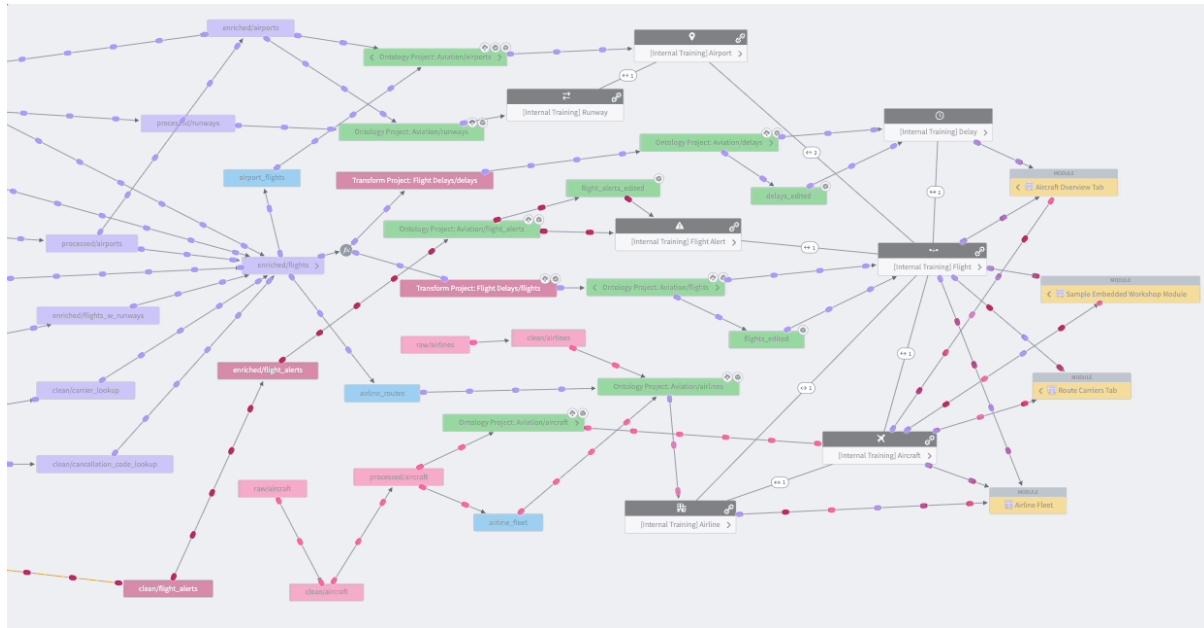


Data Lineage

Data Lineage is an interactive tool that facilitates a holistic view of how data flows through the Foundry platform.



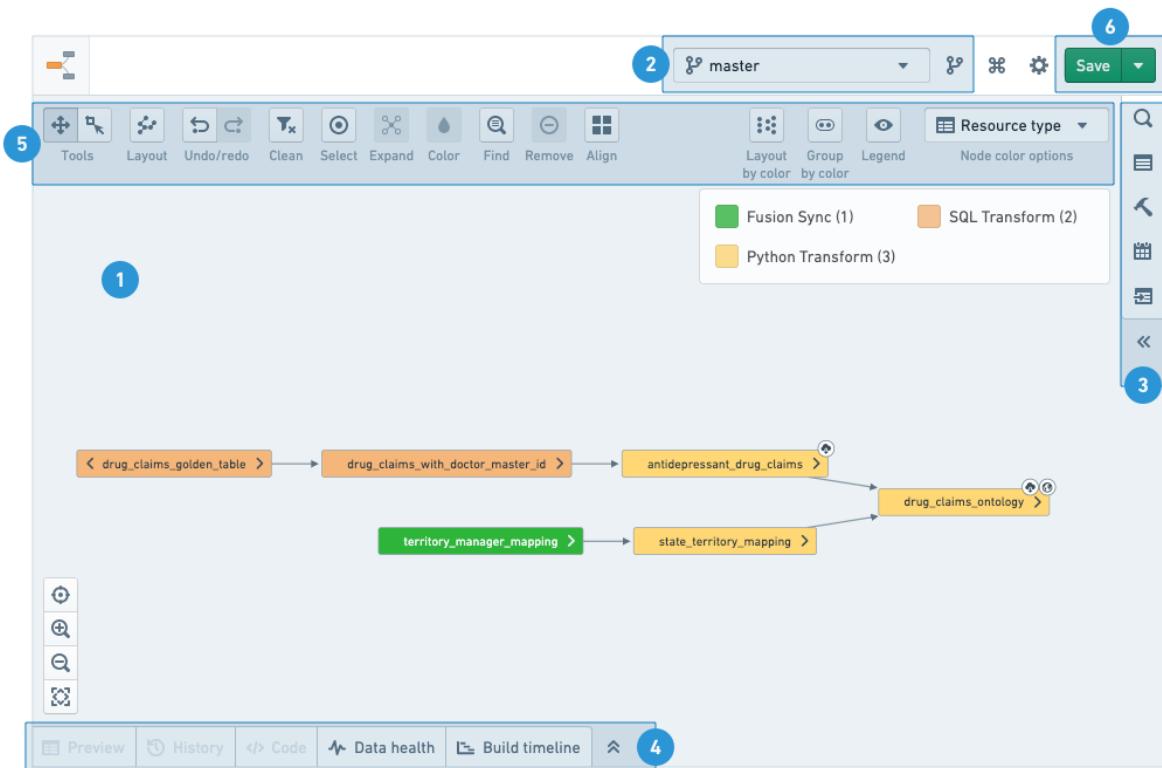
With Data Lineage, you can:

- Easily find and discover datasets
 - Search to find datasets using Project, table, and column names
 - Click through Foundry Projects to browse data
- Explore pipelines through a powerful interface
 - Expand or hide ancestors and descendants of datasets
 - View attributes of a group of tables at once
 - Visualize your pipeline through coloring (e.g. color out-of-date tables)
 - Drill into details about your data such as its schema, when it was last built, and the code that generated the data itself
- Collaborate with teammates
 - Create pipeline snapshots to share with other users

Navigation

To make the best use of the Data Lineage application, you will need to know how to navigate the graphs, use tools, and configure branch and graph properties. The following numbered sections correspond to the numbers on the screenshot below:

1. Lineage graph
2. Branch settings
3. Side panel
 - o Search & Browse
 - o Properties and Histogram
 - o Manage Builds
 - o Manage Schedules
 - o Related Artifacts
4. Node details panel
5. Graph tools
6. Save graph



Lineage graph

The graph is your workspace for arranging and manipulating nodes as you explore your data pipeline.

After adding nodes to the graph, you can add their related resources by clicking on the arrows on either side of the node or by using the **Expand** option in the graph tools.

Nodes are arranged with auto-layout by default, but you can rearrange nodes manually by clicking and dragging them. To re-enable auto-layout, choose **Layout all nodes** under the **Layout** option in the graph tools.

Click and drag to pan around the graph when in the default **Panning mode**. To use the cursor to select multiple nodes, switch to **Drag select** mode in the graph tools or hold Shift while clicking and dragging. You can select a node by clicking it, or select multiple nodes with Ctrl/Cmd + click.

Branch settings

Select a branch from the list to explore data pipelines in that branch. The graph and any of the other helpers would show information based on the selected branch. If the branch does not exist for a resource, the listed fallback branches would be used instead (in the order they appear on the list).

Side panel

Search and browse

Use the search helper to find Foundry resources and add them to the graph. Use the free-text search or browse the tree to find resources. Add a resource by clicking on it or use the buttons at the bottom of the view to add all search results (including or excluding the content of sub-folders). Use the **Advanced** tab to add filters to your search and sort your results.

Warning

When viewing a folder with subfolders, you can recursively add *all* tables in all subfolders to the graph. Adding too many nodes at once may effect the graph's performance.

Properties and histogram

When you select a single node on the graph, the properties helper shows you the details of the resource. Depending on the type of resource you select, the properties helper shows available Foundry apps under the **Actions** menu and other links and actions (reporting issues, adding descriptions, etc.).

When you select multiple nodes on the graph, you will see the histogram helper. The helper displays common properties and their values alongside the number of appearances of each value on the graph. By clicking on the values, the matching nodes are highlighted. If you want to drill down to just those resources, click on **Update selection**.

RESOURCE TYPE	#
Code Workbook Dataset	36 / 36
Uploaded Dataset	16 / 16
Fusion Sync	4
Object Type	2
Writeback Dataset	1
TIME CREATED	#
More than 30 days ago	52 / 57
TIME LAST BUILT	#
More than 30 days ago	52 / 59

X Reset
✓ Update selection

Use the **Copy names** button in the histogram to copy the names of all currently selected resources. The full names (including path) are copied to the clipboard as a comma-separated list.

Manage builds

The builds helper offers you three build strategies:

- Build only selected datasets
- Build all datasets between the selected datasets
- Build the selected datasets and all of their ancestors

Manage schedules

The schedules helper allows you to set and edit build schedules for selected resources on the graph.

When viewing and creating schedules in Data Lineage, the schedules apply to the branches (including fallback branches) configured in the graph.

Related artifacts

The related artifacts helper displays artifacts directly linked to the nodes selected on the graph. Deleted and automatically saved files are excluded from the list unless chosen

otherwise. You can also get to the same list of related artifacts by hovering over the right arrow of each node on the graph.

Node details

Click on a node to see more details:

- **Preview:** A sample of the data in the selected dataset.
- **History:** An overview of dataset change history. The overview includes tabs for logs, files, metadata, schema and job specifications.
- **Code:** If code was used to generate the dataset, it will display here
- **Data Health:** All the health checks set on the selected datasets.
- **Build timelines:** A Gantt chart of actual build time for the selected datasets.

Graph tools

The graph tools provide a set of graph exploration, navigation, and customization capabilities:

- Node coloring
- Layout
- Expand
- Find
- Selection

Node coloring

You can color the nodes on your lineage graph by several properties and metrics. Node coloring is commonly used to communicate lineage structure, troubleshoot issues, monitor pipelines health, and manage builds. You can also create your own custom coloring and arrange the graph based on the colors you assigned.

You can arrange your nodes on the graph by color group under **Layouts**.

Layout

The layout button provide various arrangement option for the nodes on the graph. **Layout all nodes** applies automatic layout for all the nodes on the graphs. When you select multiple nodes on the graph, you can apply other layouts (vertical, hierarchical, by level, etc.).

You can use various useful keyboard shortcuts in the lineage graph. View the full list under the **Keyboard shortcuts** button at the top right corner of the app.

Expand

Use the **Expand** tool to expose ancestors and descendants of nodes in the graph.

Find

Use **Find** to search for nodes on the graph. You can either search for the name of the node or column names in datasets.

Selection

The **Selection** tool allows you to easily select nodes on the graph:

- **Select All:** Selects all the nodes currently on the graph.
- **Invert selection:** De-selects all currently selected nodes and selects the rest of the nodes on the graph.
- **Select children:** Adds all the direct children of the currently selected nodes to your selection
- **Select parents:** Adds all the direct parents of the currently selected nodes to your selection.

Save graph

You can save and share your lineage graph with other Foundry users in the following ways:

- **Save / Open:** Save your Data Lineage graph and re-open it by clicking on **Open graph**.
- **Get quick share link:** Generates a shareable link that provides read-only access to your graph.
- **Export graph to SVG:** Generates a static image of your lineage graph.

Your branch choice is saved with your saved graph. If you load a graph with a different branch configuration than you currently have, you will be asked if you would like to switch branches to the saved branch configuration.

Data Lineage questions

How can I see the backing and writeback datasets for my object type in Data Lineage?

- First, add your object to the Data Lineage graph by searching for it in the right panel (the tab with a magnifying glass icon). Select **Object types** to filter your search, then enter the name of the object for which you want to view the backing and writeback datasets.

- Next, select the arrow on the left side of your Object type to show its ancestors. This should produce one ancestor node if your object type is read-only and two ancestor nodes if your object type has writeback enabled. Make sure **Resource overview** is selected in the **Node color options** dropdown to see your Writeback Dataset colored as per the legend in the top right. Backing schema dataset colors depend on the transform type used.
 - Your writeback and backing datasets for an object type will also have a small globe icon in the top right.
-

What datasets in my pipeline also have a specific column?

1. First, ensure all desired datasets in your pipeline have been added to the Data Lineage graph.
2. Next, select desired datasets by using the **Select** mode in the **Tools** toggle in the upper left corner of the canvas.
3. Then open the **Histogram of selection properties** from the right side panel. Under the section titled **Frequent columns**, you will see the most frequent columns by column name in your selection.

Selecting one of these columns will highlight the datasets in your selection that contain this column.

Who was the last person to modify a resource on this pipeline?

- First, ensure that all datasets of interest in your pipeline have been added to the Data Lineage graph.
 - Next, select datasets by using Select mode from the **Tools** toggle in the upper left corner of your screen. Then, open the **Histogram of selection properties** from the right side panel.
 - Under the **Last Modified** section, you will see the last user(s) to modify datasets in your selection. Selecting a username will highlight the datasets that user last modified within the graph.
-

How can I find which of my datasets have open transactions?

In the dropdown menu in the top right side, choose **Build Status**. Now, you should be able to see if any dataset is currently running. Any such dataset has an open transaction.

Where are most of the datasets used in the pipeline stored?

- First, ensure that all datasets of interest in your pipeline have been added to the Data Lineage graph
- Next, select all datasets of interest with the **Select** mode from the **Tools** toggle in the upper left corner of your screen. Then, open the **Histogram of selection properties** from the right side panel.
- Under the section titled **Frequent folder paths**, you will see the most common folder paths for resources in your selection.

Selecting a golden path will highlight the resources in this path on the graph. Hovering over a folder path will show you the full path.

You can select multiple properties in the **Histogram of selection properties** panel such that the graph highlights all resources that satisfy your selection.

How can I share my unsaved Data Lineage graph?

To share your unsaved Data Lineage, select the arrow in the top right corner near Save. Once there, you can see a quick share link.

Why is my dataset not up-to-date?

There are a few reasons why your dataset may not be up-to-date.

Consider the following reasons why your dataset may not be up-to-date:

- Is your dataset build failing?
- Is there an upstream dataset that has not built and is not up-to-date?
- Have you received up-to-date data from the source?

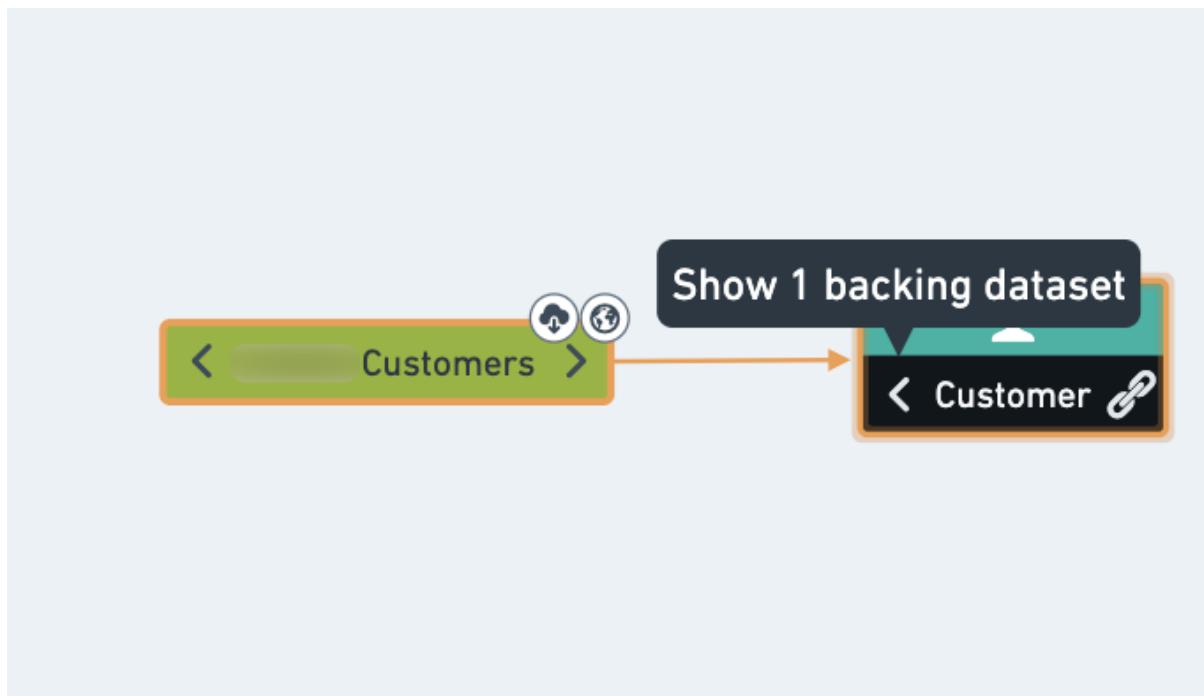
You can easily answer these questions in Data Lineage:

1. First, verify the status of each of the resources in your pipeline by opening up the dataset of interest in Data Lineage and then right-clicking on the node.
2. Then, select **Expand node....** You can view all ancestor nodes for that dataset by selecting the double left arrow above **Expand parents....**
3. Next, select the **Build status** option in the **Node color options** dropdown menu in the top right to view the build status of every resource in your pipeline. This view of your pipeline will make it easier to diagnose stale datasets.

Explore data lineage

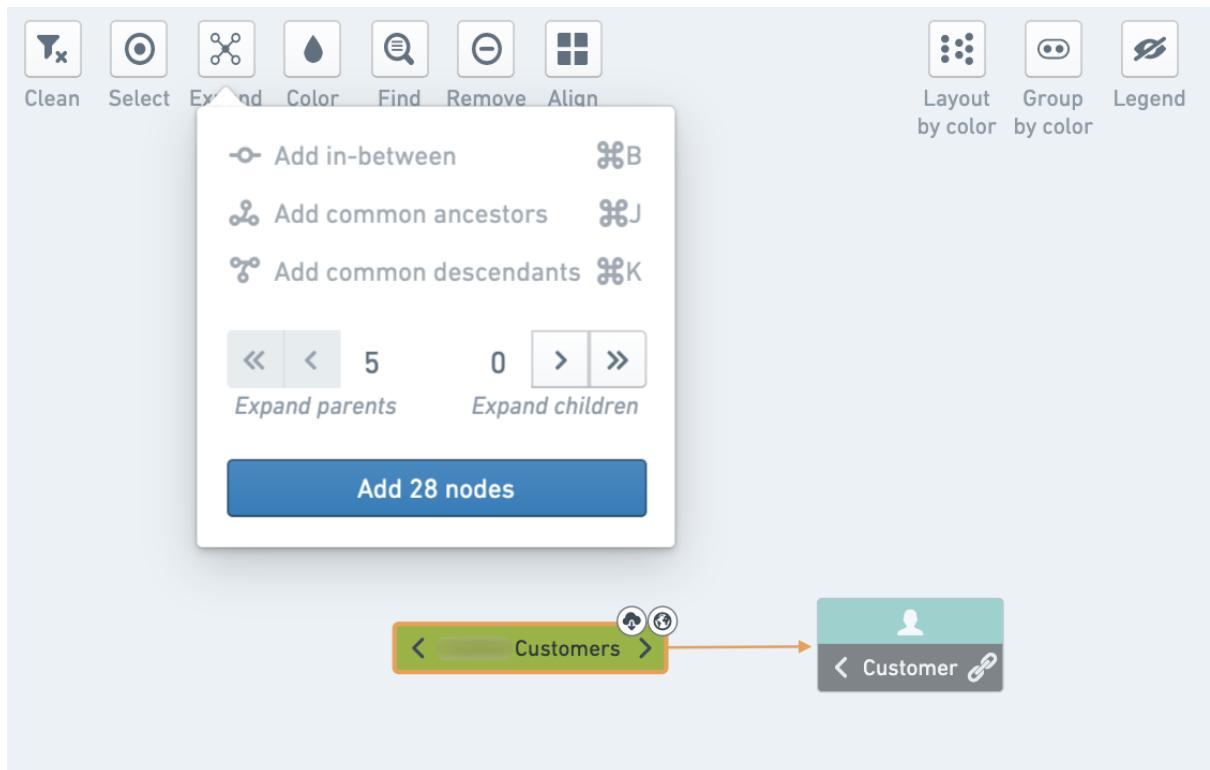
Data lineage helps you understand how your data came to be. There are various ways to explore data pipelines in the Data Lineage app. Consider one common path:

1. Using the **Search** helper, find your resource (for example, a dataset or an object type) and add it to the graph.
2. Click on the left arrow of the node to expose the direct parents of the resource.



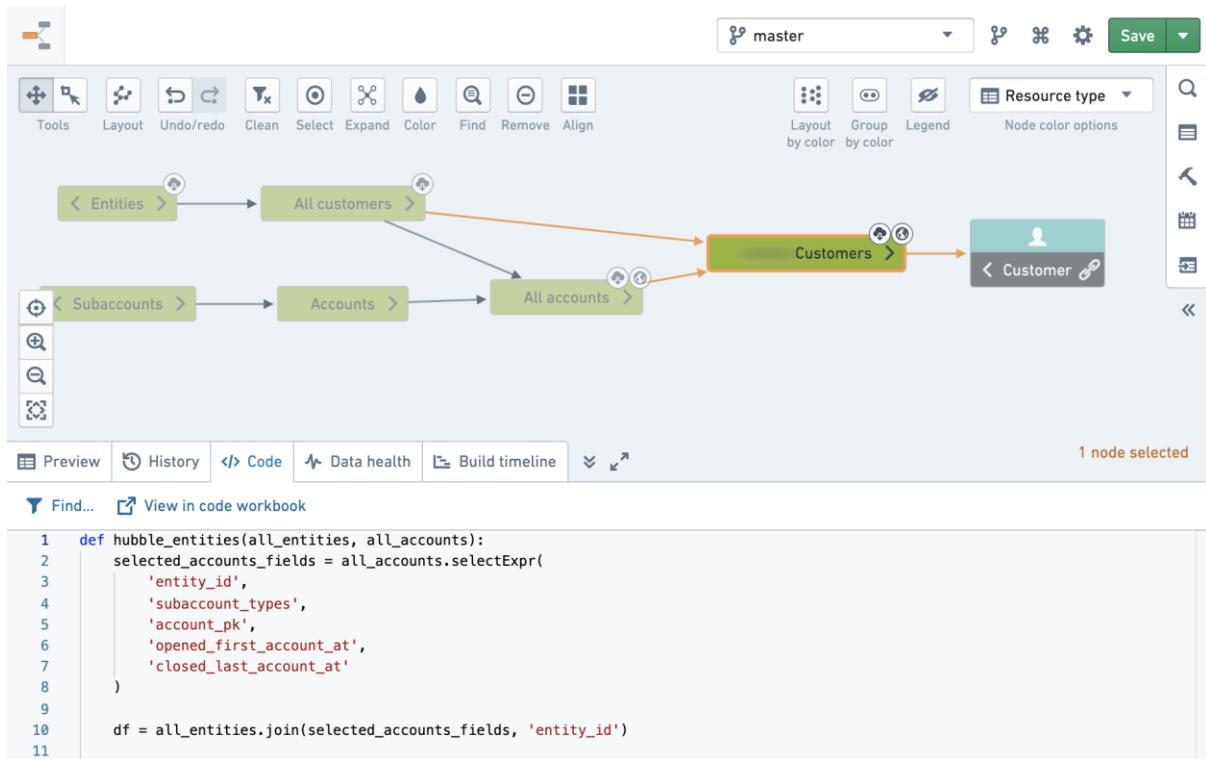
3. To expand your graph, select the next resource on the graph and click the **Expand** button in the graph tools.
4. Click on the chevron button to define the number of levels to expose. Click the double-chevron to expand all the way to the raw data (or all the way to the final descendants).

Adding too many nodes simultaneously may affect the graph's performance and usability. Keep a manageable number of nodes by checking the node count in the **Expand** tool.



You can find the relation between two nodes on the graph by selecting the **Expand** button and adding all nodes in between the resources or all common ancestors/descendants.

5. Get more information on one of the datasets by selecting the dataset and using the bottom panel to display a preview of the data.
6. Click on **Code** to view how the dataset was created.



- Click on **View in code workbook** or **View in repository** to see the original code and make changes as needed (subject to permissions).

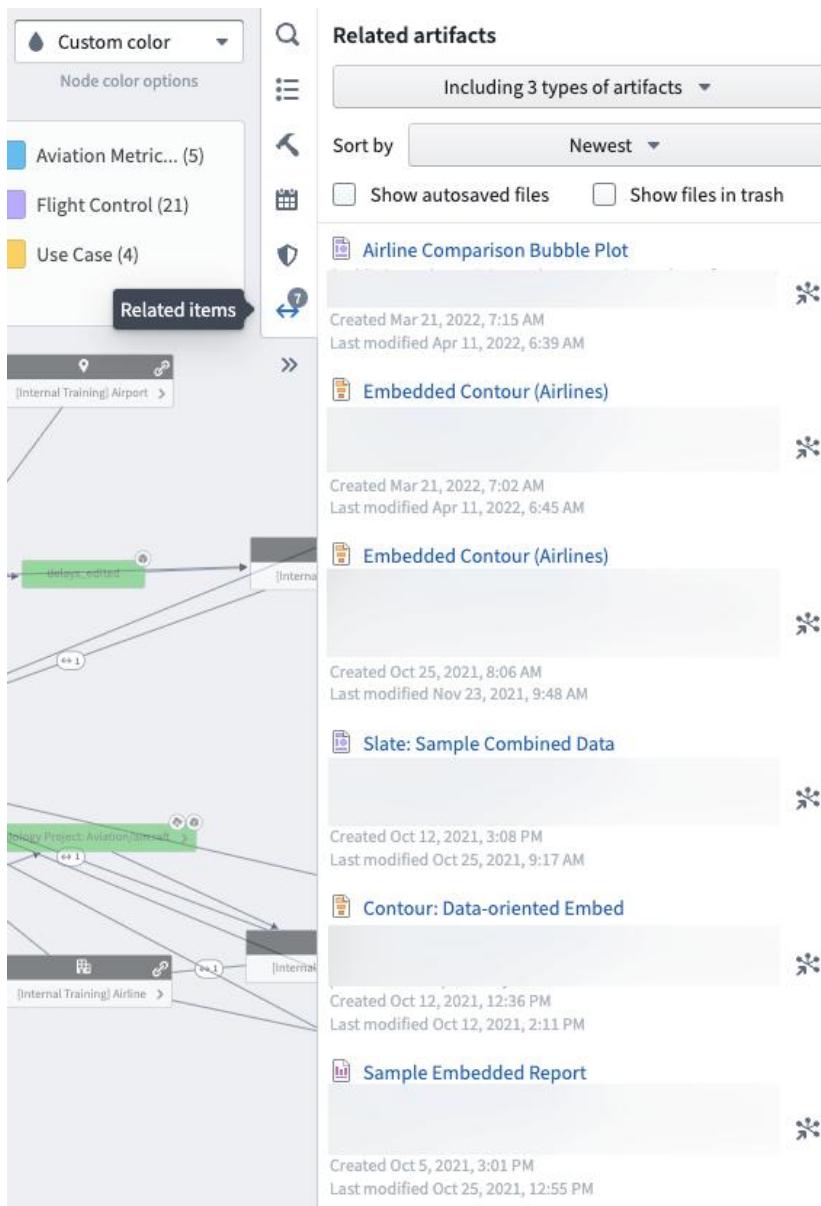
Some options may be unavailable for some datasets depending on the type of resource. For example, **Code** is only available for Code Workbook or Code Repositories. For Fusion sheet syncs with no code to show, you may have the option to view the source sheet and make your changes there (if you have appropriate permissions).

Explore artifacts and ontology entities

You can find Foundry artifacts and ontology entities related to your datasets in Data Lineage. The Data Lineage interface allows you to navigate directly to these resources and see how they fit in your ontology.

Find related artifacts

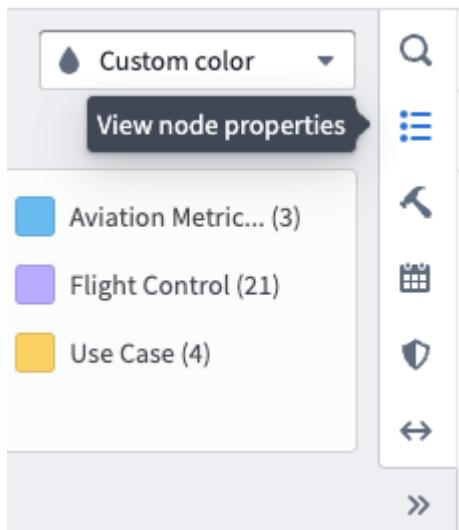
In your data lineage graph, select a dataset. Then, select **Related items** in the right sidebar to expand the **Related artifacts** panel. The **Related items** icon will show a badge with the number of artifacts related to the selected dataset. In the artifacts panel, you can see a list of related resources throughout Foundry, including Contour visualizations and Slate applications.



Click on the node icon next to a resource to zoom in on the related dataset, or click the resource to open it in the corresponding application in a new tab. You can filter the list of related artifacts to include different item types and sort the list by oldest, newest, name, path, or last modified.

Find ontology entities

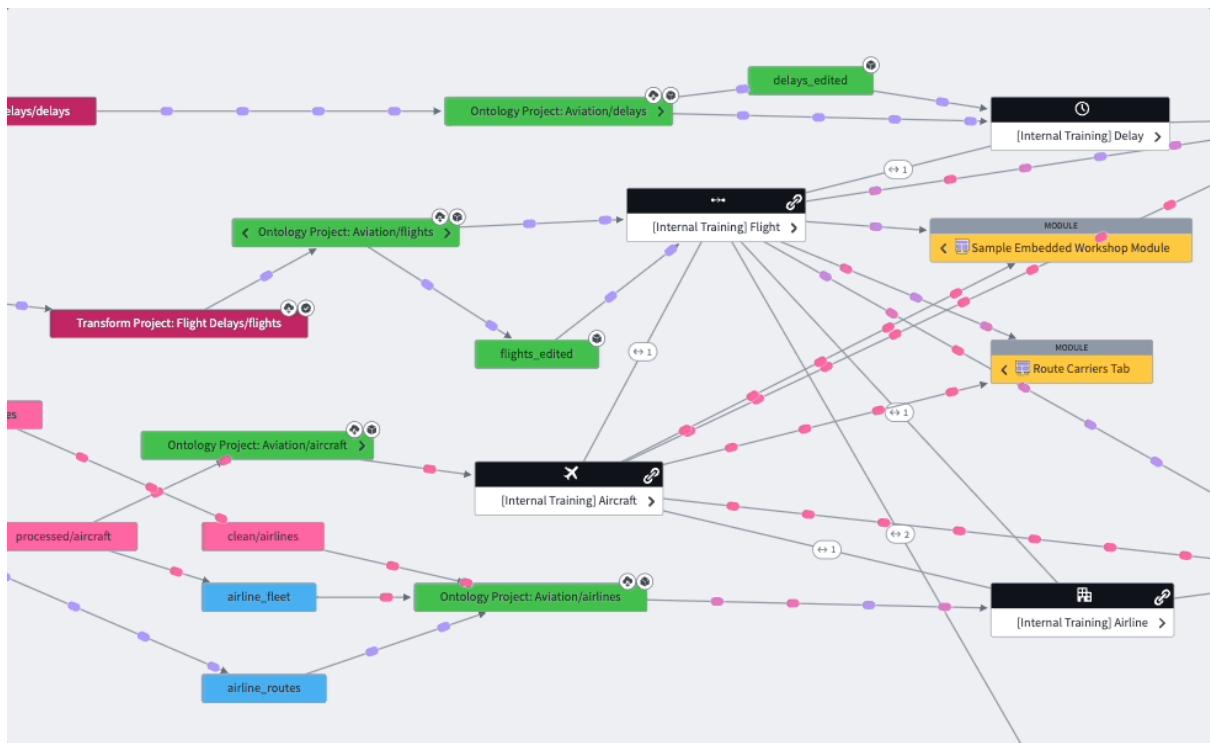
Find object types defined by datasets in your lineage graph by selecting the dataset and opening the **View node properties** panel in the right sidebar.



In the **About** tab, you will see any object types that were created with the selected dataset. Click the **Settings** icon next to the object type to view its configuration in a new Ontology manager tab.

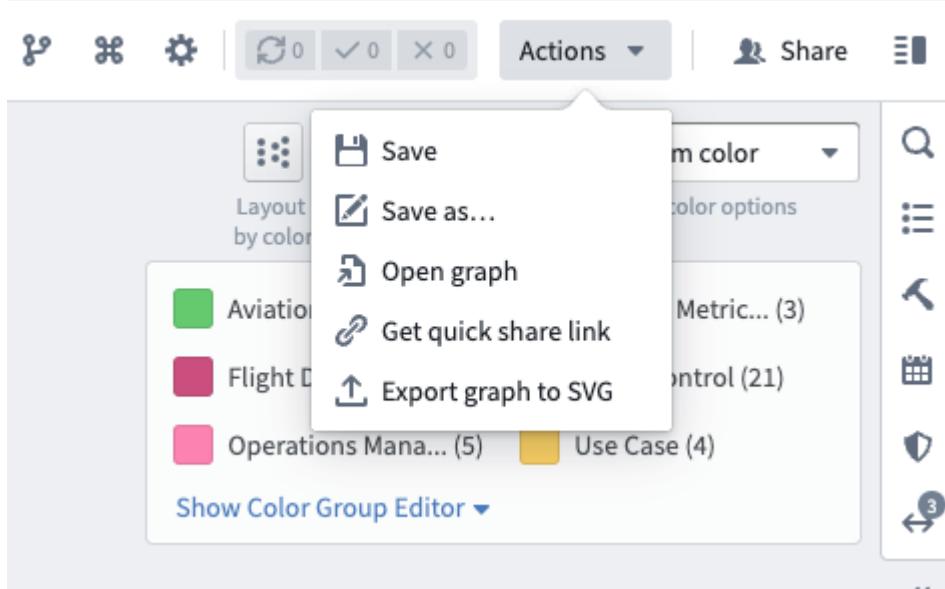
A screenshot of the 'About' tab. At the top, there are buttons for 'color options', 'Columns', and 'About'. A tooltip appears over the 'About' button stating: 'This dataset is used to create the Ontology object [Internal Training] Flight Alert'. Below this, a table lists object types: 'control (21)' and 'e (4)'. The row for 'control (21)' contains a calendar icon, a warning sign icon with the text '[Internal Training] Flight Alert', and a gear icon. The row for 'e (4)' contains a shield icon. At the bottom, it shows 'Updated Mar 21, 2022, 7:40 AM by Foundry' with a green checkmark icon.

You can also add object types to your data lineage with the **Search Foundry** tool in the right sidebar. Use a basic or advanced search to find an object type and select it from the list to add it to your graph. You can then view link types related to the object type and use the graph to visualize connections between your datasets and the newly added object type.



Save and share a graph

Data Lineage allows you to easily save your graph and share it with other users. You can find multiple ways to save and share by selecting the **Actions** tab in the upper right of the application and selecting a method from the dropdown menu.

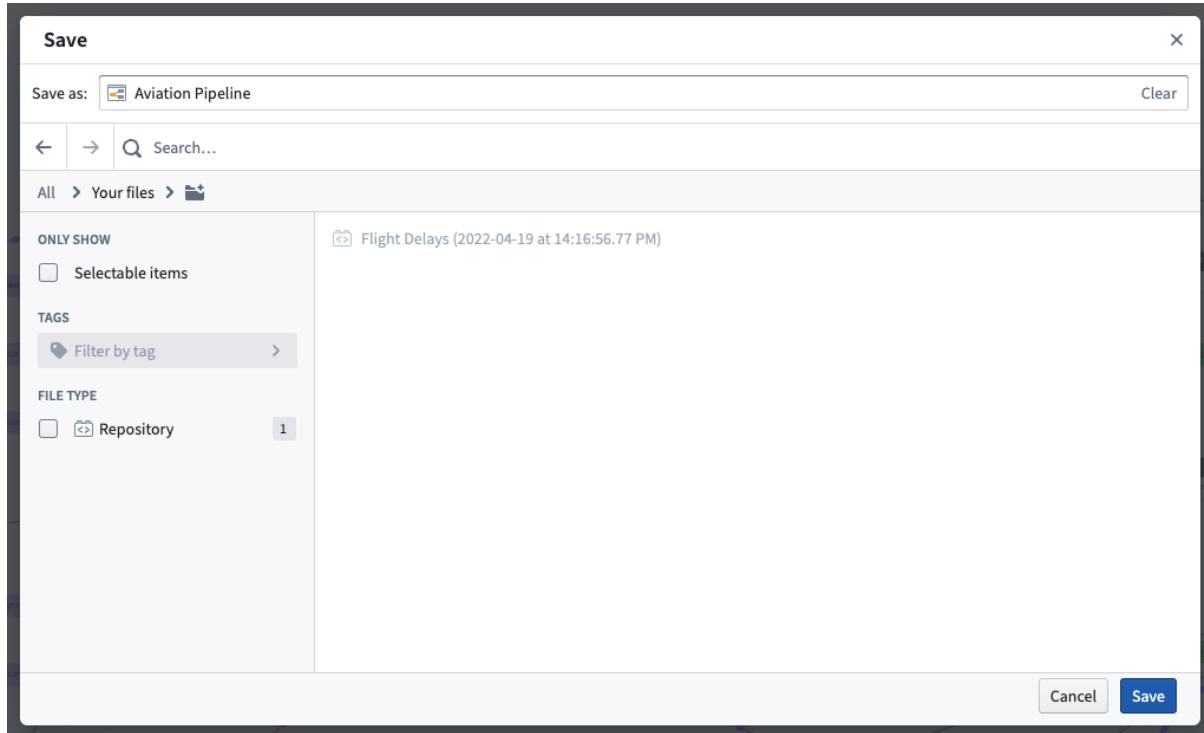


Save a graph

Select the following options in the **Actions** dropdown menu to save your Data Lineage graph:

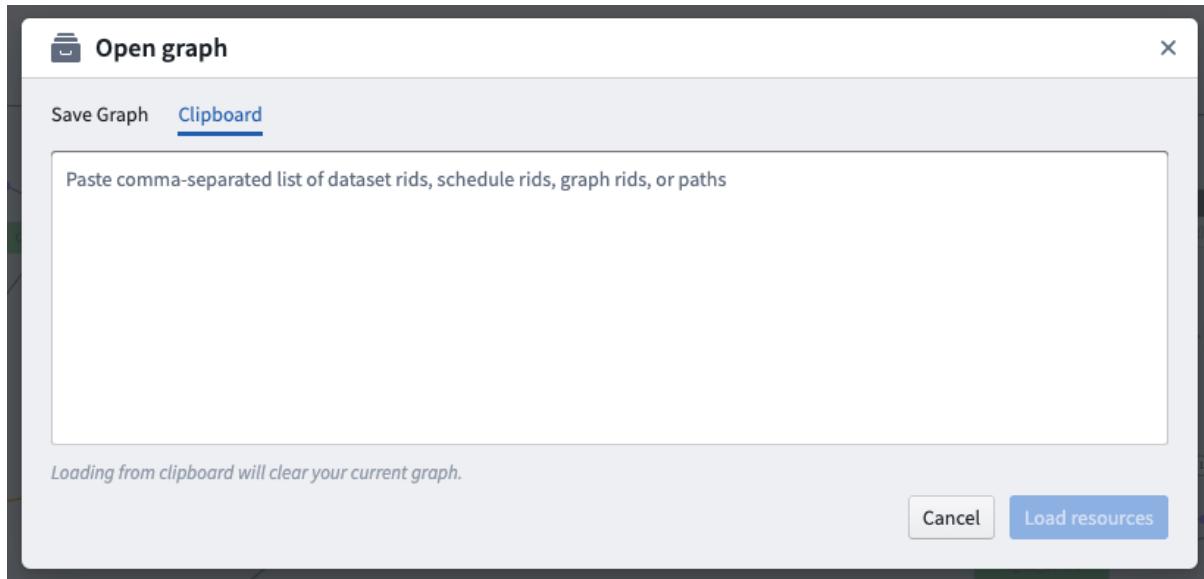
- **Save:** Save your Data Lineage graph to where it currently lives in your files or Project.

- **Save as....**: Choose a name for your lineage graph and save it to a new location in your file system.

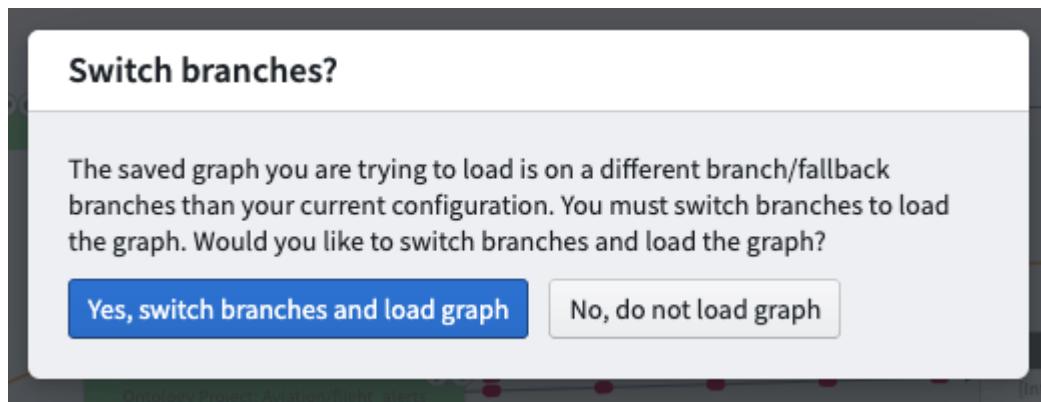


You can also open a previously saved graph with the following option:

- **Open graph**: Choose to open a different saved graph to which you have access, or open the **Clipboard** tab to enter the resource identifier (RID) of a dataset, schedule, graph, or path.



Your branch choice is saved with your saved graph. If you load a graph with a different branch configuration than you currently have, you will be asked if you would like to switch branches to the saved branch configuration.



Share a graph

You can share a graph with other users using the options below:

- **Get quick share link:** Generate a shareable link that provides read-only access to your graph. Note that this option is only available for users belonging to the same Organization. To share a graph across Organizations, ensure the graph is saved in a shared Project accessible to both Organizations.
- **Export graph to SVG:** Generate and download a static image of your lineage graph in .svg format.

You can also select **Share** in the upper right of the application to open the sidebar and view **Roles** details. From here, you can turn on link sharing or give a user or group access to your graph.

The screenshot shows a modal window titled "Aviation Pipeline" under "Details > Access > Roles". The "Link sharing" section is active, indicated by a blue toggle switch. A URL field contains "https://", and a dropdown menu next to it is set to "Viewer". Below this, a note states: "People who visit the link will be granted the **Viewer** role. They will still need to meet access requirements to be able to access the file." The "Roles" section below has a search bar with a magnifying glass icon and placeholder text "Add a user or group...".

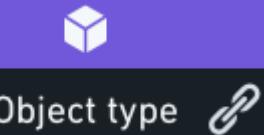
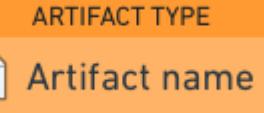
Node coloring

There are several built-in options for coloring graph nodes to give you more information about your pipeline:

Coloring option	Description
No color	Would remove coloring altogether
Custom color	Allows you to select nodes and assign them a color by clicking on the Color button
Data Catalog	Nodes would be colored based on the Data Catalog collection they are in. If the node belongs to more than one collection it would be colored as "In multiple collections"
Folder	Colors the nodes by name of the folder the resource is located in
Issues	Colors the nodes by the number of Foundry issues assigned to them. This option would also allow you to filter by issues labels.

Coloring option	Description
Permissions	Colors the nodes by the level of access the user has to the data or the resource. If you have access to the resources on the graph, this view also allows you to choose any Foundry user and view their permissions.
Project	Colors the nodes based on the Foundry Project they are in.
Repository	Colors the nodes based on the code repository used to create them. You can either color the nodes by the name of the repository, or by its type (e.g. Code Repository, Code Workbook).
Resource overview	This view colors the node by details of the resource. The details of the resource generally refer to the way the resource was created (such as in Contour, Code Workbook, Fusion spreadsheet sync, by Upload, etc.).
Resource type	Colors the nodes by Foundry resource types.
Build status	Indicates the current build status of each dataset on the graph. If the nodes are grouped the more severe status would be presented.
Data Health	Indicates the status of resources health checks with the ability to filter to only watched health checks. If the nodes are grouped, the color of the group would indicate the most severe health check status of the group.
Out-of-date	<p>This option would indicate if the data or logic is out of date relative to the dataset ancestors.</p> <p>Out-of-date with parent means a direct parent of the resource had been updated and the resource itself hasn't yet updated accordingly.</p> <p>Out-of-date with ancestor means the resource is up-to-date with its direct parents, but there is a resource upstream that is more updated. This options allows you to filter for two types of updates: Data and Logic.</p> <p>Data out-of-date means the data was updated in an ancestor and the resource hasn't yet picked up the update in a build.</p> <p>Logic out-of-date means job-specs has changed.</p>
Schedule count	Indicates the amount of build schedules set on a dataset with the option to filter out paused schedules.
Sync status	If there are syncs set on the dataset, this option would indicate the status of the sync

Coloring option	Description
Time last built	Indicates the time since the last time the dataset was successfully built.
Build duration	According to the most recent successful build of each resource, this option would indicate the approximate build duration
Files	Colors the nodes on the graph by files-related metrics: Average file size, count of files and dataset size
Row count	Colors nodes by the number of rows in each dataset. If row count does not exist, it could be calculated in the dataset details helper or in the dataset view in Foundry (dataset app).
Spark usage	Colors each node by Executor run/CPU time in a given period
User views	Colors nodes by the number of user views
Branch	Indicates the currently viewed branch of each node on the graph.
	Indicates the code status for this node/dataset.
	CI running means CI checks are currently running for this node.
	CI Failed means that CI checks failed on this node.
	Out of date means that the code is out of date for this node.
	Unavailable means that the node/dataset is not a stemma backend or that the user is lacking permissions.
Storage	Indicates where data is stored. Will be Foundry unless you are using Virtual Tables.
	Indicates the compute engine used by each node on the graph. For transforms run in Foundry, this will show the type of compute used (Spark or Flink, for example). For transforms that use compute pushdown, this will indicate the external compute engine used (BigQuery, Databricks or Snowflake, for example).
Transaction type	Indicates each nodes transaction type: Append or Snapshot
	Graph elements reference
	Node types

Node	Type	Description
 Data source	Data source	This is the name of the data source as it appears in Data Connection.
 Dataset name	Dataset	Foundry datasets and the lineage between them. The color of the dataset node depends on user selection. Dashed border indicates unstructured datasets.
 Object type	Object type	Ontology object types. The icon and color of the node depends on the definition of each object type. When clicking on the “link” icon next to the object type name, Data Lineage shows the relations between this object type and other object types.
 ARTIFACT TYPE Artifact name	Artifact	Data Lineage exposes different Foundry artifacts like: Contour analyses, Reports, etc. The color of the node depends on the artifact type, which is indicated at the top of the node.

Node indicators

Node indicators appear on top of dataset nodes and provide additional information about the resource.

Indicator	Type	Description
	Open issues	This indicator signals there are currently open issues associated with the node on the graph. Hovering over this signal indicates the number of open issues.
	Defines an object type	This indicator appears on datasets that are used to define Ontology object types. Hovering over the right arrow of allows you to expose those linked object types.
	Syncs	Datasets with this indicator on them have syncs to other databases or systems. You can view these syncs by selecting the node and opening the Properties panel, or by opening the “Details” tab in Dataset Preview (right click on the node and click on Open).

Indicator Type	Description
 Trashed	This indicator appears on nodes representing deleted datasets or artifacts. Deleted nodes are also partially faded with their name crossed out.

Preview and logic

The Data Lineage interface allows you to view previews of selected datasets or media sets, as well as examine the associated code to understand the logic behind the dataset or media set.

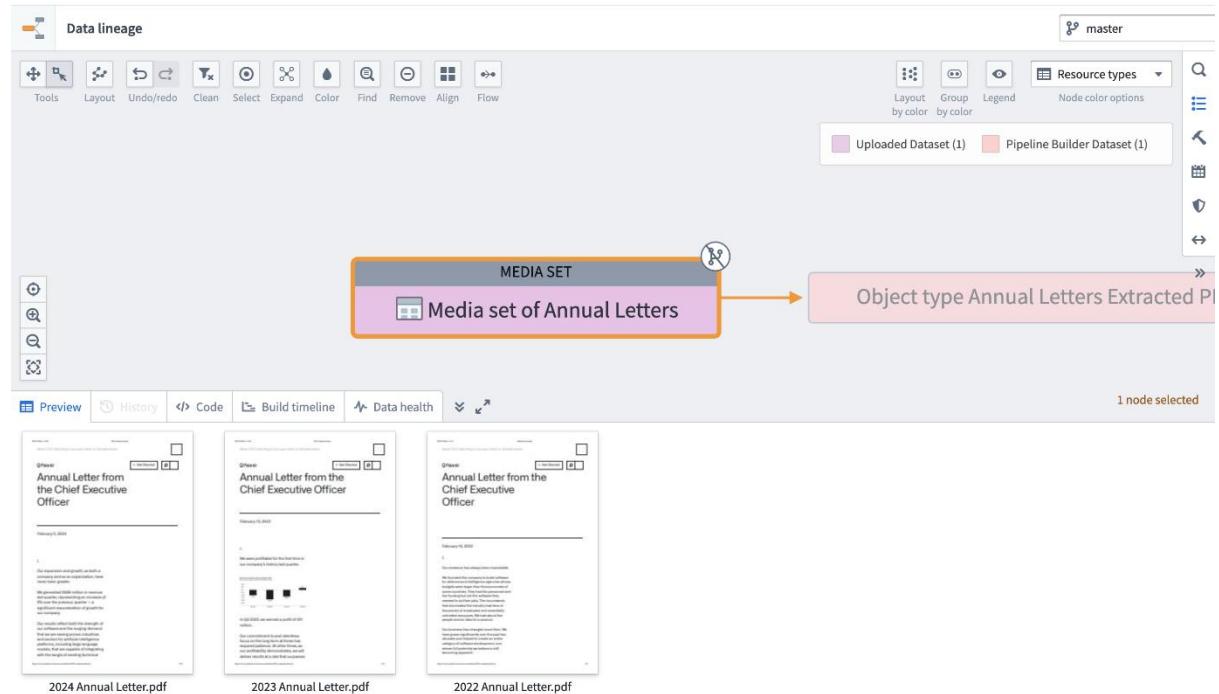
Preview

To see a preview of a dataset or media set, select it in your data lineage graph, then choose the **Preview** tab in the bottom left of the interface.

Media set

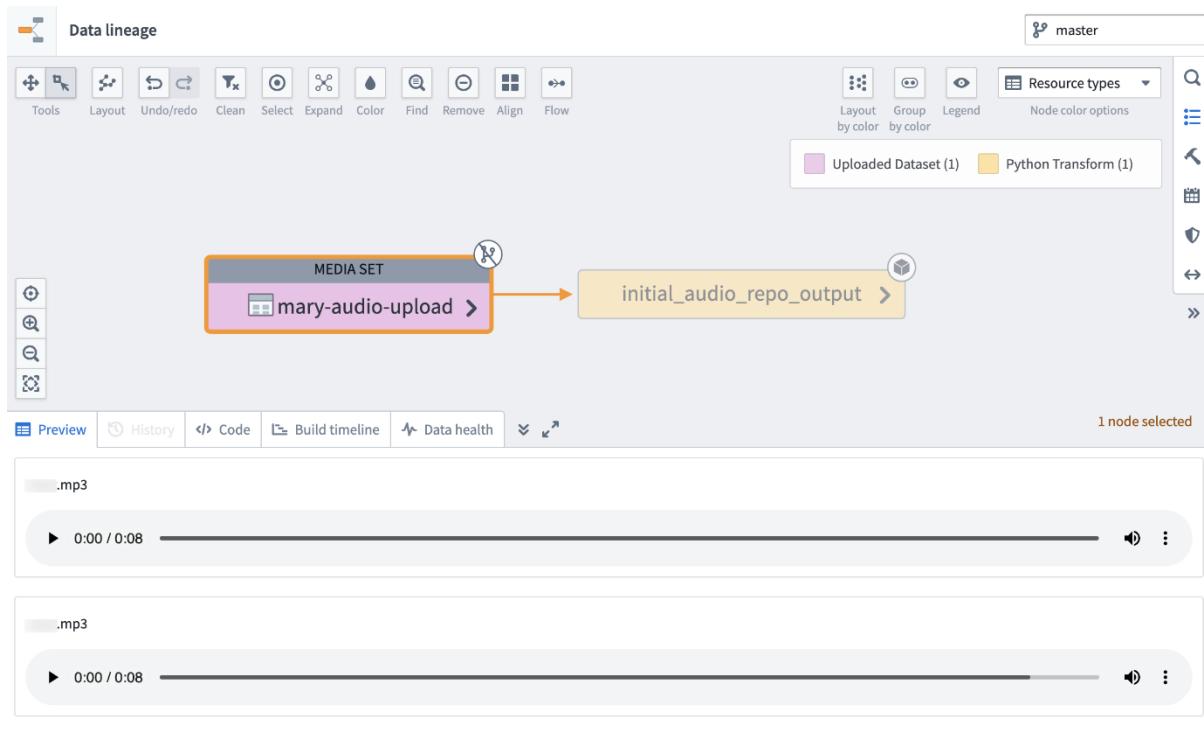
When the media set preview expands, you can view the contents of your media set.

Example of PDF preview:



The screenshot shows the Data Lineage interface with a selected node labeled "MEDIA SET" containing "Media set of Annual Letters". An arrow points from this node to a preview window displaying three PDF documents: "2024 Annual Letter.pdf", "2023 Annual Letter.pdf", and "2022 Annual Letter.pdf". The interface features a toolbar with various icons for tools like Tools, Layout, Undo/redo, Clean, Select, Expand, Color, Find, Remove, Align, Flow, and a legend. A legend panel indicates "Uploaded Dataset (1)" and "Pipeline Builder Dataset (1)". The bottom navigation bar includes tabs for Preview, History, Code, Build timeline, and Data health, with "1 node selected" displayed.

Example of audio preview:



Dataset

When the dataset preview expands, you can scroll through the first 300 rows of the selected dataset. You can also search for specific columns using the **Search columns...** field to the right of the preview window. The preview of your dataset will look different depending on the type of data within your dataset.

flights												Showing 300 of 4.2m rows	61 columns	<input type="text"/> Search columns...
	op_carrier	tail_num	op_carrier_fl_num	origin	origin_city_name	origin_state_abr	origin_state_nm	dest	dest_city_nm					
1	OO	N432SW		4712 DTW	Detroit, MI	MI	Michigan	CAK	Akron, OH					
2	YX	N731YX		3651 CLT	Charlotte, NC	NC	North Carolina	EWR	Newark, NJ					
3	F9	N702FR		1262 LAS	Las Vegas, NV	NV	Nevada	ORD	Chicago, IL					
4	AS	N403AS		450 SEA	Seattle, WA	WA	Washington	ONT	Ontario, CA					
5	OO	N138SY		5793 SAN	San Diego, CA	CA	California	LAX	Los Angeles, CA					
6	AA	N568UW		1916 CLT	Charlotte, NC	NC	North Carolina	ORD	Chicago, IL					
7	B6	N599JB		2074 TPA	Tampa, FL	FL	Florida	EWR	Newark, NJ					
8	AS	N525VA		1598 SFO	San Francisco, CA	CA	California	PSP	Palm Springs, CA					
9	OO	N947SW		5157 SPI	Springfield, IL	IL	Illinois	ORD	Chicago, IL					
10	YX	N882RW		5885 LGA	New York, NY	NY	New York	CHS	Charleston, SC					
11	UA	N19141		341 EWR	Newark, NJ	NJ	New Jersey	DEN	Denver, CO					
12	DL	N312DN		2137 TPA	Tampa, FL	FL	Florida	ATL	Atlanta, GA					
13	AA	N941NN		1097 SEA	Seattle, WA	WA	Washington	ORD	Chicago, IL					
14	UA	N69847		599 BOS	Boston, MA	MA	Massachusetts	ORD	Chicago, IL					
15	B6	N958JB		51 BOS	Boston, MA	MA	Massachusetts	MCO	Orlando, FL					

Logic

Select the **Code** tab to view the code logic of the selected dataset or media set. From the **Code** view, you can make quick edits, search for items, or open the code in the repository or other application used to derive the data.

The screenshot shows the Data Lineage interface with a code editor containing Python code for a transformation. The code imports various libraries like `transforms.api`, `pyspark.sql`, and `itertools`. It defines a function `transform_df` that takes `Input` and `Output` parameters, configures a profile, and performs operations such as reading from a datasource, performing joins, and writing back to another datasource. A preview panel on the right shows the resulting lineage graph.

```

1 from transforms.api import transform_df, Input, Output, configure
2 from pyspark.sql import functions as F, types as T
3
4 from datetime import datetime
5 from dateutil.relativedelta import relativedelta
6 import pytz
7 from timezonefinder import timezonefinder
8 import pandas as pd
9 from itertools import chain
10
11
12 @configure(profile=['EXECUTOR_MEMORY_MEDIUM'])
13 @transform_df
14     Output("//Public/Foundry Training and Resources/Foundry Reference Project/Datasource Project: Flight Control System/flights/enriched/flights"), # noqa
15     flights=Input("//Public/Foundry Training and Resources/Foundry Reference Project/Datasource Project: Flight Control System/flights/processed/flights"),
16     carrier_lookup=Input("//Public/Foundry Training and Resources/Foundry Reference Project/Datasource Project: Flight Control System/flights/clean/carrier_lookup"),
17     cancellation_code_lookup=Input("//Public/Foundry Training and Resources/Foundry Reference Project/Datasource Project: Flight Control System/flights/clean/cancellation_code_lookup"),
18     distance_group_lookup=Input("//Public/Foundry Training and Resources/Foundry Reference Project/Datasource Project: Flight Control System/flights/clean/distance_group_lookup"),
19     delay_online_lookup=Input("//Public/Foundry Training and Resources/Foundry Reference Project/Datasource Project: Flight Control System/flights/clean/delay_online_lookup")

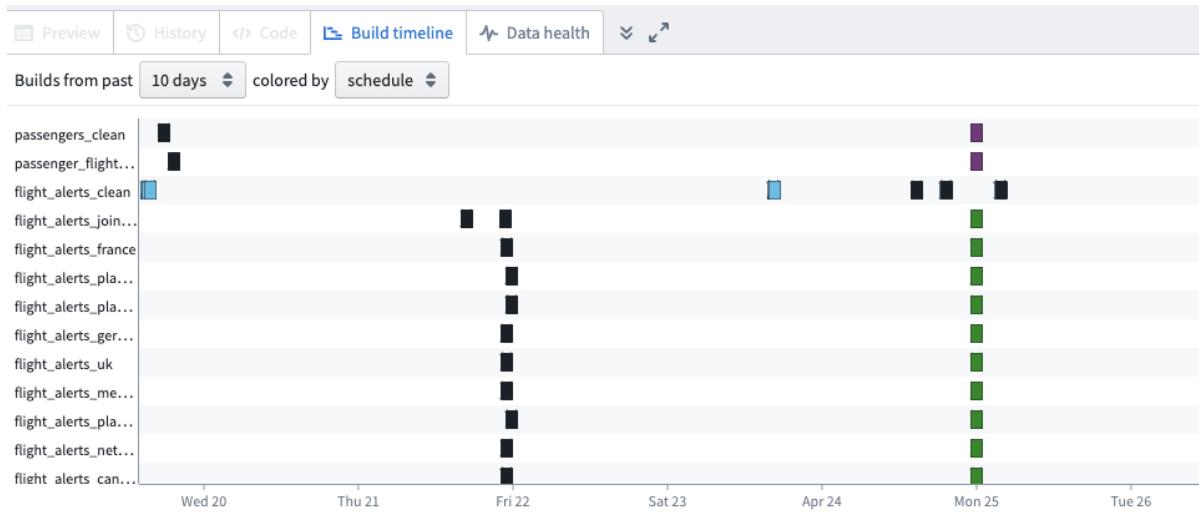
```

Uploaded and writeback datasets do not have associated code to view in Data Lineage.

View build timeline

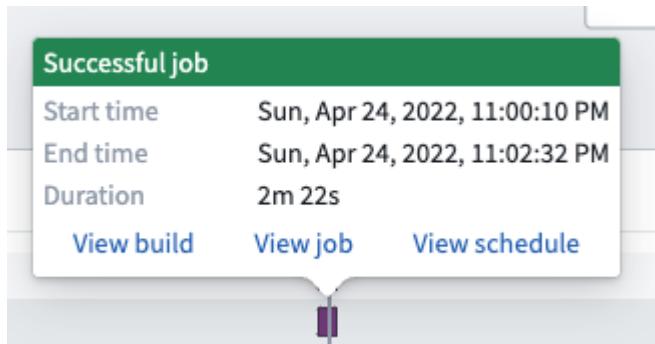
Use the **Build timeline** tool in Data Lineage to view the build history of your datasets.

In Data Lineage, click on **Build timeline** in the bottom left of your window. This action expands the view panel to display a Gantt chart of builds that took place during the period of time of your choice. You can select the number of days or hours you would like to view in your timeline, ranging from one hour to ten days. You can also choose to display the builds by color based on schedule or job status.



To view the build timeline of a specific dataset, select the dataset on your graph. Select multiple datasets with the **Drag select mode** tool or by holding Ctrl / Command while clicking.

To view details of a job in the build timeline, click on the job in the Gantt chart. You will see information about the job status, start and end time, and duration.



See more details about the build, job, and schedule by clicking on the links within the job information window.

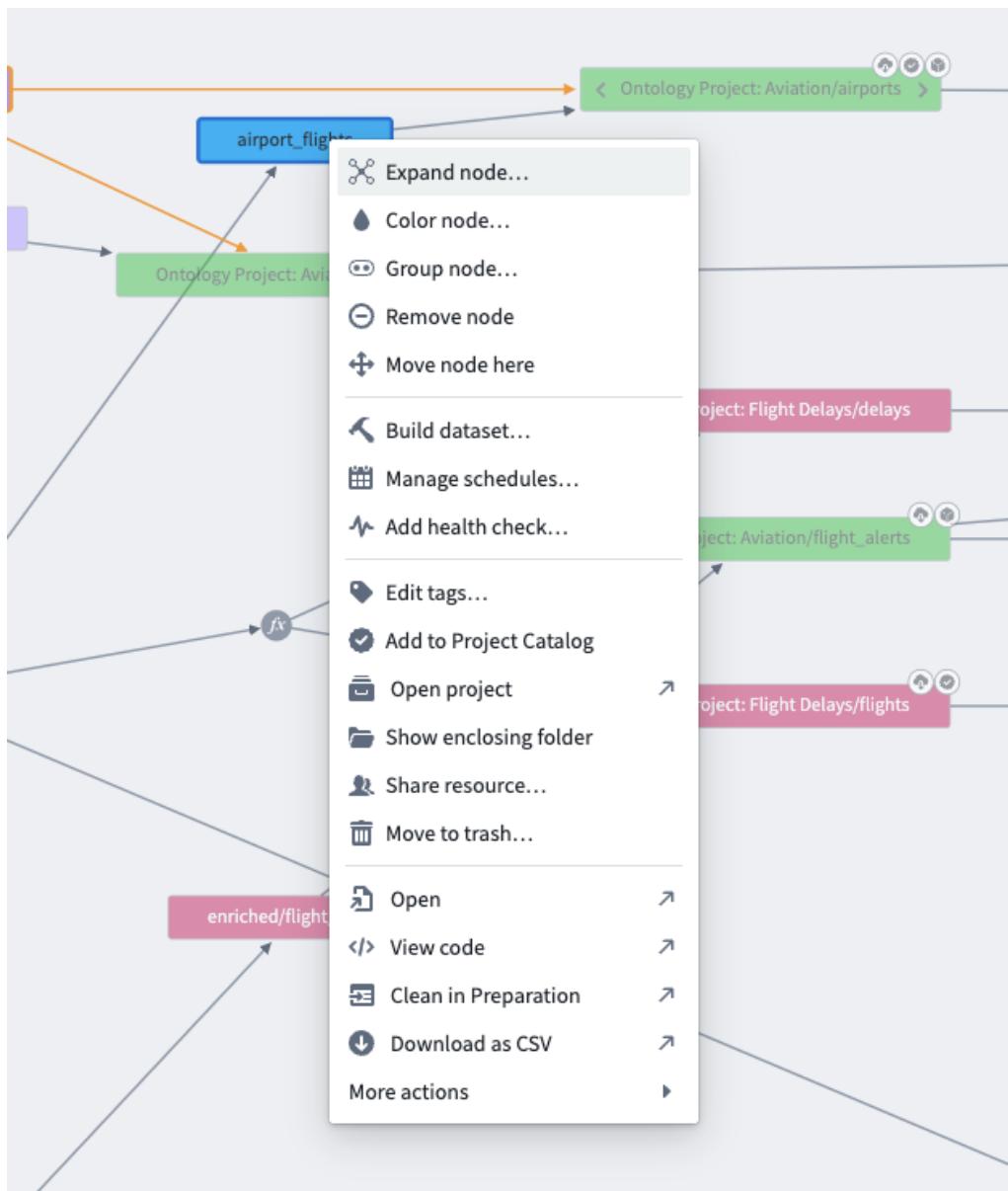
Understand out-of-date datasets

There are a few reasons why your dataset may not be up to date. Common scenarios to explore are:

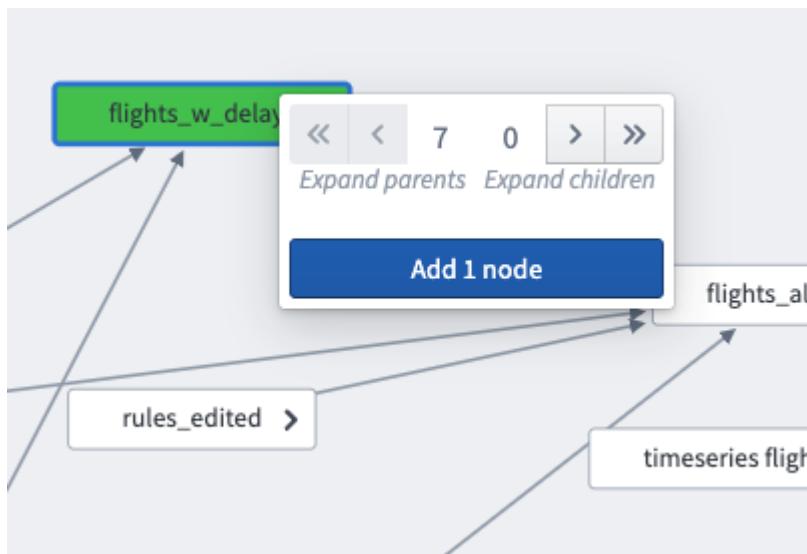
- Is my dataset build failing?
- Is there an upstream dataset that hasn't built and isn't up to date?
- Have we received up-to-date data from the source?

You can easily answer these questions by using Data Lineage.

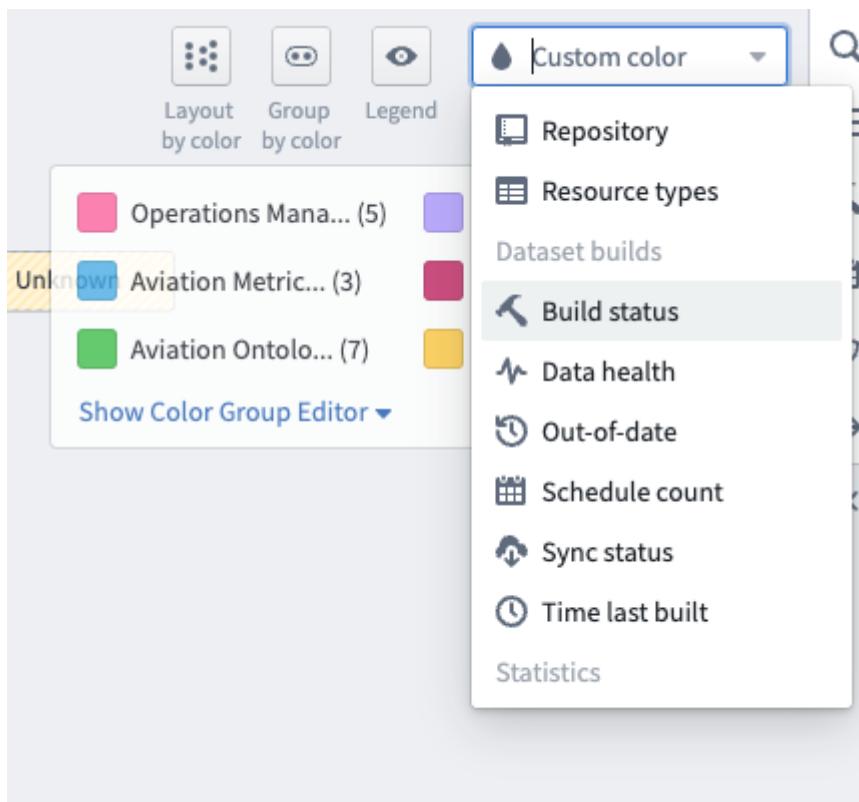
- First, verify the status of each of the resources in your pipeline by opening up the dataset of interest in Data Lineage and right-clicking on the node.



- Then, select **Expand node**. You can see all of the ancestor nodes for that dataset by clicking the double left arrow above **Expand parents**.



- Next, select the **Build status** option in the **Node color options** dropdown in the top right of Data Lineage to see the build status of every resource in your pipeline. This view of your pipeline will make it much easier to diagnose stale datasets.

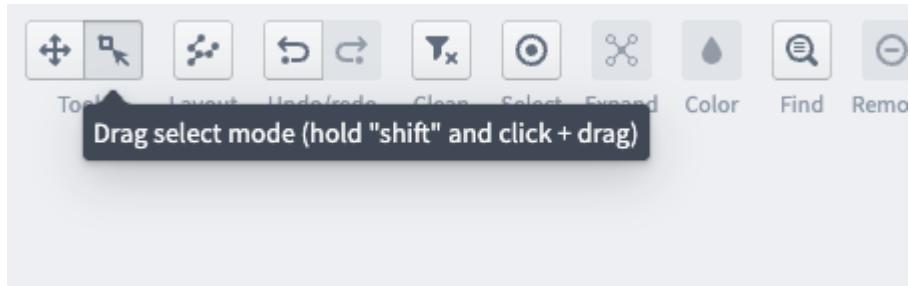


Find datasets with a given column

You can easily search for specific dataset columns within your Data Lineage graph:

- First, ensure you added all datasets of interest in your pipeline to your lineage graph.

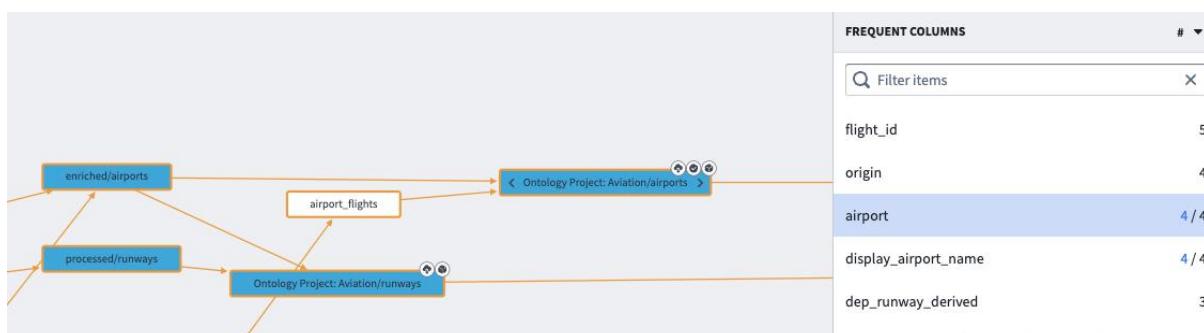
- Next, select all datasets of interest by using **Drag select mode** in the Tools toggle in the upper left hand corner of the app. You can also hold down Ctrl / Command to select multiple nodes at once, or use Ctrl / Command + A to select all nodes.



- Then, select **View histogram of selection properties** from the Data Lineage sidebar.

The screenshot shows the Data Lineage sidebar. On the left, there are buttons for 'Custom color' and 'View histogram of selection properties' (which is highlighted with a red box). Below these are groupings: 'Ontolo...' (10), 'Flight Delays' (5), 'Flight Control Mana...' (5), and a 'Group Editor' dropdown. On the right, the main panel displays '10 resources selected' with a 'Copy...' button. It shows three sections: 'CUSTOM GROUPS' (Flight Control 7, Aviation Ontology 1, Aviation Metrics 1, Flight Delays 1), 'BRANCH' (master 10), and 'CREATED BY' (no items listed).

- Under the **Frequent Columns** section, you can see the most frequent columns by name in your selection.
- Click one of the columns to highlight the datasets in your selection that contain this column.



Build datasets

You can use the Data Lineage graph to see which datasets in your pipeline are out of date, and then use the Builds helper to start builds directly from Data Lineage.

Builds triggered from Data Lineage always apply to the branches (including fallback branches) configured in the graph.

The following are a few common build workflows:

- Build all ancestors
- All transforms in between selected datasets
- Selected dataset(s) only

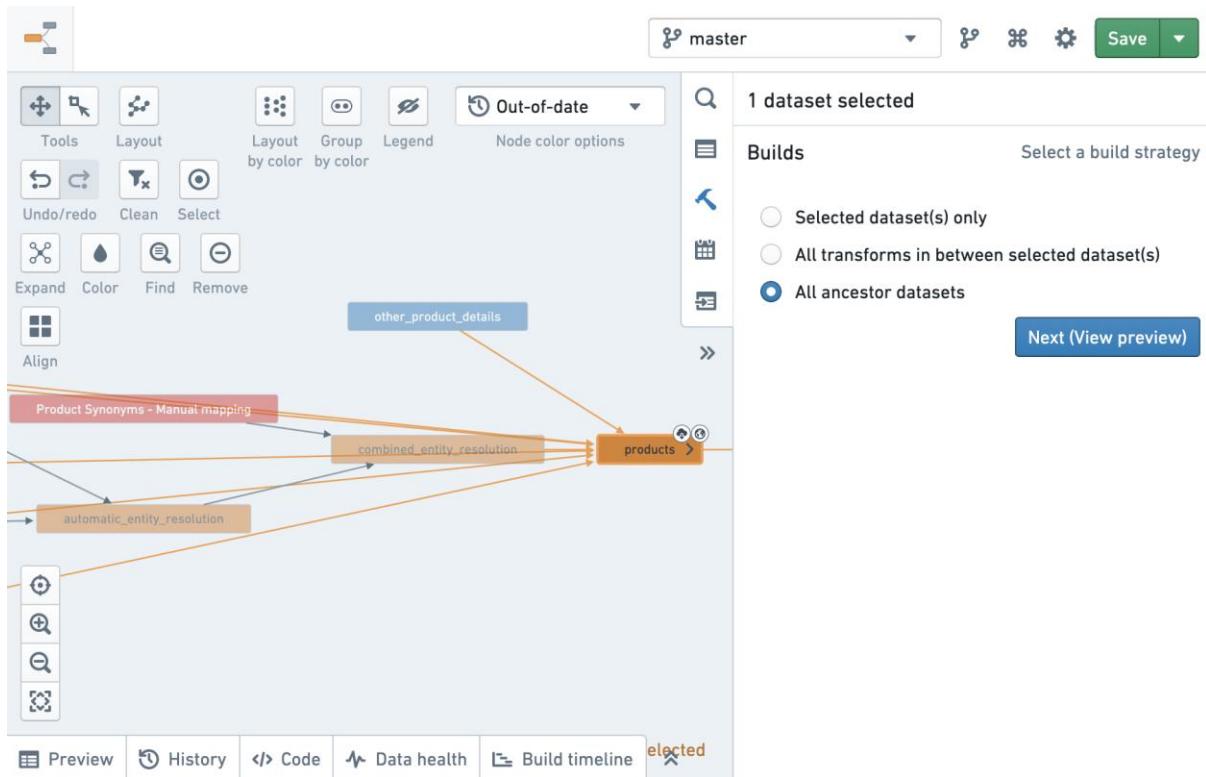
Build All Ancestors

This strategy builds the selected datasets and all ancestor datasets, to ensure that the selected datasets become completely up to date.

By default, this builds only ancestors that are out of date, but you can choose to force a re-build of up-to-date datasets. Forcing a re-build can be expensive in terms of build time and resources.

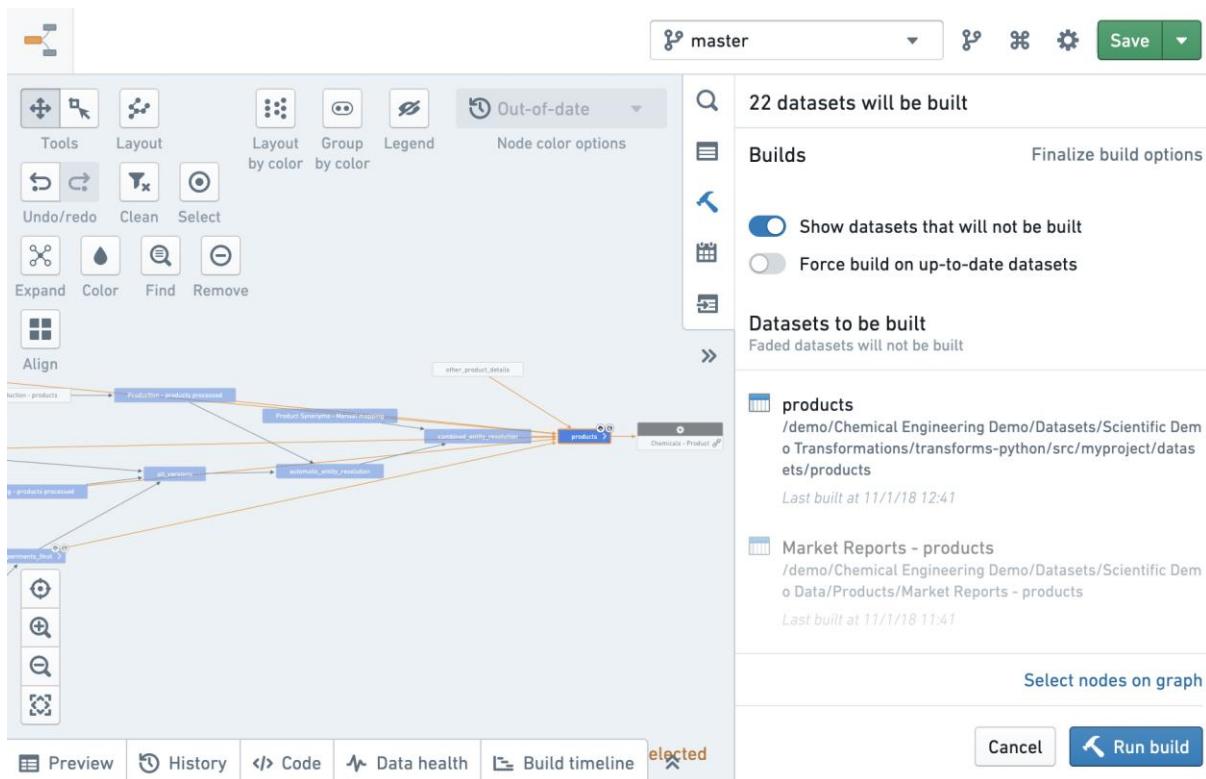
1. Add datasets to the graph or open a saved snapshot.
2. Select the dataset that you want to build.
3. In the Builds helper, choose **All ancestor datasets**, then click **Next**.

Clicking **Next** will *not* trigger any builds yet. You will simply see a preview of the datasets to be built.



4. If you want to force a re-build of up-to date datasets, click **Force build** on up-to-date datasets.
5. After examining the list of datasets to be built, click **Run build** to trigger the builds.

If you decide you do not want to build *all* out-of-date ancestors, you must click **Cancel** on the current build preview, then change the nodes you have selected. You cannot change your selection from the build preview screen.



All transforms in between selected datasets

This strategy lets you bind your builds to a subset of your pipeline. A common use case for this strategy can occur when new raw data regularly lands in your pipeline and there is a particular dataset that you want to update to reflect the new data, but you don't want to build *all* out-of-date ancestors. You can then use Data Lineage to determine which other datasets need to be built to bring your dataset of interest more up to date.

1. Add the dataset you ultimately want to build to the graph.
2. Add any raw datasets to the graph (or any upstream dataset)
3. Select all nodes.
4. In the Builds helper, choose the **All transforms in between selected dataset(s)** strategy, then click **Next**.

Clicking **Next** will *not* trigger any builds yet. You will simply see a preview of the datasets to be built based on the nodes you have selected. You can now see exactly what needs to be built to update your dataset of interest. You may not want to build *all* datasets – maybe there is a very large derived dataset that should only build once a day – so click **Add all to graph** at the bottom of the list.

Selected Datasets

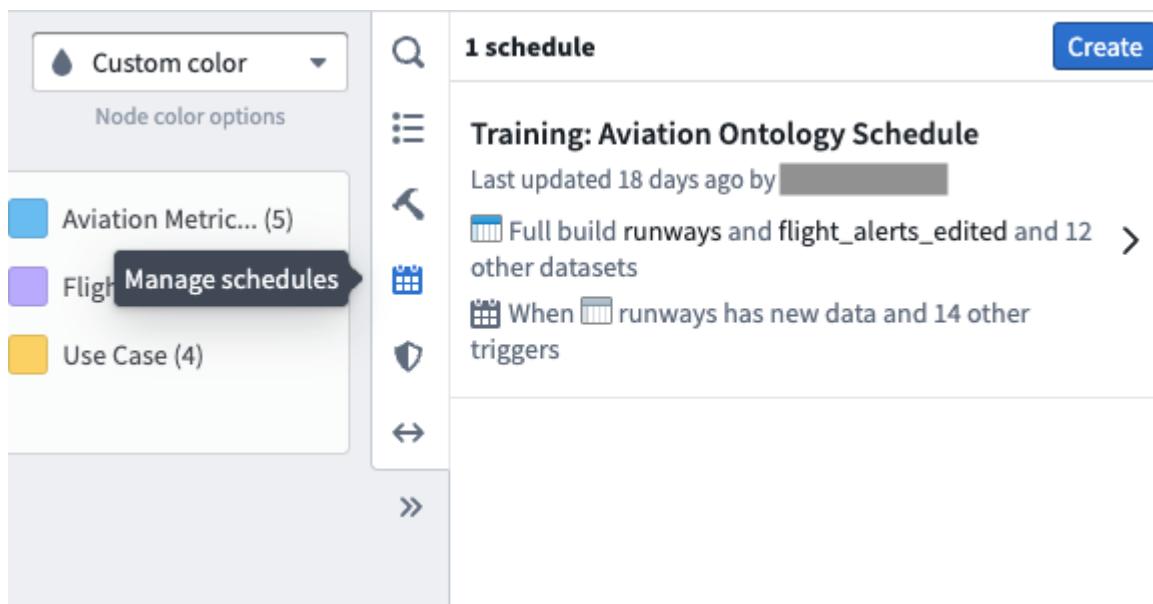
This strategy allows you to pick individual datasets that you want to build. If there are dependencies between the datasets, builds would be executed in the right order to assure descendants are built after their ancestors were built.

If you want to change the datasets you are building, you must click **Cancel** on the current build preview, change the nodes you have selected, then enter a new preview. You cannot change your build selection from the build preview screen.

After examining the final list of datasets to be built, click **Run build** to trigger the builds.

Manage schedules

Data Lineage allows you to easily manage build schedules within your lineage graph. In the right sidebar, select **Manage schedules** to open the schedule details pane.



You will see the schedules related to selected datasets in your graph. Click on a schedule to see more details:

The screenshot shows a detailed view of a build schedule named 'Training: Aviation Ontology Schedule'. The top navigation bar includes a search icon, a back button labeled 'Back to schedules list', and a refresh button. Below the header, the schedule name is displayed along with a note that the latest run was ignored and it was last updated 18 days ago by a user whose name is partially obscured. There are three buttons: 'Learn more', 'Metrics', and 'Share'. The main content area is divided into sections: 'Target datasets', 'When to build', and 'Build scope'.
Target datasets: Lists downstream datasets including 'runways', 'flight_alerts_edited', 'airlines', 'passenger_flight_alerts_edited', 'flights_edited', 'route_airline_mapping', and 'Plus 32 upstream datasets'.
When to build: Set to trigger 'When any of these datasets update' and lists specific datasets: 'route_alerts', 'aircraft', 'passenger_flight_alerts', 'timeseries_event_flight_delay', 'distance_group_lookup', 'us_state_mappings_v4', and 'Plus 32 upstream datasets'.
Build scope: Set to 'Foundry Training and Resources'.

- **Latest run:** The status of the latest run of the schedule.
- **Last update:** A timestamp of when the last update took place and the user who made changes
- **Target datasets:** A list of downstream datasets included in the build schedule.
- **When to build:** Displays the build schedule trigger determined when creating the build schedule. For example, a build schedule can be set to run **when specific datasets update**.
- **Build scope:** Defines the Project or user datasets included in the build and the permissions used to run the build.

Roll back a pipeline

Rolling back a pipeline is currently in the beta phase of development. Functionality may change during active development.

When building your pipeline, you may need to roll back a dataset and all of its downstream dependents to an earlier version. There can be many reasons for this, including the following:

- You identified a mistake in the logic required to build a dataset and need to revert it.
- Incorrect data was pushed into your pipeline from an upstream source.
- An outage occurred, and you want to quickly navigate back to an earlier state of your pipeline.

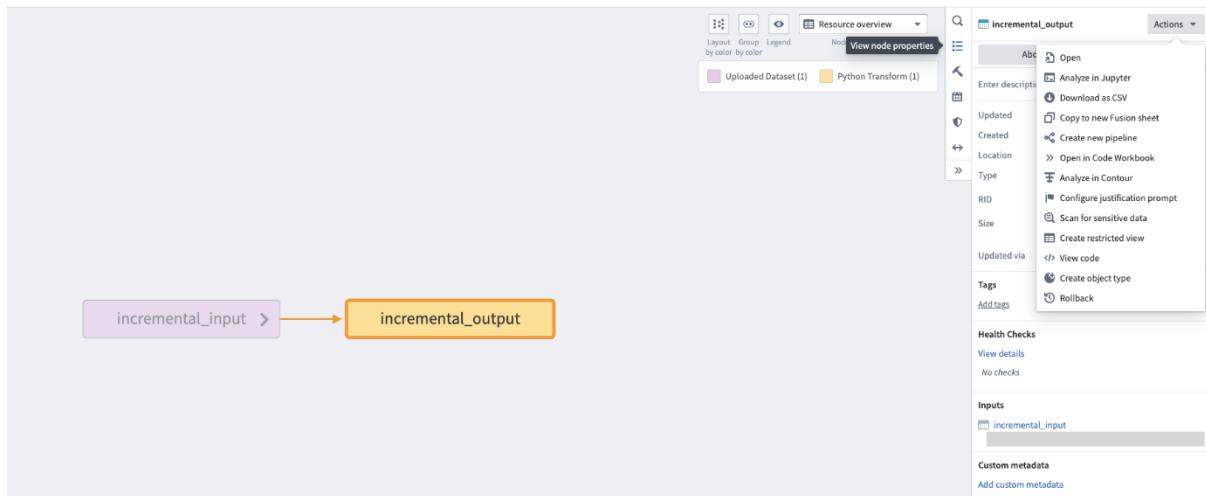
The pipeline rollback feature allows you to revert back to a transaction of an upstream dataset. When performing a rollback, the data provenance of the upstream dataset transaction is used to identify its downstream datasets and their corresponding transactions to create a final pipeline rollback state. Typically, this process would require several steps to properly roll back each affected dataset. With pipeline rollback, this is reduced to a few simple steps discussed below, along with the ability to preview the final pipeline state before confirming and proceeding with the rollback. Pipeline rollback also ensures that the incrementality of your pipeline is preserved.

As you set up your rollback, you can choose to exclude any downstream datasets; these datasets will remain unchanged as the pipeline is rolled back to the selected transaction.

If a dataset backs an object type stored using Object Storage V2, manual intervention is required to ensure that the object type is reindexed with a successful run of the replacement pipeline to reflect the state after the rollback.

Execute a pipeline rollback

1. Navigate to a Data Lineage graph containing the upstream dataset you would like to roll back.
2. Select the dataset in the graph. Then, from the branch selector at the top of the graph, select the branch on which you would like to perform the rollback.
3. Select **View node properties** in the panel on the right.



4. Select **Actions**, then **Rollback**.
5. Under **Selected transaction**, choose the transaction to which you would like to roll back.

Selected transaction

Selected transaction: Thu, Mar 27, 12:11 AM [Select](#)

Current Transaction: Thu, Mar 27, 12:11 AM

All transactions: [Wed, Feb 5, 11:32 PM](#)

After choosing the transaction, downstream datasets will automatically be found and the states they will revert to if the rollback is actioned will be displayed.

Resource types that are unable to be rolled back, including streaming datasets, media sets, and restricted views, will be displayed under the **unsupported resources** section. Transactional datasets on which you do not have Edit access will also be included in this list.

6. Select the timestamp under each dataset to navigate to the **History** page of the input, where the corresponding transaction will be highlighted.

← Exit rollback

⚠ Only transactional datasets are supported for rollbacks. Any other downstream resources, including media sets, streaming datasets, or virtual tables will remain unchanged. See documentation

Selected dataset to rollback

input

Selected transaction

Wed, Feb 5, 2025, 11:32 PM Select

Downstream Rollback ⓘ

✖ 6 unsupported resources ^

output_media_set_1

output_media_set_2

output_media_set

even_duplicate ✕
Wed, Feb 5, 11:33 PM —

even ✕
Wed, Feb 5, 11:34 PM —

adds_timestamp_duplicate ✕
Wed, Feb 5, 11:48 PM —

input ✕
Wed, Feb 5, 11:32 PM —

adds_timestamp ✕
Wed, Feb 5, 11:33 PM —

media_set_downstream ⚠
—

test_output ⚠
—

Datasets excluded from rollback ⓘ

Remove nodes from auto-rollback to keep current.

Rollback

7. Select any datasets to exclude from the rollback by selecting — to the right of the dataset name. Once excluded from rollback, the dataset will appear in the **Datasets excluded from rollback** section.

Downstream Rollback ⓘ

 even_duplicate	-
Wed, Feb 5, 11:33 PM	
 even	-
Wed, Feb 5, 11:34 PM	
 input	-
Wed, Feb 5, 11:32 PM	
 adds_timestamp_duplicate	-
Wed, Feb 5, 11:48 PM	
 adds_times	Exclude from rollback
Wed, Feb 5,	-

8. To add an excluded dataset back to the rollback, select + to the right of the dataset name.

Datasets excluded from rollback ⓘ

 adds_timestamp	Add to rollback	+
Wed, Feb 5, 11:33 PM		

9. After finalizing the state of your desired rollback, select **Rollback**. A confirmation dialog will appear.

Confirm rollback

X

A You may not have permissions to discover all downstream outputs of a dataset. Rolling back may not update the outputs to which you do not have access, resulting in failing builds, unexpected transactions (such as snapshot instead of incremental), or other consequences.

This action will rollback transactions on **5 datasets** and reset incremental state on **2 datasets**.

To confirm rollback, enter the branch name: **master**

Cancel

Confirm rollback

10. Enter the branch name as confirmation, then select **Confirm rollback** to proceed.

[← Exit rollback](#)

Rolled back 7 datasets

	even_duplicate Wed, Feb 5, 11:33 PM	✓
	even Wed, Feb 5, 11:34 PM	✓
	adds_timestamp_duplicate Wed, Feb 5, 11:48 PM	✓
	input Wed, Feb 5, 11:32 PM	✓
	adds_timestamp Wed, Feb 5, 11:33 PM	✓
	media_set_downstream	✓
	test_output	✓

Rollback successful

[Exit](#)

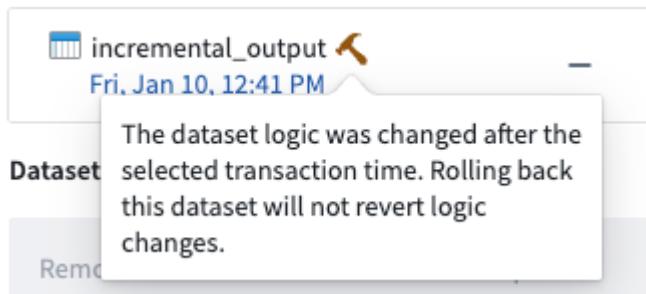
- Once the rollback is complete, navigate to the **History** tab of the datasets and notice that the rolled back transaction is now crossed out, as shown below:



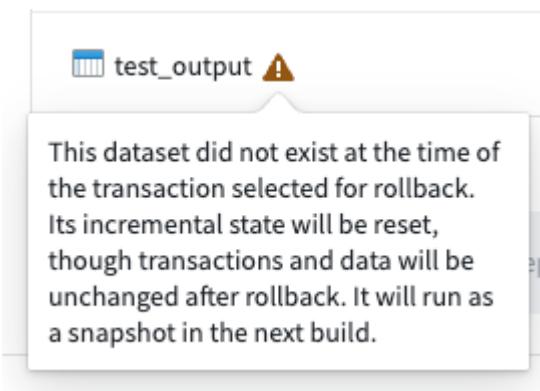
Warnings

Depending on how your pipeline changed between the current state and the rollback transaction, you may receive one of the warnings shown below on datasets selected for downstream rollback:

- If the rollback transaction was built using a different JobSpec (logic) than the latest transaction, the transaction will show the following warning: The dataset logic was changed after the selected transaction time. Rolling back this dataset will not revert logic changes. If you receive this warning, you may need to inspect how the logic changed over time and determine if you can apply the logic again after the rollback and before any new dataset builds.



- If a downstream dataset did not exist at the time of the selected rollback transaction, the dataset will show the following warning: This dataset did not exist at the time of the transaction selected for rollback. Its incremental state will be reset, though transactions and data will be unchanged after rollback. It will run as a snapshot in the next build. If you receive this warning, note that the rollback will reset the dataset's job history as if it was never built. The next build on the dataset will not be run incrementally (if the dataset is an incremental dataset) since it will be treated as a completely new dataset being built for the first time.



Considerations and limitations

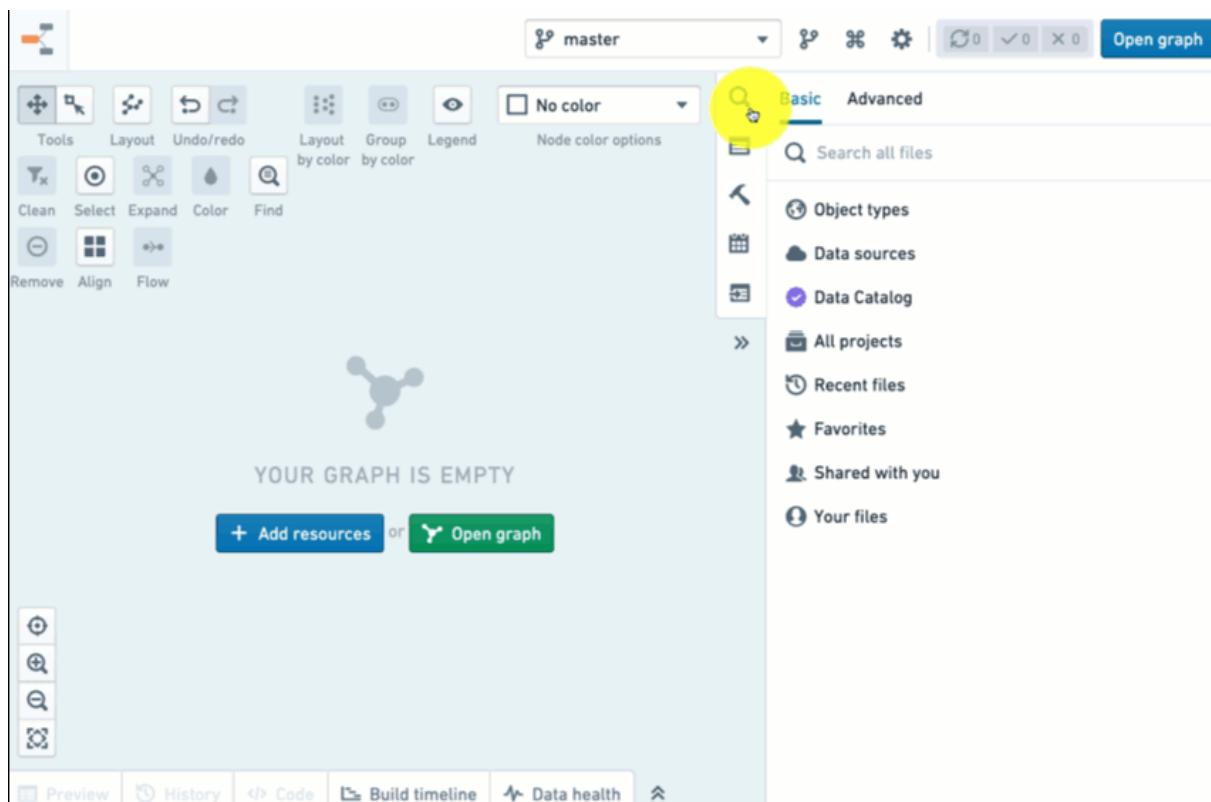
When rolling back a pipeline, keep the following considerations in mind:

- Only transactional datasets are supported for rollbacks. Any other downstream resources, including media sets, streaming datasets, or virtual tables will remain unchanged.
 - These resources will be displayed under the **unsupported resources** section of the rollback preview panel.

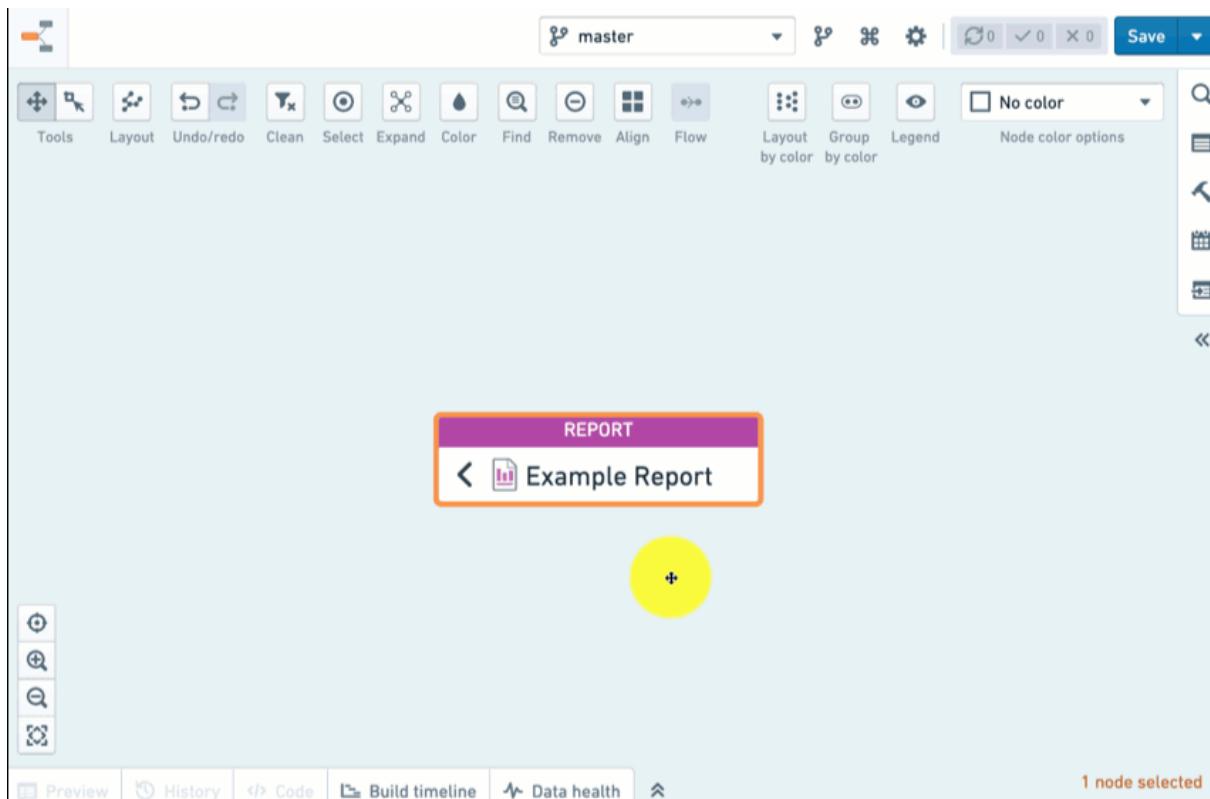
- You are only able to roll back to a *successful* transaction.
- It is not possible to roll back to a retained transaction
 - However, you can roll back to a DELETE retention transaction.
- You can only roll back datasets to which you have access. If you do not have access to certain downstream datasets, they **will not** be rolled back.
 - It is possible to have Edit access on an upstream dataset that is then used to build a downstream dataset to which you only have View access. In this case, the downstream dataset will be considered an "unsupported resource".
- Only transactional datasets with a JobSpec can be rolled back; uploaded datasets are not supported for rollback.
- A maximum of **50 transactions** can be rolled back on the upstream dataset at a given time.

Check resource permissions

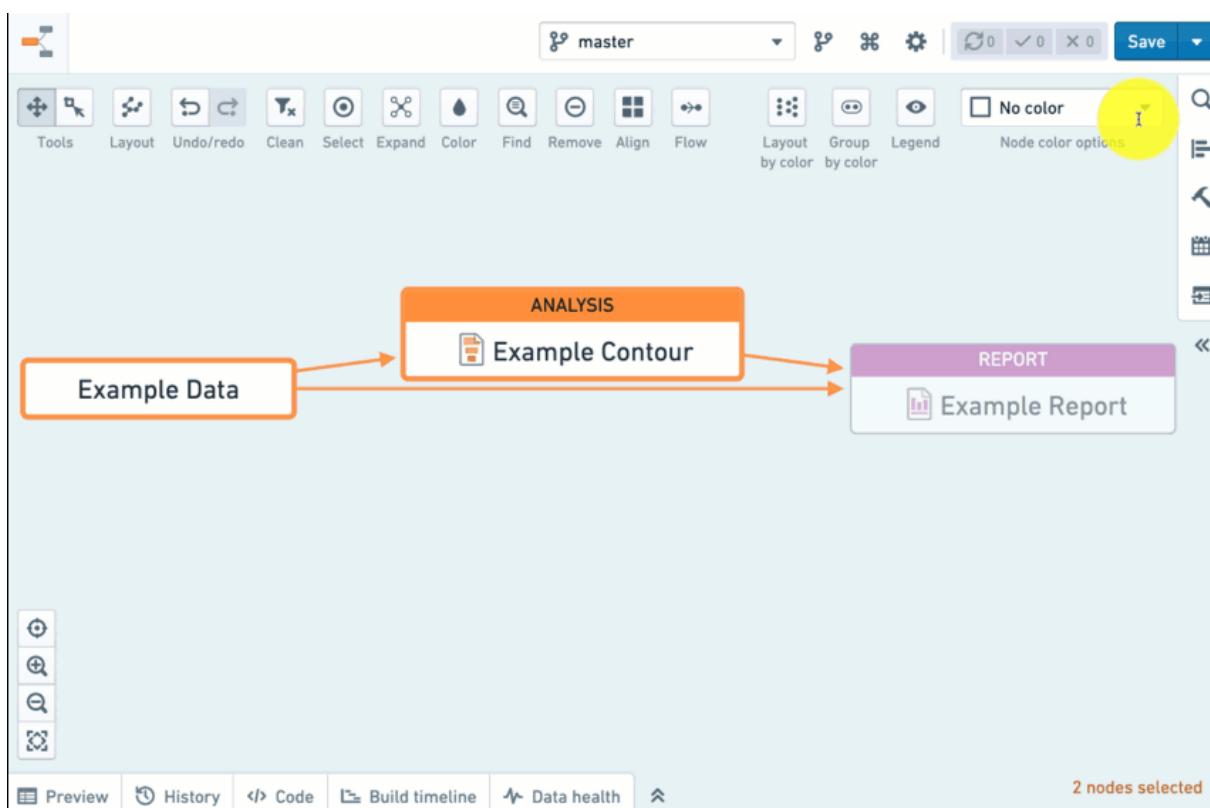
You can use Data Lineage to check users' permissions to view datasets or artifacts using the "Permissions" coloring option. To do that, start by adding nodes to the graph. You can do so using the search helper on the side panel.



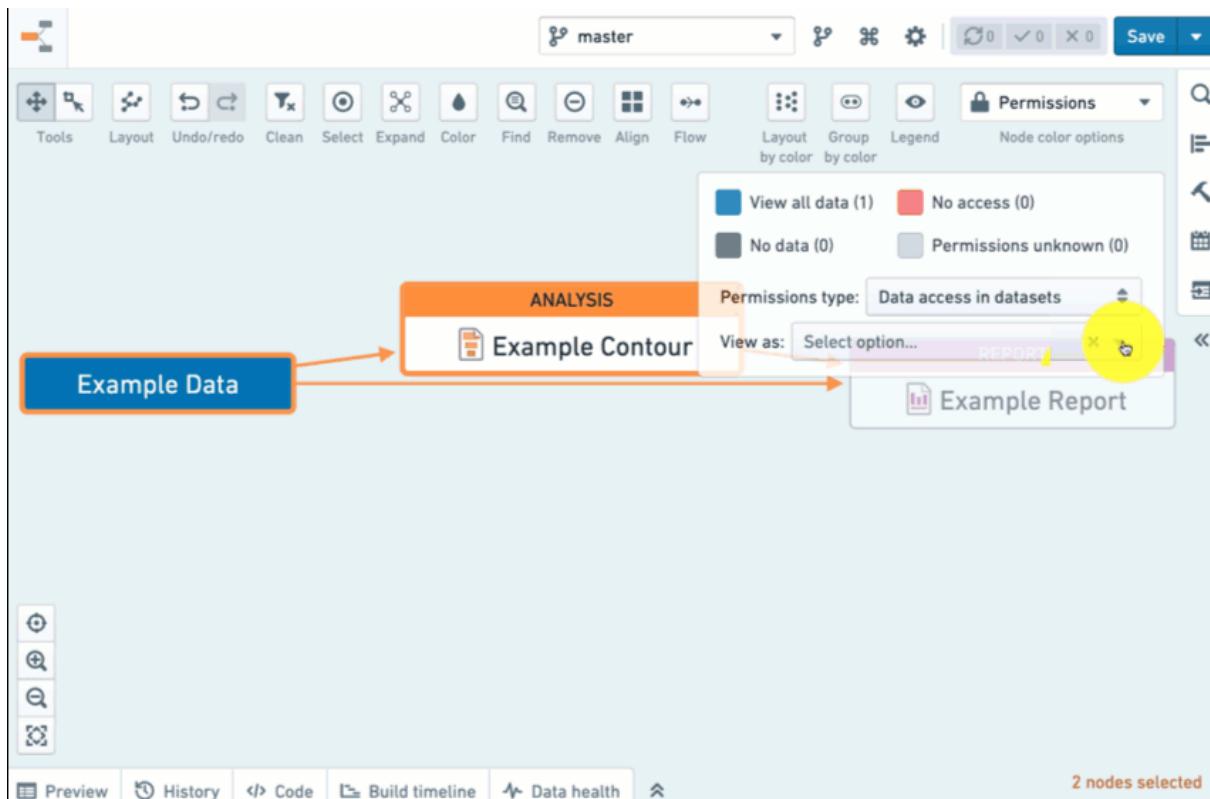
Then expand the graph to view the lineage leading to your resource (read more about exploring lineage).



Once you have done this, use the **Node color options** dropdown to select the **Permissions** color scheme.

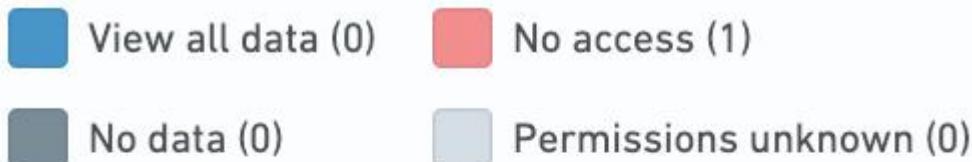


Select the user's name from the **View as** dropdown. This will allow you to see the user's permissions to each of the nodes on the graph.



There are two permission types you can color by:

- Data access in datasets
- Resource access



You are only able to view a user's access level if
you have that access level



Data access in datasets

Use this option to troubleshoot permissions issues. Remember that a user's data access is affected by data lineage. By coloring your nodes based on the user's access to data, you can easily see what the upstream datasets are that may restrict the user's access to data.

Note that this option only works on dataset nodes.

Resource access

This will allow you to see the role (such as Editor, Viewer, etc.) that is set for the selected user on the selected resource.

Use this option to view the level of access users have to your artifacts.

Roles do not correspond to data lineage the same way that data access does. For example, user being an "Editor" on a Contour Analysis does not guarantee they have permissions to see the data that the analysis depends on. Make sure your users can access the underlying data when sharing a resource with them.

See the impact of Marking changes

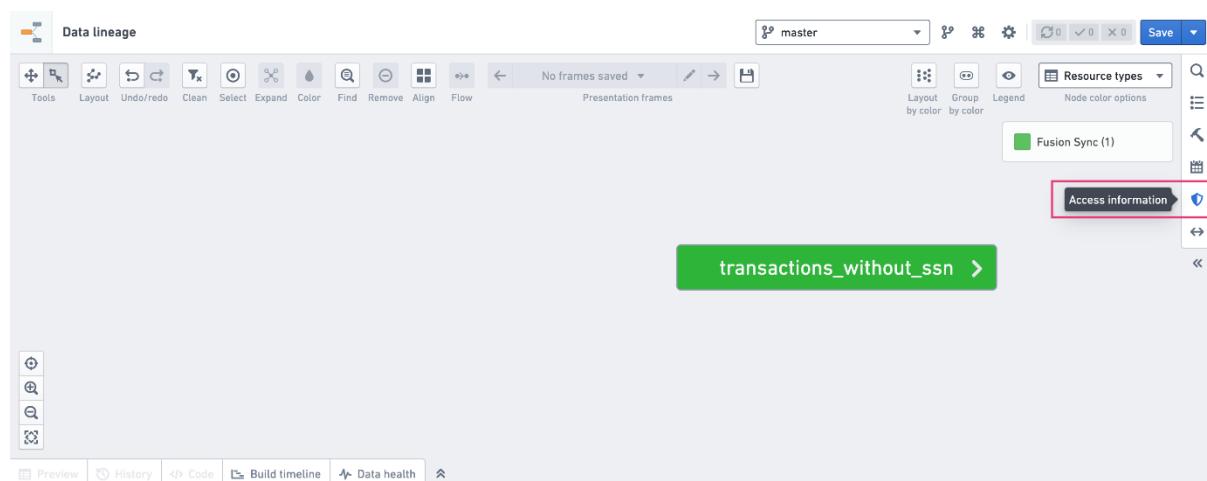
You can use Data Lineage to evaluate how changes to dataset Markings can impact derived datasets. This can be useful when removing Markings.

Warning

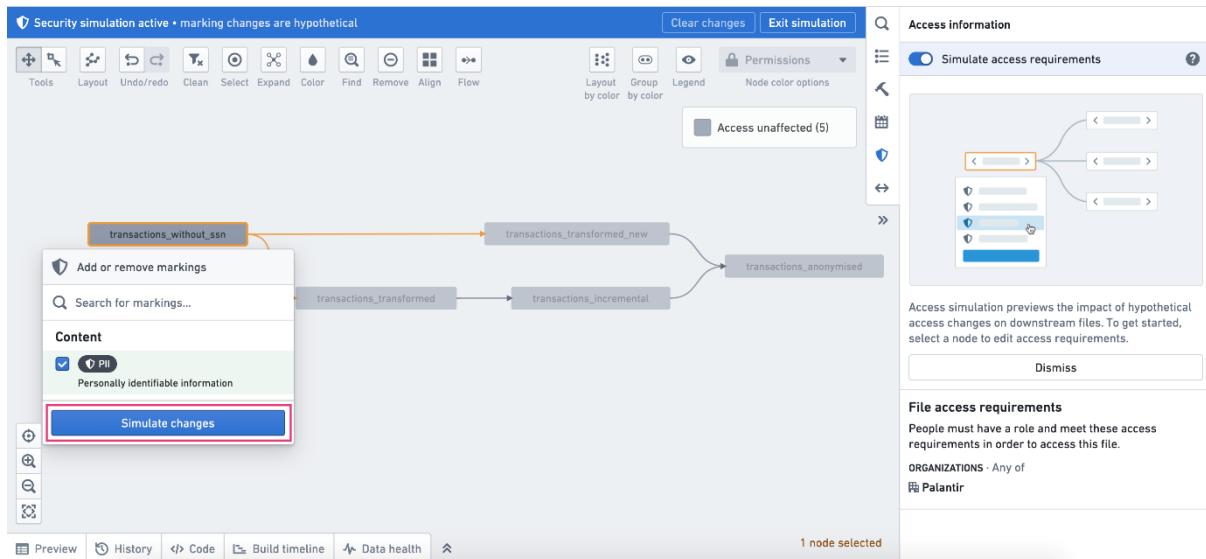
Marking simulation relies on the most recent dataset builds and does not account for changes that are not yet finalized. Confirm that you are working with the most up-to-date version of your data.

Access simulation mode

1. Open the **Access information** side panel.
2. Toggle on **Simulate access requirements**.
3. Select any dataset on the graph.
4. Click **Edit markings**.



Simulate Marking changes

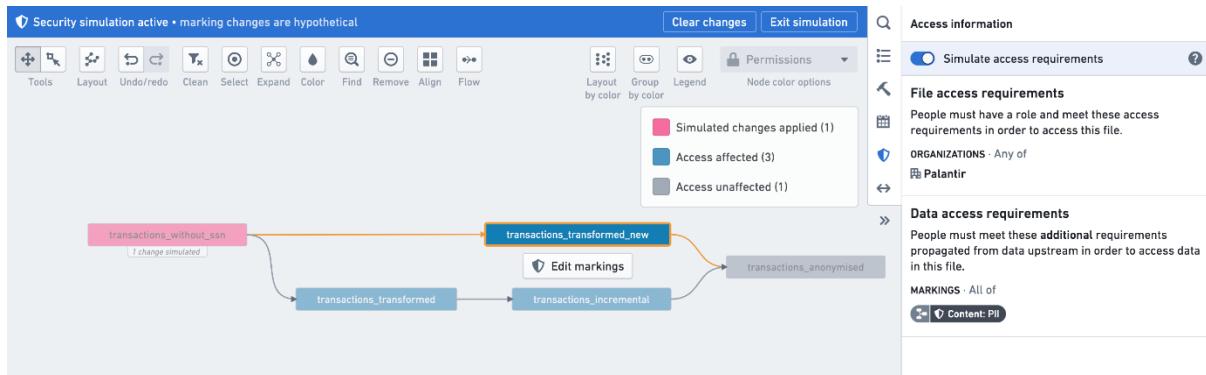


To simulate Marking application, search for the Marking you want to apply, check the box next to the Marking, and then select the **Simulate changes** button.

Markings that are already applied on a dataset will appear as selected. To simulate Marking removal, uncheck the box next to the Marking and click **Simulate changes**.

You can only remove Markings that were applied directly on the dataset. Removal of Markings that were inherited through a dataset's lineage or from the parent Project cannot be simulated.

Analyze the simulated graph



When in simulation mode, the graph coloring will indicate the datasets affected by the Marking changes. The graph colors are labeled in the interface and can represent the following dataset statuses:

- **Simulate changes applied** appears on the datasets to which you applied changes.
- **Access affected** appears on datasets for which the Markings before and after the change will be different.
- **Access unaffected** appears on datasets for which the Markings before and after the change will remain the same.

- **No visible transactions** appears on datasets that have not been built yet, or where you do not have permission to see transactions.

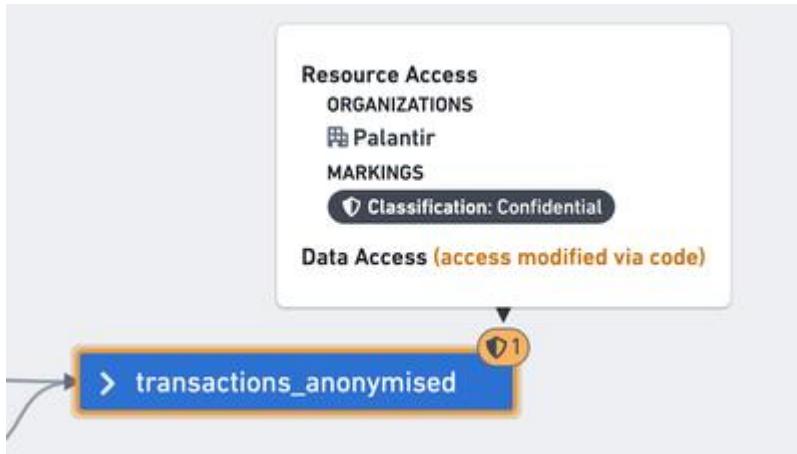
By selecting any of the datasets, the **Access information** side panel will show the simulated access requirements. You can toggle simulation mode on and off to view differences without losing any of the simulation changes.

Tips for understanding changes

Before making changes, we suggest consulting the Markings documentation to learn more about the impact of Markings on users.

When simulating Markings, consider the following:

- Datasets can stop propagating Markings *via code*.



- In the **Permissions** coloring, nodes on the Data Lineage graph that stop propagating Markings show that data access was *modified via code*. This message will also appear in the **Access information** section of the node properties side panel.
- In the Code Helper, you can check the code for a dataset to see if it stops propagating Markings by using the term `stop_propagating`.
- Datasets can have Markings propagated to them from *other inputs*; expand the dataset inputs by clicking on the left arrow in the dataset node.
- Markings can be applied on the *parent Project or folder*; Markings will have a folder icon on their left when simulation mode is not enabled, and will show a folder icon in the Marking simulation menu when simulation mode is enabled.