Cloudera Data Science Workbench

Model Governance

Date published: 2020-02-28
Date modified: 2021-06-24

CLOUDƎRA

Legal Notice

Contents

Enabling Model Governance

Cloudera Data Science Workbench utilizes Atlas lineage to help you understand the source and impact of data and changes to data over time and across all your data.

Before you begin

You must install the following services on your CDP cluster:

Note: These services should not be installed on the CDSW nodes to avoid impacting compute resources on those nodes.

- Atlas
- Ranger
- Ranger KMS
- Kafka
- ZooKeeper
- SOLR

About this task

You must enable governance to capture and view information about your CDSW projects, models, and builds centrally from Apache Atlas (Data Catalog) for a given environment. If you do not select this option while provisioning workspaces, then integration with Atlas will not work.

Procedure

1. Go to Cloudera Manager.
2. Choose your CDSW cluster.
3. Click the Configuration tab.
4. Check the Enable Model Governance Support checkbox for your CDSW cluster.

   Wait for Cloudera Manager to detect and validate your change. Your change will result in a Stale Configuration and Cloudera Manager will require that you restart your cluster.
5. If you want to review your changes:
   a) Click the Restart button.

      Cloudera Manager displays a list of your changes.
   b) Click the Restart Stale Services button at the bottom of the screen.
   c) Confirm your changes by clicking the checkbox next to Re-deploy client configuration.
6. If you do not want to review your changes, choose Restart from the Actions menu.

Viewing lineage for a model deployment in Atlas

Lineage information helps you understand the origin of data and the transformations it may have gone through before arriving in a model. In Atlas, if transformations occurred in services that provide process metadata, a lineage graph shows how data was generated. This relationship is stored as a vertex in Atlas' graph database. It is displayed as a lineage graph in the details of each entity.
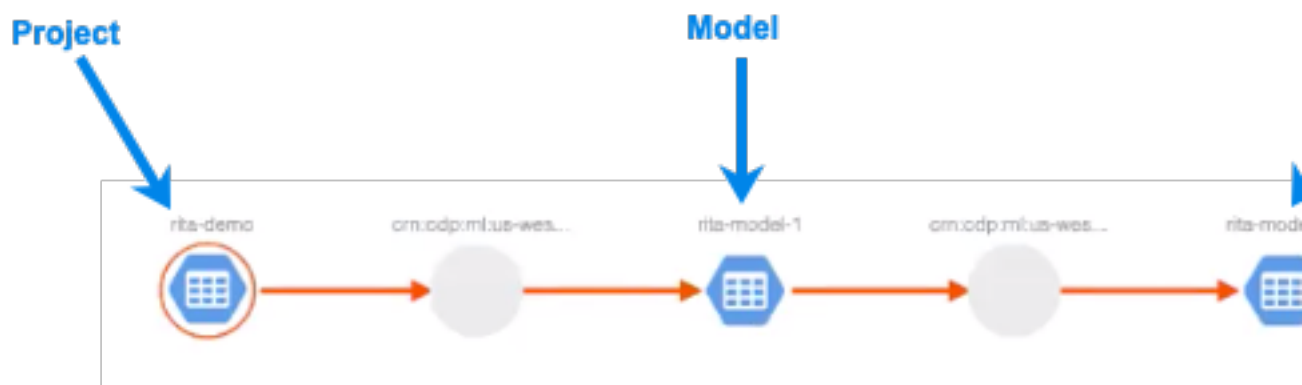
About this task

You can view the lineage information for a particular model deployment and trace it back to the specific data that was used to train the model through the Atlas' Management Console.

Procedure

1. Navigate to the Atlas web UI.
2. In the Search By Type field, select ml_project and click Search.
3. In the Search by Text field, enter the name of your project and click Search.
4. Click your project name in the list of found projects.
5. Click the Lineage tab.

   Atlas displays a visualization of lineage information for the particular model deployment. You can use the visualization to trace the model deployment back to the specific data that was used to train the model.

You can also search for a specific table, click through to its Lineage tab, and see if the table has been used in any model deployments.

Registering training data lineage using a linking file

The Cloudera Data Science Workbench projects, model builds, model deployments, and associated metadata are tracked in Apache Atlas, which is available in the environment's SDX cluster. You can also specify additional metadata to be tracked for a given model build. For example, you can specify metadata that links training data to a project through a special file called the linking file (lineage.yaml).

About this task

The lineage.yaml file defines additional metadata and the lineage relationships between the project's models and training data. You can use a single lineage.yaml file for all the models within the project.

> **Note:** Your lineage file should be present in your project before you create a model build. You must parse the lineage file and attach the metadata during the model build process.

Procedure

1. Create a YAML file in your CDSW project called lineage.yaml. If you have used a template to create your project, a lineage.yaml file should already exist in your project.

2. Insert statements in the file that describe the relationships you want to track between a model and the training data. You can include additional descriptive metadata through key-value pairs in a metadata section.

YAML	YAML Structure	Description
Model name	Top-level entry	A ML model name associated with the current project. There can be more than one model per linking file.
hive_table_qualified_names	Second-level entry	This pre-defined key introduces sequence items that list the names of Hive tables used as training data.
Table names	Sequence items	The qualified names of Hive tables used as training data enclosed in double quotation marks. Qualified names are of the format db-name.table-name@cluster-name
metadata	Second-level entry	This pre-defined key introduces additional metadata to be included in the Atlas representation of the relationship between the model and the training data.
key:value	Third-level entries	Key-value pairs that describe information about how this data is used in the model. For example, consider including the query text that is used to extract training data or the name of the training file used.

The following example linking file shows entries for two models in your project: modelName1 and modelName2:

```
modelName1:                      # the name of your model
  hive_table_qualified_names:                    # this is a predefined key to
 link to
                                    # training data
    - "db.table1@namespace"       # the qualifiedName of the hive
_table
                                    # object representing training
data
    - "db.table2@ns"
  metadata:                        # this is a predefined key for
                                   # additional metadata
    key1: value1
    key2: value2
    query: "select id, name from table"   # suggested use case: query used
 to
                                    # extract training data
    training_file: "fit.py"       # suggested use case: training f
ile
                                    # used
modelName2:                        # multiple models can be spec
ified in
                                    # one file

  hive_table_qualified_names:
    - "db.table2@ns"
```