In **Palantir Foundry**, orchestrating a full data flow from **ingestion → transformation → insight** is all about connecting and automating the lifecycle of data. Foundry’s platform is built for this kind of **end-to-end data orchestration**, and it combines low-code/graphical tools with code-based flexibility.

Let’s break down how this works in **5 steps**, with tooling and tips for each stage:

**🔁 1. Data Ingestion**

**📥 Tools:**

* **Data Connection Manager**: Connect to external sources (databases, APIs, S3, Kafka, etc.)
* **Ingest Workflows**: Define how often data is pulled (batch or streaming)
* **Schema Inference & Validation**: Automatically structure raw data and check for schema issues

**Example:**

You define an ingestion pipeline that pulls daily sales data from an AWS S3 bucket or SAP system.

🛠 You set up:

* A connection (e.g., JDBC/SFTP/S3)
* A pipeline that lands data into Foundry’s object store
* Optional preprocessing like format normalization or timestamp standardization

**🧬 2. Data Transformation**

**🔧 Tools:**

* **Code Workbooks or Transform Graphs**
* **Spark, SQL, Python, R, or no-code blocks**
* **Templated pipelines** (parameterized logic, reusable steps)
* **Dependency management** across datasets

**What Happens Here:**

* Data is cleaned, filtered, enriched
* Joins, aggregations, and business logic are applied
* You create **intermediate and refined datasets**

**Example:**

You join sales data with product metadata, apply currency conversion, and calculate KPIs like total\_sales, conversion\_rate, etc.

You can visualize this in the **Transform Graph**, where nodes represent processing steps and edges represent data flow.

**📊 3. Insights / Analytics**

**🧠 Tools:**

* **Quiver Notebooks**: For exploration and advanced analytics (Python, R, SQL)
* **Applications (e.g., Contour or Slate)**: For dashboards or internal tools
* **Operational Dashboards**: For monitoring KPIs in real time
* **Ontology Models (Objects & Actions)**: For business-user-friendly access

**What Happens:**

* Analysts and business users consume the output
* Insights are visualized as dashboards, apps, or shared reports
* Decision-making or automated workflows are triggered based on results

**⚙️ 4. Orchestration & Automation**

**🚀 Tools:**

* **Job Scheduler**: Define when pipelines run (e.g., hourly, daily)
* **Dependency Triggers**: Pipelines can be triggered when upstream data updates
* **Global Parameters**: Pass date ranges, regions, or config values dynamically
* **Health Checks & Monitoring**: Built-in alerts for failures or anomalies

**Example Flow:**

S3 → Ingest pipeline (daily) → Transform nodes (clean + enrich + aggregate) → Dataset output → Dashboard updates

Each stage is **linked via dependency**, so updates flow through automatically.

**🔒 5. Governance, Security, and Collaboration**

**✅ Tools:**

* **Access Controls**: Define who can see or modify data
* **Lineage Tracking**: Full visibility of data journey
* **Versioning**: Rollback capability for all datasets and code
* **Collaboration Tools**: Comments, reviews, and shared workspaces

**🧩Putting It All Together – Example Flow**

1. **Raw Ingest Dataset:** raw\_sales\_data
2. **Cleaned Dataset:** cleaned\_sales\_data
3. **Enriched Dataset:** sales\_with\_product\_info
4. **Aggregated Insight Dataset:** sales\_summary\_metrics
5. **Slate Dashboard:** “Daily Sales Overview” powered by sales\_summary\_metrics

Let’s walk through a **real-world example** of orchestrating a full **Foundry data flow**, end-to-end. This is a typical **Sales Analytics Use Case**, and I’ll structure it like a pipeline you'd actually build in Foundry.

**🧭 Use Case: Daily Sales Insights Pipeline**

**🧩 Goal:**

Provide a dashboard that shows **daily sales performance**, segmented by **region**, **product category**, and **sales channel**.

**📌 Step-by-Step Flow**

**✅ 1. Ingest Raw Data**

| **Source** | **Dataset Name** | **Description** |
| --- | --- | --- |
| S3 bucket | raw/daily\_sales | Raw JSON/CSV of transactions |
| Internal ERP | ref/products | Product master data |
| CRM | ref/customers | Customer metadata |

📥 Set up ingest pipelines:

* Use **Connection Manager** to link S3/ERP/CRM
* Schedule ingest every night at midnight
* Enable schema inference & data validation rules

**✅ 2. Transform & Clean Data**

**Node 1: cleaned\_sales\_data**

* Parse date fields, enforce data types
* Drop invalid or incomplete records

df = spark.read.json("raw/daily\_sales")

df = df.filter("transaction\_amount IS NOT NULL")

df = df.withColumn("transaction\_date", to\_date("transaction\_ts"))

**Node 2: sales\_enriched**

* Join cleaned\_sales\_data with ref/products and ref/customers
* Add region, category, and customer segment info

df = df.join(products, "product\_id").join(customers, "customer\_id")

**Node 3: sales\_summary\_metrics**

* Group by region, category, date
* Compute metrics: total\_sales, avg\_order\_value, transactions\_count

SELECT

region,

category,

transaction\_date,

COUNT(\*) AS transactions\_count,

SUM(transaction\_amount) AS total\_sales,

AVG(transaction\_amount) AS avg\_order\_value

FROM sales\_enriched

GROUP BY region, category, transaction\_date

**✅ 3. Publish Insights**

**Dataset: sales\_summary\_metrics**

* Final dataset used for analytics
* Partitioned by date for performance
* Marked as **“Ready for Use”** in the catalog

**✅ 4. Build a Dashboard**

**Tool: Slate or Contour**

* Build a “**Daily Sales Dashboard**”
* Filters: Date, Region, Category
* Charts: Line chart (trends), bar chart (category split), map (regional breakdown)

**✅ 5. Orchestrate Everything**

| **Component** | **Description** |
| --- | --- |
| **Job Scheduler** | Triggers at 01:00 AM every day |
| **Pipeline Dependencies** | sales\_summary\_metrics runs after sales\_enriched, which runs after cleaned\_sales\_data, etc. |
| **Parameterization** | Pass ${TODAY.minusDays(1)} as default filter date |
| **Monitoring** | Alerts if any ingest or transform node fails |

**✅ 6. Bonus: Governance & Collaboration**

* Tag sales\_summary\_metrics as **“Certified”** by the analytics team
* Add documentation and assumptions inline in Slate/Notebook
* Grant access to sales leads, execs, and data stewards

**🧠 End Result:**

A fully automated data flow that:

* Ingests new data daily
* Applies transformations
* Outputs business-ready insights
* Updates dashboards automatically
* Can be reused for other markets (via parameterization)

