

Atlas Use Case: Extending Metadata

Date published: 2019-09-23

Date modified: 2021-09-09



Legal Notice

© Cloudera Inc. 2025. All rights reserved.

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

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

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

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

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

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

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

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

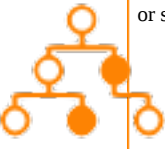
Contents

Top-down process for adding a new metadata source.....	4
---	----------

Top-down process for adding a new metadata source

You can create entity types in Atlas to represent data assets, operations, or other types of artifacts from sources other than those supported by default.

Atlas' data model is designed to be flexible enough to represent a wide variety of data assets and the processes that generate them. You can design your own entities to collect metadata from sources beyond what comes predefined in Atlas. The high-level process for creating entities is as follows:

1		Think about how you want to model the core assets from your source. The predefined entity types inherit basic attributes from the DataSet and Process entity types. Using these general types as your starting point ensures that your entities can display lineage in the Atlas Dashboard and take advantage of other predefined rules and hierarchies built into the models. That said, new entity types are not limited to this data/process model.
2		Define the relationships you want to track among the data assets. To have your entities appear in lineage graphs in the Atlas Dashboard, include relationship attributes for "inputToProcesses" and "outputFromProcesses".
3		List all the metadata you want to track for each entity type. If necessary or convenient, define enumerations or structures to describe metadata specifically for your entities.
4		Use REST API or create an Atlas client to build the entity type definitions in Atlas. (build JSON)
5		Validate the model by manually filling in metadata in the types to validate that they behave the way you expect. You can do this using the Atlas Dashboard or REST API and then test that lineage, search, and Ranger tag-based-policies do what you expect.
6		Write a hook, bridge, or both to collect metadata automatically from your source.