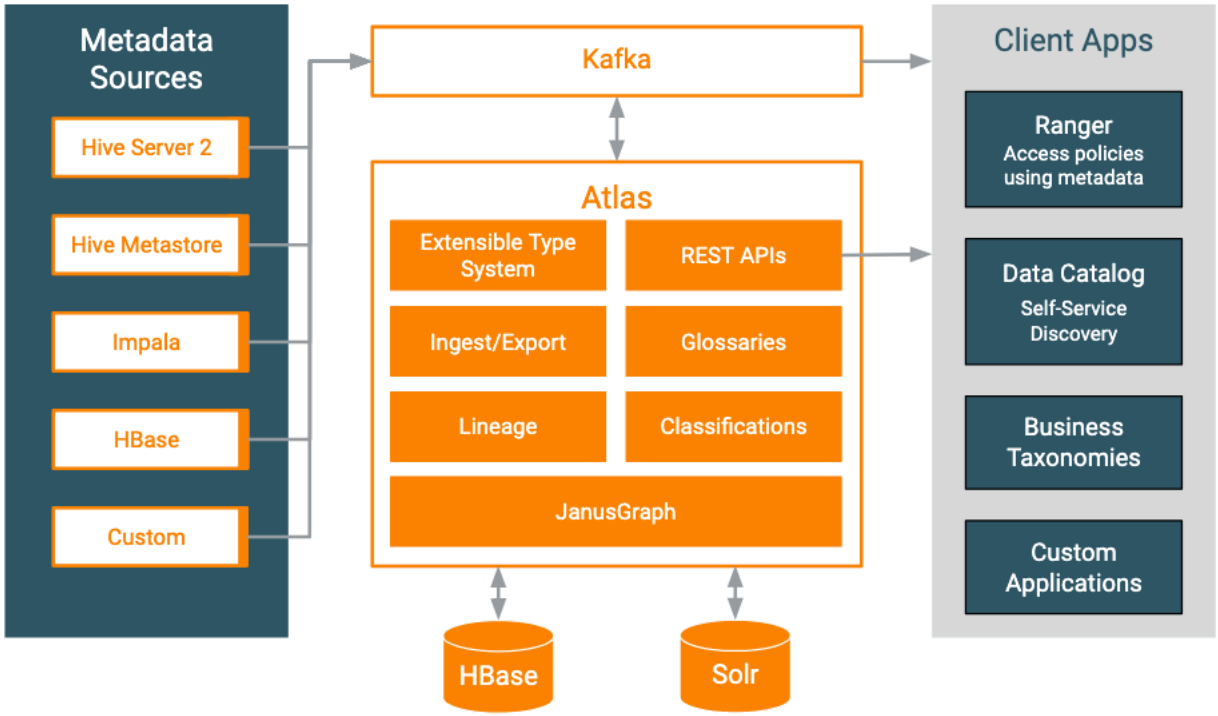
Your tags enhance your metadata, make it easier to search

Atlas manages labels, or “classifications,” that you associate with entities in your data lake. You can create and organize labels to use for anything from identifying data cleansing stages to recording user comments and insights on specific data assets. When you use classifications, the Atlas Dashboard makes it easy to search, group, report, and further annotate the entities you label. Classifications themselves can be organized into hierarchies to make them easier to manage.

Atlas also provides an infrastructure to create and maintain business ontologies to label your data assets. Atlas’ “glossaries” include “terms” so you can build agreed-upon lists for department- or organization-wide vocabulary to identify and manage data. Adding a term gives you a single-click report of entities identified by that term.

Apache Atlas architecture

Atlas runs as an independent service in a Hadoop environment. Many Hadoop data processing and storage services include Atlas add-ons that publish metadata for the services' activities to a Kafka message topic. Atlas reads the messages and stores them in JanusGraph to model the relationships among entities. The datastore behind JanusGraph is HBase, though it can be configured with Cassandra. Atlas stores a search index in Solr to take advantage of Solr's search functionality.



Many Hadoop services include addons or "hooks" that pass metadata to Atlas as the service performs actions against data. The Atlas hooks push metadata to a Kafka topic; the Atlas service reads these messages from the queue and creates the entities. Pre-defined hooks exist for Hive, Impala, HBase, Kafka, Spark, and Sqoop.

Atlas also provides "bridges" that import metadata for all of the existing data assets in a given source. For example, if you start Atlas after you've already created databases and tables in Hive, you can import metadata for the existing data assets using the Hive bridge. Bridges use the Atlas API to import the metadata.

If you need a hook or bridge to automate collecting metadata from another source, use the Atlas Java API to create one.

Data Stewardship with Apache Atlas

Concepts for collecting, creating, and using metadata.

The value of Atlas metadata for organizing and finding data increases when you augment the generated "technical" metadata by classifying and labeling data assets using your organization's business vocabulary. Here's how you would do that:

- Tools. Atlas gives you a hierarchy of classifications, attributes, and a glossary of terms. The glossary allows you to identify synonyms so that the vocabulary from different teams doesn't get in the way of identifying the same data.
- Planning. Figure out who and how to apply the tools: set up an overall plan for what kinds of metadata you want to apply, design some conventions for how to apply them and who can apply them. Design some processes to oversee metadata as it collects to make sure teams are using the appropriate labels; identify synonyms and antonyms.

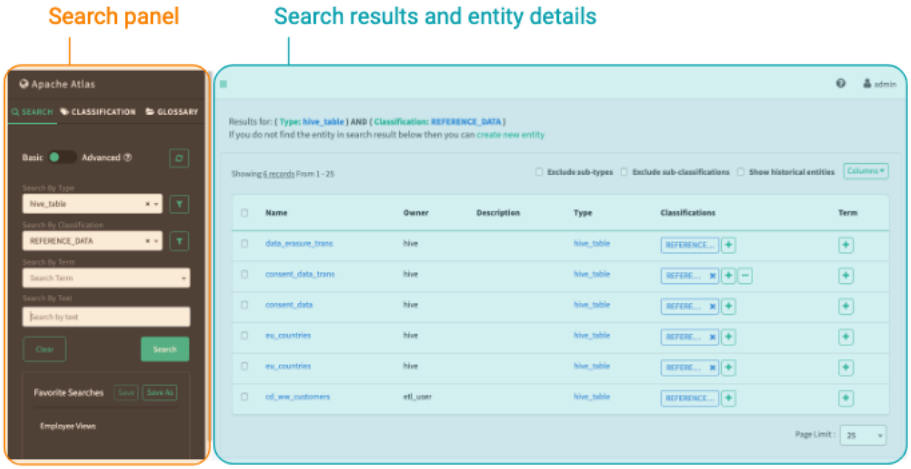
- Examples. This document includes examples of ways you can organize your metadata; strategies that describe how to optimize for specific use cases.

Atlas dashboard tour

Quick introduction to the Atlas user interface and terms.

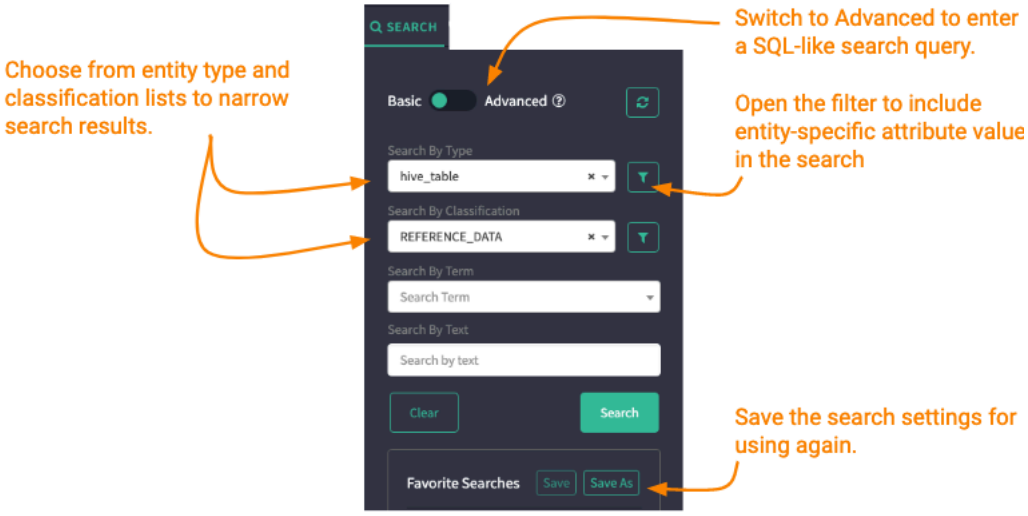
The Atlas user interface is a metadata dashboard with two parts:

- Search panel on the left.
- Detail panel on the right where search results appear, and, when you drill into a specific entity, the details for that entity are displayed. Each detail page has a header section and a series of tabbed panels, all of which are oriented to the metadata for that entity type.



Searching

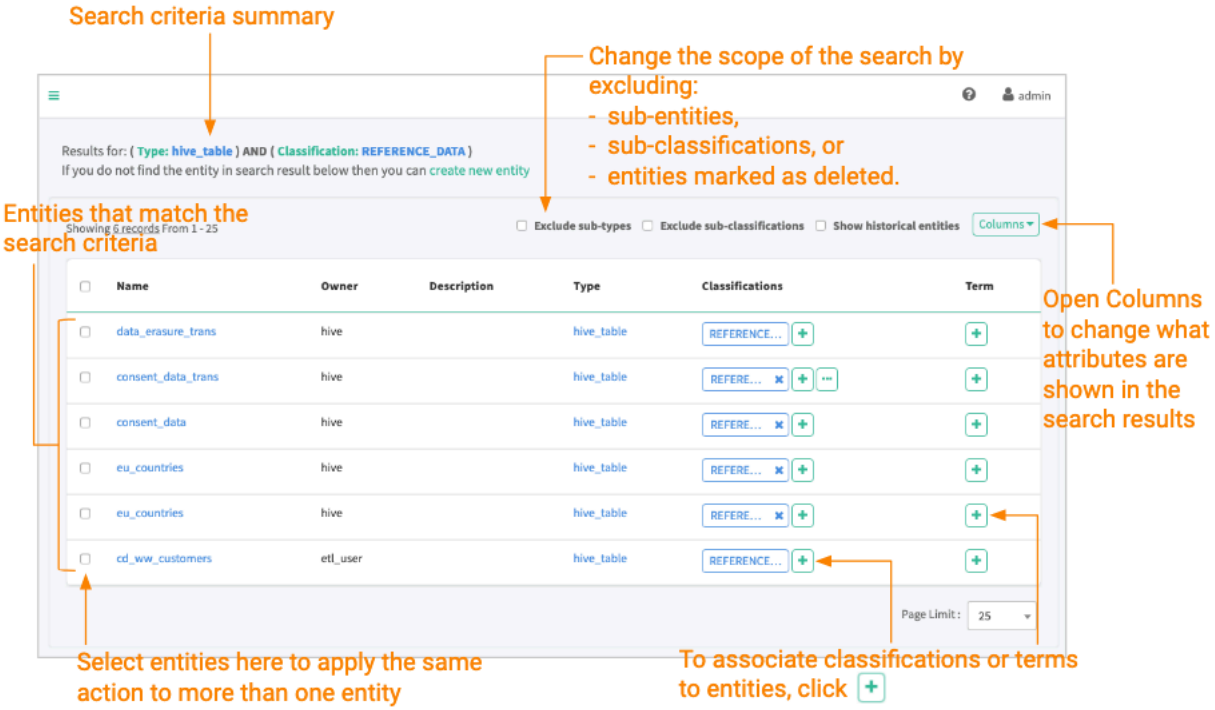
The search panel has three tabs for searching: the general Search tab, and predefined searches based on Classifications and Glossary terms. In the general Search tab, you choose from existing lists of metadata types to narrow the search results. Switching to the Advanced search lets you enter specific search queries; both basic and advanced searches can be saved for easy reuse.



In the Classification tab, selecting a classification displays all the entities that are marked with that classification. Find a specific classification using the search box or browse through the classification hierarchy that you define when you create classifications.

In the Glossary tab, selecting a term displays all the entities that are marked with that term. Find a specific term using the search box or browse through terms by glossary. You can also find specific terms using the category view: browse through the hierarchy you build of your organization's business glossary; when you select a category, the display pane shows the terms assigned to the category. When you select one of those terms, Atlas displays the entities associated with that term.

When you run a search and Atlas returns results, you see a paged-list of entities that match the search criteria. From here, you can go back to the search options and further refine your search or use controls to change how the search results are presented.



Apache Atlas metadata collection overview

Actions performed in cluster services create metadata in Atlas.

Atlas provides addons to many Hadoop cluster services to collect metadata when the service performs certain operations. The Atlas addon or "hook" assembles a predefined set of information and sends it to the Atlas server. The Atlas server reads through the metadata and creates entities to represent the data sets and processes described by the metadata. Atlas may create one or many entities for each event it processes. For example, when a user creates a namespace in HBase, Atlas creates a single entity to represent the new HBase namespace. When a user runs a query in Hive Server 2, Atlas may create many entities, including entities to describe the query itself, any tables involved in the query, entities for each column for each table involved in the query, and so on.

The following table lists the services that are integrated with Atlas by default. For each service, the table lists the events produced by the service that Atlas acknowledges and the entities Atlas produces in response to each event. Note that there isn't always a one-to-one relationship between the event and an entity: the entities produced from a single event depend on the event itself.

Source	Actions Acknowledged	Entities Created
Hive Server 2	ALTER DATABASE CREATE DATABASE DROP DATABASE	hive_db, hive_db_ddl

Source	Actions Acknowledged	Entities Created
	ALTER TABLE CREATE TABLE CREATE TABLE AS SELECT DROP TABLE	hive_process, hive_process_execution, hive_table, hive_table_ddl, hive_column, hive_column_lineage, hive_storagedesc, hdfs_path
	ALTER VIEW ALTERVIEW_AS_SELECT CREATE VIEW CREATE VIEW AS SELECT DROP VIEW	hive_process, hive_process_execution, hive_table, hive_column, hive_column_lineage, hive_table_ddl
	INSERT INTO (SELECT) INSERT OVERWRITE	hive_process, hive_process_execution
HBase	alter_async	hbase_namespace, hbase_table, hbase_column_family
	create_namespace alter_namespace drop_namespace	hbase_namespace
	create table alter table drop table drop_all tables	alter table (create column family), alter table (alter column family), alter table (delete column family)
	alter table (create column family) alter table (alter column family) alter table (delete column family)	hive_process, hive_process_execution
Impala*	CREATETABLE_AS_SELECT	impala_process, impala_process_execution, impala_column_lineage, hive_db hive_table_ddl
	CREATEVIEW	impala_process, impala_process_execution, impala_column_lineage, hive_table_ddl
	ALTERVIEW_AS_SELECT	impala_process, impala_process_execution, impala_column_lineage, hive_table_ddl
	INSERT INTO INSERT OVERWRITE	impala_process, impala_process_execution
Spark*	CREATE TABLE USING CREATE TABLE AS SELECT, CREATE TABLE USING ... AS SELECT	spark_process
	CREATE VIEW AS SELECT,	spark_process

Source	Actions Acknowledged	Entities Created
	INSERT INTO (SELECT), LOAD DATA [LOCAL] INPATH	spark_process

*For these sources, Atlas collects the corresponding asset metadata from HMS.

Related Information

Hive Server 2 metadata collection

HBase metadata collection

Impala metadata collection

Atlas Metadata model overview

Atlas' model represents cluster data assets and operations, and is flexible enough to let you represent objects from other sources.

The flexibility Atlas' metadata model lets you represent whatever objects and relationships among them that you want to create a map of your data lake. Atlas lets you create new instances of predefined entity types and lets you define new types of entities so you can represent data assets and actions from additional data sources or even services that do not reside in Hadoop. Atlas' building blocks are entities, relationships, classifications, enumerations, and structures.

Entities are a collection of attributes that model or represent a data asset or data action. Entities are the unit that Atlas returns in search results or shows as nodes in a lineage diagram. Use Classifications to add metadata to entities; create Relationships connect entities.

Relationships describe connections between two entities. Because relationships are their own type in the Atlas data model, you can create new relationships with custom attributes to represent behaviors that are specific to your organization.

Classifications are reusable labels that can be attached to entities. Atlas supports two separate systems of labels: classifications can be used to describe data, clarify field names, identify status, and other manual or automated metadata. Glossary terms—which are implemented as classifications but managed separately—are used to associate data assets with formal names for agreed-upon business concepts and in business contexts. When you build a department or company-wide glossary and use its terms to label data, you create a search structure that allows everyone to access data with a common language. Creating and applying classifications and terms to entities lets you group data assets, mark them based on sensitivity or other access requirements, and label them to allow easier searching. The Atlas user interface leverages these labels to make it easy to find data assets marked with a given classification or term.

Atlas supports defining custom enumerations and data structures as well, similar to those constructs in structured programming languages. Enums can be used in attribute definitions to store lists of predetermined values; structs can be used in attribute definitions and relationship endpoints to identify more complex groupings.