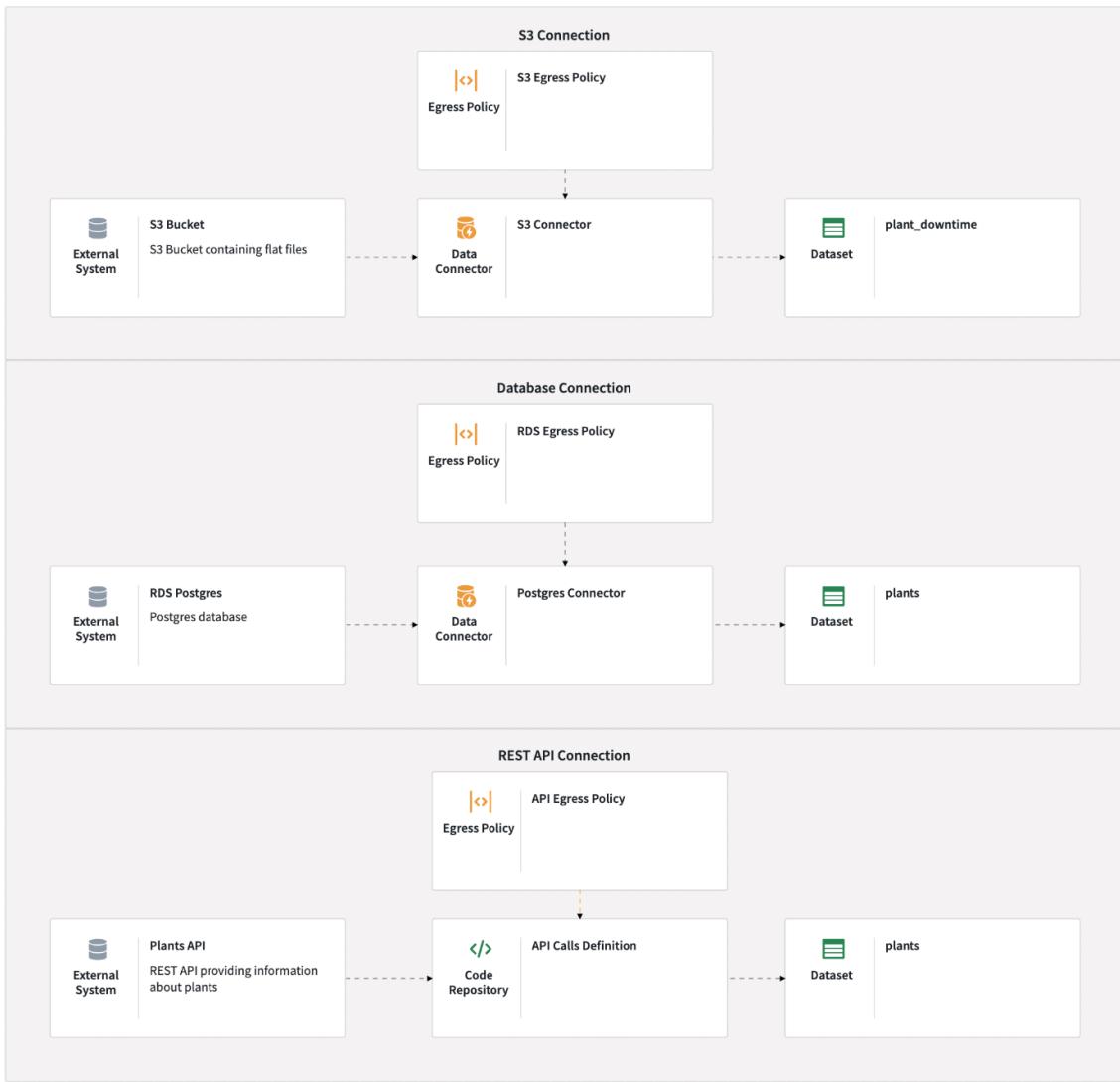


Creating Your First Data Connection

Introduction

You will learn how to efficiently set up data connections in Foundry by connecting to **real** data sources. **You do not need to provide your own datasources. You can connect to Palantir provided datasources for this training.** You will be guided through the three most common types of data sources.



Connecting to data sources in Foundry is often the first step to leveraging value from the platform. Foundry's data connection tools support an extensive array of standard enterprise data sources, including cloud-based object stores, file systems, databases, and data warehouses. This versatility enables organizations to integrate a variety of data types—structured, unstructured, or semi-structured—using different data transfer methods such as batch, micro-batch, or streaming.

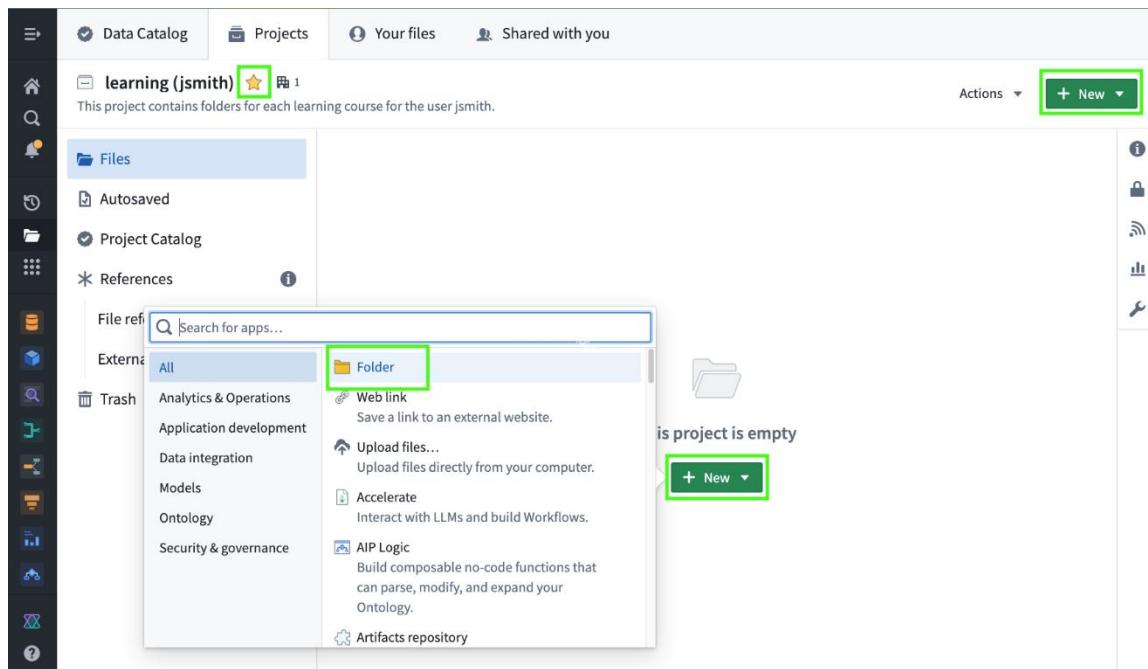
By connecting to these data sources, users gain the ability to access and manage data pipelines via an intuitive frontend interface, simplifying complex technical tasks. Additionally, Foundry's open architecture promotes interoperability with third-party software and open-source tools, enhancing the ease of working with data across different formats and systems.

Setting up your Folder

Create a Course-Specific Training Folder

Step 1: Create a folder

1. In the top left, click on the star to favorite your project. This will allow you to find it quickly later on.
2. Click on **New**
3. Click on **Folder**



Step 2: Name your folder

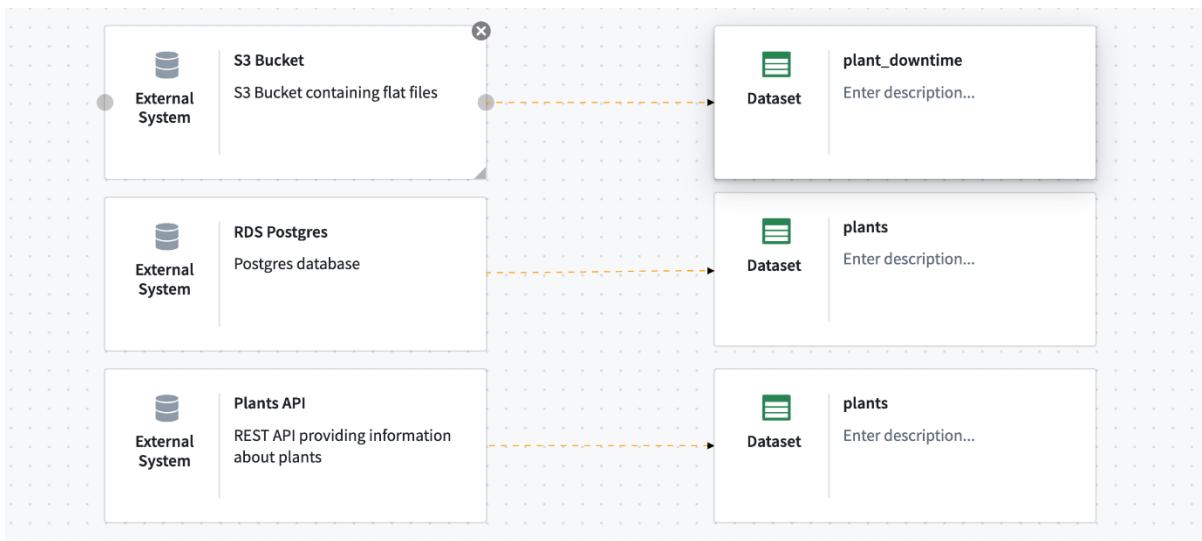
1. Name the new folder after the current learning course. For example: Creating Your First Data Connection
-

Data Connection: Getting Data Into Foundry

Basic Concepts

Foundry Data Connection is an application designed to synchronize data from external systems, for use and integration within the Foundry platform. It streamlines the management of data connections from external systems to Foundry, by abstracting the connection complexities into a straightforward frontend interface. Data Connection is compatible with a broad array of standard enterprise data sources, including cloud-based object stores, file systems, databases, and data warehouses.

The diagram below showcases possible data flows from various external sources, such as S3 buckets or RDS PostgreSQL databases, into Foundry datasets.

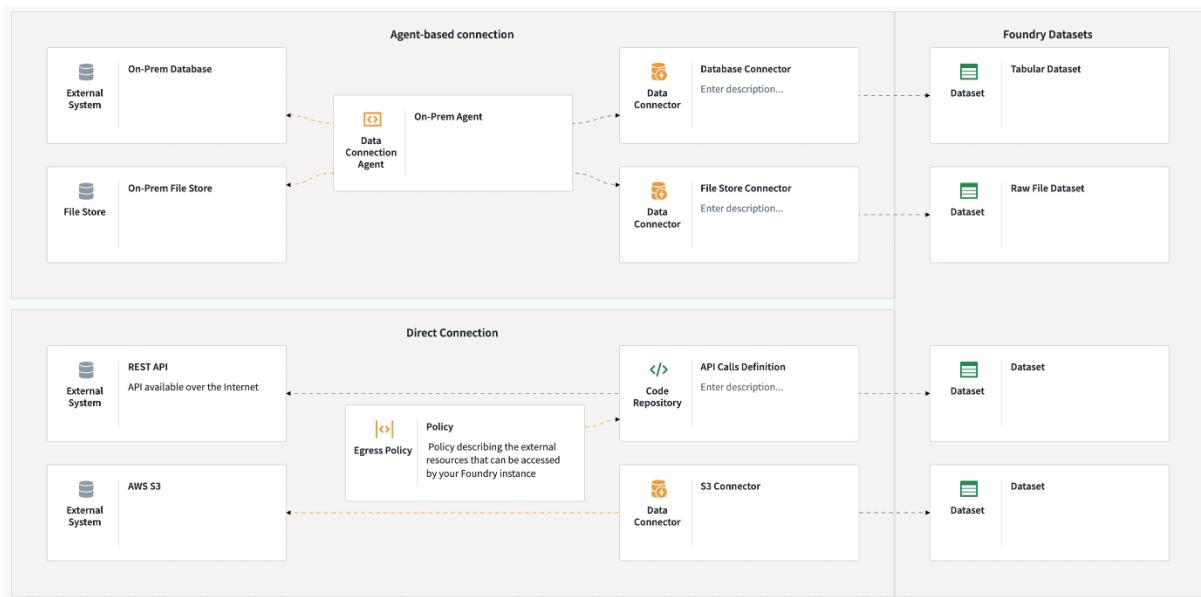


The Data Connection tool emphasizes robustness, extensibility, and user-friendliness, thereby enabling a wider range of users to engage in data connection tasks and facilitating seamless integration with diverse data systems.

Consistent with the rest of the Foundry platform, Data Connection enforces robust security measures to ensure that data access is appropriately controlled, while still allowing users to work efficiently within the platform.

Agents vs Direct Connection

Foundry Data Connection has two methods used to establish connections with data sources: Agents and Direct Connection.



Agents are downloadable programs installed within your organizational network and managed from Foundry's Data Connection interface. They are primarily used to read data from sources within your organizational network and securely ingest it into Foundry Agents which are required to access sources running on private networks or on-premises systems. They communicate with the Coordinator via encrypted, unidirectional outbound requests from the customer network into the Foundry platform.

Direct Connection is used to connect to data sources accessible over the Internet, such as REST APIs, SFTP servers, or cloud storage accounts. Direct Connection offers several advantages, including no need to provision, configure, and manage an Agent, avoiding routing Internet-to-Foundry through your network, and providing excellent uptime and performance as cloud-based Syncs do not depend on an Agent software package or its host.

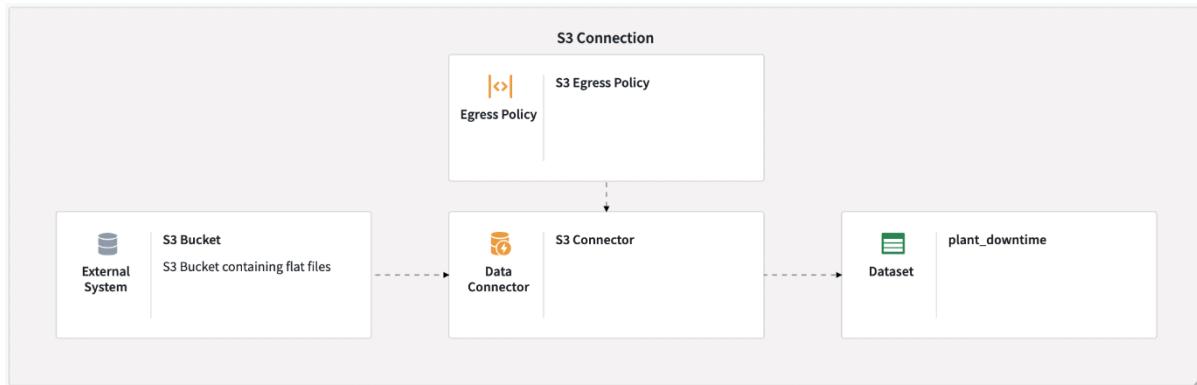
In summary, Agents are used for connecting to private networks and on-premises data sources, while Direct Connection is preferred for connecting to data sources over the public network.

For the purpose of this exercise we will be connecting to data sources using the **Direct Connection** method.

Data Connection: S3

Introduction

In this lesson, you will learn how to establish a connection from Foundry to an AWS S3 bucket. Your objective is to retrieve data for use within the Foundry platform. The architecture of this connection is illustrated in the diagram below:



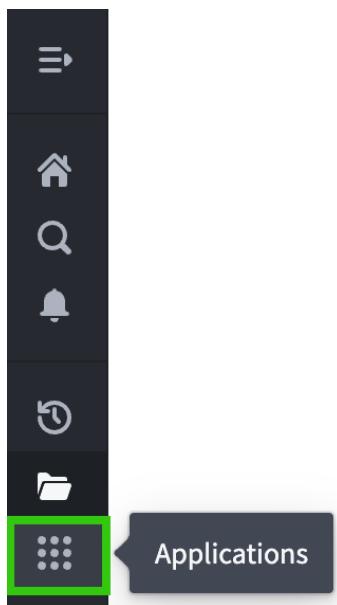
Amazon Simple Storage Service (AWS S3) is a scalable, durable, secure, and cost-effective object storage service offered by Amazon Web Services (AWS). It enables users to store and retrieve virtually unlimited amounts of data, from small files to extensive datasets, at any time and from any location. Due to these advantages, AWS S3 is a popular choice among many enterprises.

Please note that while this training focuses on AWS S3, the methodology for connecting to other cloud object storage solutions—such as Google Cloud Storage (GCS), Microsoft Azure Blob Storage, or other S3-compatible services—is quite similar. This means the skills you acquire here will be broadly applicable across different cloud storage platforms.

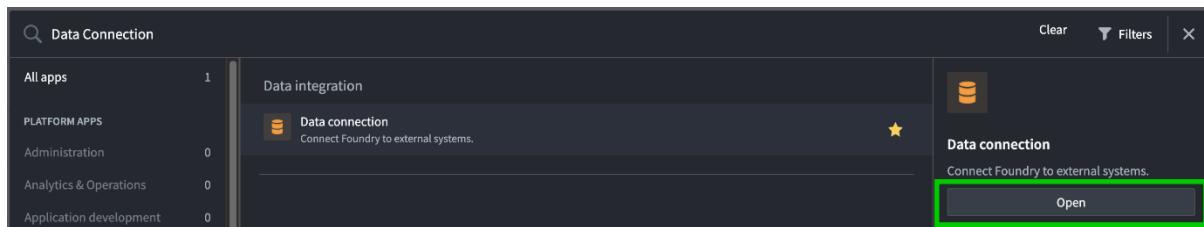
Open the Data Connection Application

Step 1: Open Data Connection application

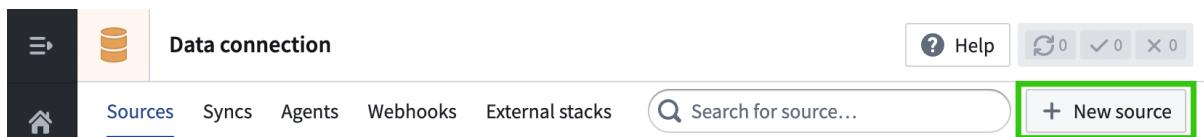
1. Click on the **Applications** button in the Foundry menu.



2. Search for Data Connection and click **Open**.



3. Inside the Data Connection application, click **New source**.



Select Source Type

Step 1: Set-up an S3 source

1. Choose the S3 connector type. This will initiate the new source creation workflow.

Select source type

Sources

To connect to data from the Internet or an on-premises source, select from the listed source types.

The screenshot shows a list of source types. The 'S3' option is highlighted with a green border. Below it, there are four sub-options: 'Batch syncs', 'Media syncs', 'Virtual tables', and 'Batch exports'. The 'Batch syncs' option is also highlighted with a green border.

2. Click **Continue** in the bottom right corner to proceed to the next step in the process.

The screenshot shows the 'New Source' setup wizard in progress. The sidebar on the left lists steps: 1. Overview (selected), 2. Connection type, 3. Name and location, 4. Connection details, 5. Export configuration, and 6. Summary. The main area is titled 'Overview' and shows a diagram of a cloud icon connected to a database icon with a plus sign, labeled 'Creating a New Source'. A note states: 'A Source is any data system that you connect to Foundry. Source systems may support a variety of capabilities including imports, streaming, and webhooks.' Below this is a section titled 'What we're going to do' with a diagram showing the flow from 'SOURCE' through 'AGENT / EGRESS POLICY' to 'SYNC' and finally to 'FOUNDRY'. A note below explains: 'In this wizard, you will set up your Source via the following steps:' followed by a numbered list: 1. Set up your source metadata, 2. Connect to Source using your credentials, 3. Ensure you can access the Source using either an Agent or an Egress policy, 4. Confirm your Source is connected and ready to use. At the bottom, a note says: 'Note: connecting a Source does not automatically download your data to Foundry. Data must be imported before it is used. A Sync is how you import data to Foundry, and Webhooks allow you to make user-click-driven outbound requests from applications built in Foundry. After you are done configuring your Source in this setup process, you have the option to continue by configuring a Sync or setting up a Webhook.' At the very bottom, there are buttons for 'Need help?', 'Open', 'Ask AIP Assist', and a large blue 'Continue →' button.

Select Connection Type

Step 1: Set-up a Direct Connection

1. Choose **Direct connection**, since the S3 resources used in this tutorial are accessible over the public Internet.
2. Click **Continue** in the bottom right corner to move forward.

The screenshot shows the 'Data connection > New Source' interface. On the left, a sidebar lists steps: 1 Overview, 2 Connection type (selected), 3 Name and location, 4 Connection details, 5 Export configuration, and 6 Summary. The main area is titled 'Connect to your data source'. It compares two options: 'Direct connection' (selected) and 'Through an Agent'. The 'Direct connection' section includes icons for a cloud, a person, and a gear, and text explaining it connects directly over the internet. It also lists requirements: an allowlisted source, an egress policy, and an agent connector. A 'Selected' button is at the bottom. The 'Through an Agent' section includes icons for a cloud, a person, and a gear, and text explaining it connects through an agent. It also lists requirements: an agent installed on a host, current agents, and creation of new agents if needed. A 'Select' button is at the bottom. At the bottom of the dialog are 'Need help?' and 'Ask AIP Assist' buttons, and 'Back' and 'Continue →' buttons. The 'Continue →' button is highlighted with a green border.

Create Source

Step 1: Create source and save to your Project

1. Assign a name to your source, for example, Data Connection Training - S3 (your_name).

The screenshot shows the 'Name and Project' dialog. At the top, the title 'Name and Project' is displayed. Below it is a 'Source name' input field containing the text 'Data Connection Training - S3 (your_name)', which is highlighted with a green border. Below the input field is a descriptive text: 'Provide a unique, distinguishable name, such as the server name or the database name.' At the bottom of the dialog are 'Back' and 'Continue →' buttons. The 'Continue →' button is highlighted with a green border.

2. Choose your working project as the location to save your new Source. If you have created a new folder within your Learning Project specifically for this training, use that folder as your location.

Project to save Source in

Choose a location

Choose a folder in an existing Project or Create a new Project

Anyone with Editor or Owner roles on your selected Project will be able to read or write to the data source.

[How to choose a location for your Source](#) Expand >

3. Click **Create source and continue** in the bottom right corner to finalize the creation process and proceed.

Create source and continue →

Input Connection Details

Step 1: Update the S3 connection settings

1. Enter the following S3 connection details:

- o URL: **s3://s3-bucket.sandbox.training.palantir.com/**
- o Region: **eu-west-1**

Source Setup

Connection settings

URL
`s3://s3-bucket.sandbox.training.palantir.com/`

URL of the S3 bucket. Data connection supports the s3a protocol. Should contain a trailing slash.

S3 endpoint
`s3.amazonaws.com`

The endpoint to use to access S3.

S3 endpoint signing region

The region used when constructing the S3 client using a custom endpoint. This is often not required and would only be needed if you are using the S3 connector with an S3-compliant third-party API, and are also setting a custom endpoint that requires a non-default region.

Region
`eu-west-1`

The AWS region to use when configuring AWS services. This is required when using STS roles. Warning: Providing region together with an s3Endpoint also containing the region can cause failures.

2. Select **Access key and secret** as method of providing credentials. Use the following values:

- o Access key ID: **AKIAWNSQVQU4RPCWEESQ**
- o Secret access key: **tWpDRP2QcCUIV1GLkdiRVP9qlE3o4jjKSTXR5m98**

Credentials

None Access key and secret Cloud identity

Access key ID (required)

.....

Encrypted

Secret access key (required)

.....

Encrypted

Set-up an Egress Policy

For security purposes, you need to create or import a network egress policy. This policy whitelists the data source, ensuring that your Foundry instance can access it securely. This is a one-time operation—if a previous user has already completed this training and created the policy, you can simply import it.

Foundry will suggest the recommended next steps based on the URL you've inputted in the previous step.

Step 1a: Select an existing Egress Policy

1. If the policy already exists, import it by clicking **Import existing policy**.

Network Connectivity Switch to connect via Agent



Select a policy

A network egress policy is required to connect to sources outside Foundry's secure network. Select an existing policy if one already exists, otherwise create a new policy. [Learn more](#)

+ [Use existing policy](#) ✚ [Create new policy](#)

★ **Suggested network egress policies**
These suggested egress policies are required for Foundry to connect to your source.

⚠ In addition to specifying network egress policies, S3 buckets in the same AWS region where your Foundry instance is hosted must also be added to the S3 bucket policies allow-list on the network egress page in Control Panel.

 s3-bucket.sandbox.training.palantir.com.s3.eu-west-1.amazonaws.com (Port 443) S3 Training	Import existing policy
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------

Step 1b: Create a new Egress Policy

1. If no suitable policy exists, you will need to create one. Click on **Create this policy** to initiate the policy creation workflow.

The screenshot shows the 'Network Connectivity' page. At the top, there is a cloud icon and a link to 'Switch to connect via Agent'. Below the cloud icon, the text 'Select a policy' is displayed. A note states: 'A network egress policy is required to connect to sources outside Foundry's secure network. Select an existing policy if one already exists, otherwise create a new policy. [Learn more](#)'.

Two buttons are present: '+ Use existing policy' and 'Create new policy' (with a gear icon).

A section titled 'Suggested network egress policies' contains the text: 'These suggested egress policies are required for Foundry to connect to your source.' It includes a warning message: '⚠️ In addition to specifying network egress policies, S3 buckets in the same AWS region where your Foundry instance is hosted must also be added to the S3 bucket policies allow-list on the network egress page in Control Panel.'

Below this, a box contains the text: 's3-bucket.sandbox.training.palantir.com.s3.eu-west-1.amazonaws.com' and 'AWS S3 domain for bucket: s3-bucket.sandbox.training.palantir.com'. To the right of this box is a button labeled 'Create this policy' with a gear icon, which is highlighted with a green border.

2. Assign a descriptive name to your policy, such as S3 Training
3. Click **Next**.
4. The **Address** field should be pre-populated. If it is empty, enter
s3-bucket.sandbox.training.palantir.com.s3.eu-west-1.amazonaws.com.

Add network egress policy

1 Configure egress

Policies must be applied to sources in Data Connection to take effect.

2 Configure permissions

Description
Describe the type of traffic and/or destination that this policy will allow.

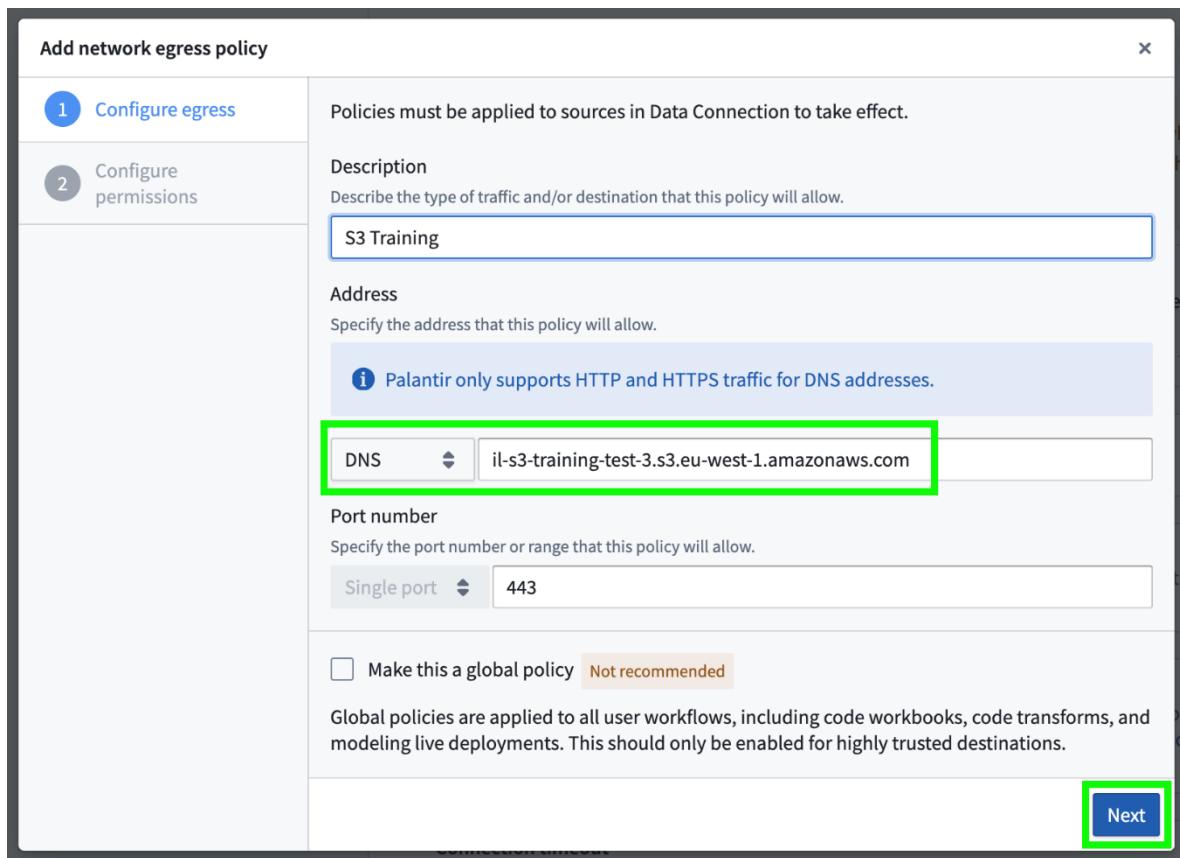
Address
Specify the address that this policy will allow.
(i) Palantir only supports HTTP and HTTPS traffic for DNS addresses.

DNS

Port number
Specify the port number or range that this policy will allow.
Single port

Make this a global policy Not recommended
Global policies are applied to all user workflows, including code workbooks, code transforms, and modeling live deployments. This should only be enabled for highly trusted destinations.

Next



5. Click **Save** to save the policy. It will be automatically applied to the source configuration

Add network egress policy

1 Configure egress

2 Configure permissions

Roles
Manage who has permission to view and import this policy

Add a user or group...

Training Policy Users Importer ▾

Organizations
Policies must have at least one organization marking

Select all organizations in **Palantir**

Search for organizations...

Palantir

Back Save

6. On the configuration screen click **Save and continue**.

Data connection > New Source

Connection timeout
The amount of time to wait (in milliseconds) when initially establishing a connection before giving up and timing out. Add

Socket timeout
The amount of time to wait (in milliseconds) for data to be transferred over an established, open connection before the connection times out and is closed. Add

Max connections
The maximum number of allowed open HTTP connections, per sync. Add

Max error retries
The maximum number of retry attempts for failed retryable requests (ex: 5xx error responses from services). Add

Client KMS key
A KMS key name or alias used to perform client-side data encryption with the AWS SDK. Using this option on an agent in PCloud requires proxy changes. Add

Match subfolder exactly
* Optionally match the path specified under subfolder as an exact subfolder in S3. If set to false, both s3://bucket-name/foo/bar/ and s3://bucket-name/foo/bar_baz/ will be matched with a subfolder setting of foo/bar/. Add

Proxy configurations Add

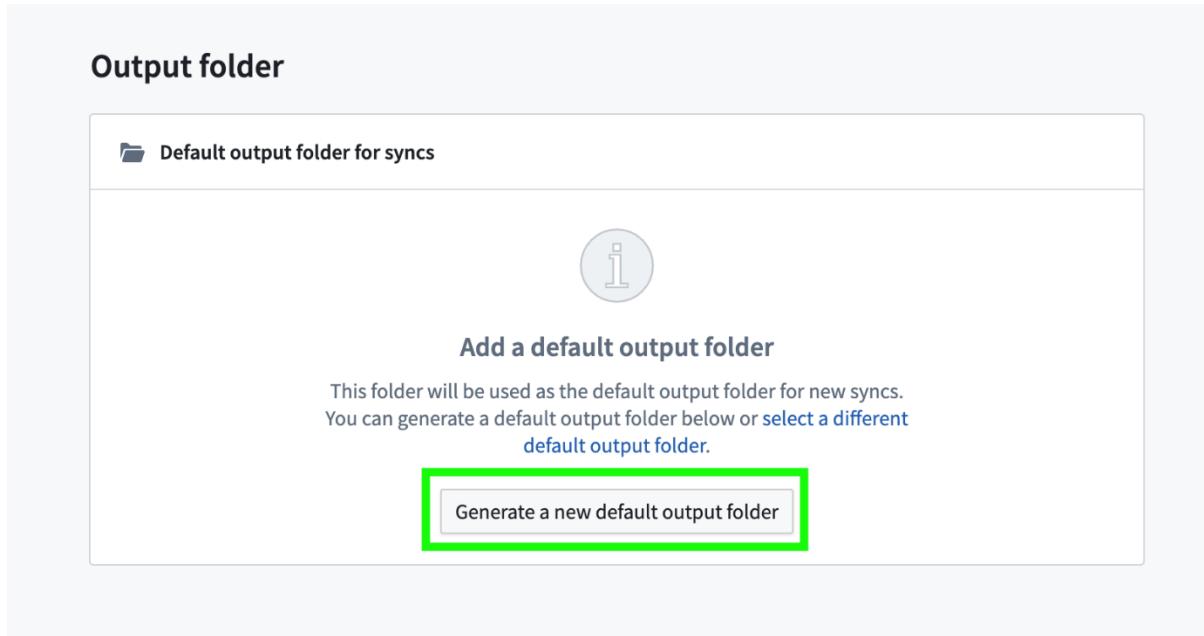
Need help? Check out the Source setup documentation Open ↗ Ask AIP Assist

Back Save and continue ➔

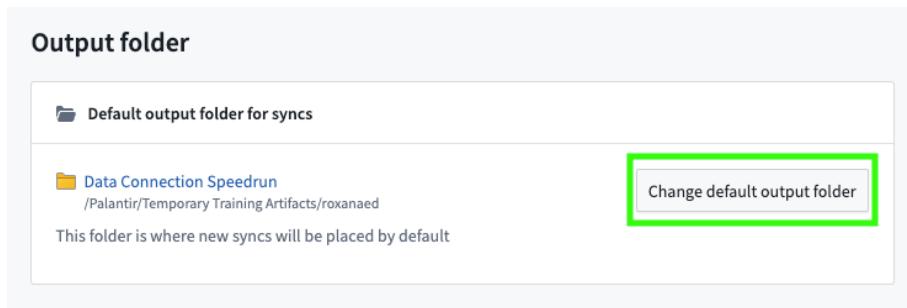
Select an Output Folder

Step 1: Set-up an output folder

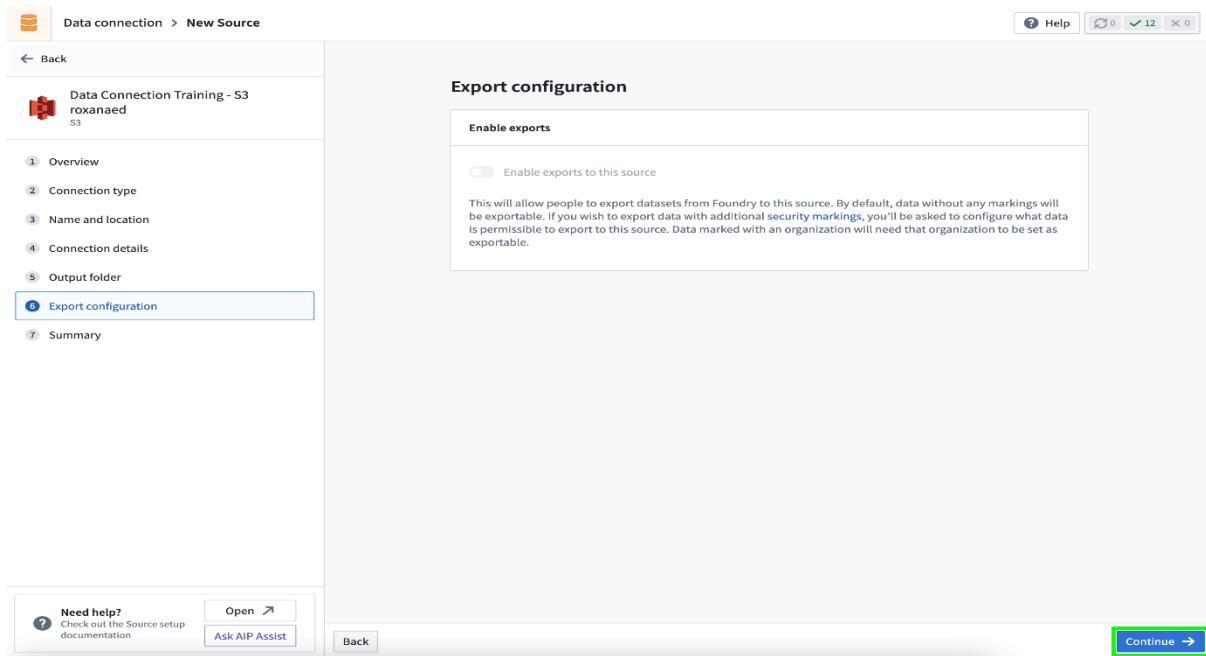
1. On the Output folder screen, select **Generate a default output folder**.
2. Click **Generate a default output folder** again on the following screen.



3. Note you can change the default output folder for the Source after it is live. In this case, we will leave the setting as defaulted.
4. Click **Next**.



5. On the Export configuration screen click **Continue**. We will cover Export Configurations in a following tutorial.
6. If the **Code import configuration** tab is visible, leave the settings as default and click **Continue**.



Configure a Data Sync

Step 1: Explore the Data Source and configure an Export

1. On the summary screen, click **Explore** to access and browse the S3 source you have just created.

The screenshot shows the 'Summary' page. At the top, it says 'What to do next'. Below are two options: 'Explore and import Source data' (with a green box around the 'Explore →' button) and 'Go to Source page' (with a green box around the 'Open Source →' button). Both buttons are rectangular with rounded corners and a slight shadow.

2. You will be directed to the Source Explorer (Explore source) view. Here you will view the datasets within the source that you have access to. Note it may take up to a few minutes for the source content to load on the initial connection.

Explore this source

You are previewing resources on the remote system that have not been synced to Foundry yet. Add your desired files to a sync to import them to Foundry.

1 Select a folder

/

2 Preview

Create a sync with root and 0 filters Add sync →

Filter files... *

NAME ^	PATH	SIZE	LAST UPDATED
plant_downtime_new.csv	plant_downtime_new.csv	3.37kB	Jan 8, 2024, 10:19 PM

Files or folders to sync into Foundry

Add files or folders to create a Sync

A Sync is how data is synchronized between an external source system and Foundry. Add files or folders here to begin the Sync creation process.

- To import the `plant_downtime_new.csv` file from the S3 bucket into a Foundry dataset, locate the file and click on the plus sign (+) next to it.

Preview

Create a sync with root and 0 filters Add sync →

Filter files... *

NAME ^	PATH	SIZE	LAST UPDATED
plant_downtime_new.csv	plant_downtime_new.csv	3.37kB	Feb 14, 2024, 4:28 PM

Sync only this file

Files or folders to sync into Foundry

To complete synchronizing the data into Foundry, create a Sync.

Create sync for 1 dataset →

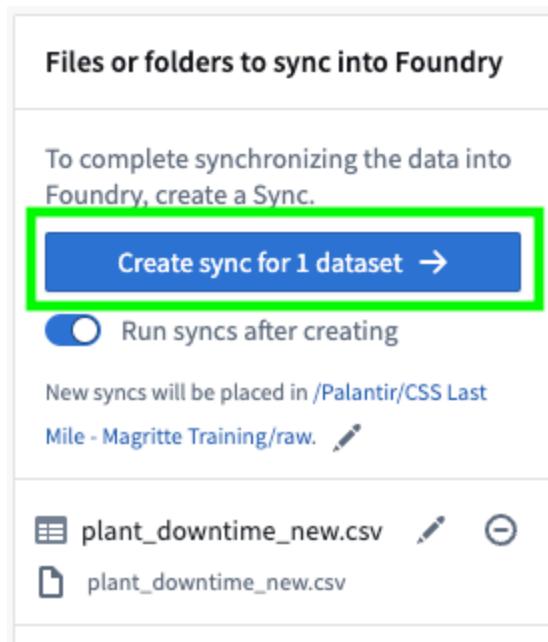
Run syncs after creating

New syncs will be placed in /Palantir/Temporary Training Artifacts/roxanaed/Data Connection Speedrun.

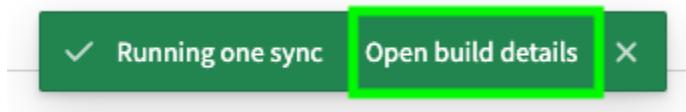
plant_downtime_new.csv

- Then, click on **Create sync for 1 dataset**. By doing so, you will establish a new data sync and trigger the initial import operation.

- Note: In Foundry, any operation that alters the contents of datasets is referred to as a dataset build.



5. After creating the sync, a pop-up will appear. Click on the **Open build details** button within this pop-up to monitor the build's progress.



6. Wait for the build to complete. You should eventually observe a successful build status, indicating that the data has been successfully imported into Foundry.

Build info		Build progress	
Status	Succeeded	Queued	Gantt chart
Duration	38s	Waiting	Job status
Estimated	--	Running	Dataset path...
Started	Today at 4:45 PM	Succeeded	86 K
Ended	Today at 4:46 PM	Failed	
Started by	Tomasz Musielak	Canceled	
Progress	1 of 1 job succeeded		
Build ID	cc-9d62-00025464c		

Datasets		Start time	Duration	Status	
plant_downtime_new.csv	Today at 4:45 PM	33s	Succeeded	Today at 4:46 PM	Compare Logs Actions
/Palantir/CSS Last Mile - Magritte Training/raw/plant_downtin		No previous runs			

7. In the Data Connection tab, if your screen looks like the below, you can return to source by clicking on the left arrow at the top left side of the screen.

The screenshot shows the Data connection interface with the following details:

- Header:** Data connection > 1 new sync
- Left sidebar:** Syncs (1) - plant_downtime_new.csv
- Right panel title:** Configure 1 sync
- Message:** 1 sync created, but needs to be configured
- Text:** You have successfully created 1 sync. We recommend configuring this sync so it runs on a regular schedule moving forward.
- Recommended next steps:**
 - Select a sync from the left: Configure the dataset if it should be imported regularly.
 - Add a schedule: Choose how often you want to import that dataset into Foundry, based on your desired data freshness for your workflows.
 - Add a transaction type: Transaction types are how Foundry adds new, updated data to your existing dataset.
- Buttons:** Start configuring 1 sync (button), Start → (button)

View Imported Data

Step 1: Open imported data as a Foundry dataset

1. To access the newly-created dataset, click on **Actions** and then select **Open**. This action will take you to the dataset view.

The screenshot shows the Foundry interface with the 'Job status' tab selected. A context menu is open over a specific job entry, listing various actions such as 'Open', 'Job details', 'Build history', 'View Spark Job details', 'Data connection', 'Explore data lineage', 'Manage Schedules', 'More', and 'View logs'. The 'Actions' option is highlighted with a green box.

2. In the dataset view, you will be able to confirm that the file has been successfully imported. Since the imported file contains structured data, Foundry is capable of displaying it in a tabular format. To facilitate this, click on the **Apply a schema** button.

The screenshot shows the dataset view for 'plant_downtime_new.csv'. The 'Apply a schema' button is highlighted with a green box. Other visible buttons include 'Explore pipeline', 'All actions', 'Build', 'Import', and 'Show hidden files'.

3. Once the schema is applied, you will see the data presented in an organized table, indicating that it is now ready for use within Foundry.

File ▾ Help ▾ 1 master ▾

Preview History Details Health Compare

Analyze in Contour Explore pipeline All actions Build

plant_downtime_new.csv

About Columns Schedules

Enter description...

Updated 12 minutes ago by Tomasz Musielak

Created 13 minutes ago by Tomasz Musielak

Location /Palantir/CS Last Mile - Magritte Trainin...

Type Dataset

Size 6 columns • 2 files • 112KB Calculate row count

Updated via Data connection • Edit Schema Show more

Tags Add tags

Health Checks View details There are no health checks on this dataset.

Inputs Explore data lineage

plant_downtime_new.csv

Showing 36 rows 6 columns Search columns...

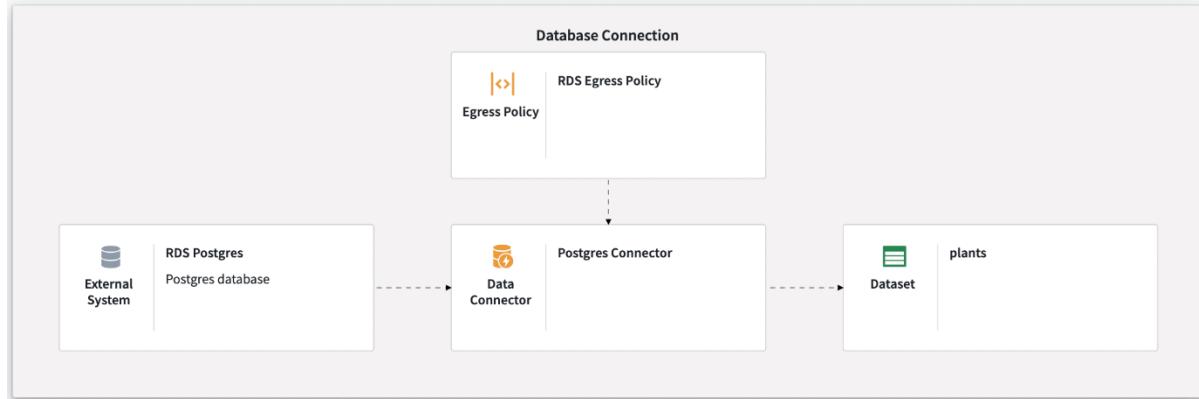
	downtime_plant_id	downtime_value	ingestionTime_time...	ingestionTime_ori...	timestamp_timestamp...	timestamp_origin...
1	titan_downtime_tdp_11	100	2023-10-10T22:40:38.1 +00:00		2023-10-10T22:40:34.1 +00:00	
2	titan_downtime_tdp_10	99	2023-10-10T22:40:38.1 +00:00		2023-10-10T22:40:33.1 +00:00	
3	titan_downtime_tdp_6	100	2023-10-10T22:40:33.1 +00:00		2023-10-10T22:40:32.1 +00:00	
4	titan_downtime_tdp_5	99	2023-10-10T22:40:33.1 +00:00		2023-10-10T22:40:31.1 +00:00	
5	titan_downtime_tdp_3	0	2023-10-10T22:40:33.1 +00:00		2023-10-10T22:40:30.1 +00:00	
6	titan_downtime_tdp_7	101	2023-10-10T22:40:33.1 +00:00		2023-10-10T22:40:29.1 +00:00	
7	titan_downtime_tdp_6	100	2023-10-10T22:40:33.1 +00:00		2023-10-10T22:40:28.1 +00:00	
8	titan_downtime_tdp_1	101	2023-10-10T22:40:28.1 +00:00		2023-10-10T22:40:27.1 +00:00	
9	titan_downtime_tdp_3	75	2023-10-10T22:40:28.1 +00:00		2023-10-10T22:40:26.1 +00:00	
10	titan_downtime_tdp_3	74	2023-10-10T22:40:48.1 +00:00		2023-10-10T22:40:43.1 +00:00	
11	titan_downtime_tdp_3	75	2023-10-10T22:40:43.1 +00:00		2023-10-10T22:40:42.1 +00:00	
12	titan_downtime_tdp_15	99	2023-10-10T22:40:43.1 +00:00		2023-10-10T22:40:41.1 +00:00	
13	titan_downtime_tdp_3	75	2023-10-10T22:40:43.1 +00:00		2023-10-10T22:40:40.1 +00:00	
14	titan_downtime_tdp_15	99	2023-10-10T22:40:43.1 +00:00		2023-10-10T22:40:39.1 +00:00	
15	titan_downtime_tdp_3	75	2023-10-10T22:40:43.1 +00:00		2023-10-10T22:40:38.1 +00:00	
16	titan_downtime_tdp_10	99	2023-10-10T22:40:38.1 +00:00		2023-10-10T22:40:37.1 +00:00	
17	titan_downtime_tdp_11	100	2023-10-10T22:40:38.1 +00:00		2023-10-10T22:40:36.1 +00:00	
18	titan_downtime_tdp_10	99	2023-10-10T22:40:38.1 +00:00		2023-10-10T22:40:35.1 +00:00	
19	titan_downtime_tdp_2	100	2023-10-10T22:40:53.1 +00:00		2023-10-10T22:40:52.1 +00:00	
20	titan_downtime_tdp_1	99	2023-10-10T22:40:53.1 +00:00		2023-10-10T22:40:51.1 +00:00	
21	titan_downtime_tdp_2	100	2023-10-10T22:40:53.1 +00:00		2023-10-10T22:40:50.1 +00:00	
22	titan_downtime_tdp_1	99	2023-10-10T22:40:53.1 +00:00		2023-10-10T22:40:49.1 +00:00	
23	titan_downtime_tdp_2	100	2023-10-10T22:40:53.1 +00:00		2023-10-10T22:40:48.1 +00:00	
24	titan_downtime_tdp_3	74	2023-10-10T22:40:48.1 +00:00		2023-10-10T22:40:47.1 +00:00	
25	titan_downtime_tdp_3	75	2023-10-10T22:40:48.1 +00:00		2023-10-10T22:40:46.1 +00:00	Calculate row count

You've now successfully completed the steps to import data to Foundry from an S3 source! Your resulting Foundry dataset can now be used as an input into pipelining and analysis tools, as well as the Ontology.

Data Connection: Relational Database

Introduction

In this lesson, you will learn how to establish a connection from Foundry to a Postgres relational database. Your objective is to retrieve data for use within the platform. The architecture of this connection is illustrated in the diagram below:



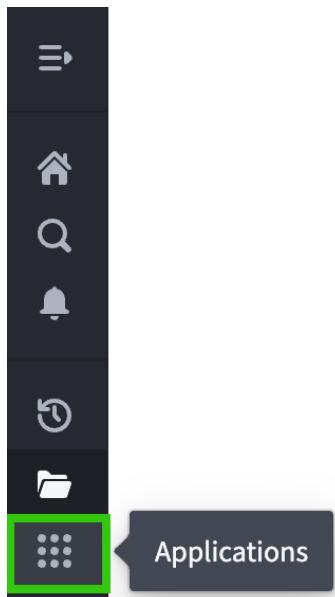
A relational database is a database management system (DBMS) that organizes data in a structured format, typically using tables, columns, and rows. These tables represent various entities and their relationships, which facilitates efficient data retrieval and manipulation.

Please note that, for this example, you will be using an AWS RDS hosted Postgres database. However, the process you'll learn for connecting to this database is similar to the steps you would follow for any other database that supports the JDBC protocol.

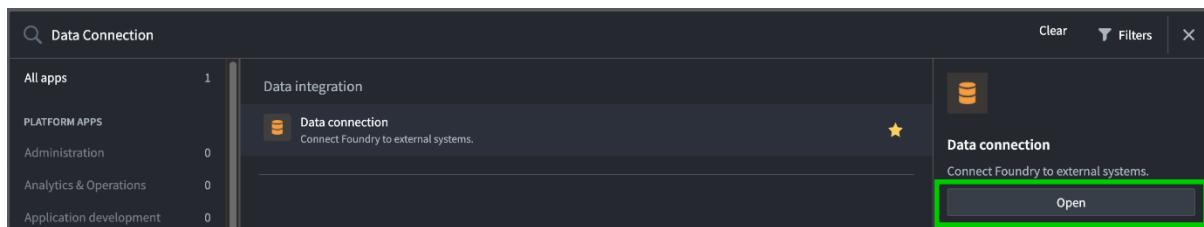
Open the Data Connection Application

Step 1: Open Data Connection application

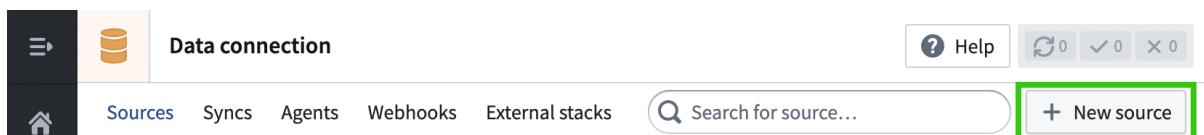
1. Click on the **Applications** button in the Foundry menu.



2. Search for Data Connection and click **Open**.



3. Inside the Data Connection application, click **New source**.



Select Source Type

Step 1: Select the PostgreSQL source type

1. Select **PostgreSQL** as the source type. Use the search box to help narrow down your selection if needed.

Select source type

Sources

To connect to data from the Internet or an on-premises source, select from the listed source types.

Postgres

The screenshot shows a list of source types under the heading 'Sources'. The 'Postgres' option is highlighted with a blue border. Below it are other options: 'Batch syncs', 'CDC syncs', and 'Streaming exports'. Each option has a small icon to its left.

2. This action will initiate the new source creation workflow. Click **Continue** in the bottom right corner.

The screenshot shows the 'Overview' step of the 'New Source' setup wizard. The sidebar on the left lists steps: 1. Overview (selected), 2. Connection type, 3. Name and location, 4. Connection details, 5. Export configuration, and 6. Summary. The main area shows the 'Creating a New Source' section with a diagram of a cloud icon connected to a database icon. Below it is a note: 'A Source is any data system that you connect to Foundry. Source systems may support a variety of capabilities including imports, streaming, and webhooks.' The 'What we're going to do' section shows a flowchart: SOURCE SETUP (Current setup) → SOURCE → AGENT / EGRESS POLICY → SYNC → FOUNDRY. A note below says: 'In this wizard, you will set up your Source via the following steps:' followed by a numbered list: 1. Set up your source metadata, 2. Connect to Source using your credentials, 3. Ensure you can access the Source using either an Agent or an Egress policy, 4. Confirm your Source is connected and ready to use. A note at the bottom states: 'Note: connecting a Source does not automatically download your data to Foundry. Data must be imported before it is used. A Sync is how you import data to Foundry, and Webhooks allow you to make user-click-driven outbound requests from applications built in Foundry. After you are done configuring your Source in this setup process, you have the option to continue by configuring a Sync or setting up a Webhook.' At the bottom, there are 'Need help?' links, an 'Open' button, an 'Ask AI/PA Assist' button, and a 'Continue →' button.

Select Connection Type

Step 1: Configure the Data Connection settings

1. Select the **Direct connection** option. Just as with the previous S3 step, the database you are connecting to is accessible over the internet.
2. Click **Continue** to proceed to the next step.

The screenshot shows the 'Data connection > New Source' interface. On the left, a sidebar lists steps 1 through 7. Step 2, 'Connection type', is currently selected and highlighted with a blue border. The main area is titled 'Connect to your data source'. It displays two options: 'Direct connection' (selected) and 'Through an Agent'. The 'Direct connection' section includes a description, icons, and requirements. A green box highlights the 'Selected' button at the bottom of this section. To the right, the 'Through an Agent' section is partially visible. At the bottom right, a 'Continue →' button is also highlighted with a green box.

Create Source

Step 1: Create the Source connection

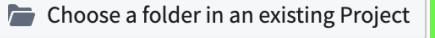
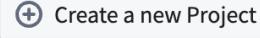
1. Provide a name for your source. For example, Data Connection Training - Postgres (your_name).

The screenshot shows the 'Name and Project' step of the source creation process. It has a 'Source name' field containing 'Data Connection Training - Postgres (your_name)'. Below it is a 'Name' field with the same value. A descriptive note below the name field says: 'Provide a unique, distinguishable name, such as the server name or the database name.' At the bottom right of the form, there is a 'Continue' button.

2. Select the Project and folder where you want to save the Source. If you have created an existing Learning Project, continue to use that Project for this course.

Project to save Source in

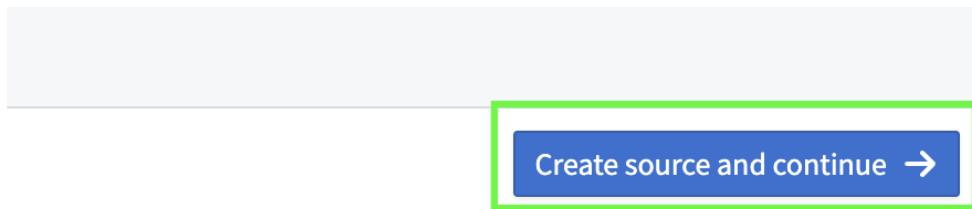
Choose a location

 Choose a folder in an existing Project or  Create a new Project

Anyone with Editor or Owner roles on your selected Project will be able to read or write to the data source.

 How to choose a location for your Source Expand >

3. Click **Create source and continue** in the bottom right corner to finalize the creation process and proceed.



Input Connection Details

Step 1: Configure the connection settings

1. Input the following database connection details. Use the screenshot below for guidance.
 - **Host type:** Select 'hostname' from the options.
 - **Hostname:** Enter database.sandbox.training.palantir.com
 - **Port:** Type in 5432
 - **Database name:** Use postgres as the database name.

Source Setup

i If you're using a version of postgres older than 8.2 (which was released in 2006), then please use the general JDBC source type with the proper driver.

Connection settings

Select host type:

Hostname

IPv4

IPv6

Hostname

database.sandbox.training.palantir.com

Port

5432



Database name

postgres

2. Next, enter the access credentials provided below.

- **Username:** postgres_user
- **Password:** speakfriendandenterdatabase

Authentication

Username (required)

postgres_user

Password (required)

.....

Encrypted

Set-up Egress Policies

As with the S3 connection, you must ensure the PostgreSQL database is securely accessible to Foundry by creating or importing a network egress policy for the connection. This policy whitelists the data source, allowing secure access from your Foundry instance. This step is only required once—if someone has already set up the policy during a previous training session, you can reuse it.

Foundry will suggest the recommended next steps based on the URL you've inputted in the previous step. **Note that we are creating two Egress Policies in this section.**

Step 1a: Select an existing Egress Policy

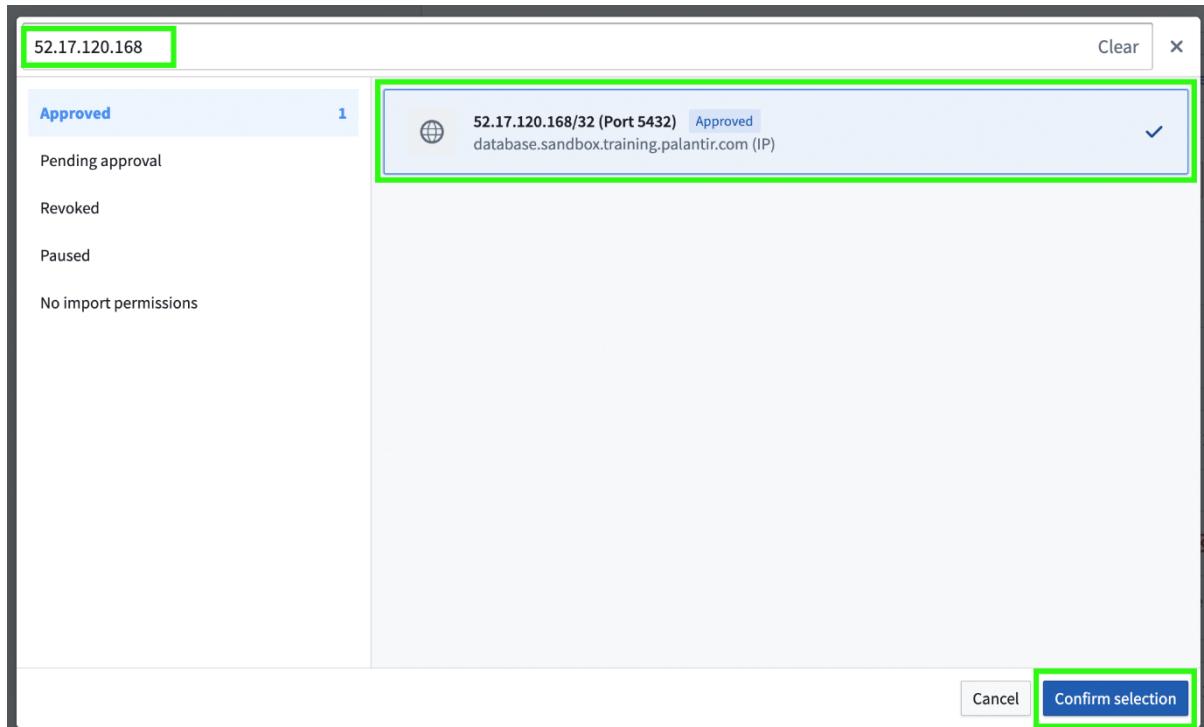
1. If the database.sandbox.training.palantir.com policy already exists, import it by clicking **Import existing policy**.

The screenshot shows the 'Network Connectivity' page. At the top right is a 'Switch to connect via Agent' button. Below it is a cloud icon. The main heading is 'Select a policy'. A sub-instruction says: 'A network egress policy is required to connect to sources outside Foundry's secure network. Select an existing policy if one already exists, otherwise create a new policy.' It includes 'Learn more' and 'Use existing policy' and 'Create new policy' buttons. Below this is a section titled 'Suggested network egress policies' with the sub-instruction: 'These suggested egress policies are required for Foundry to connect to your source.' It lists 'database.sandbox.training.palantir.com (Port 5432)' and 'database.sandbox.training.palantir.com'. The 'Import existing policy' button for the first item is highlighted with a green border.

2. Click **Add another policy**.

The screenshot shows the 'Network Connectivity' page after adding a policy. It displays a list of policies: 'database.sandbox.training.palantir.com (Port 5432) Approved' and 'database.sandbox.training.palantir.com'. To the right of the approved policy is a 'Remove' button. Below the list are 'Add another policy' and 'Request new policy' buttons. The 'Add another policy' button is highlighted with a green border.

3. Search 52.17.120.168 in the search bar to see if the second policy exists. If it exists, click on the entry and then on **Confirm selection**. If it does not exist, move on to **Step 1b.6** below.



Step 1b: Create a new Egress Policy

1. If neither of the two policies exist, you will need to create them. Click on **Create this policy** to initiate the policy creation workflow for the database.sandbox.training.palantir.com policy.

Network Connectivity Switch to connect via Agent



Select a policy

A network egress policy is required to connect to sources outside Foundry's secure network. Select an existing policy if one already exists, otherwise create a new policy. [Learn more](#)

+ Use existing policy Create new policy

Suggested network egress policies
These suggested egress policies are required for Foundry to connect to your source.

database.sandbox.training.palantir.com	 Create this policy
----------------------------------------	--------------------------------------------------------------------------------------------------------------------------

2. Assign a descriptive name to the policy, such as Postgres Training.
3. If not automatically populated, enter the DNS database.sandbox.training.palantir.com and the port 5432.
4. Click **Next** to proceed.

Add network egress policy

1 Configure egress Policies must be applied to sources in Data Connection to take effect.

2 Configure permissions

Description
Describe the type of traffic and/or destination that this policy will allow.
Postgres Training

Address
Specify the address that this policy will allow.
i Palantir only supports HTTP and HTTPS traffic for DNS addresses.

DNS database.sandbox.training.palantir.com

Port number
Specify the port number or range that this policy will allow.
Single port 5432

Make this a global policy Not recommended
Global policies are applied to all user workflows, including code workbooks, code transforms, and modeling live deployments. This should only be enabled for highly trusted destinations.

Next

- Finally, click **Save** to save the policy. It will then be automatically applied to your source configuration.

Add network egress policy

1 Configure egress

2 Configure permissions

Roles
Manage who has permission to view and import this policy

Add a user or group...

Training Policy Users Importer ▾

Organizations
Policies must have at least one organization marking

Select all organizations in Palantir

Search for organizations...

Palantir

Back **Save**

6. You will also need to create a second egress policy to access the database using its IP address 52.17.120.168. Click **Request new policy**.

Network Connectivity

Switch to connect via Agent

	database.sandbox.training.palantir.com (Port 5432)	Approved	Remove
database.sandbox.training.palantir.com			

+ Add another policy **Request new policy**

7. Assign a descriptive name to the policy, such as Postgres Training (IP) .
8. If not automatically populated, enter the IP address 52.17.120.168 and the port 5432.
9. Click **Next** to proceed.

Request egress policy

1 Configure egress Policies must be applied to sources in Data Connection to take effect.

2 Configure permissions Description Describe the type of traffic and/or destination that this policy will allow.

Description Postgres Training (IP)

Address Specify the address that this policy will allow.

IP 52.17.120.16

Port number Specify the port number or range that this policy will allow.

Single port 5432

Next

10. Finally, click **Save** to save the policy. It will then be automatically applied to your source configuration.

Add network egress policy

1 Configure egress Roles Manage who has permission to view and import this policy

2 Configure permissions

Roles Add a user or group... Training Policy Users Importer

Organizations Policies must have at least one organization marking

Select all organizations in Palantir Search for organizations... Palantir

Back **Save**

Configure Certificates

For the PostgreSQL connection, you also need to configure certificates. SSL certificates are used to establish a secure, encrypted connection between the client (in this case, Foundry) and the server (the PostgreSQL database). This ensures that the data transmitted over the network cannot be intercepted or tampered with, which is critical for sensitive information. The SSL certificate also verifies the identity of the server to prevent man-in-the-middle attacks.

You didn't need to configure certificates before because S3 connections are secured using AWS's built-in mechanisms.

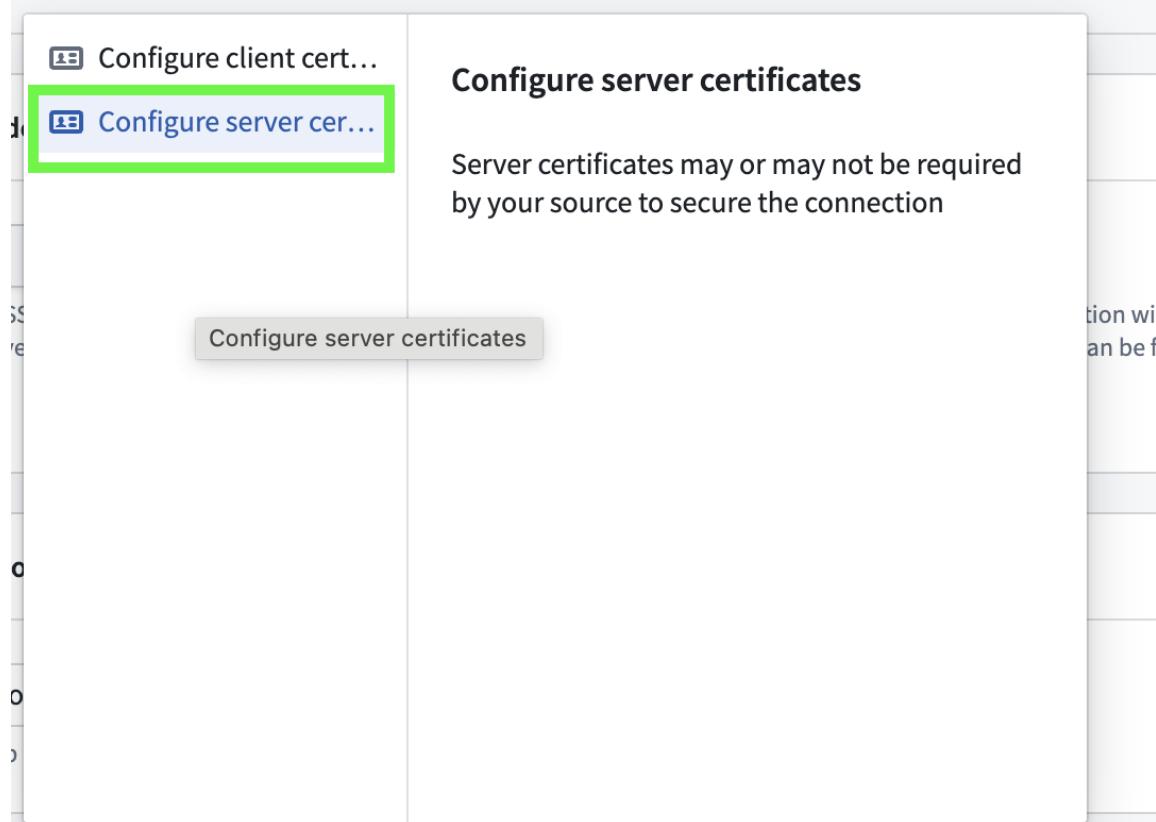
Step 1: Add Certificate details to the connection

1. Still on the **Connection Details** tab, scroll to the bottom of the page and select **+ More Options**.

The screenshot shows two configuration panels stacked vertically. The top panel is titled "SSL mode" and contains a dropdown menu set to "require". A descriptive text below explains the SSL settings, mentioning "prefer" as the default setting. The bottom panel is titled "JDBC properties" and includes a "Add property" button and a note about properties passed to the JDBC driver. A large green rectangular box highlights the "More options" button at the bottom right of the JDBC properties panel.

2. Select **Configure server certificates** from the options on the left panel.

So connections from Foundry come from?



Configure client cert...

Configure server cer...

Configure server certificates

Server certificates may or may not be required by your source to secure the connection

3. Add the certificate details. To get the server certificate, open the file attached to this lesson in a text editor and copy its entire contents into the text box.
 - **Alias:** server.pem
 - **Certificate content:** server-certificate.txt (attached to this lesson, download at the bottom of this page)



Add server certificate...

Alias

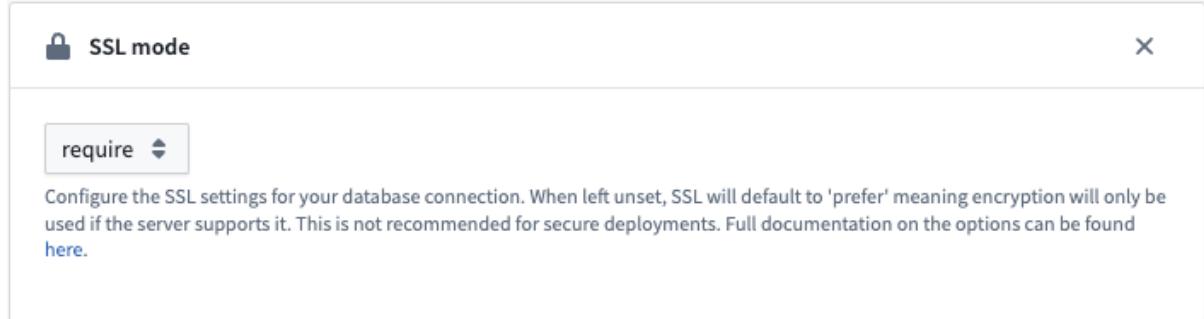
server.pem

Certificate content

-----END CERTIFICATE-----

Add

4. Set the SSL mode to **require**.



5. On the configuration screen, click **Save and continue** in the bottom right corner.

A screenshot of the "Data connection > New Source" configuration screen. The left sidebar shows the following tabs: Back, Data Connection Training - Postgres (selected), Overview, Connection type, Name and location, Connection details (selected), Output folder, Export configuration, and Summary. The main area contains several configuration sections: "Username (required)" set to "postgres", "Password (required)" masked, and an "Encrypted" link. Below this is the "Network Connectivity" section, which lists a policy for "database.sandbox.training.palantir.com (Port 5432)" with status "Approved" and a "Remove" button. There are also "Add another policy" and "Request new policy" buttons. The "Configure server certificates" section shows a certificate named "server.pem" with a red "X" icon and a "Delete" link. At the bottom right, a blue button labeled "Save and continue →" is highlighted with a green box.

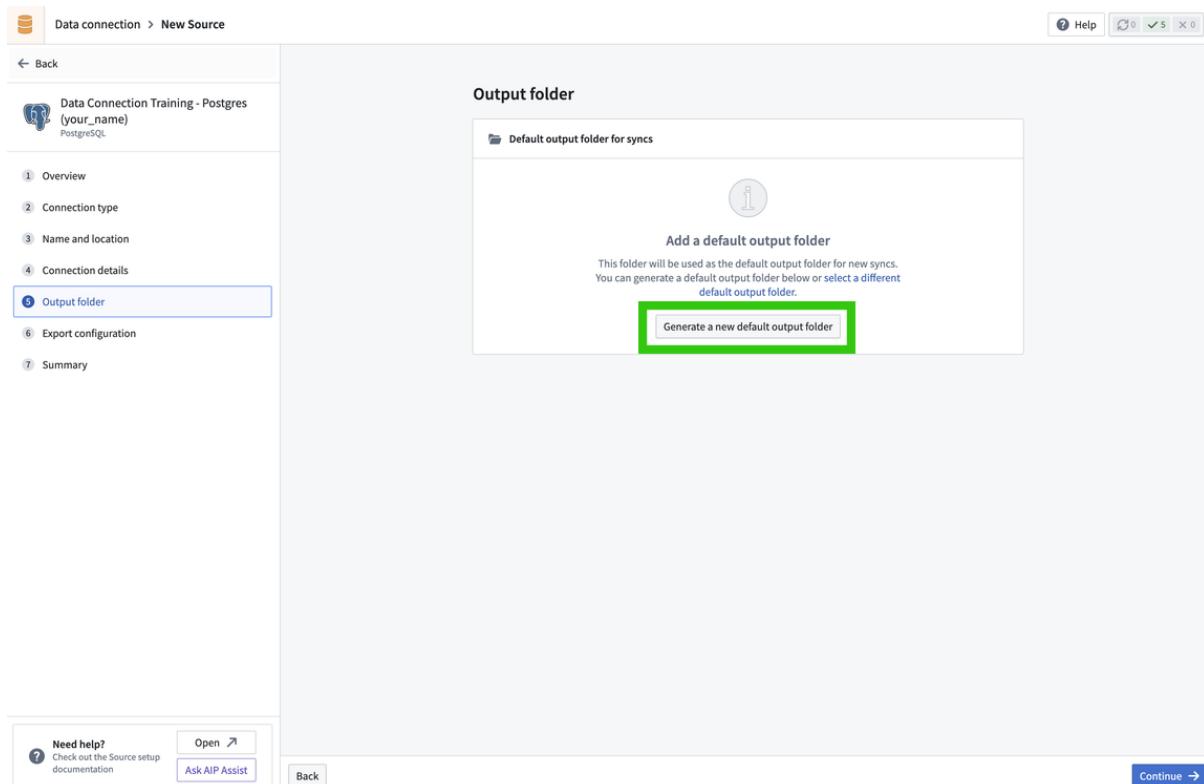
Downloads

- [server-certificate.txt](#)

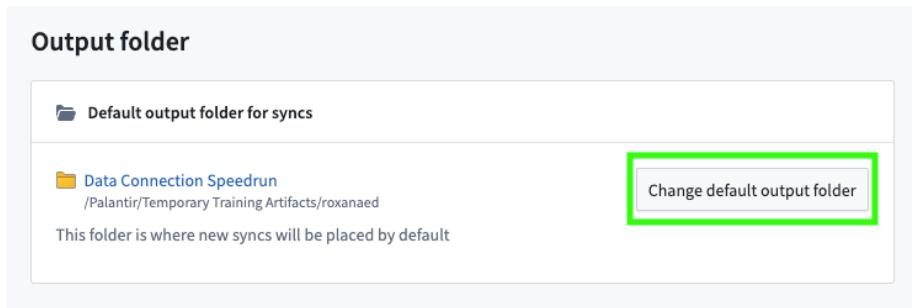
Select an Output Folder

Step 1: Set-up an output folder

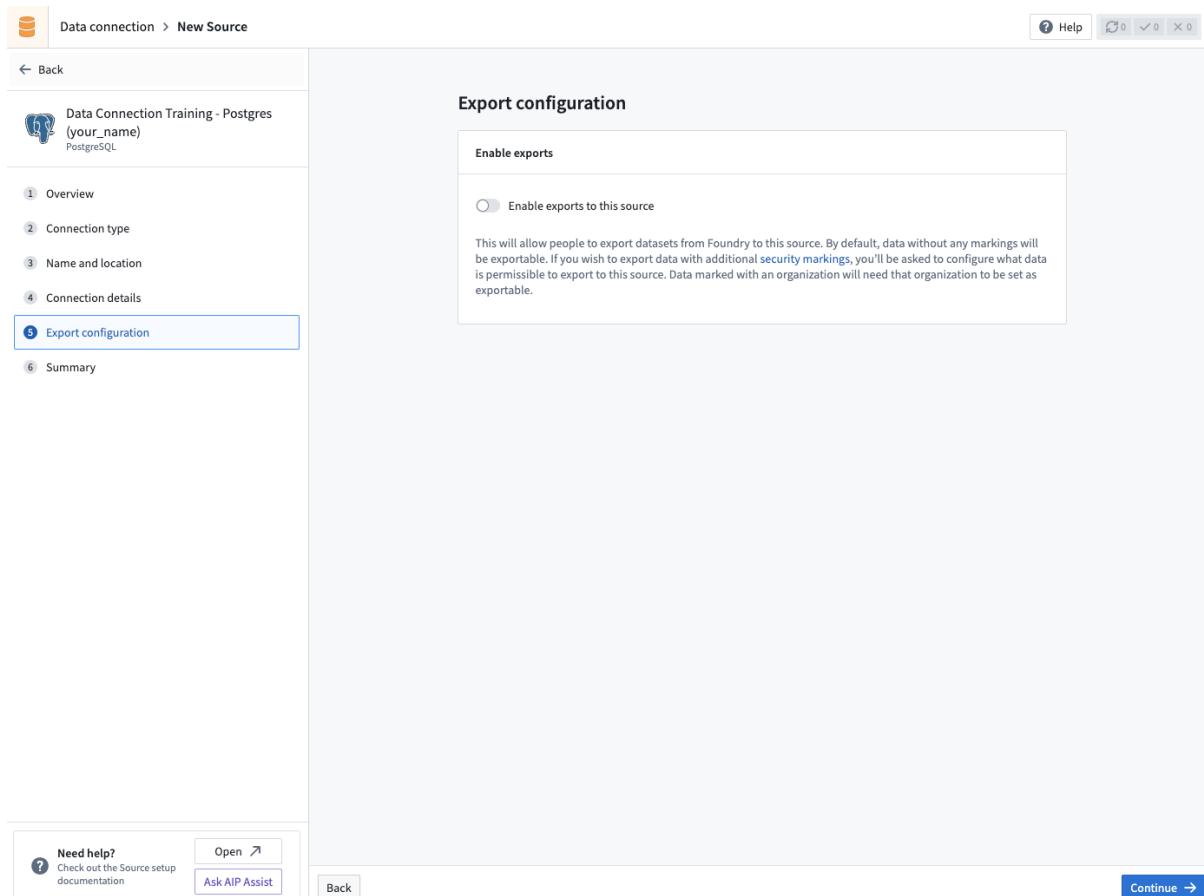
1. On the Output folder screen, select **Generate a default output folder**.
2. Click **Generate a default output folder** again on the following screen.



3. Note you can change the default output folder for the Source after it is live. In this case, we will leave the setting as defaulted.
4. Click **Next**.



5. On the Export configuration screen click **Continue**. We will cover Export Configurations in a following tutorial.
6. If the **Code import configuration** tab is visible, leave the settings as default and click **Continue**.



Configure a Data Sync

Step 1: Explore the Data Source and configure an Export

1. On the summary screen, click **Explore** to access and browse the Relational Database source you have just created.

Summary

What to do next

Explore and import Source data

Explore the data available on this Source and select data to synchronize into Foundry.

[Explore →](#)

Go to Source page

Go to Source page to monitor Source health and syncs.

[Open Source →](#)

You will be directed to the Source Explorer (Explore source) view. Here you will view the datasets within the source that you have access to. Note it may take up to a few minutes for the source content to load on the initial connection.

The screenshot shows the Foundry interface with the 'Explore source' tab selected (highlighted by a green box). The left sidebar displays a preview of resources from a remote system, specifically the 'public' folder which contains 'plant_types' and 'plants'. A 'Filter by name...' search bar is available. The right side shows a list of resources to sync into Foundry, with a prominent 'Add resources to create a Sync' button and a detailed description of what a Sync is.

Overview Connection settings CDC syncs Edit sync **Explore source** Logs

You are previewing resources on the remote system that have not been synced to Foundry yet. Start by selecting resources from the resource list and then create a sync to import the data into Foundry.

Previewing source Refresh

These resources are a preview of what can be synced into Foundry.

Filter by name...

public

- plant_types +
- plants +

Select a resource to preview data

Select a resource from the left sidebar or the graph below to preview it here. To synchronize it into Foundry, start by clicking "Add dataset to Sync".

Resources to sync into Foundry

Add resources to create a Sync

A Sync is how data is synchronized between an external source system and Foundry. Add resources here to begin the Sync creation process.

3. Locate and click on the **plants** table to preview its contents. Clicking on the **+** sign next to the table name will schedule the data for a Foundry sync (same functionality as clicking **Add dataset to sync**).

i You are previewing resources on the remote system that have not been synced to Foundry yet. Start by selecting resources from the resource list and then create a sync to import the data into Foundry.

Previewing source (Refresh)

These resources are a preview of what can be synced into Foundry.

Filter by name...

▼ public

- plant types (+)
- plants** (+) [highlighted]

	id INTEGER	name STRING	type STRING	color STRING	sunlight STRING
1	1	Daisy	flower	white	full sun
2	2	Tulip	flower	red	full sun
3	3	Fern	houseplant	green	partial shade
4	4	Aloe	succulent	green	full sun
5	5	Palm	tree	green	full sun
6	6	Rose	flower	pink	full sun
7	7	Lemon Tree	tree	green	full sun
8	8	Orchid	flower	purple	indirect light
9	9	Spider Plant	houseplant	green	indirect light
10	10	Cactus	succulent	green	full sun
11	11	Lavender	flower	purple	full sun
12	12	Bamboo	tree	green	partial shade
13	13	Ivy	houseplant	green	partial shade
14	14	Sunflower	flower	yellow	full sun
15	15	Maple	tree	orange	full sun
16	16	Snake Plant	houseplant	green	low light
17	17	Echeveria	succulent	green	full sun
18	18	Daffodil	flower	yellow	full sun
19	19	Oak	tree	green	full sun
20	20	Ficus	houseplant	green	indirect light

Add dataset to Sync →

Resources to sync into Foundry

+

Add resources to create a Sync

A Sync is how data is synchronized between an external source system and Foundry. Add resources here to begin the Sync creation process.

4. Click on **Create sync for 1 dataset**. This action will finalize the data sync setup and trigger the initial import operation.

✓ Added to Sync

	type STRING
	flower
	flower
	houseplant
	succulent
	tree
	flower
	tree
	flower
	houseplant
	succulent
	flower
	tree
	houseplant
	flower
	tree
	houseplant

Tables to sync into Foundry

To complete synchronizing the data into Foundry, create a Sync.

Create sync for 1 dataset → [highlighted]

Run syncs after creating

New syncs will be placed in /Palantir/Temporary Training Artifacts/roxanaed/Data Connection

Speedrun. (edit)

"public"."plants" (edit) (remove)

5. Once the sync is created, a pop-up will appear. Click on the **Open build details** button within this pop-up to monitor the build's progress.



6. After some time, you should observe a successful build indicating that the data import is complete.

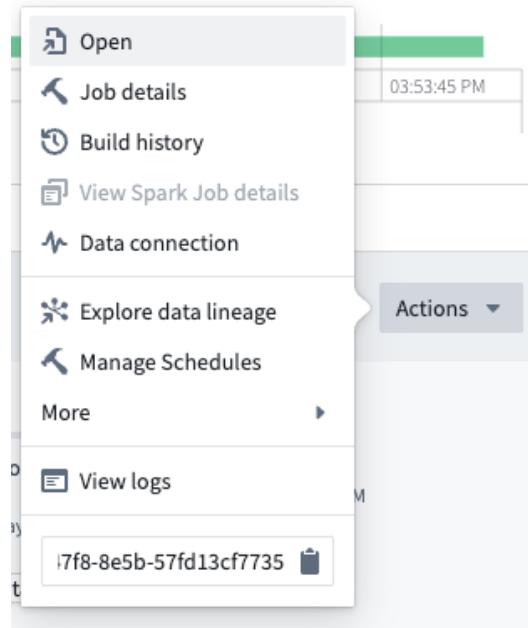
Datasets	Start time	Duration	Status
"public'"plants" /Palantir//your_username Data Connection Training/raw/"public"	Today at 3:53 PM	38s Typically 38s	✓ Succeeded Today at 3:53 PM

7. In the Data Connection tab, if your screen looks like the below, you can return to source by clicking on the left arrow at the top left side of the screen.

View Imported Data

Step 1: Open imported data as a Foundry dataset

1. To access the newly-created dataset, click on **Actions** and then select **Open**. This action will take you to the dataset view.



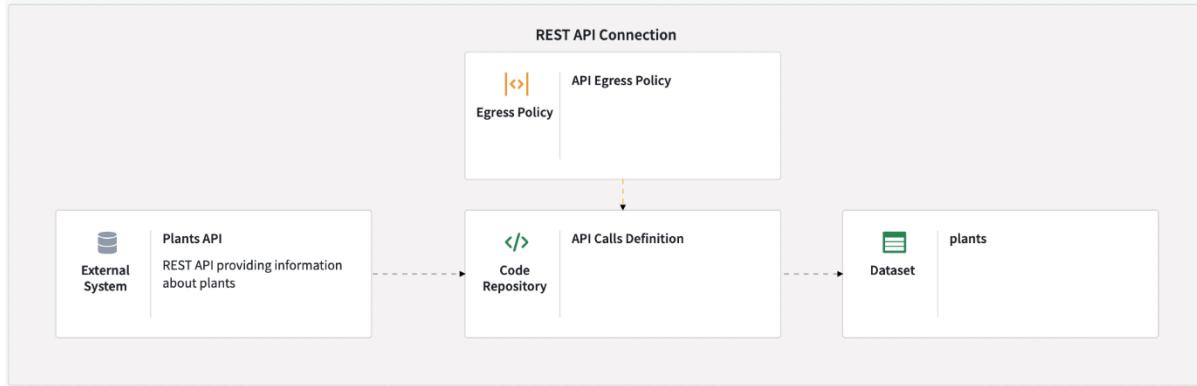
2. In the dataset view, you will be able to confirm that the file has been successfully imported. Since the imported file contains structured data, Foundry is capable of displaying it in a tabular format. To facilitate this, click on the **Apply a schema** button.

You've now successfully completed the steps to import data to Foundry from a relational database! Your resulting Foundry dataset can now be used as an input into pipelining and analysis tools, as well as the Ontology.

Data Connection: REST API

Introduction

In this lesson, you will learn how to establish a connection from Foundry to a REST API, with a view to retrieving data for use within the platform. The architecture of this connection is illustrated in the diagram below:



REST APIs, or Representational State Transfer Application Programming Interfaces, are a prevalent architectural style for creating networked applications. They are the backbone of web communication, enabling client and server applications to exchange data seamlessly using a set of well-defined principles, conventions, and constraints.

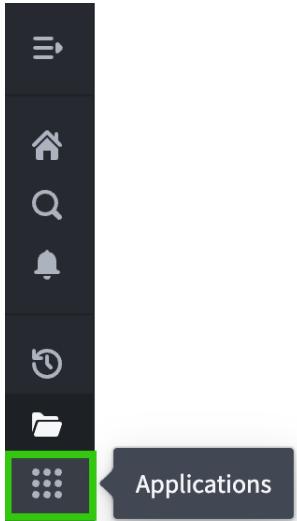
By leveraging HTTP protocol methods (GET, POST, PUT, and DELETE), REST APIs interact with resources identified by URLs. Their simplicity, scalability, and broad compatibility make REST APIs especially useful when interacting with the outside world, as many external systems use APIs to provide access to public data and services.

It's important to note that the example provided here focuses on a REST API, but the general process for connecting to any system that exposes HTTP endpoints will be similar.

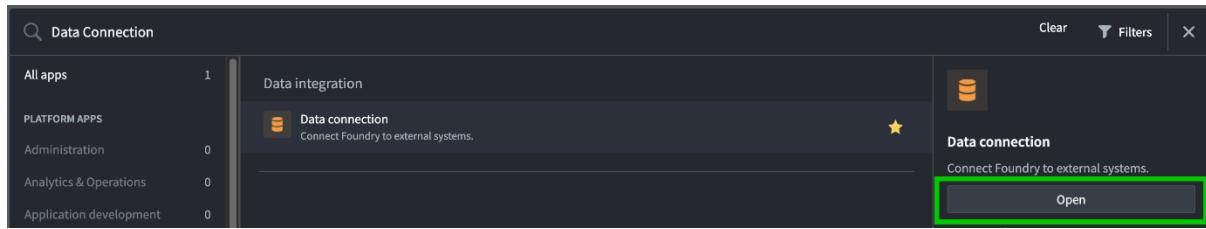
Open the Data Connection Application

Step 1: Open Data Connection application

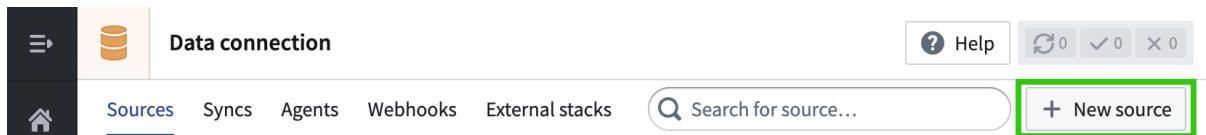
1. Click on the **Applications** button in the Foundry menu.



2. Search for Data Connection and click **Open**.



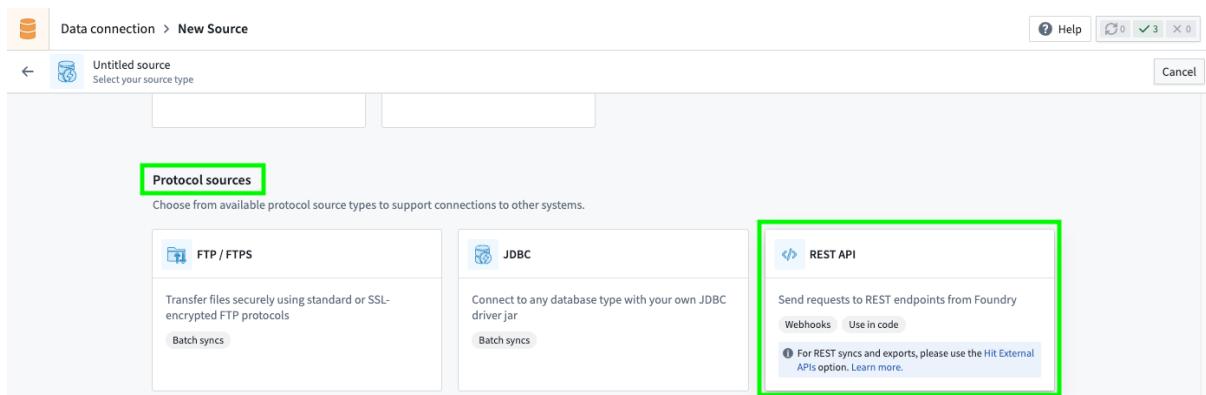
3. Inside the Data Connection application, click **New source**.



Select Source Type

Step 1: Set-up Rest API source

1. Scroll down through the list of supported sources until you reach the **Protocol sources** section, and then select **REST API**. This action will initiate the new source creation workflow.



2. Click **Continue** to proceed to the next step.

Select Connection Type

Step 1: Set-up a Direct Connection

1. Choose **Foundry worker**, since the S3 resources used in this tutorial are accessible over the public Internet.
2. Click **Continue** in the bottom right corner to move forward.

Create Source

Step 1: Create source and save to your Project

1. Assign a name to your source, for example, Data Connection Training - REST API (your_name).

Name and Project

Source name

Name

Provide a unique, distinguishable name, such as the server name or the database name.

2. Choose your working project as the location to save your new Source. If you have created a new folder within your Learning Project specifically for this training, use that folder as your location.

Project to save Source in

Choose a location

Choose a folder in an existing Project or Create a new Project

Anyone with Editor or Owner roles on your selected Project will be able to read or write to the data source.

i [How to choose a location for your Source](#) [Expand >](#)

3. Click **Create source and continue** in the bottom right corner to finalize the creation process and proceed.

Input Connection Details

Step 1: Update the S3 connection settings

1. Input the following API credentials.

- **Domain base URL:** Enter api.sandbox.training.palantir.com.
- **Authentication:** Select None - authentication is going to be handled in the logic for making the calls.
- **Port:** Type in 443, which is the standard port for HTTPS connections.

Source Setup

Domains

https://api.sandbox.training.palantir.com	No auth	Port 443	Remove	▼
-------------------------------------------	---------	----------	--------	---

Domain base URL
HTTPS ▾ api.sandbox.training.palantir.com

Authentication
None

Port (optional)
443 ▾ X

Add new domain

ⓘ Based on the selected authentication mode, you may need different inputs. To learn more about HTTP authentication, please visit the [MDN documentation](#).

2. Input the following into the 'Additional secrets' section:

- Name: Secret
- Secret value: secretToken

 **Additional secrets**
Define additional encrypted secrets that can be referenced in code and webhook configurations.

additionalSecretSecret	:	secretToken
------------------------	---	-------------

Secret  

+ Add Secret

Set-up an Egress Policy

As with the previous sources, you must ensure that Foundry can connect to your REST API by creating or importing a network egress policy for the connection. This policy whitelists the REST API address, allowing secure access from your Foundry instance.

Foundry will suggest the recommended next steps based on the URL you've inputted in the previous step.

Note: Skip the Additional secrets section that comes before this section.

Step 1a: Select an existing Egress Policy

1. If the policy already exists, import it by clicking **Import existing policy**.

Network Connectivity

Switch to connect via Agent

Cloud icon

Select a policy

A network egress policy is required to connect to sources outside Foundry's secure network. Select an existing policy if one already exists, otherwise create a new policy. [Learn more](#)

[+ Use existing policy](#) [Create new policy](#)

+ Suggested network egress policies

These suggested egress policies are required for Foundry to connect to your source.

 api.sandbox.training.palantir.com (Port 443) Magritte Tutorial REST Source	Import existing policy
-----------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------

Step 1b: Create a new Egress Policy

1. If no suitable policy exists, you will need to create one. Click on **Request new policy** to initiate the policy creation workflow.
2. Leave 'How to reach an address' as Direct connection.
3. Provide a descriptive name for the policy, such as REST Training.

4. Enter the DNS api.sandbox.training.palantir.com and specify the port 443.
5. Leave 'Transport protocol' as TLS.
6. Click **Next** to continue.

Request egress policy

1 Configure egress

Policies must be applied to sources in Data Connection to take effect.

2 Configure permissions

How to reach an address

Direct connection

Description

Describe the type of traffic and/or destination that this policy will allow.

REST Training

Address

Specify the address that this policy will allow.

DNS api.sandbox.training.palantir.com

Port number

Specify the port number or range that this policy will allow.

Single port 443

Transport protocol

TLS

Next

5. Click **Save** to save the new policy, which will then be automatically applied to your source configuration.

Add network egress policy

1 Configure egress

2 Configure permissions

Roles
Manage who has permission to view and import this policy

Add a user or group...

Training Policy Users Importer ▾

Organizations
Policies must have at least one organization marking

Select all organizations in Palantir

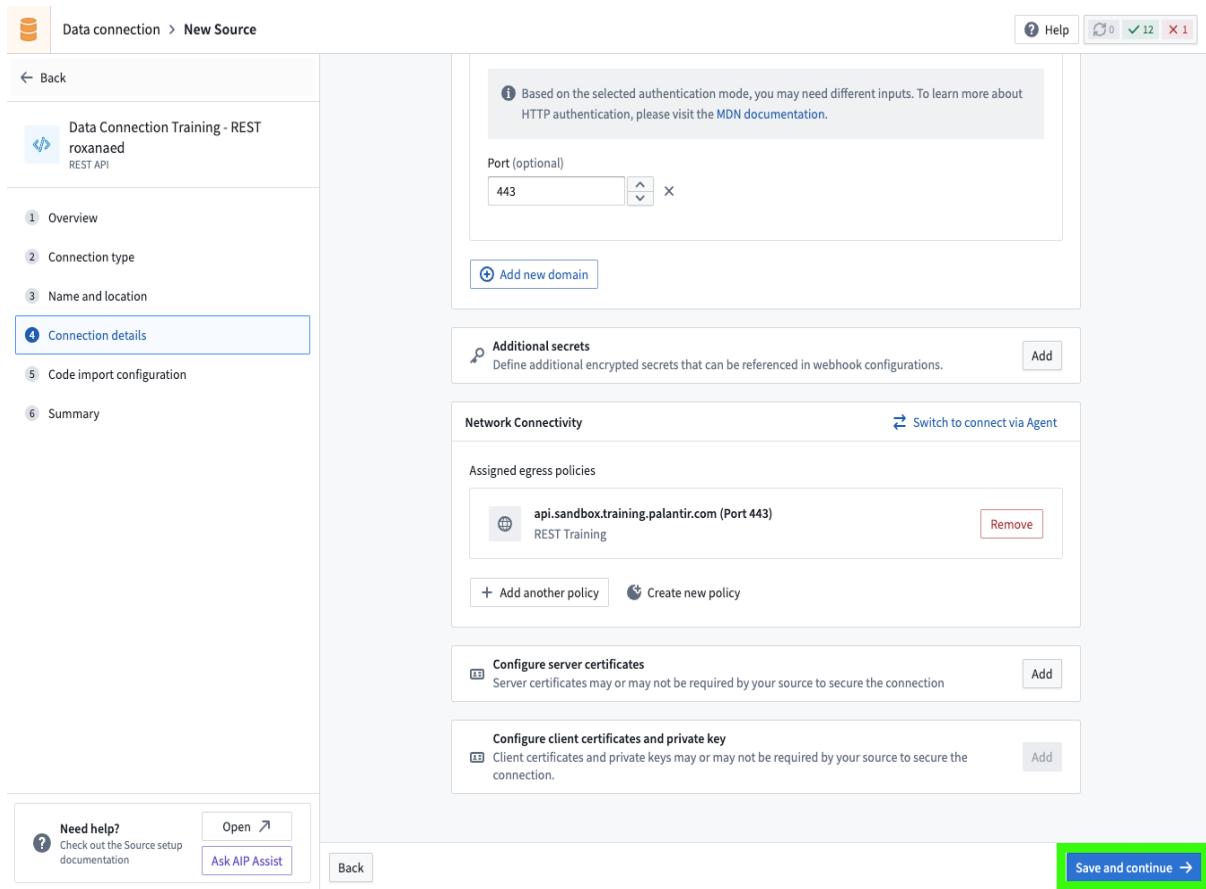
Search for organizations...

Palantir

Back **Save**

x

6. On the configuration screen, click **Save and continue** to move forward with the setup.



Configure Code Import Configuration

Note: If you see the 'Export Configuration' step before this section, leave as default and click Continue.

Step 1: Prepare the Source to be accessed in Code Repositories

To prepare the source for retrieving data from the REST APIs, you need to make it available for import into Code Repositories in Foundry.

1. Click on **Allow this source to be imported into code repositories** to enable this feature.
2. Leave the below selections as default, see image for reference.

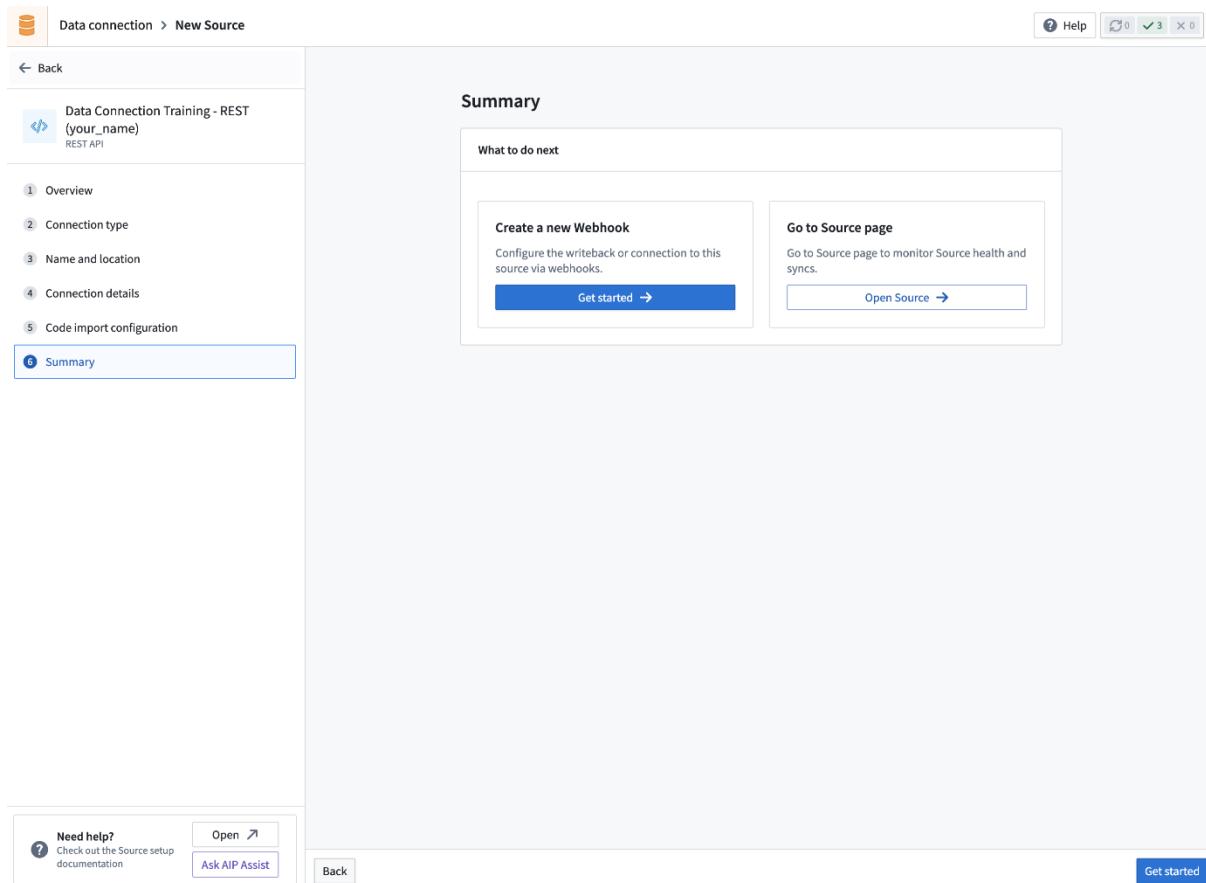
The screenshot shows the 'Code import configuration' page for a REST API source named 'Data Connection Training - REST API2 (your_name) REST API'. The left sidebar lists steps 1 through 7, with 'Code import configuration' highlighted. The main content area is titled 'Code import configuration' and contains several sections with toggle switches. The first section, 'Enable code imports', has a switch labeled 'Allow this source to be imported into code repositories' which is checked and highlighted with a green border. Below it is a descriptive text about allowing users to write custom code. The other sections include 'Allow the usage of source webhooks in code repositories', 'Allow this source to be imported into compute modules', 'Allow this source to be imported into pipelines for UDFs', and 'Allow this source to be imported into pipelines for virtual table inputs and outputs', each with their respective descriptions.

3. Write a unique API name, such as FirstLastnameRestApiSource (camel case).

The screenshot shows the 'API name' configuration page. The title is 'API name' and the subtitle is 'Serves as an alias to reference your source in code'. A text input field contains the value 'JSmithRestApiSource', which is highlighted with a green border. Below the input field is a warning message in a yellow box: '⚠️ Editing the API name will break existing references in code, which will have to be updated manually.' There is also an 'X' button to clear the input.

4. Click **Continue** to proceed.

5. You have now successfully created the REST source. We'll now proceed to access this source via an API call in Code Repositories.

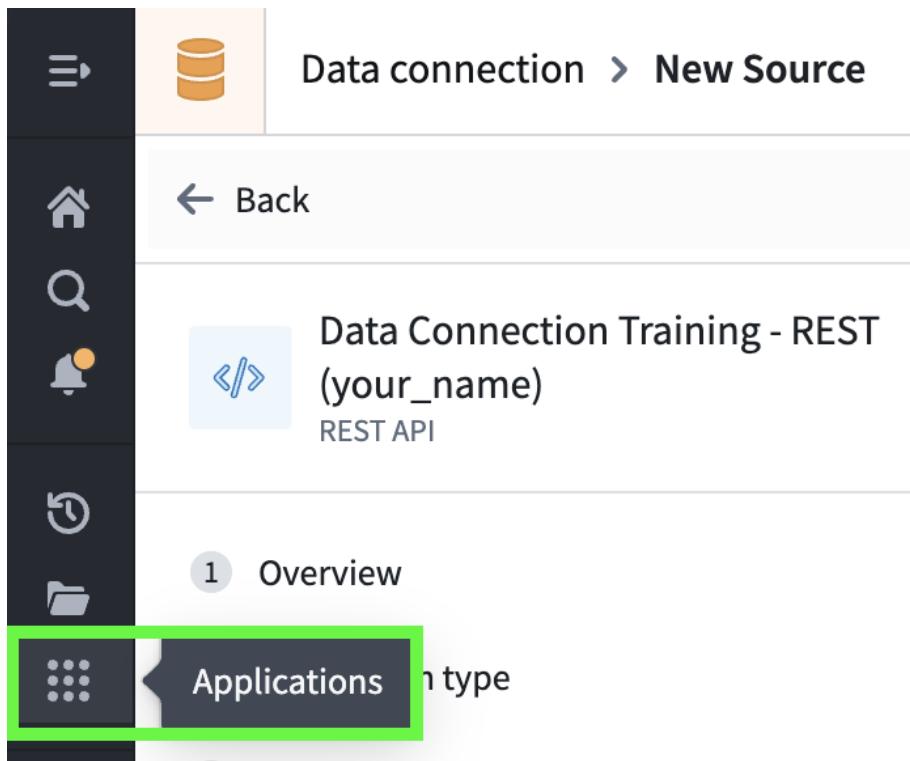


Create a Code Repositories Resource

Step 1: Configure a new Code Repositories file

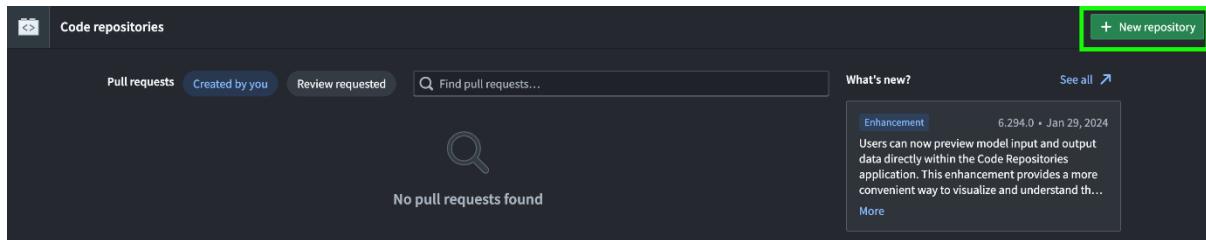
To retrieve data from REST APIs, you'll need to write a Foundry Transform, which requires a Code Repositories file to store your logic. Follow the below steps to create one.

1. Click on **Applications** in the Foundry menu bar.



2. Choose **Code Repositories** from the options presented. This action will take you to the Code Repositories landing page.

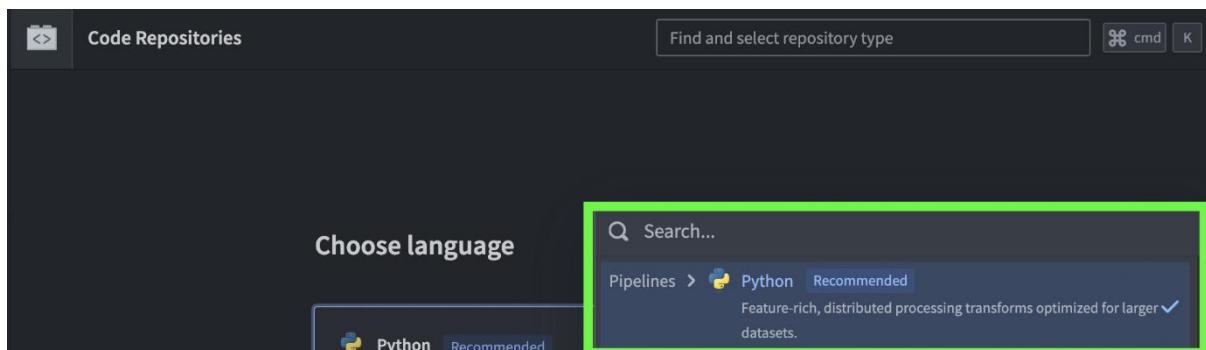
To create a new Code Repository, click on **+ New repository**.



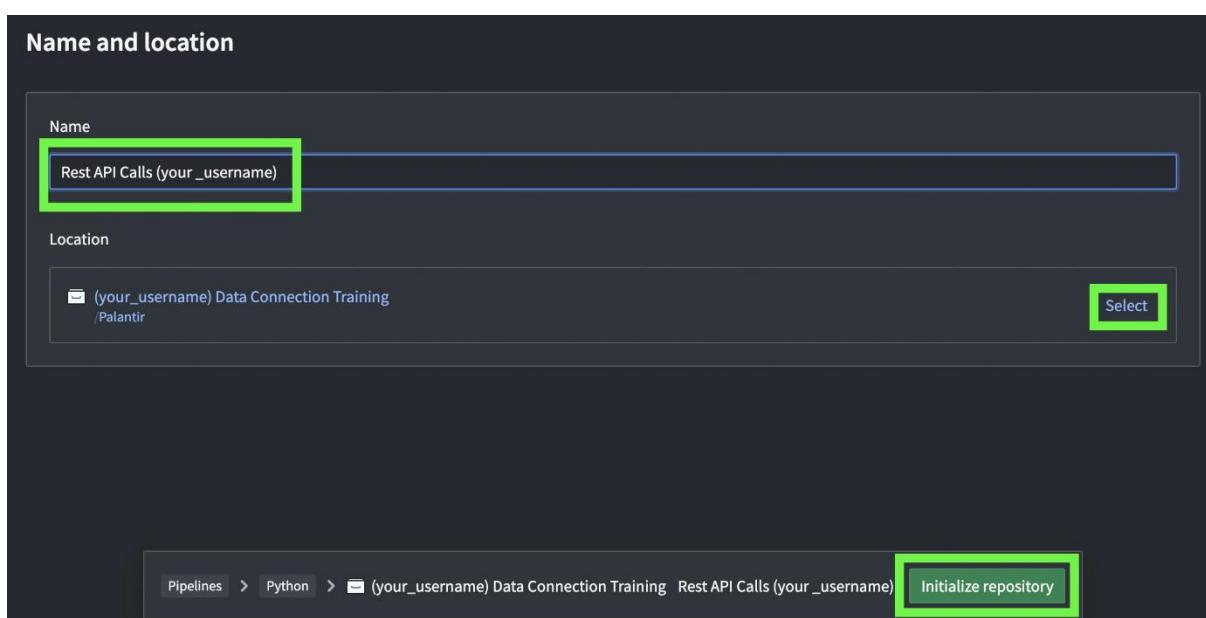
4. Click **Find and select repository type**



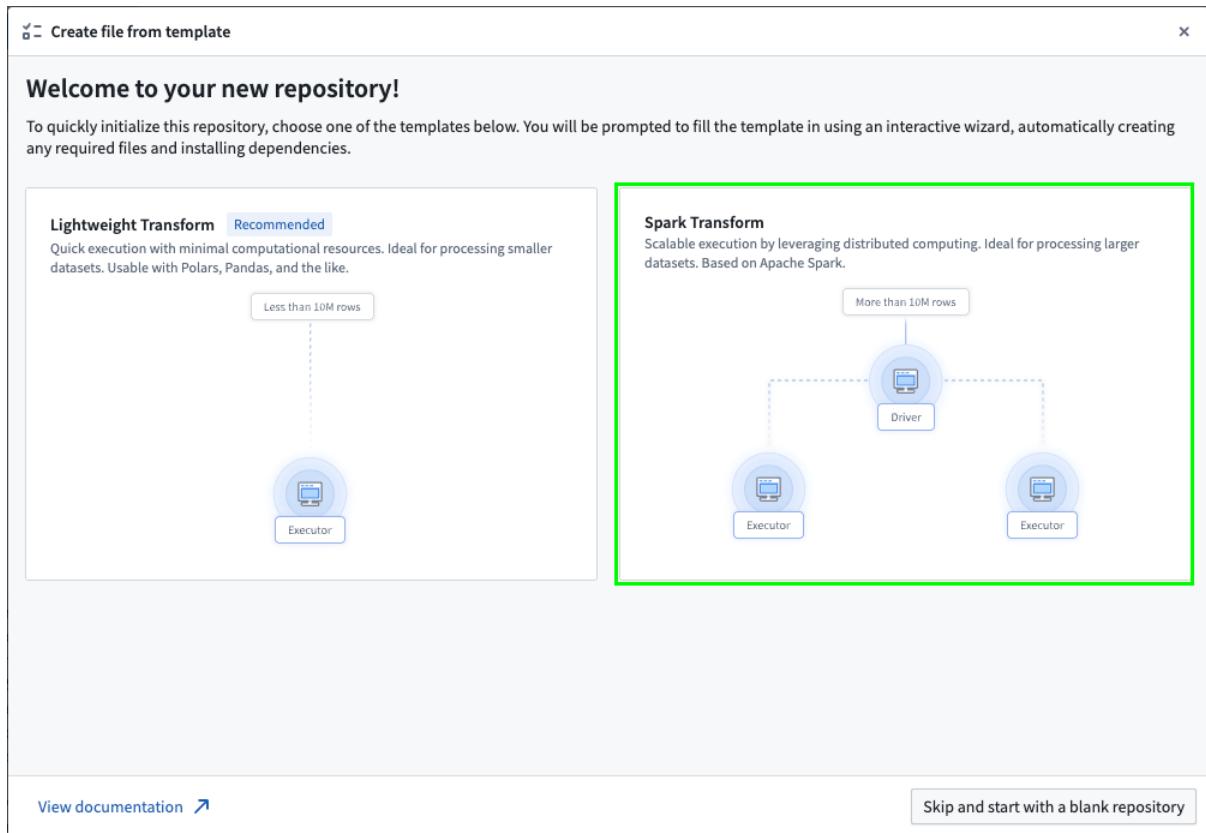
5. Select **Pipelines > Python**



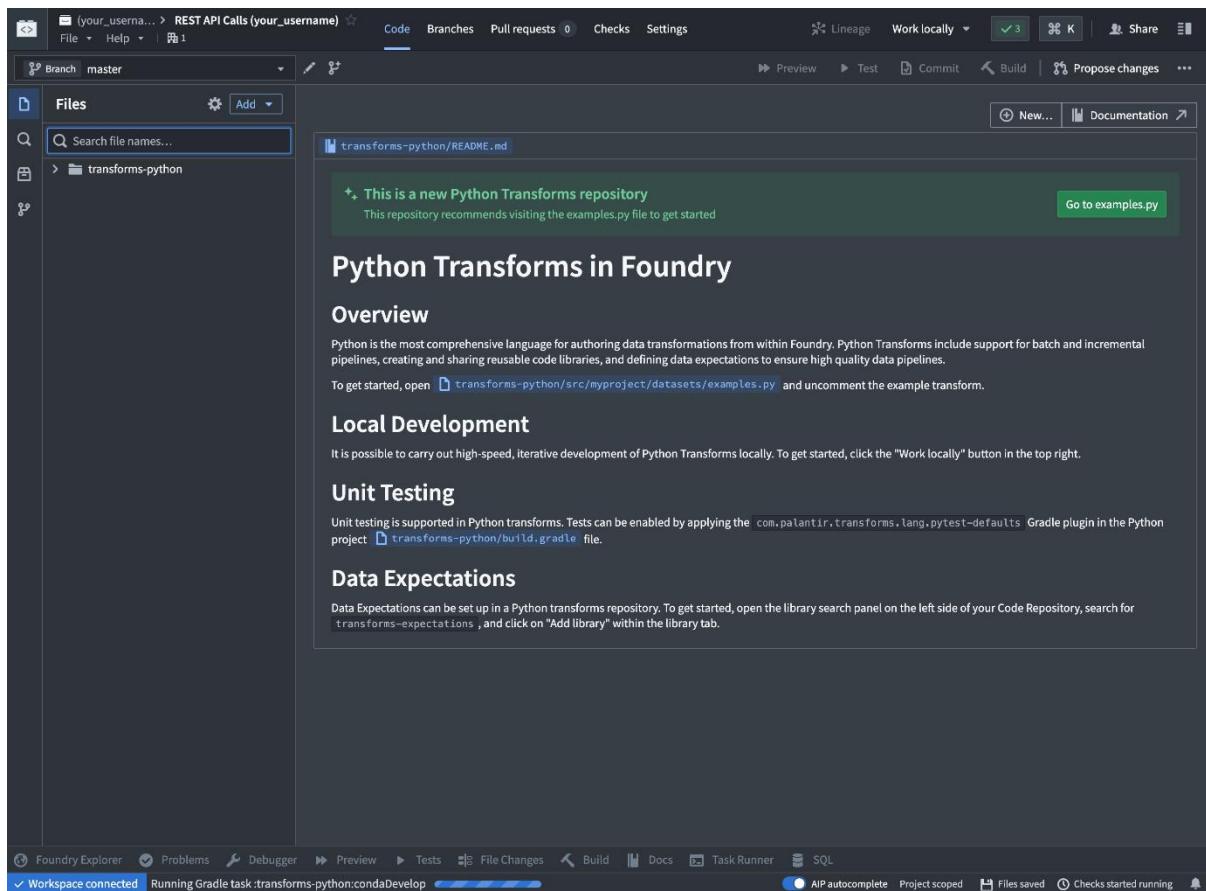
6. Name your new repository, for example, REST API Calls (your_username), replacing (your_username) with your actual username.
7. Select the location for your repository, which should be the same Project you have been using for the earlier steps.
8. Click on **Initialize repository** to create the repository.



9. Select the **Spark Transform** option for your repository.



10. Add an output variable in the bottom left of the modal. Set the Output dataset location in the same folder we have been working in.
11. Name the file "plants", click **Save**, and then click **Add** (leave the Parameter name unchanged.) This name and associated path will represent this dataset in code.
12. Now that we've configured the 'Outputs' section, click **Generate file**.
13. You can now view your new Code Repository.



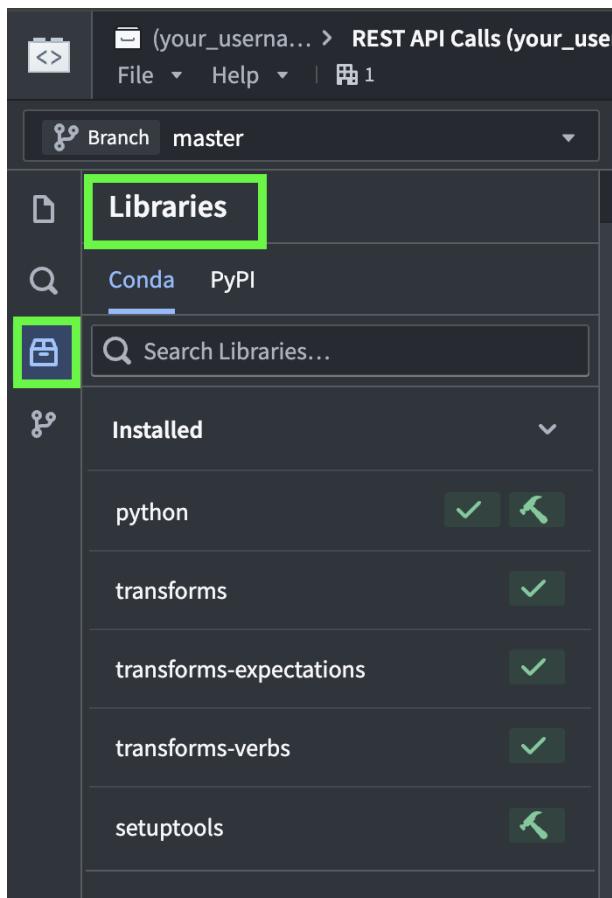
Note: If you see the following message: "A new commit has been made to this branch that may conflict with your uncommitted changes" at the top of the repository, click 'Merge in Changes' to merge the changes as soon as possible to prevent further issues.

Add External Transform Library

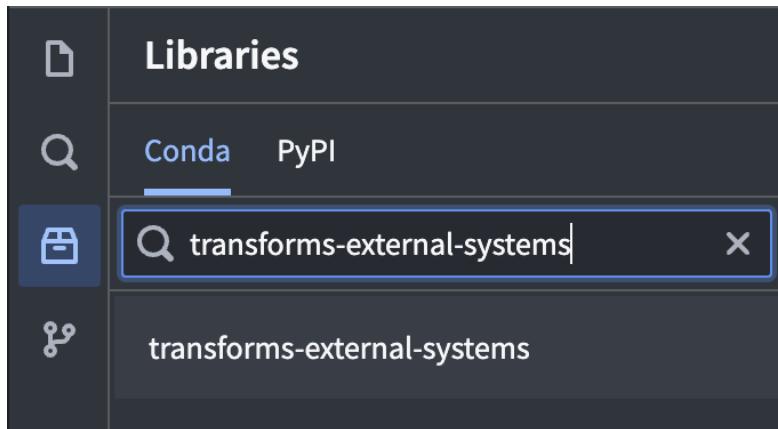
Step 1: Install library for accessing REST API connections

To configure your repository to allow interaction with the REST API, you'll need to install a necessary library.

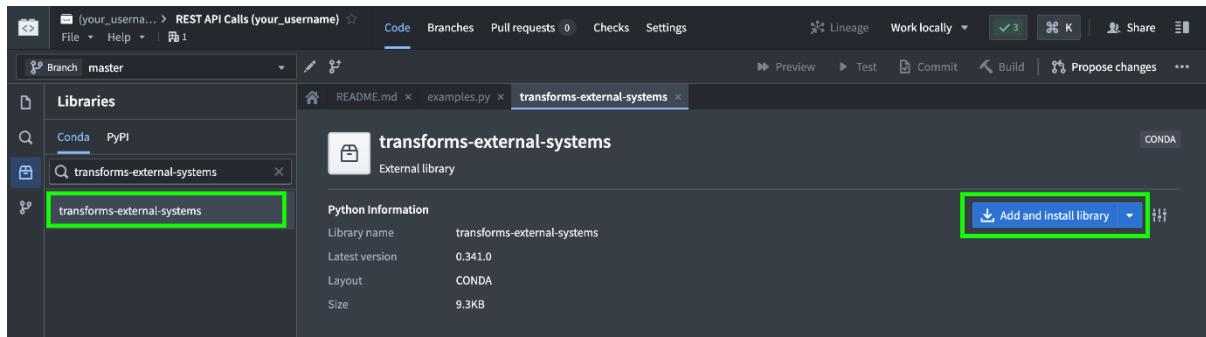
1. Navigate to the library settings from the left-side panel.



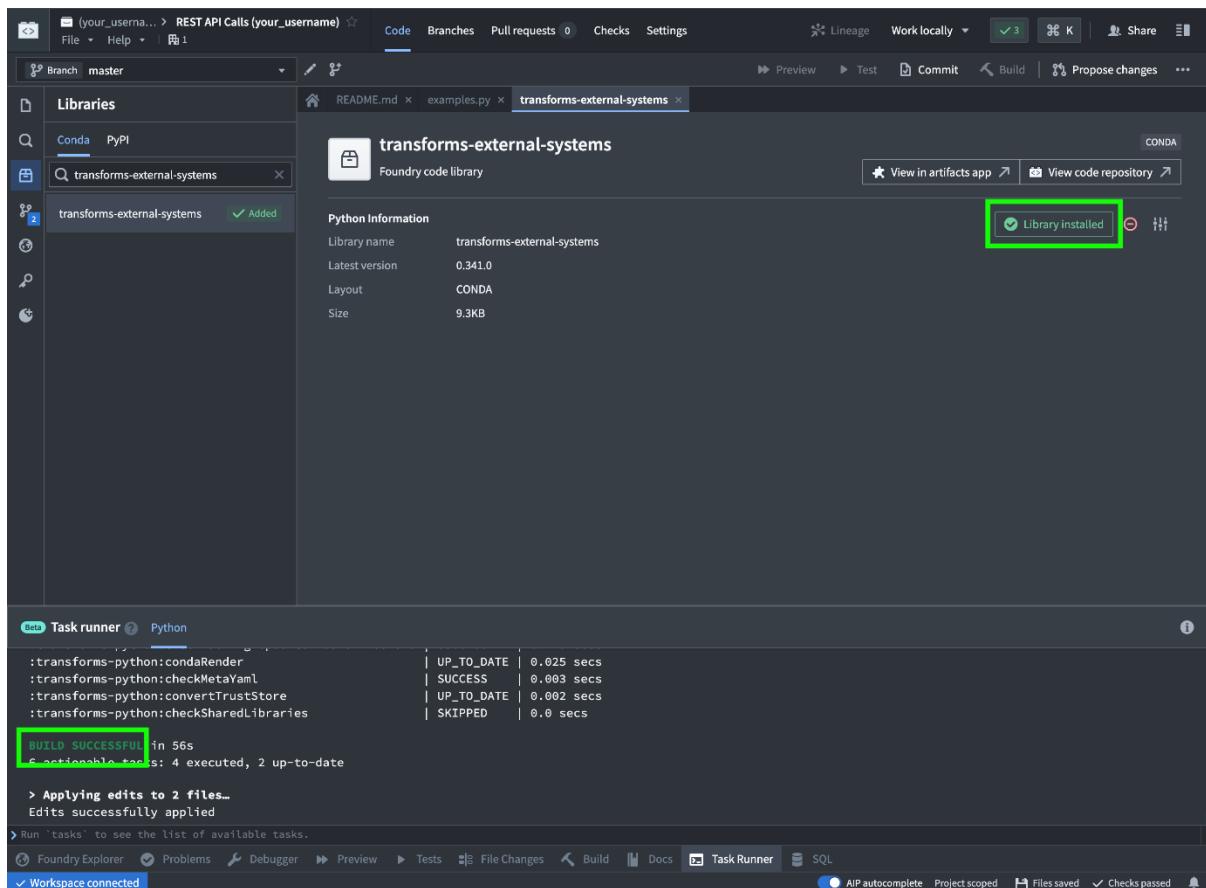
2. In the **Search Libraries** box, type transforms-external-systems.



3. Click on the first search result to open the library details pane, then click **Add and install library**.



4. Wait for the library installation to complete. You'll know it's done when you see the **BUILD SUCCESSFUL** output in the Task runner window and a **Library installed** status on the library details pane.



Import Rest API Source

Step 1: Find and import your Rest API Source

1. Click the external systems icon (globe) on the left.
2. Click the **Add** button and select **Existing connections**
3. Search for the Source you created earlier, select it, and then click **Confirm selection**.
4. Your source is now added to the Code Repository and can be utilized in code.

External systems ?

Add ▾

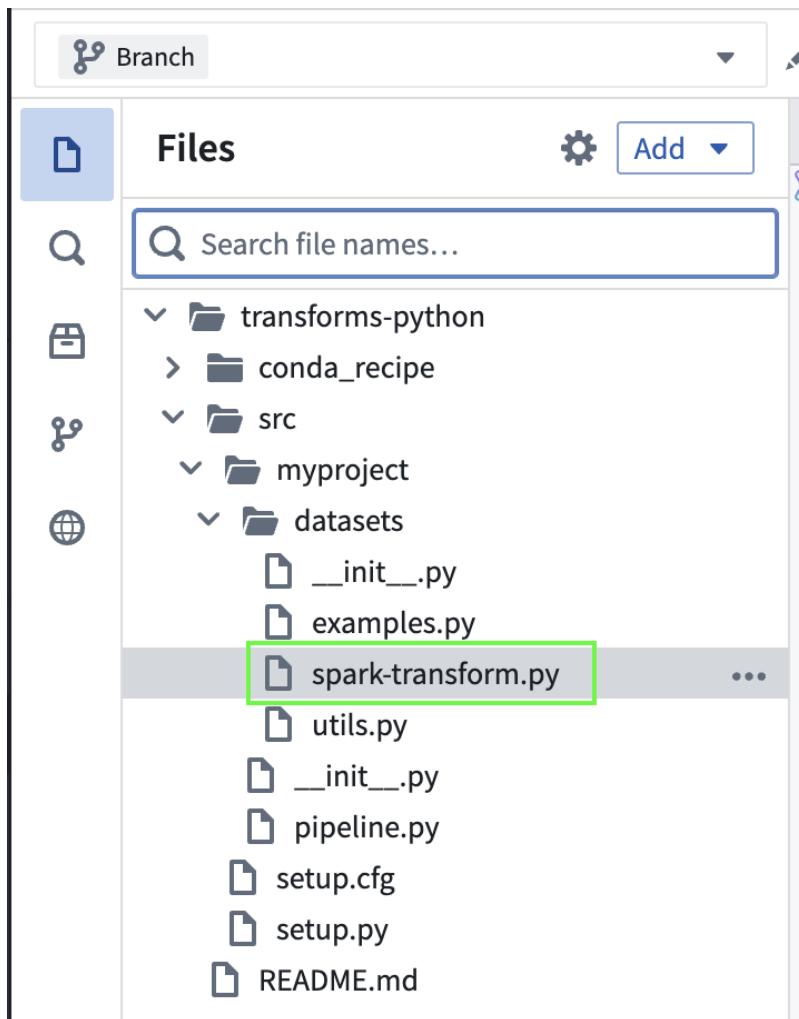
Data Connection Training - REST API
REST API

Configure Code for Accessing Data

Step 1: Add code to access data via API

You are now ready to write the code that will retrieve data from the API.

1. Open the **Files** panel and select the spark-transform.py file.



Copy down the path of the Output dataset as you will need to reference it shortly. Depending on your working project and folder, it should look like this:

/Organization/(your_username) Data Connection Training/plants

```

Branch master
Files
spark-transform.py x
1 from transforms.api import transform, TransformContext, Output, TransformOutput, configure, ComputeBackend
2
3
4 @configure(profile=["EXECUTOR_MEMORY_MEDIUM", "EXECUTOR_MEMORY_OFFHEAP_FRACTION_HIGH"], backend=ComputeBackend.VELOX)
5 @transform(output="/Organization/(your_username) Data Connection Training/TARGET_DATASET_PATH")
6 def compute(ctx: TransformContext, output: TransformOutput):
7     df_custom = ctx.spark_session.createDataFrame([{"Hello": "World"}], schema=["phrase"])
8     # Remove comment to write contents of df_custom to output on build
9     # output.write_dataframe(df_custom)
10

```

The screenshot shows a code editor with the 'spark-transform.py' file open. The code imports from 'transforms.api' and defines a function 'compute' that creates a DataFrame with one row ('Hello' and 'World') and writes it to a specified output path. A comment in the code specifies the output path as '/Organization/(your_username) Data Connection Training/TARGET_DATASET_PATH'.

2. Remove all the code in this file and replace it with the following code snippet:

```

from transforms.api import transform_df, Output
from transforms.external.systems import external_systems, Source

```

```

@external_systems(
    rest_api_source=Source("{{SOURCE RID}}")
)
@transform_df(
    Output("{{OUTPUT_PATH}}")
)
def compute(rest_api_source, ctx):
    url = rest_api_source.get_https_connection().url + "/secretPlants"
    auth_token = rest_api_source.get_secret("additionalSecretSecret")
    # client is a pre-configured Session object from Python `requests` library.
    client = rest_api_source.get_https_connection().get_client()

    headers = {
        "Authorization": auth_token
    }
    response = client.get(url, headers=headers)
    response.raise_for_status()
    json_content = response.json()

    # the data fetched and parsed from the external system are written to the output dataset
    return ctx.spark_session.createDataFrame(json_content)

```

4. Replace the <>SOURCE RID<> with the rid of your source. This can be found in the external system tab (globe).
 5. Replace <>OUTPUT_PATH<> with your dataset path, for example /Organization/(your_username) Data Connection Training/plants.
-

After making these replacements, your code should be configured like the screenshot below to retrieve data from the REST API and save it to the specified dataset. Note that you may need to indent your code from line 11 downward (as pictured) to avoid syntax errors.

```

sparktransform.py x
1  from transforms.api import transform_df, Output
2  from transforms.external.systems import external_systems, Source
3
4  Replace paths with RIDS
5  @external_systems(
6      rest_api_source=Source("Data Connection Training - REST API (371537d00461")
7  )
8  @transform_df(
9      output="dataset warning (.....)/path/to/output/dataset"
10 )
11 def compute(rest_api_source, ctx):
12     url = rest_api_source.get_https_connection().url + "/secretPlants"
13     auth_token = rest_api_source.get_secret("additionalSecretSecret")
14     # client is a pre-configured Session object from Python 'requests' library.
15     client = rest_api_source.get_https_connection().get_client()
16
17     headers = {
18         "Authorization": auth_token
19     }
20     response = client.get(url, headers=headers)
21     response.raise_for_status()
22     json_content = response.json()
23
24     # the data fetched and parsed from the external system are written to the output dataset
25
26     return ctx.spark_session.createDataFrame(json_content)

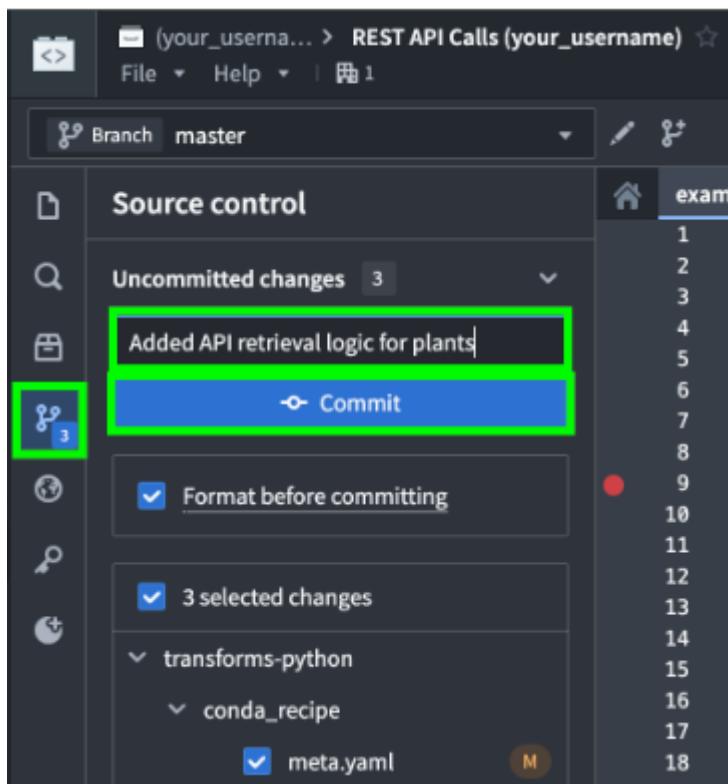
```

Commit and Build Your Dataset

Step 1: Build your dataset

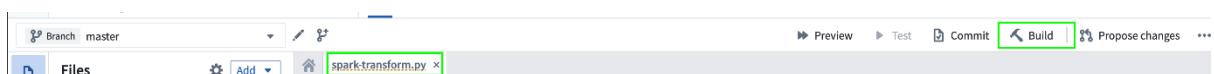
To apply the changes you've made to the code, you need to commit them.

1. Navigate to the **Source Control** panel on the left and enter a commit message, such as "Added API retrieval logic for plants," in the provided field.
2. Click **Commit** to save your changes to the repository.

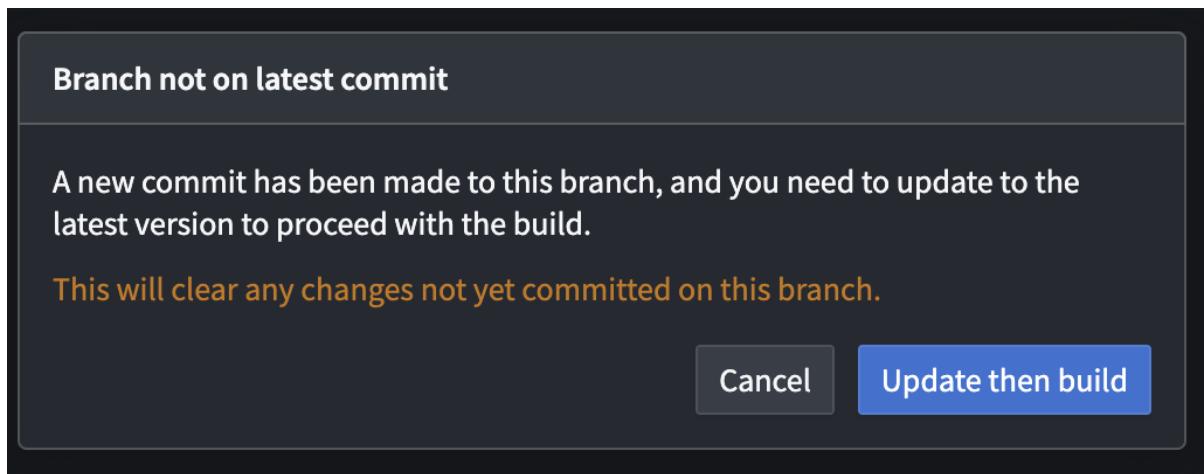


Now, let's build the dataset. You need to ensure that the spark-transform.py file is active in your editor.

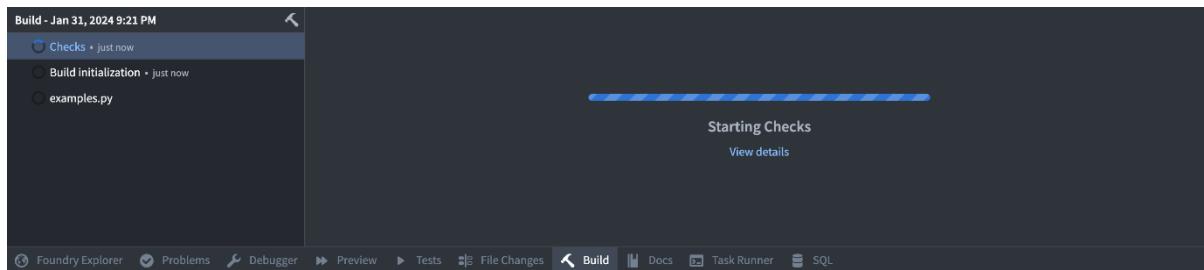
3. Click **Build** to start the data retrieval process and store it in the dataset.



4. If you see a message prompting you to update, click on **Update then build** to continue.



5. The build's progress will be displayed in the **Build** tab at the bottom of the screen. Keep in mind that the initial build may take a little longer since Foundry is packaging and deploying your logic for the first time. Subsequent builds will typically be faster.



6. After a few minutes, the build should complete and you will be able to view the data retrieved from the REST API.

	color	id	name	sunlight	type
	String	Long	String	String	String
1	silver		Glimmerleaf	partial shade	flower
2	blue		Dazzlepetal	full sun	flower
3	green		Aether Fern	indirect light	houseplant
4	rainbow		Prism Cactus	full sun	succulent
5	green		Whispering Palm	full sun	tree
6	blue		Cosmic Snake Plant	low light	houseplant
7	green		Stardust Echeveria	full sun	succulent
8	green		Aurora Daffodil	full sun	flower
9	blue		Celestial Oak	full sun	tree
10	green		Mystic Ficus	indirect light	housepl...
..

7. To view the dataset, click on the three dots menu next to the plants dataset. Select **Open** to open the full dataset view.

Build - Today at 4:43 PM

- ✓ Checks • 1 hour ago
- ✓ Build initialization • 59 minutes ago
- ✓ spark-transform.py
 - ✓ dataset

Build finished Preview

Open

View build details

...	1	silver
	2	blue
	3	green
	4	rainbow

8. Now you can see that the data has been successfully retrieved from the REST API and is now ready for downstream use in pipelining tools and the Ontology.

(your_username... > plants ★

File • Help • 1 master •

Preview History Details Health Compare

Analyze in Contour Explore pipeline All actions Build

plants Report issue

About Columns Schedules

Enter description...

Updated 52 seconds ago by

Created Jan 31, 2024, 9:19 PM by Foundry

Location /Palantir/(your_username) Data Connection...

Type Dataset

Size 5 columns + 20 rows + 3 files + 354KB

Updated via REST API Calls (your_username)

Show more

Tags Add tags

Health Checks

View details

Inputs Explore data lineage

No inputs

Custom metadata Add custom metadata

plants

Showing all 20 rows 5 columns Search columns...

	color	id	name	sunlight	type
	String	Long	String	String	String
1	silver		1 Glimmerleaf	partial shade	flower
2	blue		2 Dazzlepetal	full sun	flower
3	green		3 Aether Fern	indirect light	houseplant
4	rainbow		4 Prism Cactus	full sun	succulent
5	green		5 Whispering Palm	full sun	tree
6	golden		6 Sparkle Rose	full sun	flower
7	purple		7 Dreamberry Tree	full sun	tree
8	violet		8 Nebula Orchid	indirect light	flower
9	green		9 Glowvine	low light	houseplant
10	orange		10 Meteor Cactus	full sun	succulent
11	white		11 Moonflower	partial shade	flower
12	green		12 Star Bamboo	partial shade	tree
13	multicolor		13 Galaxy Ivy	partial shade	houseplant
14	yellow		14 Solar Sunflower	full sun	flower
15	purple		15 Twilight Maple	full sun	tree
16	blue		16 Cosmic Snake Plant	low light	houseplant
17	green		17 Stardust Echeveria	full sun	succulent
18	green		18 Aurora Daffodil	full sun	flower
19	blue		19 Celestial Oak	full sun	tree
20	green		20 Mystic Ficus	indirect light	houseplant