**Databricks Workshop (Lab1): Data Ingestion**

**Workshop Objective**

By the end of this workshop, participants will be able to:

* Create a catalog in Databricks
* Create a volume within the catalog
* Upload CSV files (customer, product, order) into the volume using the upload data feature
* Understand basic data organization concepts in Databricks

**1. Introduction to Databricks Catalogs and Volumes**

* **Catalog**: A top-level container to organize data assets such as databases, tables, and volumes.
* **Volume**: A storage abstraction that holds files and data objects, similar to a folder or container.
* **Upload Data Feature**: A UI tool in Databricks to upload local files to the Databricks File System (DBFS) or a volume.

**2. Step 1: Create a Catalog called training**

**Using SQL or Databricks UI:**

CREATE CATALOG training;

* Open a notebook in Databricks and call it **Lab1**
* Run the above SQL command to create a catalog named training.
* Alternatively, you can create catalogs via the Databricks Data Explorer UI.

**3. Step 2: Create a Volume called landing inside the training catalog**

**Using SQL:**

CREATE VOLUME training.default.landing;

* This creates a volume named landing inside the training catalog and default schema.
* Volumes are used to store files and data assets.

**4. Step 3: Upload CSV files (customer, product, order) into the landing volume**

**Using the Upload Data Feature:**

1. **Navigate to Data Tab:**
   * In Databricks workspace, click on the **Catalog** icon on the sidebar.
2. **Select the Volume:**
   * Expand the training catalog.
   * Select the landing volume.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Upload Files:**
   * Click the **Upload to volume** button.
   * Choose the CSV files from your local machine: customer.csv, product.csv, and order.csv.
   * Upload each file one by one.
   * The files will be stored inside the landing volume.
2. **Verify Upload:**
   * After upload, you should see the files listed inside the landing volume.
   * You can also verify by running:

LIST '/Volumes/training/default/landing/';

**A screenshot of a computer

AI-generated content may be incorrect.**

**5. Read the data using SQL**

**Using SQL:**

SELECT \* FROM csv.`/Volumes/training/default/landing/customer.csv` WITH (header = "true") LIMIT 5;

* The above command will read the first 5 rows from csv file.

A screenshot of a computer

AI-generated content may be incorrect.

**6. Bonus: Reading the Uploaded CSV Files in a Notebook**

After uploading, you can read the CSV files into Spark DataFrames for analysis.

%python

# Reading customer.csv

customer\_df = spark.read.format("csv") \

.option("header", "true") \

.load("/Volumes/training/default/landing/customer.csv")

# Reading product.csv

product\_df = spark.read.format("csv") \

.option("header", "true") \

.load("/Volumes/training/default/landing/product.csv")

# Reading order.csv

order\_df = spark.read.format("csv") \

.option("header", "true") \

.load("/Volumes/training/default/landing/order.csv")

# Show sample data

customer\_df.show(5)

product\_df.show(5)

order\_df.show(5)

**A screenshot of a computer

AI-generated content may be incorrect.**

**Summary**

|  |  |  |
| --- | --- | --- |
| Step | Action | Command/Instruction |
| 1 | Create Catalog training | CREATE CATALOG training; |
| 2 | Create Volume landing | CREATE VOLUME training.default.landing; |
| 3 | Upload CSV files to volume | Use Databricks UI **Upload Data** feature |
| 4 | Read CSV files into SQL | SELECT \* FROM CSV. |
| 5 | Read CSV files into DataFrames | Use Spark .read.format("csv") with volume path |