A picture containing text, sign, clock

Description automatically generated

Adobe Experience Platform

Data Pipeline

# Lab Overview

Ingest Orders historical data into AEP in batch mode using Data Landing Zone (this lab) and stream orders live data using Streaming Source HTTP API (next lab).

This lab will introduce some complex data transformations and JSON data processing

**Expected time: 60 minutes**

# Learning Objectives

What should you walk away with after taking this Lab?

* Understand ingestion of JSON data
* Use Data Landing Zone as a source
* Use Data Prep to Map the non-XDM data to XDM
* Scheduling batch workflows

# Lab Tasks – Orders – Historical data

In this exercise, we will load the Orders data from Data Landing Zone to AEP Data Lake and Profile.

**Pre-requisites**

1. Orders JSON file uploaded in the Azure ADLS Directory
2. Orders Schema and Dataset are already created

**Steps**

Go to Adobe Experience Platform à **Sources** à **Catalog** à **Cloud storage**. Click on **Setup** / **Add Data** for the Data Landing Zone.

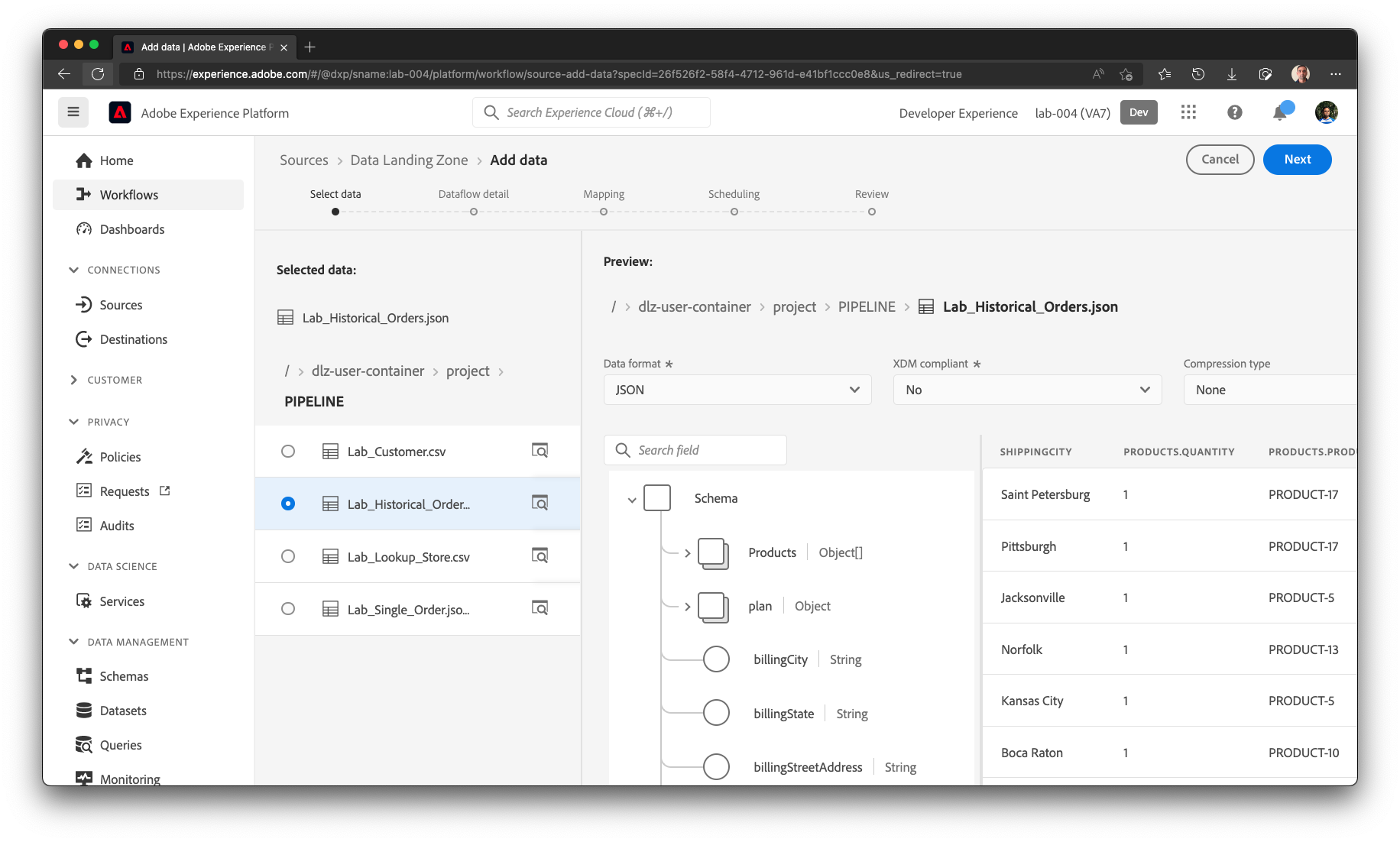
|  |  |
| --- | --- |
| Tip | If at least one connection exists for that source, you will see “**Add data**” as the default action. If no connections exist for that source, you will see “**Setup**” as the default action. |

Graphical user interface, application, Teams

Description automatically generated

## Select Source data

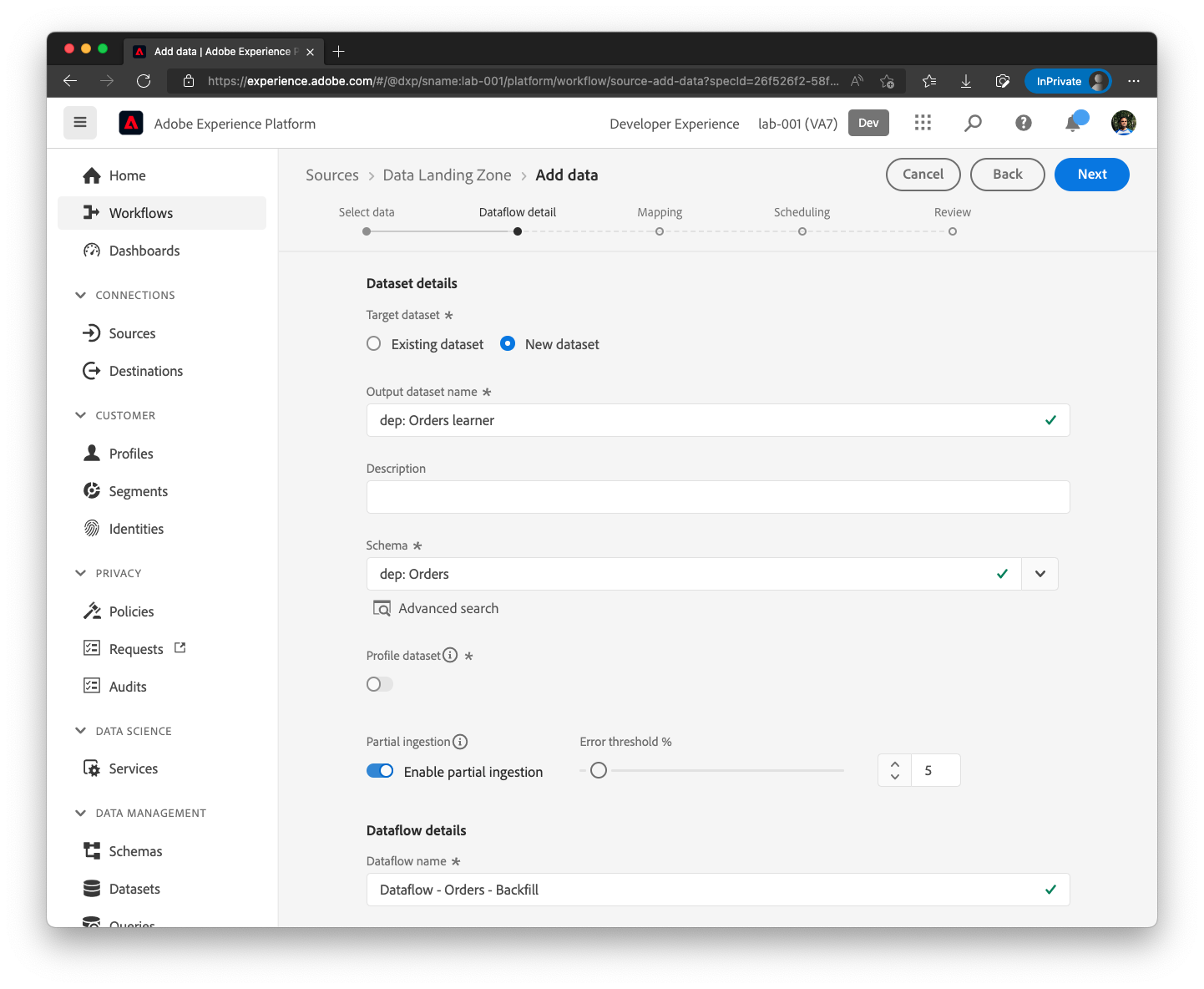
In the **Select data** screen, navigate to **dlz-user-container** ⟶ **project** ⟶ **PIPELINE** and choose **Lab\_Historical\_Orders.json**. On the right-hand side, choose data format as **JSON.** Preview of the selected file is automatically displayed. Click **Next**.



## Define the Target dataset

In the **Dataset details** step, choose ***New dataset*** option. Name the output dataset as **dep: Orders learner**. Set the schemas as **dep: Orders**.

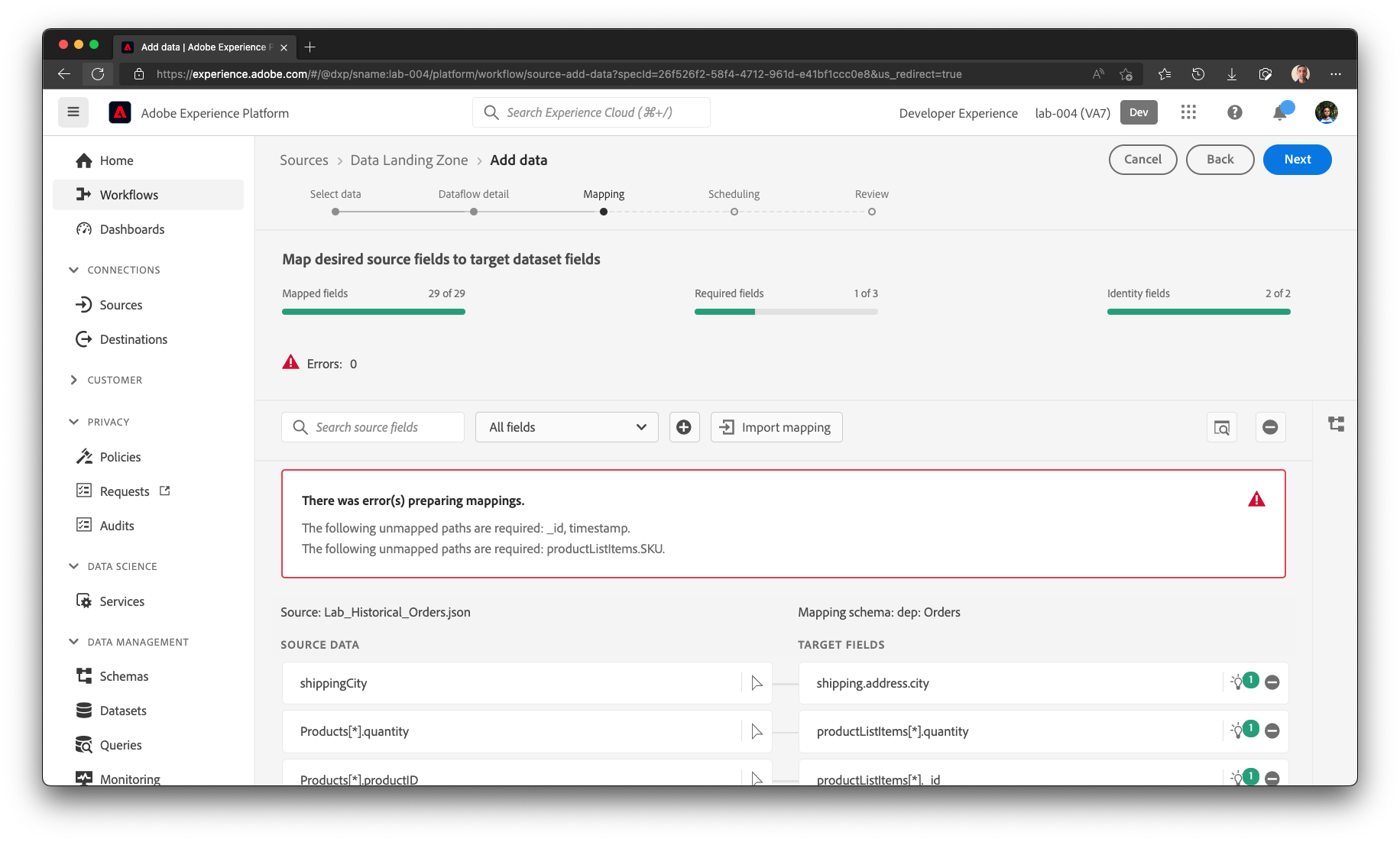
**Turn OFF** the **Profile dataset** option. Enable **Partial ingestion**. Name the Dataflow as **Dataflow – Orders – Backfill**.



|  |  |
| --- | --- |
| Tip | When Partial Ingestion is enabled, Error diagnostics are automatically enabled and hence the toggle box will disappear. |

## Data Prep / Transformation

In the Data Prep (Mapping) step, ML Recommendations will automatically map most attributes. However, you will also see several errors. The initial screen will look similar to below.



Please ensure the mappings are accurate. The initial recommendations loaded may have the following **invalid** mappings

## Remove the following mappings

* Mapping to XDM attribute **billing.address.lastVerifiedDate**



## Create Calculated Fields

### Mapping to \_id

Create the following calculated fields by clicking on Graphical user interface, application

Description automatically generated with medium confidence and click on Add Calculated Field.

Graphical user interface, text, application, chat or text message

Description automatically generated

Write the following expression and click Preview

|  |
| --- |
| concat(orderID, "-", lastOrderStatusUpdate) |

Calculated field will look similar to this. Click Save to save the calculated field.

Graphical user interface, application

Description automatically generated

Map the calculated field to **\_id**



### Mapping to order.\_dxp.acqSource

Create the following calculated field by clicking on Graphical user interface, application

Description automatically generated with medium confidence and click on Add Calculated Field.

Graphical user interface, text, application, chat or text message

Description automatically generated

Write the following expression and click Preview

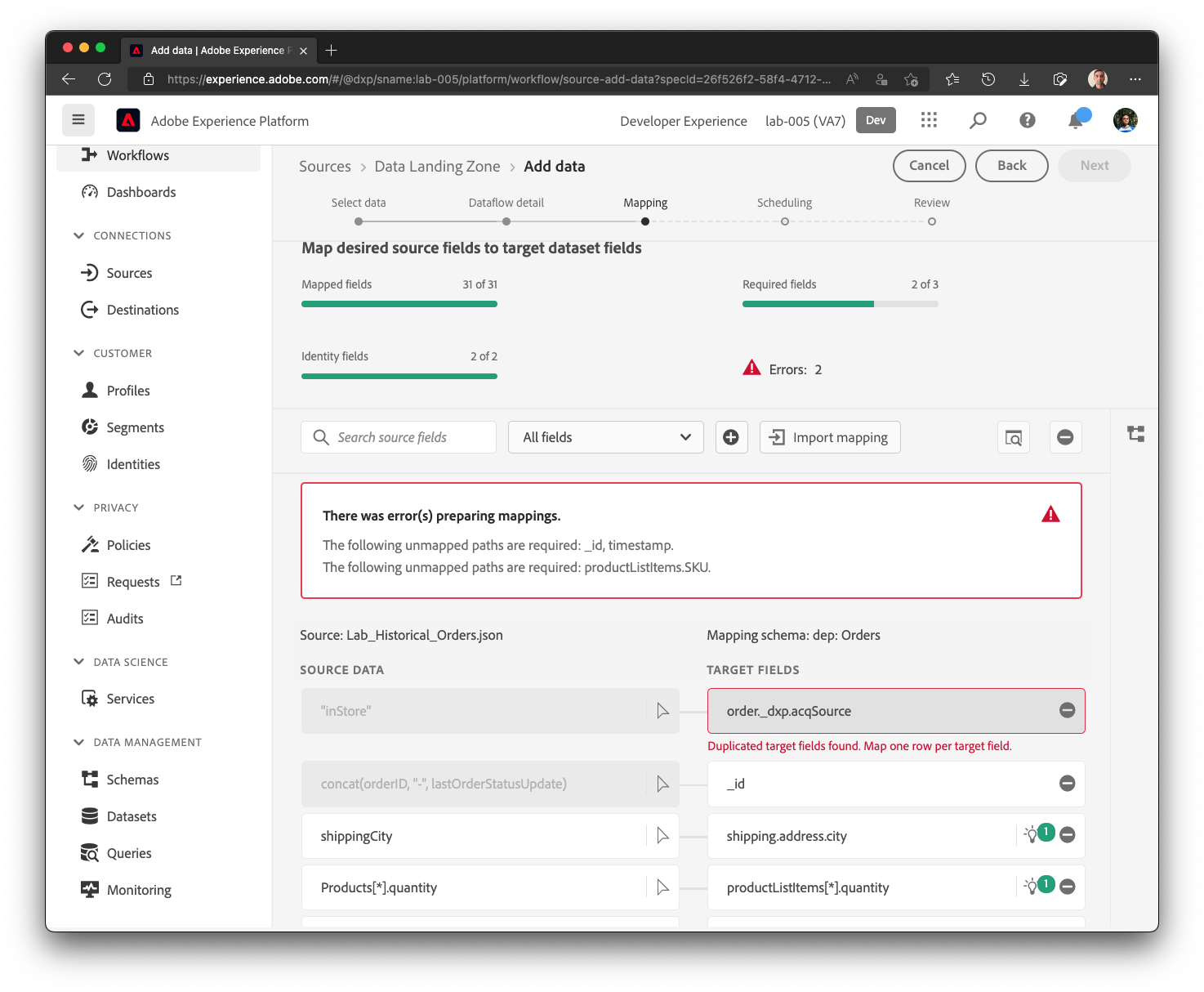
|  |
| --- |
| “inStore” |

Calculated field will look similar to this. Click Save to save the calculated field.

Graphical user interface

Description automatically generated

Map the calculated field to **order.\_dxp.acqSource**. The Data Prep screen will complain there is a duplicate mapping.



Remove the other duplicate mapping for order.\_dxp.acqSource by clicking on the (-) symbol next to the row.

Graphical user interface, diagram

Description automatically generated

## Add new pass-through mappings

Add the following passthrough mappings by clicking New field type (A picture containing text

Description automatically generated) and Add new field for each row here. Some may already be present due to ML Recommendations.

| Source Column | XDM Column |
| --- | --- |
| orderStatus | eventType |
| lastOrderStatusUpdate | Timestamp |
| products[\*] | productListItems[\*] |
| products[\*].productID | productListItems[\*].SKU |

## Product List Items

We will remove the mappings to XDM attributes **productListItems[\*].quantity** and **productListItems[\*].currencyCode**. Change the other mappings to **productListItems[\*]**

|  |  |
| --- | --- |
| NOTE | You will have to manually type in [\*] in the target attributes after making a selection for **Products** source object array or **productListItems** XDM attribute. |

|  |  |  |
| --- | --- | --- |
| Source Column | XDM Column | Action |
| products[\*] | productListItems[\*] | Add |
| products[\*].productID | productListItems[\*].SKU | Add |
| products[\*].productID | productListItems[\*].\_id | No change |
| products[\*].make | productListItems[\*].\_dxp.make | Change |
| products[\*].model | productListItems[\*].\_dxp.model | Change |
| products[\*].price | productListItems[\*].priceTotal | No change |
| products[\*].quantity | productListItems[\*].quantity | Remove |
| products[\*].currencyCode | productListItems[\*].currencyCode | Remove |

The resultant mappings for ProductListItems[\*] should look like this.

Graphical user interface, application

Description automatically generated



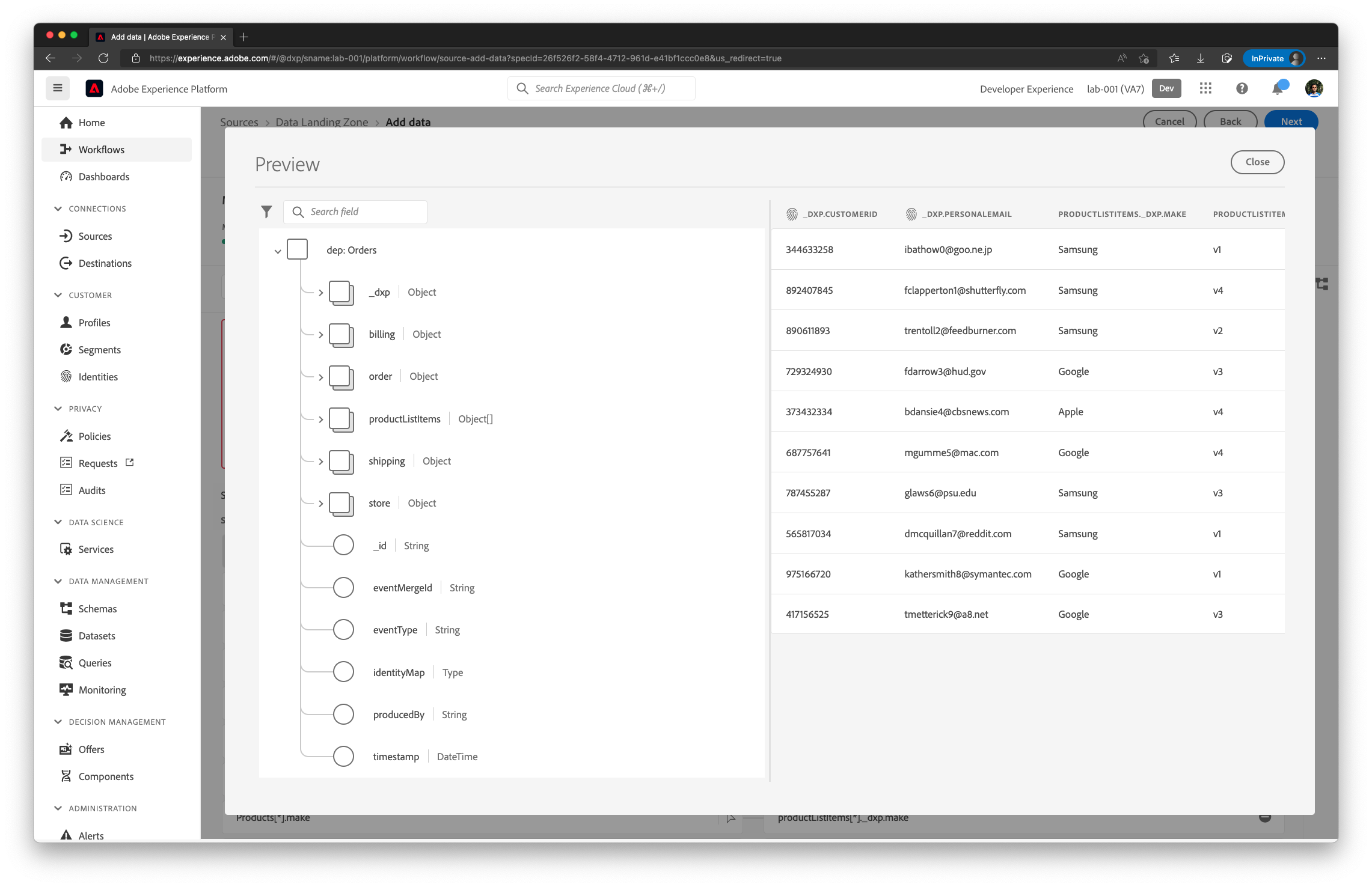
## Final Mapping Set

The final mapping set should look like this:

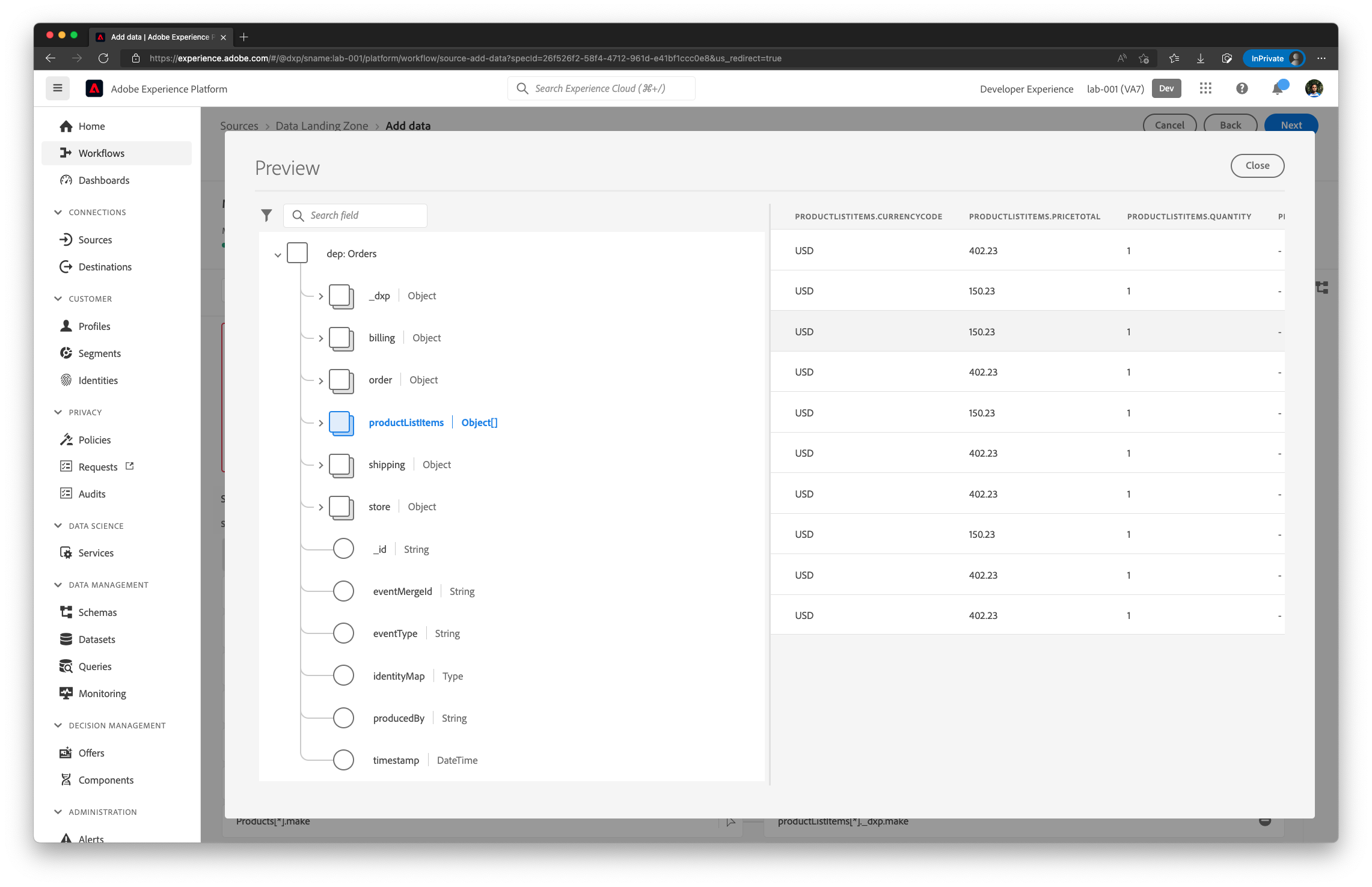
|  |  |
| --- | --- |
| Source Column | XDM Column |
| orderStatus | eventType |
| lastOrderStatusUpdate | timestamp |
|  |  |
| orderID | order.orderID |
| orderDate | order.orderDate |
| orderTotal | order.priceTotal |
|  |  |
| paymentType | order.payment.paymentType |
| paymentAmount | order.payment.paymentAmount |
| paymentCurrencyCode | order.payment.currencyCode |
| paymentTransactionID | order.payment.transactionID |
|  |  |
| plan.ID | order.\_dxp.plan.planID |
|  |  |
| customerID | \_dxp.customerID |
| personalEmail | \_dxp.personalEmail |
|  |  |
| storeID | store.storeID |
|  |  |
| shippingStreetAddress | shipping.address.street1 |
| shippingCity | shipping.address.city |
| shippingState | shipping.address.state |
| shippingZip | shipping.address.postalCode |
| shippingMethod | shipping.shippingMethod |
| shippingAmount | shipping.shippingAmount |
| shippingDestination | shipping.shippingDestination |
|  |  |
| billingStreetAddress | billing.address.street1 |
| billingCity | billing.address.city |
| billingState | billing.address.state |
| billingZip | billing.address.postalCode |
|  |  |
| products[\*] | productListItems[\*] |
| products[\*].productID | * productListItems[\*].\_id * productListItems[\*].SKU |
| products[\*].make | productListItems[\*].\_dxp.make |
| products[\*].model | productListItems[\*].\_dxp.model |
| products[\*].quantity | productListItems[\*].quantity |
| products[\*].price | productListItems[\*].priceTotal |
| products[\*].currencyCode | productListItems[\*].currencyCode |
|  |  |
| Calculated Fields | |
| concat(orderID, "-", lastOrderStatusUpdate) | \_id |
| "inStore" | order.\_dxp.acqSource |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Preview the data

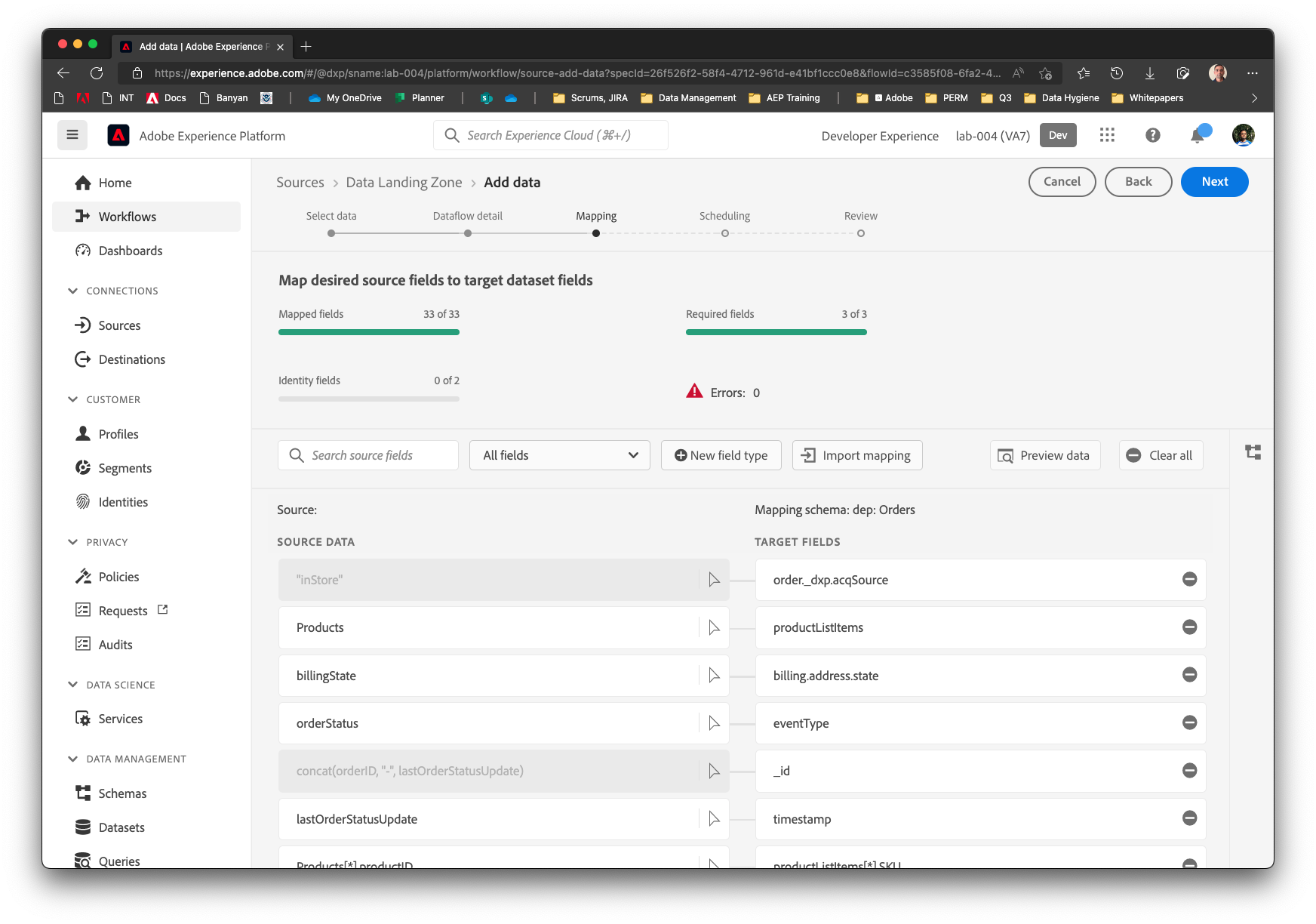
Preview the mapping output. Scroll through all the attribute to ensure there is no red exclamation next to any of the attributes on the right hand side. Preview will look similar to the below screenshot



On the left hand side navigation of the Preview, select **productListItems** object array. Right hand side will update to show only the attributes in that object array. Notice that **productListitems.currencyCode** and **productListitems.quantity** are automatically populated (even after removing the mappings). This happens because **productListItems** as a parent object is mapped.



Completed mapping will look similar to the following screenshot



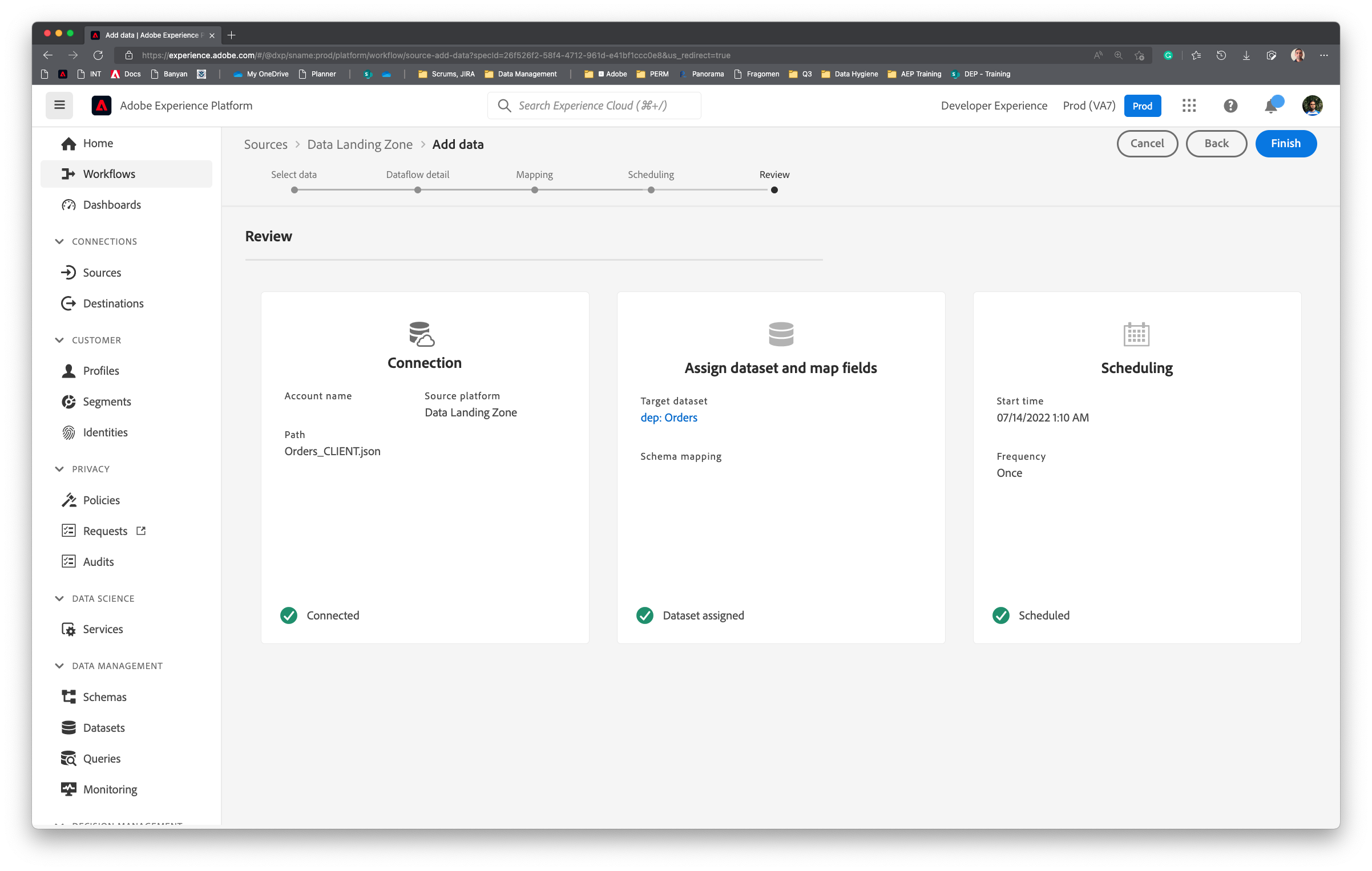
## Schedule

Set the schedule to run every 30 days by setting the Frequency as **Day** and Interval as **30**.

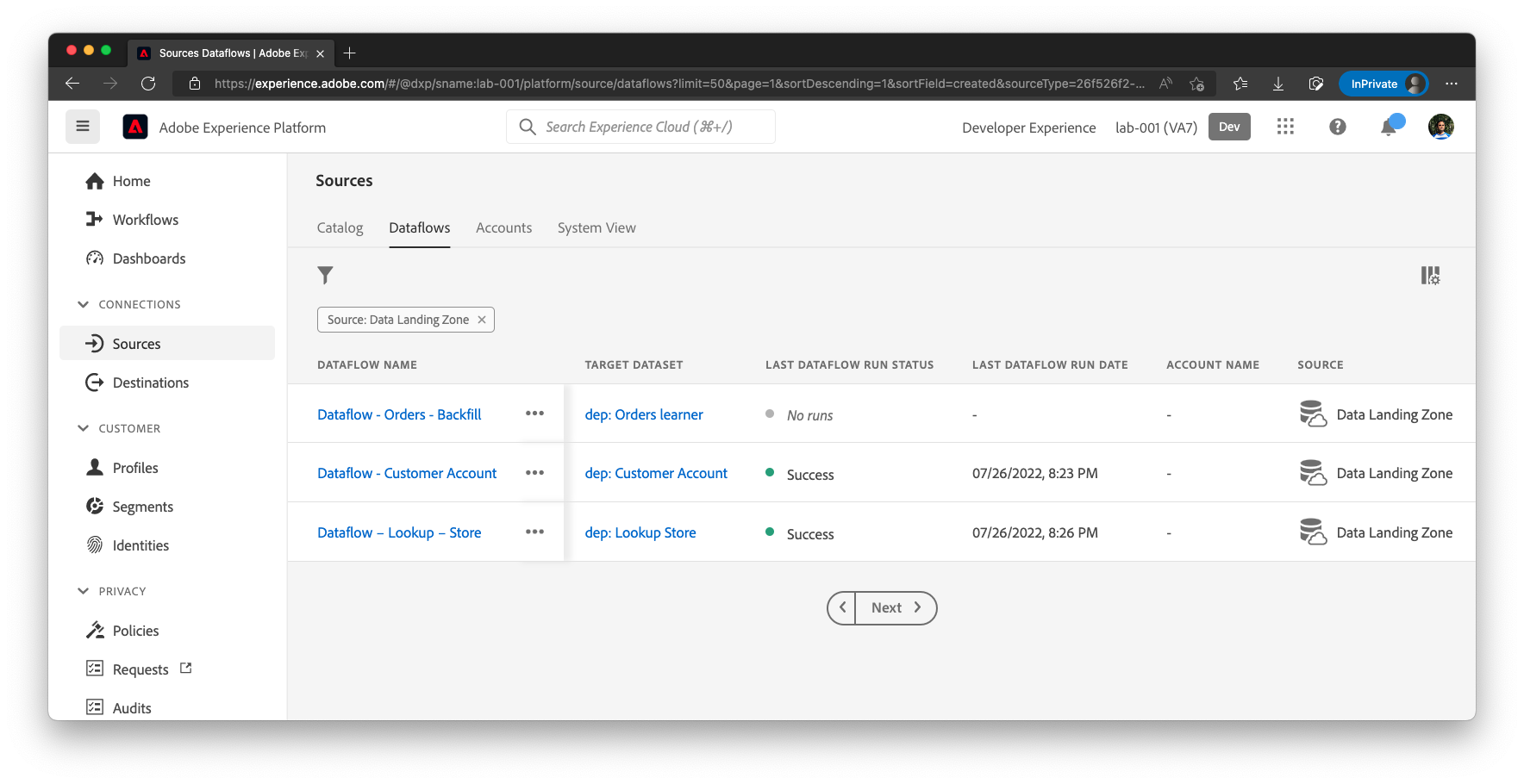
Graphical user interface, application

Description automatically generated

Review the flow and click Finish

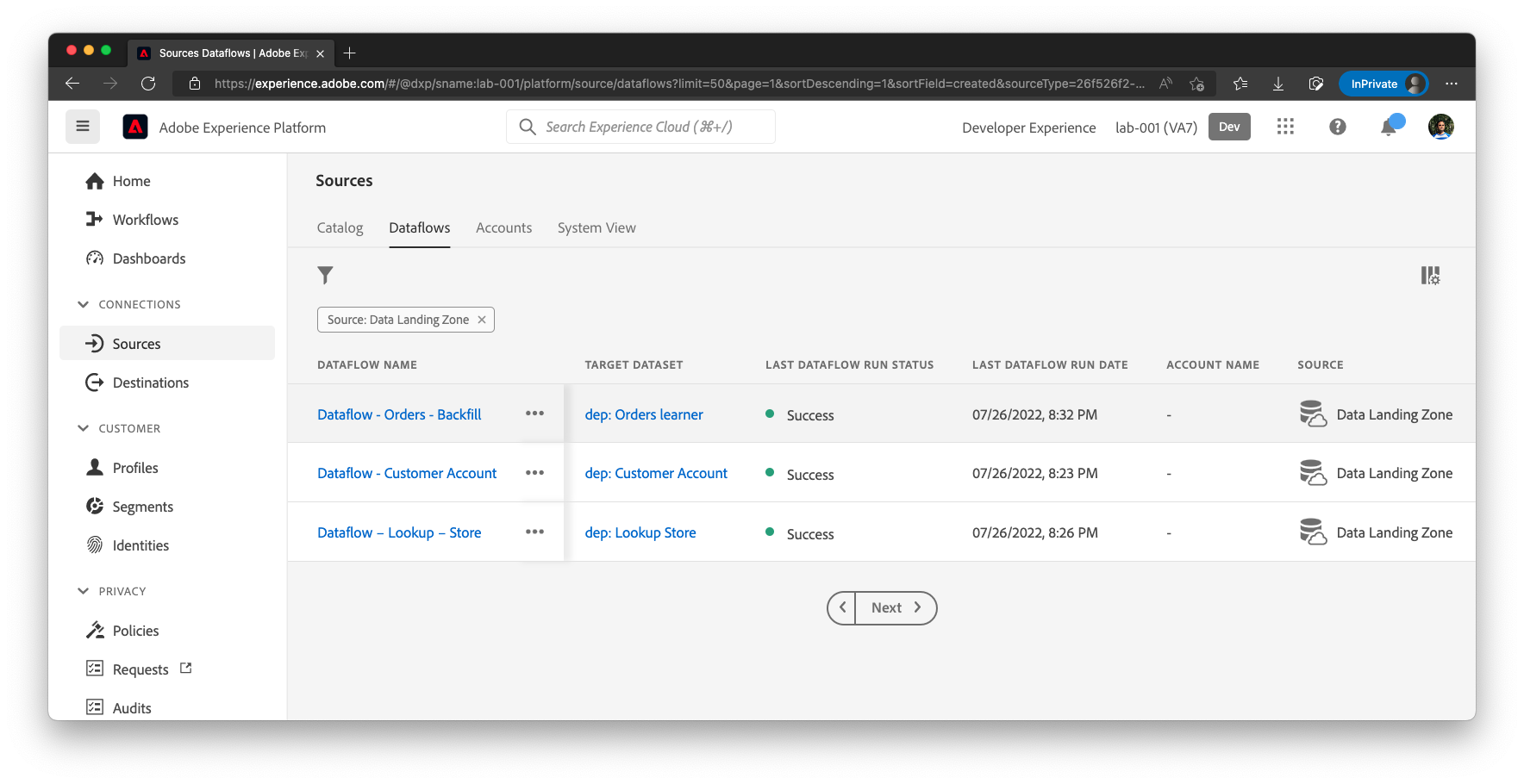


Backfill / Historical load Dataflow will be created. Dataflow execution will not start immediately and will take few minutes. So, last Dataflow Run Status will be set to “*No runs*”.

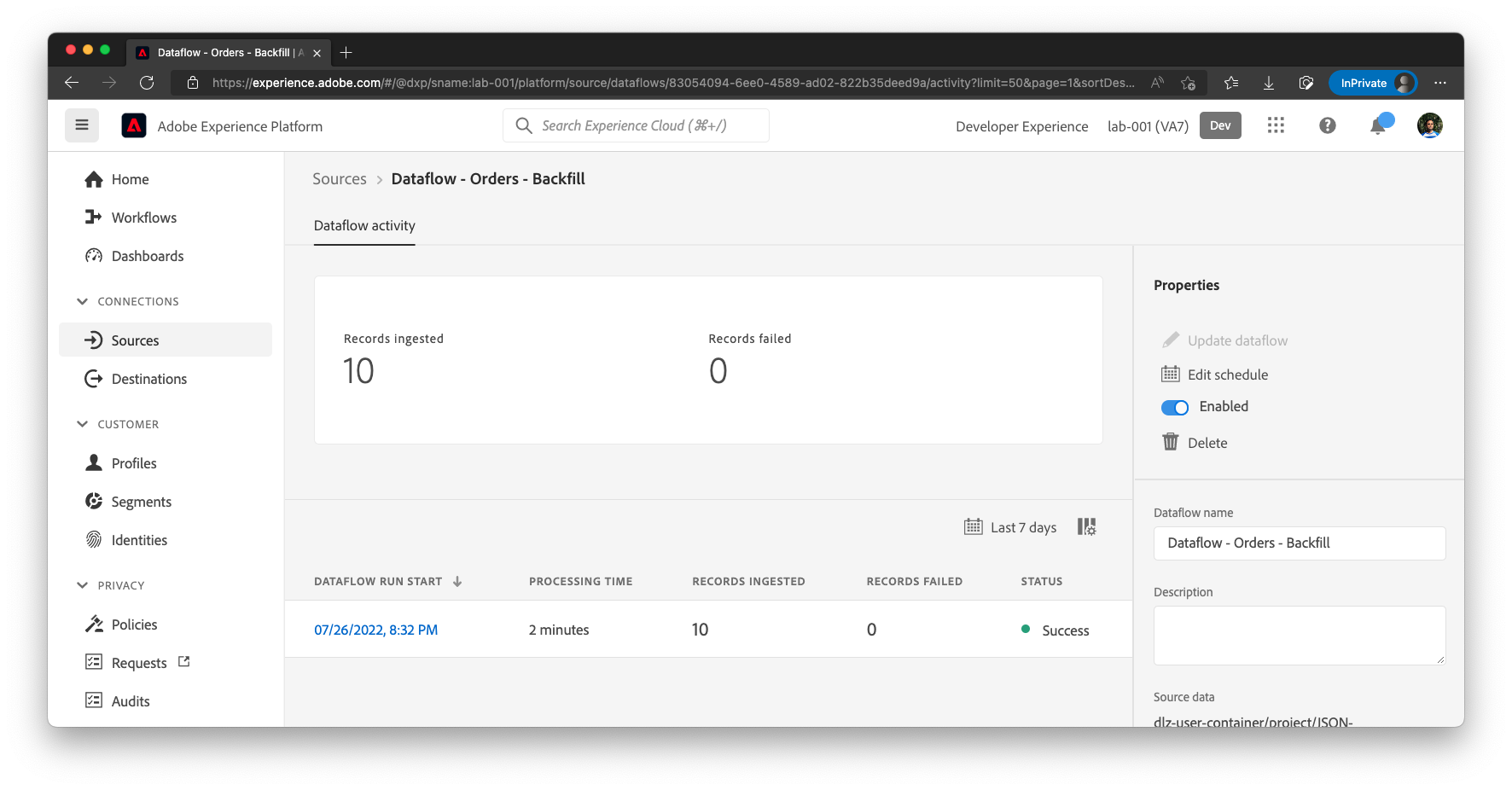


## Check scheduled execution

After few minutes, the following will appear. Notice the **Last Dataflow Run Status** and **Last Dataflow Run Date**.



Click on the Dataflow name to get a list of Dataflow Runs. 10 Records should be ingested.



Click on the Dataflow Run Start time to see error diagnostic details.

A screenshot of a computer

Description automatically generated

## Verify the data

In the Left Nav bar, Go to Datasets in Platform and click on **dep: Orders learner**

Graphical user interface, application

Description automatically generated

Click on the **Preview Dataset.**

Graphical user interface, application

Description automatically generated

Notice that **productListitems.currencyCode** and **productListItems.quantity** are auto populated.

Graphical user interface, application

Description automatically generated