Highlighting OpenMetadata: Unified Data Catalog & Governance

OpenMetadata is an open-source metadata management platform that provides a unified data catalog, data lineage, and data quality capabilities. It acts as the central hub for data discovery, understanding, and governance within your data platform. By consolidating metadata from various sources (databases, streaming platforms, data lakes, APIs), OpenMetadata empowers users to find, trust, and collaborate on data.
This guide will demonstrate basic and advanced use cases of OpenMetadata, leveraging your **Advanced Track** local environment setup and its integration with Spark, Spline, and other components.
Reference: This guide builds upon the concepts and setup described in **Section 4.2. Core Technology Deep Dive** of the **Core Handbook** and the **Progressive Path Setup Guide Deep-Dive Addendum**, specifically OpenMetadata's role in the **Orchestration & Governance Layer**.

Basic Use Case: Data Discovery and Browsing

Objective: To demonstrate how users can easily discover and browse various data assets (Kafka topics, Delta tables, PostgreSQL tables, FastAPI endpoints) through the OpenMetadata UI.
Role in Platform: Facilitate self-service data discovery for data analysts, scientists, and engineers, enabling them to quickly find relevant datasets without needing to understand the underlying infrastructure.
Setup/Configuration (Local Environment - Advanced Track):

1. **Ensure all Advanced Track services are running:** docker compose up --build -d from your project root. This includes openmetadata, postgres, kafka, minio, fastapi_ingestor, and spark.
2. **Verify OpenMetadata is accessible:** Navigate to http://localhost:8585 in your web browser. (Initial login: admin/admin if not configured otherwise).
3. **Ensure initial metadata ingestion has occurred:** Your Airflow DAGs (e.g., those in airflow_dags/ designed for OpenMetadata ingestion) should have run at least once to populate the catalog. These DAGs typically call scripts in openmetadata_ingestion_scripts/ that use OpenMetadata's Python client to extract metadata from different sources.
    - You might need to manually trigger the Airflow DAGs responsible for metadata ingestion (e.g., openmetadata_ingestion_dag) from the Airflow UI (http://localhost:8080).

Steps to Exercise:

1. **Access OpenMetadata UI:** Open your web browser and go to http://localhost:8585.
2. **Browse Data Assets:**

    - From the left-hand navigation, click "Explore".
    - You can browse by "Services" (e.g., kafka_broker, minio_s3, my_postgres_db, my_fastapi_service) or by "Entities" (e.g., Tables, Topics).
    - **Click on a Service:** For example, click on your minio_s3 service. You should see listed tables like raw_data_bucket.financial_data_delta and raw_data_bucket.insurance_data_delta (assuming your Spark jobs have created these Delta tables).
    - **Click on an Entity:** Navigate to a specific table or topic (e.g., raw_data_bucket.financial_data_delta).
    - **Explore Details:** Observe the "Schema" tab to see column names and data types, and the "Sample Data" tab (if profiling is enabled and run).

Verification:

- **OpenMetadata UI:** The UI successfully loads and displays a list of services (Kafka, S3/MinIO, PostgreSQL, FastAPI), and under each service, you can find and browse the associated data assets (topics, tables). The schema and basic details for these assets are visible, demonstrating basic data discovery.

Advanced Use Case 1: Automated Data Lineage Visualization

Objective: To demonstrate how OpenMetadata automatically captures and visualizes end-to-end data lineage, showing how data flows through your Spark transformation jobs from source Kafka topics to target Delta Lake tables.
Role in Platform: Provide transparency into data origins, transformations, and dependencies, crucial for impact analysis (e.g., "what breaks if I change this column?"), debugging data quality issues, and fulfilling compliance requirements.
Setup/Configuration:

1. **Ensure Basic Use Case setup is complete.**
2. **Ensure Spark jobs are running/have run:** Your Spark streaming jobs (Kafka to raw Delta) and batch transformation jobs (raw to curated Delta) must have executed at least once (e.g., from Airflow DAGs).
3. **Ensure Spline is running:** Spline (http://localhost:8081) must be collecting lineage from your Spark jobs.
4. **Ensure OpenMetadata Lineage Ingestion is active:** The Airflow DAG responsible for ingesting lineage from Spline into OpenMetadata should have run successfully. This DAG typically triggers the Spline connector in OpenMetadata.

Steps to Exercise:

1. **Access OpenMetadata UI:** Go to http://localhost:8585.
2. **Search for a Curated Data Asset:** Use the search bar (e.g., search for financial_data_curated or insurance_data_curated) and click on one of your curated Delta Lake tables.
3. **Navigate to the "Lineage" Tab:**
    - Once on the asset's detail page, click the "Lineage" tab.

   - **Observe:** You should see a visual graph depicting the data flow. This graph will typically show:
     - An upstream Kafka topic (e.g., raw_financial_transactions).
     - A Spark transformation job (represented as a process node).
     - The intermediate raw Delta Lake table (e.g., raw-data-bucket.financial_data_delta).
     - Another Spark transformation job.
     - The final curated Delta Lake table (e.g., curated-data-bucket.financial_data_curated).
   - **Explore Column-Level Lineage:** Click on individual columns within the graph. OpenMetadata (if configured and data exists from Spline) can show which source columns contribute to which target columns.

Verification:

- **OpenMetadata UI:** The "Lineage" tab for your curated Delta tables accurately displays the upstream Kafka topics, intermediate raw Delta tables, and the Spark jobs as process nodes, confirming end-to-end data lineage visualization. Column-level lineage provides fine-grained dependency tracking.

Advanced Use Case 2: Active Data Governance (Tags, Ownership, Descriptions)

Objective: To demonstrate how data stewards and owners can enrich metadata within OpenMetadata by adding business-friendly descriptions, assigning ownership, and applying classification tags (e.g., PII, sensitive data).
Role in Platform: Enable active data governance, improve data understanding across the organization, facilitate compliance efforts (e.g., identifying PII), and foster data ownership.
Setup/Configuration:

1. **Ensure OpenMetadata UI is accessible** and data assets are imported.

Steps to Exercise:

1. **Assign an Owner to a Table:**
   - In OpenMetadata UI, navigate to raw_data_bucket.financial_data_delta.
   - Click the "Manage" tab or look for an "Add Owner" button/section.
   - Assign yourself or a dummy "Data Team" as the owner.
   - **Verify:** The owner is now displayed on the table's overview page.
2. **Add a Business Description to a Table:**
   - On the raw_data_bucket.financial_data_delta overview page, find the "Description" section.
   - Click the "Edit" icon and add a meaningful business description (e.g., "This table contains raw, unvalidated financial transaction data directly ingested from source systems. Used for initial historical record keeping.").
   - **Verify:** The description is saved and visible.
3. **Add a PII Tag to a Column:**
   - On the raw_data_bucket.financial_data_delta table, go to the "Schema" tab.

    - Locate a column that might contain sensitive information (e.g., account_id or user_id if present in raw).
    - Click the "Tags" icon next to the column name.
    - Search for and select a PII.Sensitive or Personal.Sensitive tag (OpenMetadata has a default set of classification tags).
    - **Verify:** The tag is now associated with the column and is visible next to its name.
    - **Search by Tag:** Go back to the "Explore" page and try searching for the tag (e.g., PII.Sensitive). Your table should appear in the search results, demonstrating discoverability by governance classifications.

Verification:

- **OpenMetadata UI:** The changes (owner, description, tags) are immediately reflected on the asset's detail page. Searching by tags successfully filters relevant assets, demonstrating effective data governance capabilities.

Advanced Use Case 3: Data Quality Profiling & Monitoring Integration (Conceptual)

Objective: To demonstrate (conceptually) how OpenMetadata integrates with data quality tools and displays data quality metrics and profiles directly within the data catalog.
Role in Platform: Surface data quality issues and insights directly alongside data discovery and lineage, empowering data consumers to trust the data and data owners to quickly address issues.
Setup/Configuration (Conceptual Discussion):

1. **Ensure OpenMetadata is running.**
2. **Conceptual Integration with Data Quality Tools (e.g., Great Expectations):**
    - In a production setup, your Spark ETL jobs would typically integrate with a data quality framework like Great Expectations. After a Spark job runs, Great Expectations would generate data quality "expectations" (tests) and "validation results" (pass/fail for those tests).
    - An OpenMetadata ingestion pipeline would then be configured to collect these Great Expectations results and link them to the relevant tables in the data catalog.
    - OpenMetadata can also run its own "Profiler" workflows, which automatically calculate column-level statistics (e.g., min, max, average, null count, distinct count) and data quality metrics (e.g., completeness, uniqueness) based on data stored in MinIO/S3 (Delta Lake).

Steps to Exercise (Conceptual/Discussion):

1. **View "Profiler" Tab:**
    - In OpenMetadata UI, navigate to one of your Delta Lake tables (e.g., curated-data-bucket.financial_data_curated).
    - Click the "Profiler" tab (if enabled and run).
    - **Observe:** You should see various statistical profiles for each column (e.g., column_name, data_type, null_count, distinct_count, min_value, max_value, average_value).

    - **Discuss Data Quality Metrics:** Explain how these profiles can be used to assess data quality (e.g., a high null_count for a mandatory column indicates incompleteness).
2. **Discuss "Data Quality" Tab (Conceptual with Great Expectations):**
    - If Great Expectations integration were fully set up and results ingested, you would see a "Data Quality" tab.
    - **Discuss:** This tab would show a summary of failed/passed data quality tests, potentially with links to detailed Great Expectations validation reports.

Verification (Conceptual):

- **OpenMetadata UI:** The "Profiler" tab (even with basic auto-profiling) demonstrates OpenMetadata's capability to provide statistical insights into data quality. A successful integration with a tool like Great Expectations would show explicit pass/fail results for defined data quality rules, highlighting OpenMetadata's role as a central hub for data quality monitoring and trust. This allows data consumers to quickly assess the reliability of a dataset before using it.

This concludes the guide for OpenMetadata.