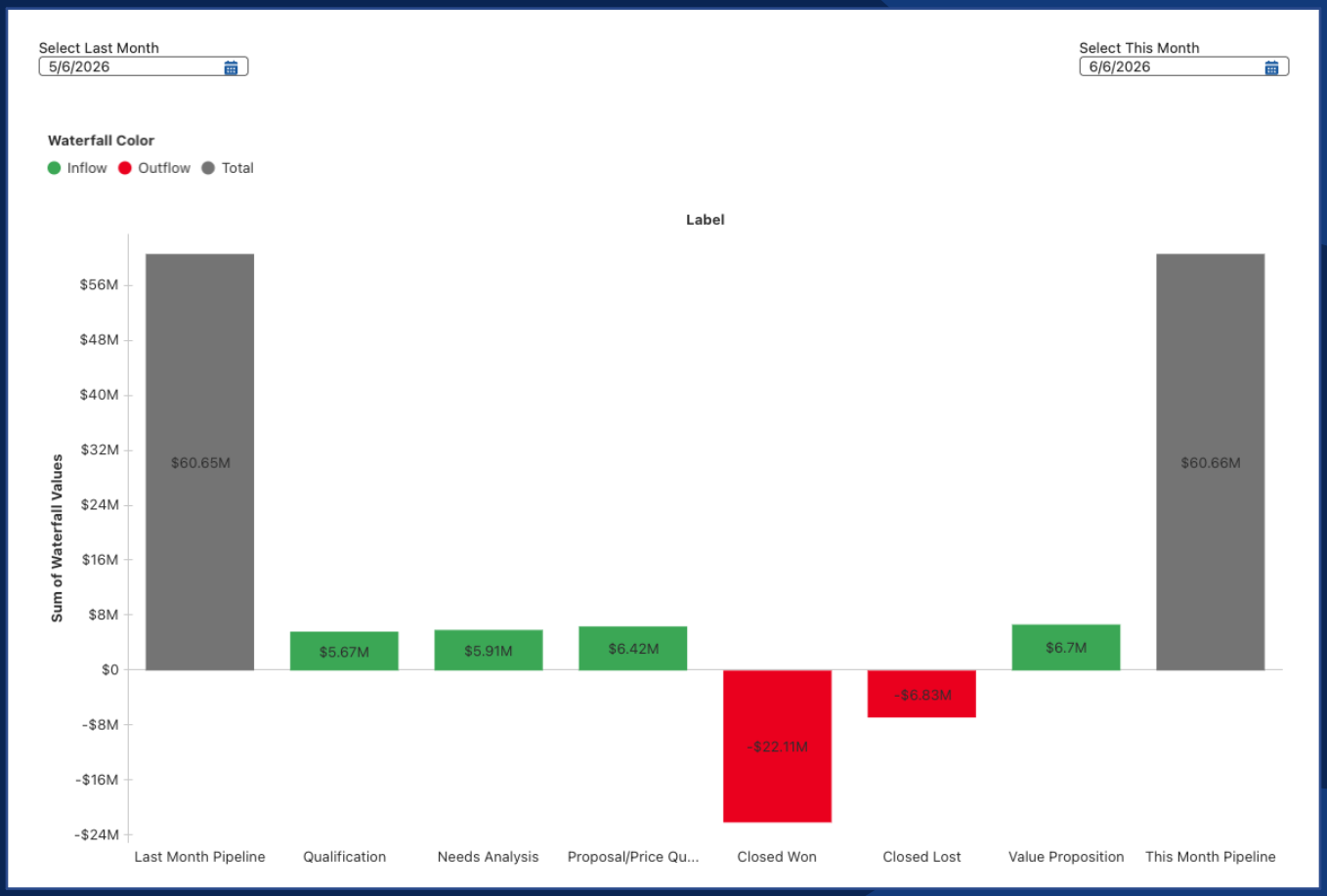
# Snapshot Analytics & Pipeline Waterfall

A complete implementation guide for point-in-time pipeline analysis in Data Cloud and Tableau Next

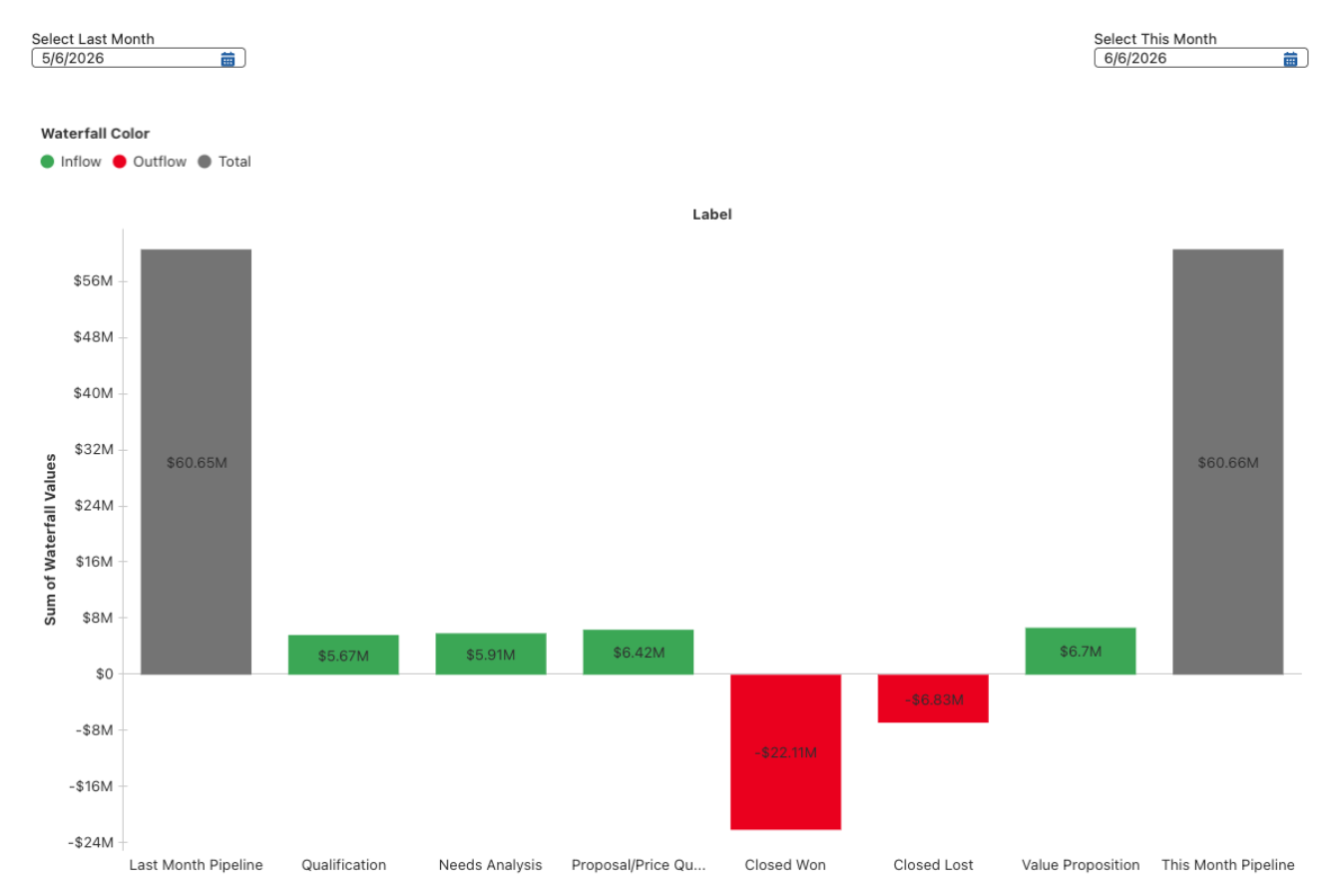


## Why This Matters

Sales leaders need to answer a deceptively hard question: **Why is our pipeline different today than it was last month?** Standard Salesforce reports show current state. They cannot show movement — what came in, what grew, what churned out, what closed.

Answering that question requires two things: **point-in-time snapshots** of opportunity data, and a **waterfall visualization** that breaks the delta between two dates into its component movements.

| ■ **Bars 1 & 9** | ■ **Bars 2–5** | ■ **Bars 6–8** |
|---|---|---|
| Start & End pipeline totals | Stage inflows | Closed Won / Closed Lost outflows |




*Final result: Pipeline Waterfall dashboard with date parameters — gray total bars, green inflow bars, red outflow bars. All bars render from zero; the CASE formula handles directionality.*

## Prerequisites

- Data Cloud provisioned and connected to your Salesforce org

  Note: Data Cloud was rebranded to **Data 360** in October 2025. You may see "Data 360" in the UI — the features and navigation paths are identical.

- Tableau Next enabled

- Data Cloud Admin access to create Data Streams, Data Lake Objects, and Data Transforms
- Tableau Next Creator license
- A configured Data Space in Data Cloud
- A live Salesforce CRM Data Stream connected to the Opportunity object

## Architecture Overview

The pipeline waterfall requires a **Snapshot Store** — a single Data Lake object containing one row per opportunity *per snapshot date*. A **Batch Data Transform** runs daily, stamping that day's date onto every opportunity and writing it into the store. The Tableau Next semantic model then compares two selected dates to compute each waterfall bar.

**Data Pipeline Architecture** single linear flow

```
Salesforce Opportunity (live)
↓
Batch Data Transform
— stamps SnapshotDate = today
— generates CompoundId = OpportunityId + SnapshotDate
— writes CrossJoinKey = 1
↓ (Upsert, daily)
Snapshot Store DLO ← one row per opp per date, grows daily
↓
Tableau Next Semantic Model
— Snapshot Store joined to Waterfall Labels (cross-join on CrossJoinKey)
— Parameters: Select Last Month / Select This Month
— Calculated field: Waterfall Values (single CASE field, all 9 bars)
↓
Waterfall Visualization
```

> **Key insight:** The Snapshot Store must contain rows from at least **two distinct dates** before the waterfall shows any movement. A single-date store only supports current-state reporting.

PART 1 OF 2

Build the Snapshot Store — add the cross-join key, create the consolidated DLO, and run the daily Batch Data Transform.

## Step 1 — Add CrossJoinKey to Your Opportunity DLO

Before building the Snapshot Store, add a formula field to your source Opportunity DLO. This carries a constant value of `1` through the batch transform, enabling the cross-join to the Waterfall Labels table in the semantic model.

Navigate to your Opportunity DLO: Data Cloud Setup → **Data Lake Objects** → [your Opportunity DLO] → **Fields** → **New Field**

| Field Label | Field API Name | Data Type | Formula / Default |
|---|---|---|---|
| CrossJoinKey | `CrossJoinKey__c` | Number | `1` |

> **Why this field?** The SDM cross-join between the Snapshot Store and the Waterfall Labels table requires a shared key where every row on both sides carries the same value. A static `1` on every opportunity row guarantees every row joins to every label row — producing all 9 bars.

Also confirm these fields exist and note their exact API names — every formula downstream references them verbatim:

| Required Field | Common API Names |
|---|---|
| Opportunity ID | `Id__c` |
| Revenue / Amount | `Amount__c` |
| Stage | `Stage__c`, `StageName__c` |

> **Note on waterfall bars 2–8:** These bars can be driven by either **Opportunity Type** or **Stage name**. If your org only has one Type value (e.g. all opps = `'New Business'`), use Stage name instead — it produces more meaningful breakdowns. Confirm what distinct values exist before building formulas.

# Step 2 — Create the Snapshot Store DLO

Create a new, standalone Data Lake Object with only the fields needed for the waterfall. This keeps the Snapshot Store lean and purpose-built.

## 1 New Data Lake Object

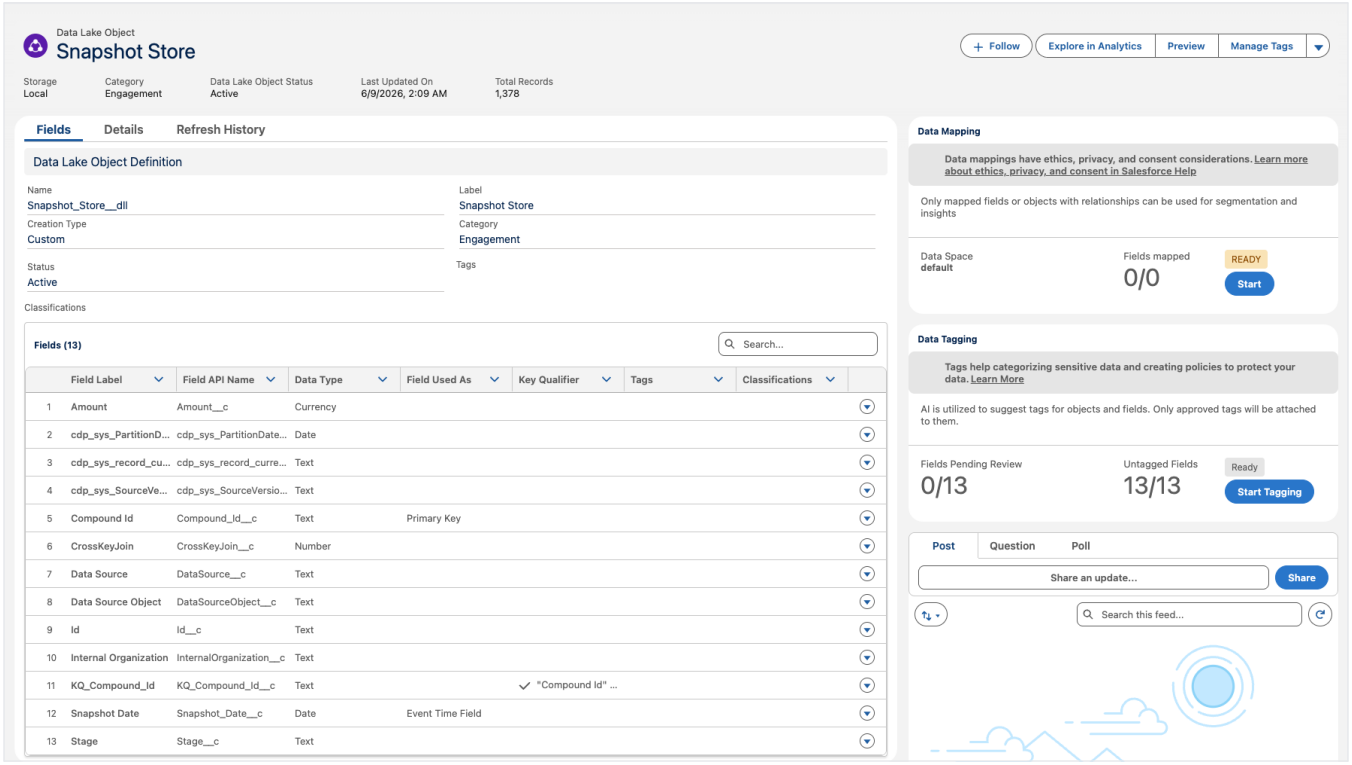
Data Cloud Setup → **Data Lake Objects** → **New**. Name it `Snapshot Store`. Add these six fields:

| Field Label | Field API Name | Data Type | Notes |
|---|---|---|---|
| Id | `Id__c` | Text (255) | Opportunity record ID |
| Amount | `Amount__c` | Number | Opportunity revenue value |
| Stage | `Stage__c` | Text (255) | Opportunity stage name |
| Snapshot Date | `Snapshot_Date__c` | Date | Stamped by the daily transform |
| Compound Id | `Compound_Id__c` | Text (255) | Upsert key: OpportunityId + SnapshotDate |
| CrossKeyJoin | `CrossKeyJoin__c` | Number | Static 1 — enables SDM cross-join |

> **Why Snapshot Date?** Without a date stamp on each row, the DLO only holds current-state data. Snapshot Date lets you filter to any point in time and compare two dates in the semantic model. It's what turns a flat opportunity list into a time-series store.
>
> **Why Compound Id?** Each daily run writes a new snapshot for every opportunity. The upsert write mode needs a unique key per row to distinguish "new snapshot row for the same opportunity" from "update to an existing row". Concatenating `OpportunityId + SnapshotDate` gives a key that is unique across both dimensions.

## 2 Add Snapshot Store to your Data Space

Data Spaces → [your data space] → **Add Objects** → **Snapshot Store**. The DLO must be in a Data Space to be visible in Tableau Next.




*Snapshot Store DLO showing the six essential fields: Amount, Compound Id, CrossKeyJoin, Id, Snapshot Date, Stage.*

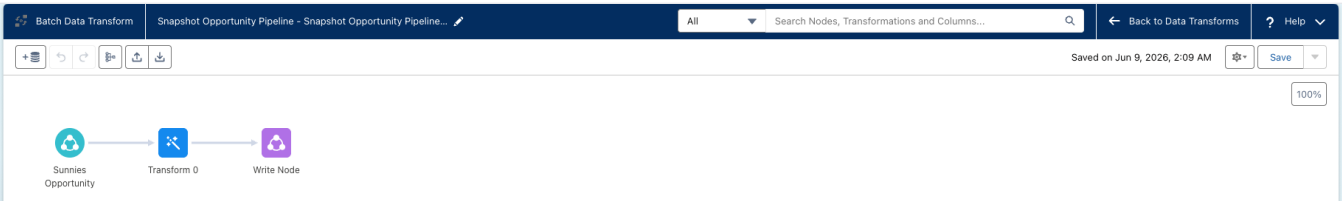
# Step 3 — Build the Batch Data Transform

Navigation: Data Cloud Setup → **More** (top of left nav) → **Data Transforms** → **New** → **Batch Data Transform**

> **Navigation note:** The menu item is **Data Transforms**, not "Data Transformations". Direct URL navigation (`/lightning/o/DataTransformation/home`) returns a permissions error — navigate via the **More menu** only.

Name it: `Snapshot Opportunity Pipeline`. The canvas uses a recipe-style node editor — the same visual language as CRMA Data Recipes:

- 🟢 **Teal circle** = source (data input)
- ★ **Blue star** = transform (formula / calculated columns)
- 🟣 **Purple circle** = output (write node)




*Batch transform canvas: Opportunity DLO (teal) → Transform 0 (blue star) → Write Node (purple).*

## 3a — Source Node

Click the teal source node. In the left panel, confirm the API name of your Opportunity DLO. This is the object the transform reads from on every run.

> **Field selection:** Select only the fields needed for the waterfall: `Id`, `Amount`, `Stage`, and `CrossJoinKey`. Including all fields from a large Opportunity DLO adds unnecessary processing time. Any `KQ_` prefixed fields auto-generated by Data Cloud on the source DLO can be excluded — the Snapshot Store manages its own upsert key via Compound Id.

## 3b — Transform Node (Calculated Fields)

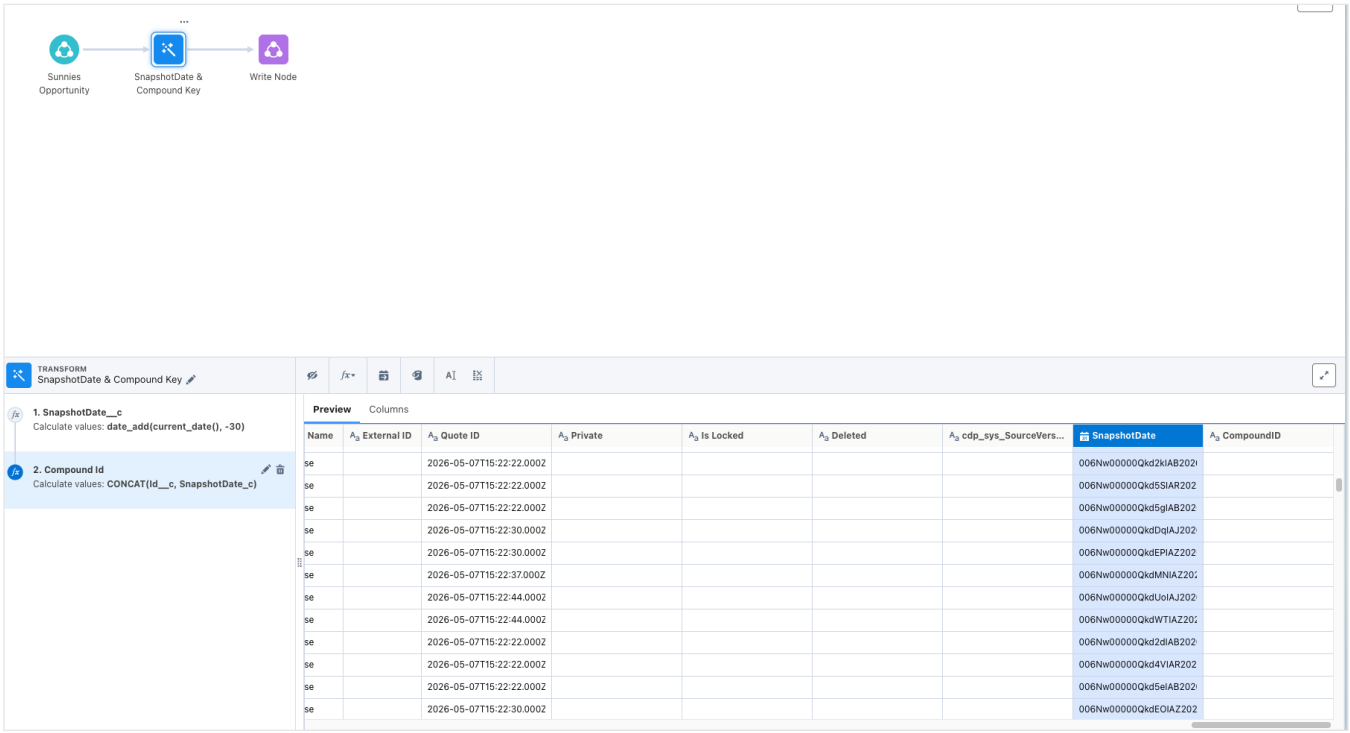
Click the blue star node. In the left panel, add two calculated columns:

**Calculated Columns** Transform node

```
Column 1 — Snapshot Date
Name: Snapshot Date
Formula: current_date()
Type: Date
Column 2 — Compound Id
Name: Compound Id
Formula: concat(Id__c, Snapshot_Date)
Type: Text
```

> **Why these fields?** The Opportunity DLO holds current records but no date stamp. The transform injects two fields on every run: **Snapshot Date** pins today's date to every row, creating the time dimension; **Compound Id** combines the opportunity ID with that date to produce a row-level upsert key. Together they convert a flat opportunity list into a growing time-series store — each daily run adds a new dated layer without overwriting history.

Replace `Id__c` with your actual Opportunity ID field name if different.




*Transform node: (1) Snapshot Date = current_date(), (2) Compound Id = concat(Id__c, Snapshot_Date). Preview shows CrossJoinKey = 1, Snapshot Date and Compound Id populated on all rows.*

> **Date formula syntax note:** In the Batch Data Transform formula editor, use:
> • `current_date()` — for today's date (permanent daily formula)
> • `current_date() - INTERVAL 1 MONTH` — for one month back (backfill use only)
> • `current_date() - INTERVAL N DAY` — for N days back (backfill use only)
> **Do NOT** use `DATE()`, `toDate()`, or `date_add()` — these either fail validation or produce blank preview values in this formula editor.

## 3c — Write Node

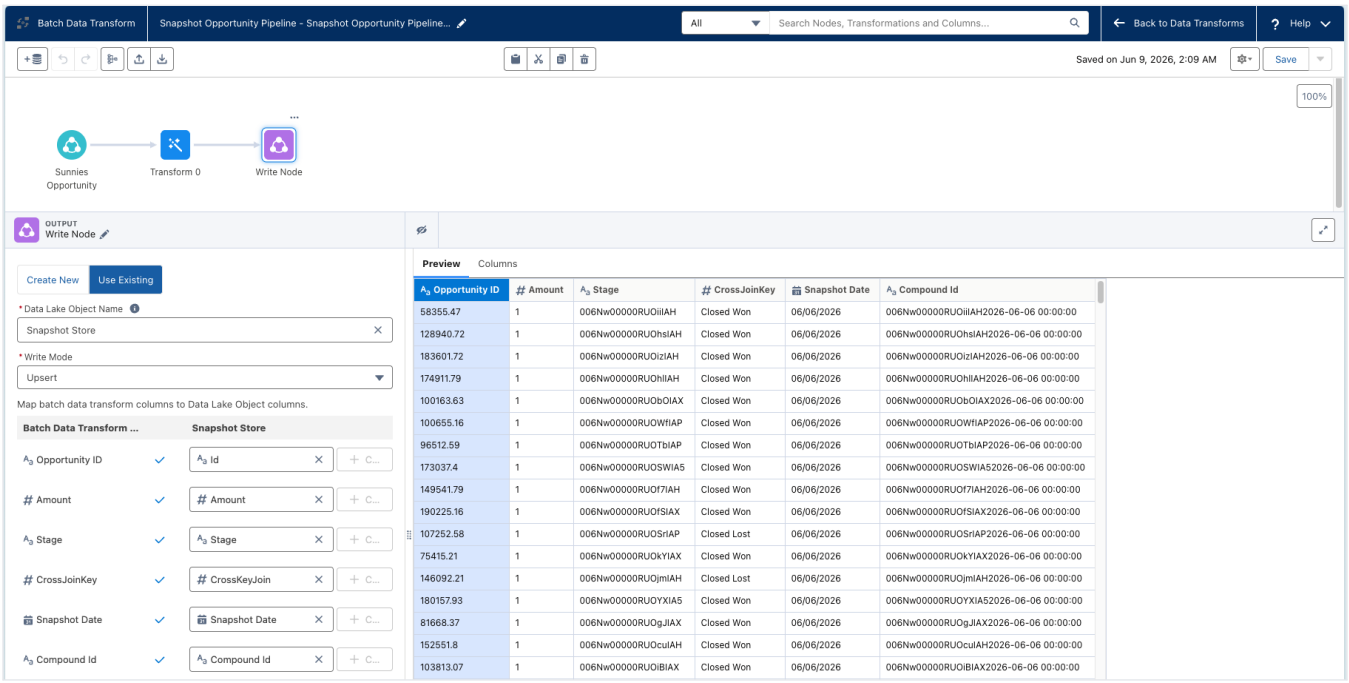
Click the purple circle node. Configure as follows:

- Click **Use Existing**
- Data Lake Object Name: `Snapshot Store`
- Write Mode: **Upsert**

The field mapping table will populate. Confirm these mappings are present, then click **Apply**:

| Batch Data Transform | Snapshot Store |
|---|---|
| Opportunity ID | Id |
| Amount | Amount |
| Stage | Stage |
| CrossJoinKey | CrossKeyJoin |
| Snapshot Date | Snapshot Date |
| Compound Id | Compound Id |

> **Upsert behavior:** The upsert key is determined by the primary key on the target DLO. Since **Compound Id** is the primary key on the Snapshot Store, the upsert write honors it automatically — same opportunity + same date updates the existing row; same opportunity + new date inserts a new row. No additional configuration is needed in the Write node.




*Write node: Use Existing selected, Snapshot Store as target, Upsert write mode, field mapping table with all six fields mapped.*

### 3d — Save and Run

**Save:** Click **Save** in the top right of the transform editor. The Run option is not available from within the editor itself.

**Run:** Return to the Data Transforms list view: Data Cloud Setup → More → **Data Transforms** → [find your transform] → dropdown chevron → **Run Now**. Alternatively, open the saved transform detail view (not the editor) and use Run Now from there.

> A dialog appears with a `"Run with all data"` checkbox:
> • For the **initial run and all backfill runs**: check "Run with all data" to ensure all existing rows are processed.
> • For **scheduled daily runs**: leave unchecked (incremental mode is sufficient for new data).

After the run completes, verify in Data Lake Objects → Snapshot Store → **Preview**:

- ■ Row count matches your Opportunity count
- ■ Snapshot Date column shows today's date on every row
- ■ Compound Id column is populated
- ■ CrossKeyJoin column shows 1 on every row

## Step 4 — Seed a Second Snapshot Date & Set Schedule

> **The bootstrap requirement:** The waterfall needs at least two distinct snapshot dates to show any movement. After your first run, you have one date. You need a second.

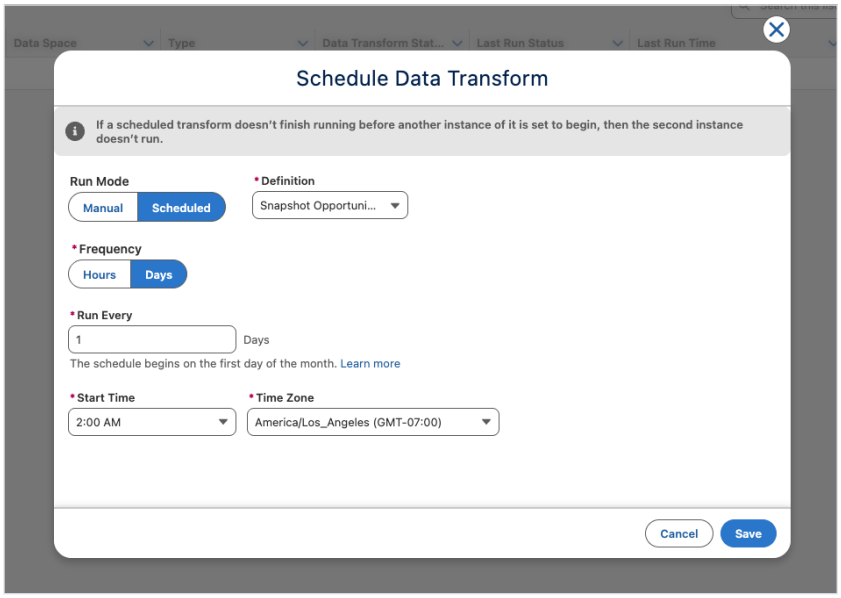
| OPTION A | OPTION B |
|---|---|
| **Wait** | **Backfill immediately** |
| Set the daily schedule (below) and run again tomorrow. On day 2, select the two dates as your parameters in Tableau Next. | Recommended for demos — temporarily stamp a past date, run, then restore current_date(). Steps below. |

### Option B — Backfill immediately (recommended for demos):

1. Open the transform → click the **Transform** node
2. Edit the Snapshot Date formula: change `current_date()` to `current_date() - INTERVAL 1 MONTH`
3. Save → Run Now → check `Run with all data`
4. Wait for completion
5. Edit the formula back to `current_date()`
6. Save → Run Now → check `Run with all data`
7. Verify in the Snapshot Store Preview: you should now see 2 distinct dates in the Snapshot Date column and approximately 2× your Opportunity count in total rows

### Set the daily schedule:

Data Transforms → find your transform → dropdown → **Schedule**




*Schedule Data Transform dialog: Run Mode = Scheduled, Frequency = Days, Run Every = 1.*

- Run Mode: **Scheduled**
- Frequency: Days
- Run Every: 1
- Start Time: choose a time after your nightly Salesforce sync completes (typically 2–4 AM in your org's timezone)
- Save

> From this point the transform is fully automated. Every morning it adds a new dated layer to the Snapshot Store. After 30 days you have 30 distinct snapshots — select any two dates in Tableau Next and the waterfall recomputes automatically.

PART 2 OF 2

Build the waterfall visualization — semantic model, cross-join labels, calculated fields, and chart configuration.

## Step 5 — Create the Waterfall Labels Data Object

The waterfall uses a 9-row lookup table to define bar labels and ordering. This is a static CSV uploaded once and joined to the Snapshot Store in the semantic model.

> **Critical:** The CSV requires **three columns** — `Points`, `Label`, and `Cross_Join_Key`. The Cross_Join_Key column (all values = 1) is required for the SDM join. A two-column CSV will produce a broken join and zero values on all bars.

Create a file named `waterfall_labels.csv` with exactly this content:

**waterfall_labels.csv** CSV

```
Points,Label,Cross_Join_Key
1,Last Month Pipeline,1
2,Qualification,1
3,Needs Analysis,1
4,Proposal/Price Quote,1
5,Negotiation/Review,1
6,Closed Won,1
7,Closed Lost,1
8,Value Proposition,1
9,This Month Pipeline,1
```

> **Note on bar labels:** Bars 2–8 (middle bars) are driven by **Stage name** in this implementation. If your org has a meaningful Opportunity Type field with values like `'New Business'`, `'Organic Growth'`, `'Contract Renewal'`, adapt the labels and Step 8 formulas accordingly. Check your actual data first — bars mapped to non-existent values will show zero.

### Upload to Data Cloud:

1. Data Cloud Setup → **Data Streams** → **New** → **File Upload**
2. Upload the CSV. Name the stream `waterfall_labels`
3. Accept auto-detected schema — confirm `Points` = Number, `Label` = Text, `Cross_Join_Key` = Number
4. Data Lake Objects → **New** → **Create from Existing** → select `waterfall_labels` stream
5. Name the DLO `waterfall_labels`
6. Data Spaces → [your data space] → **Add Objects** → `waterfall_labels`

> **Do NOT** include `waterfall_labels` in the Snapshot Transform — it is a static lookup table, not snapshot data.

# Step 6 — Build the Semantic Data Model

In Tableau Next → **New** → **Semantic Model**. Name it `Pipeline Waterfall` (or `Pipeline Snapshot`).

### 1 Add both data objects

Add **Snapshot Store** as the primary data object. Add **Pipeline Waterfall Labels.csv** as a second data object.

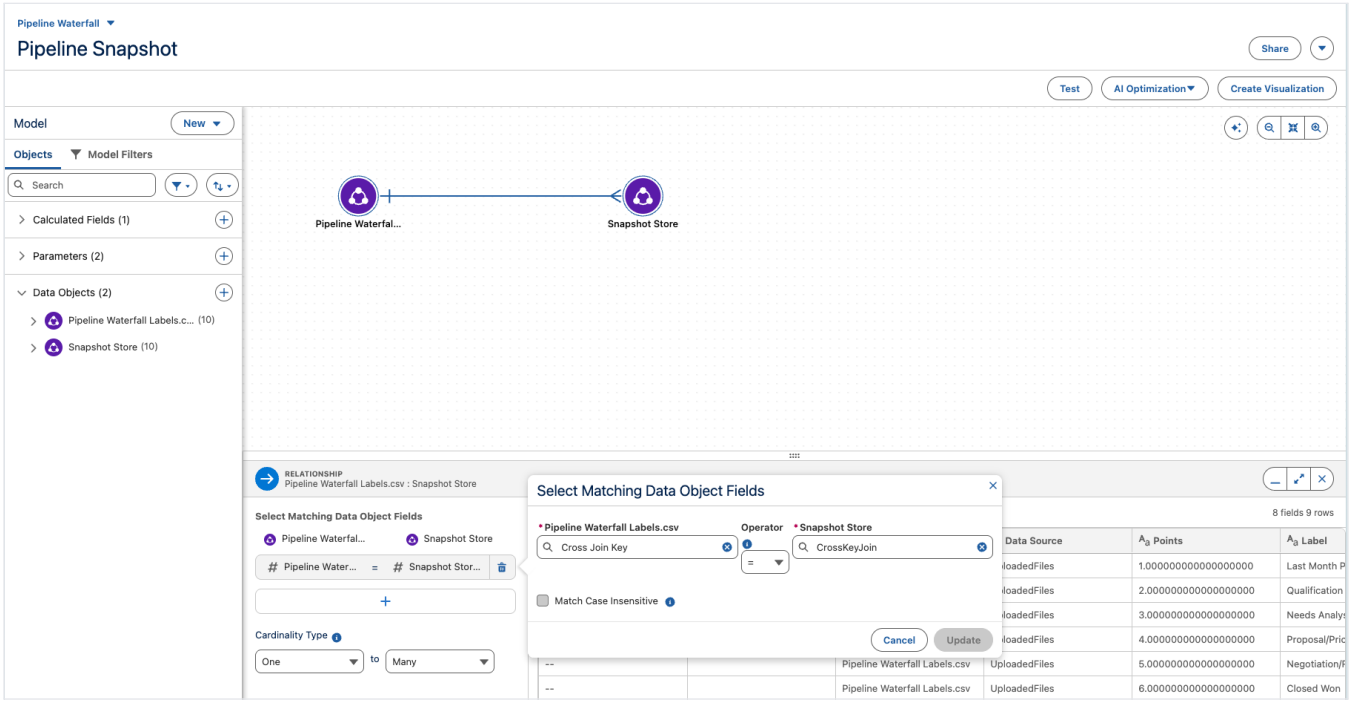
> **Important — the cross-join key must be a physical field.** In the Tableau Next SDM, you cannot create a calculated field scoped to a specific data object. The `New ▼` button creates global calculated fields only. The + button on each table circle opens the JOIN dialog, not an add-field dialog. The cross-join key must be a physical field on both DLOs — which is why you added CrossJoinKey to the Opportunity DLO in Step 1 and included Cross_Join_Key in the waterfall_labels CSV in Step 5.

### 2 Create the cross-join relationship

Click the relationship line between the two table circles on the canvas. In the join dialog:

- Left field (Pipeline Waterfall Labels.csv): `Cross Join Key`
- Operator: `=`
- Right field (Snapshot Store): `CrossKeyJoin`
- Cardinality: **One** (Labels) to **Many** (Snapshot Store)

Click **Update** → **Apply**.




*SDM join dialog: Pipeline Waterfall Labels.csv.Cross Join Key = Snapshot Store.CrossKeyJoin, Cardinality One to Many. Canvas shows the join line connecting the two data objects.*

# Step 7 — Create Parameters

In the SDM left panel → **Parameters** (+) → add two date parameters:

| Parameter Name | Type | Default Value |
|---|---|---|
| `Select Last Month` | Date — single value | Your "last month" snapshot date |
| `Select This Month` | Date — single value | Your "this month" snapshot date |

Parameter names are case-sensitive. Use exactly `Select Last Month` and `Select This Month` — all formulas reference `[Parameters].[Select Last Month]` and `[Parameters].[Select This Month]`.

**Default values must exactly match dates in your Snapshot Store.** If the parameter date doesn't match any Snapshot Date value in the DLO, **all bars return zero with no error message.** Verify your snapshot dates in Data Lake Objects → Snapshot Store → Preview before setting defaults.

# Step 8 — Create Calculated Fields

Click `New ▼` → **Calculated Field** for each field below.

> **Before writing formulas:**
> • Use `ELSE 0.0` (not `ELSE 0`) throughout. The Amount field is typed as Currency/Decimal — integer literals cause a type mismatch validation error.
> • Never write `-[Amount]`. Always write `[Amount] * -1`. The semantic layer rejects unary negation.
> • If your Amount field is typed as Currency in the DLO and you see a `"missing conversion rates"` error in the SDM, go to the Amount field properties in the SDM and change the Data Type from Currency to Number.
> • Use `[Snapshot Store].[field name]` syntax to reference Snapshot Store fields. Use `[Pipeline Waterfall Labels.csv].[field name]` for label fields.

**Last Month Pipeline** — Measure · Number

```
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select Last Month]
THEN [Snapshot Store].[Amount]
ELSE 0.0
END
```

Total pipeline value at the "last month" date.

**This Month Pipeline** — Measure · Number

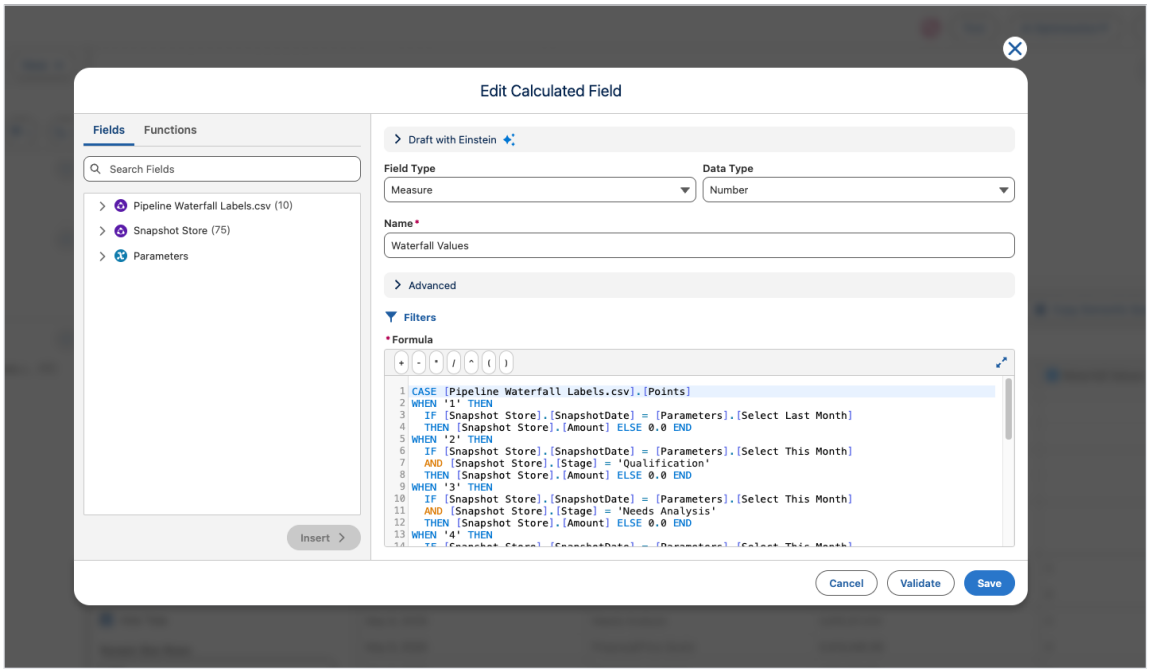
```
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select This Month]
THEN [Snapshot Store].[Amount]
ELSE 0.0
END
```

Total pipeline value at the "this month" date.

## Waterfall Values — Field Type: Measure, Data Type: Number

This single CASE field drives all 9 bars. It switches on the **Label string** from the waterfall_labels table.

> **Why Label strings and not Points integers?** Points values ingested from CSV become floats (`1.0000000000000000`). The SDM only supports Dimension (Text) or Measure (Number) field types — there is no Dimension + Number combination. Integer comparisons against float Points values silently return zero. Using Label strings avoids this entirely.




*Edit Calculated Field dialog: Field Type = Measure, Data Type = Number, Name = Waterfall Values, formula switching on [Pipeline Waterfall Labels.csv].[Label].*

**Waterfall Values** — Core Measure — all 9 bars · switches on Label

```
CASE [Pipeline Waterfall Labels.csv].[Label]
WHEN 'Last Month Pipeline' THEN
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select Last Month]
THEN [Snapshot Store].[Amount] ELSE 0.0 END
WHEN 'Qualification' THEN
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select This Month]
AND [Snapshot Store].[Stage] = 'Qualification'
THEN [Snapshot Store].[Amount] ELSE 0.0 END
WHEN 'Needs Analysis' THEN
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select This Month]
AND [Snapshot Store].[Stage] = 'Needs Analysis'
THEN [Snapshot Store].[Amount] ELSE 0.0 END
WHEN 'Proposal/Price Quote' THEN
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select This Month]
AND [Snapshot Store].[Stage] = 'Proposal/Price Quote'
THEN [Snapshot Store].[Amount] ELSE 0.0 END
WHEN 'Negotiation/Review' THEN
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select This Month]
AND [Snapshot Store].[Stage] = 'Negotiation/Review'
THEN [Snapshot Store].[Amount] ELSE 0.0 END
WHEN 'Closed Won' THEN
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select Last Month]
AND [Snapshot Store].[Stage] = 'Closed Won'
THEN [Snapshot Store].[Amount] * -1 ELSE 0.0 END
WHEN 'Closed Lost' THEN
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select Last Month]
AND [Snapshot Store].[Stage] = 'Closed Lost'
THEN [Snapshot Store].[Amount] * -1 ELSE 0.0 END
WHEN 'Value Proposition' THEN
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select This Month]
AND [Snapshot Store].[Stage] = 'Value Proposition'
THEN [Snapshot Store].[Amount] ELSE 0.0 END
WHEN 'This Month Pipeline' THEN
IF [Snapshot Store].[Snapshot Date] = [Parameters].[Select This Month]
THEN [Snapshot Store].[Amount] ELSE 0.0 END
ELSE 0.0
END
```

> **Adapting for different Stage values:** Replace the stage strings (`'Qualification'`, `'Needs Analysis'`, etc.) with the exact values in your customer's data. Stage strings are **case-sensitive**. Verify in Data Lake Objects → Snapshot Store → Preview before building formulas.

## Waterfall Color — Field Type: Dimension, Data Type: Text

**Waterfall Color** Dimension · Text

```
IF [Pipeline Waterfall Labels.csv].[Label] = 'Last Month Pipeline'
OR [Pipeline Waterfall Labels.csv].[Label] = 'This Month Pipeline'
THEN 'Total'
ELSEIF [Pipeline Waterfall Labels.csv].[Label] = 'Closed Won'
OR [Pipeline Waterfall Labels.csv].[Label] = 'Closed Lost'
THEN 'Outflow'
ELSE 'Inflow'
END
```

Categorizes each bar for color encoding: **Total** (start/end bars), **Outflow** (Closed Won / Closed Lost), and **Inflow** (all stage bars). Click **Validate** before saving each field. Save the SDM.

### Verify in the SDM Test panel before building the visualization:

Click **Test** → add these to the Outline:

- Dimension: `Snapshot Store.Snapshot Date`, `Pipeline Waterfall Labels.csv.Label`
- Measure: `Waterfall Values`

Click **Run**. You should see one row per label per snapshot date, with Waterfall Values returning real dollar amounts for the matching date/stage combinations and `0.0` for non-matching rows. Both snapshot dates should appear with labels populated.

If Waterfall Values shows `0` for all rows, check that parameter default dates exactly match the dates in your Snapshot Store.

# Step 9 — Build the Visualization

Click **Create Visualization** (top right of the SDM).

### 1 Set mark type to Bar

Mark type → **Bar**.

### 2 Columns shelf — Label, sorted by Points

Drag `Label` (from Pipeline Waterfall Labels.csv) to Columns. Right-click the Label pill → **Sort** → By Field → `Points` → Ascending. This locks the x-axis order: Last Month Pipeline → stage bars → This Month Pipeline.

### 3 Rows shelf — Waterfall Values

Drag `Waterfall Values` to Rows → confirm it shows as `SUM(Waterfall Values)`.

### 4 Apply color encoding

Drag `Waterfall Color` to the Color shelf. Assign hex colors:

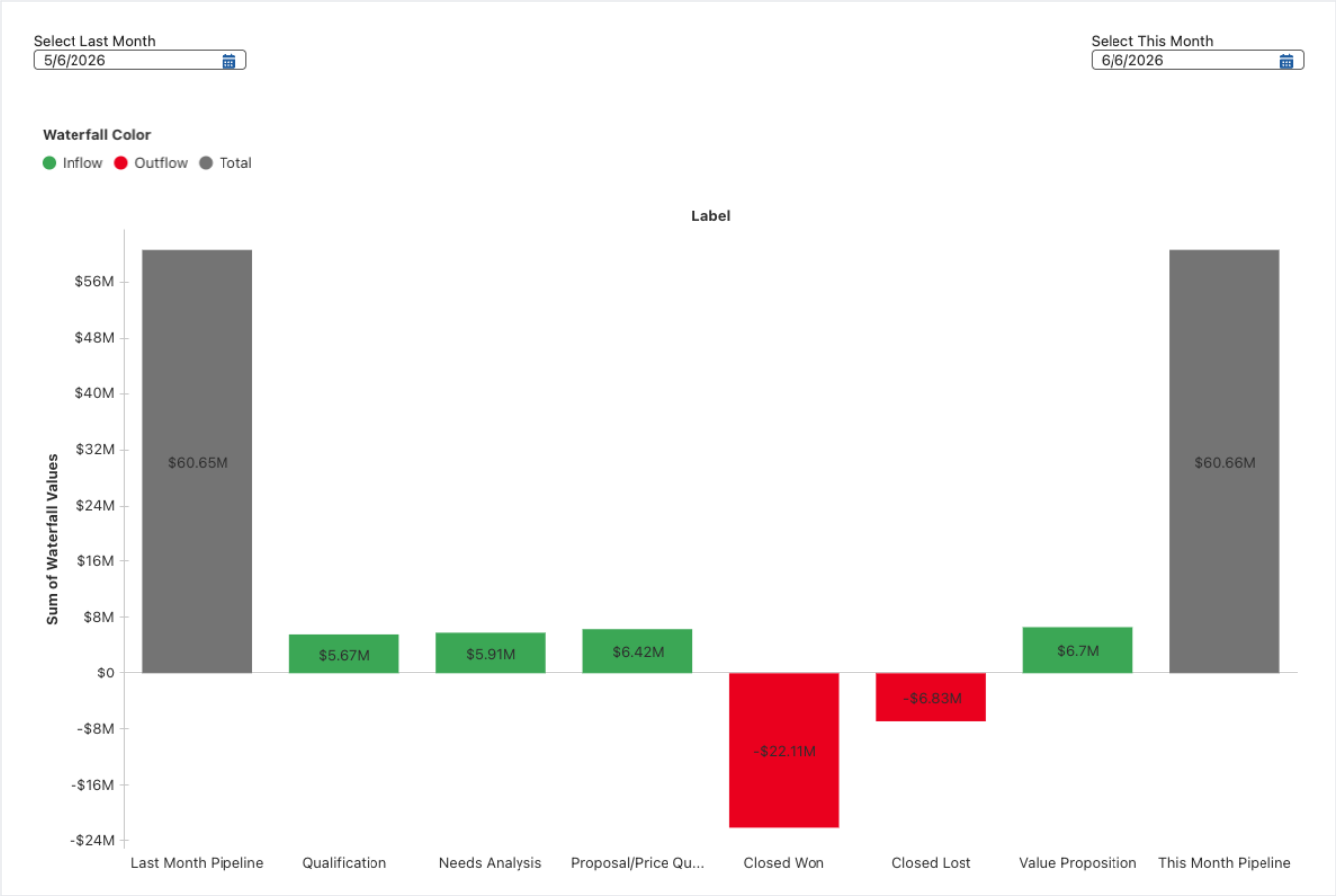
| Category | Hex | Bars |
|---|---|---|
| ■ Total | #747474 | Last Month Pipeline, This Month Pipeline |
| ■ Inflow | #3BA755 | Stage bars (green) |
| ■ Outflow | #EA001E | Closed Won, Closed Lost (red) |

### 5 Add date pickers

Drag `Select Last Month` and `Select This Month` parameters onto the canvas. These become the interactive date selectors shown at the top of the dashboard.

### 6 Save

The result: inflow bars rise above zero, outflow bars fall below zero. The CASE formula handles directionality — no floating bar encoding required.




*Final Pipeline Waterfall: Select Last Month / Select This Month date pickers, gray total bars, green stage bars (Qualification / Needs Analysis / Proposal-Price Quote / Value Proposition), red bars for Closed Won and Closed Lost.*

## Validation — Does Your Waterfall Balance?

Before sharing with stakeholders, verify with this identity. Build a plain table in the visualization with `Label` on Rows and `SUM(Waterfall Values)` as the measure. Sum all 9 rows. The total must equal **This Month Pipeline**.

**Balance Identity** Verification

```
Last Month Pipeline (positive — bar 1)
+ Qualification (positive — bar 2)
+ Needs Analysis (positive — bar 3)
+ Proposal/Price Quote (positive — bar 4)
+ Negotiation/Review (positive — bar 5)
+ Closed Won (negative — bar 6)
+ Closed Lost (negative — bar 7)
+ Value Proposition (positive — bar 8)
═══════════════════════════════════════
= This Month Pipeline (positive — bar 9)
```

> **If they don't balance:** the most common causes are a parameter date that doesn't match a stored Snapshot Date (all bars zero), a Stage string that doesn't match the DLO exactly (a stage bar zero), or the Last Month Pipeline branch pointing at the This Month parameter (bars 1 and 9 identical). See the Troubleshooting Reference on the next page.

## Troubleshooting Reference

| Symptom | Cause | Fix |
|---|---|---|
| All bars zero | Parameter dates don't match any SnapshotDate in DLO | Verify exact dates in Snapshot Store Preview; update parameter defaults to match |
| Bars 1 and 9 identical | Waterfall Values WHEN 'Last Month Pipeline' using This Month parameter | Change condition to [Parameters].[Select Last Month] |
| Stage bars all zero | Stage strings in formula don't match DLO values exactly | Check case-sensitive Stage values in Snapshot Store Preview |
| Outflow bars zero | Stage = 'Closed Won' / 'Closed Lost' not matching | Verify stage string case exactly matches DLO values |
| Type mismatch error | ELSE 0 (integer) conflicts with decimal/currency Amount field | Change all ELSE 0 to ELSE 0.0 throughout Waterfall Values |
| Currency conversion error in SDM | Amount typed as Currency, org missing conversion rates | Change Amount field Data Type from Currency to Number in SDM field properties |
| Syntax error on Points comparison | Points ingested as float — integer comparison fails | Use Label string comparisons instead of Points integer comparisons |
| CrossJoinKey null on some rows | Transform ran before CrossJoinKey was mapped in Write node | Re-run transform with "Run with all data" checked |
| Write node field not visible in mapping | Target DLO field didn't exist when Write node was opened | Add the field to the DLO first, then reopen the Write node |
| Run button not visible from editor | Transform must be run from the list view, not the editor | Data Transforms list → dropdown chevron → Run Now |
| "Data Transforms" menu not found | Wrong search term — UI says "Data Transforms" not "Data Transformations" | Navigate via More menu in Data Cloud Setup |
| Direct URL permissions error | `/lightning/o/DataTransformation/home is invalid path` | More menu navigation only |
| Viz save NullPointerException | Multiple separate SDM measures each referencing parameters (API v66 bug) | Use single Waterfall Values CASE field — do not create 9 separate measures |
| Tableau Next not in App Launcher | Not a standard App Launcher entry | Direct URL: https://[org].lightning.force.com/tableau/ |

## Schema Field Reference

Confirm these fields exist in the Snapshot Store before building SDM formulas.

| Field | Used For | Notes |
|---|---|---|
| Snapshot Date | Filtering to Last/This Month dates | Date type. Added manually during DLO creation. |
| Compound Id | Upsert primary key | Text (255). Format: `OpportunityId + SnapshotDate` |
| CrossKeyJoin | Cross-join to waterfall_labels | Number, value = 1 on every row. Required for SDM join. |
| Id | Opportunity identifier | Text. From source Opportunity DLO. |
| Amount | All waterfall bar values | Currency or Number. If Currency, change to Number in SDM to avoid conversion rate error. |
| Stage | Classifying bar 2–8 conditions | Text. Values must match CASE string literals exactly — case-sensitive. |

## Daily Operations (Post-Setup)

Once the schedule is set, no manual intervention is needed:

- **Every morning:** the transform runs — today's opportunities are written to the Snapshot Store with today's SnapshotDate.
- **In Tableau Next:** update the Select Last Month and Select This Month parameters to your desired comparison dates.
- **After 30 days:** 30 distinct snapshot dates are available — pick any two for a month-over-month comparison.
- **Backfilling historical data:** temporarily set the SnapshotDate formula to a past date, run with `Run with all data` checked, then restore `current_date()`.