

Serverless Bootcamp 2025

Introduction

In today's data-driven world, the ability to efficiently manage, analyze, and leverage vast amounts of data is paramount. Throughout this guide, we will delve into the powerful serverless options within the [Data Warehouse](#), Notebooks, Workflows and DLT capabilities offered by [Databricks' Data Intelligence Platform](#).

You will be provided the lab environment in the Databricks Academy. From there on we will start working with data and look into the serverless options that we have on Databricks.

We will start the lab by looking into the robust data governance features provided by [Unity Catalog](#), which empowers users to explore their data landscape with [Databricks SQL](#) at ease, offering AI-suggested comments and comprehensive data lineage tracking. These capabilities ensure high data quality and facilitate impact assessments, reinforcing the integrity and reliability of your data assets.

We will then execute one of the notebooks provided using both serverless and standard compute. For the latter, we'll create a compute cluster to experience the difference between the two.

Next we'll create a workflow using one of the provided examples in the lab, both on serverless and classic compute and see the difference in performance and execution time.

Then we'll create a DLT workflow using example data that comes with each workspace.

Moreover, the platform enables the creation of [AI/BI Dashboards](#), built from consolidated and well-governed data. These dashboards are not only quick to generate but also easily shareable with colleagues and stakeholders, fostering informed decision-making across the organization.

Lastly, with the innovative **Genie** feature, users can interact with their data through natural languages. This intuitive approach significantly reduces the complexity of data consumption, making it accessible to a broader audience and truly democratizing data within the enterprise.

During the workshop, you will see how [Databricks' Data Intelligence Platform](#) provides a comprehensive [Data Warehousing](#) solution designed to transform how organizations

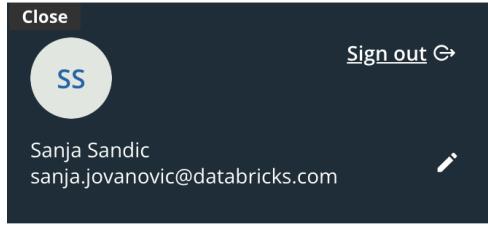
manage and utilize their data, driving efficiency, governance, and accessibility for modern data platforms for enterprises.

Prerequisites

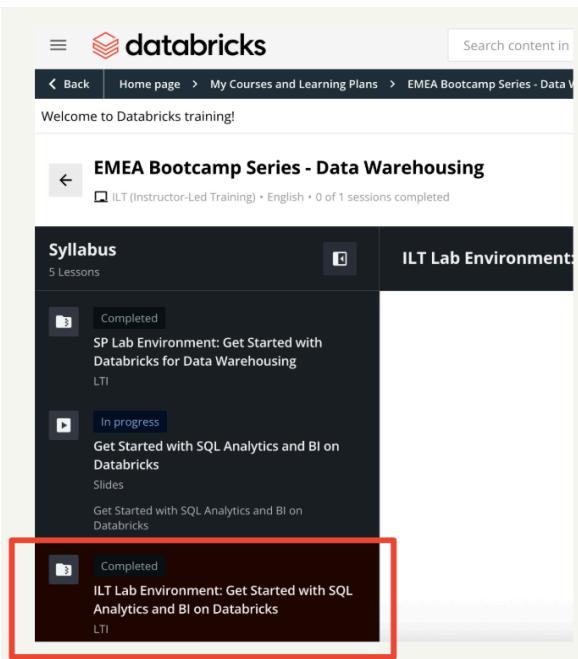
During the workshop, participants can access a Databricks workspace spin-up through the [Databricks Academy](#). Therefore, it is a prerequisite to have a Databricks Academy access using the same email that you registered for the bootcamp. If you have not yet registered for Databricks Academy let us know as we'll need to provision your environment after you register.

Log in to the environment

Log in to the Databricks Academy using your registered email address for the bootcamp. To access the lab environment, after logging in, select the menu button on the left of the page and select Calendar.



The screenshot shows the Databricks Academy login interface. It features a dark header with the Databricks logo and a "Sign out" link. Below the header, there's a circular profile picture placeholder with the letters "SS". The main area displays the user's name, "Sanja Sandic", and email, "sanja.jovanovic@databricks.com", along with a pencil icon for editing. At the bottom of this section is a "Home page" button.



The screenshot shows the syllabus page for the "EMEA Bootcamp Series - Data Warehousing". The top navigation bar includes "Home page", "My Courses and Learning Plans", and "EMEA Bootcamp Series - Data Warehousing". The main content area is titled "Syllabus" and lists "5 Lessons". The first lesson is "Completed" and titled "SP Lab Environment: Get Started with Databricks for Data Warehousing". The second lesson is "In progress" and titled "Get Started with SQL Analytics and BI on Databricks". The third lesson is "Completed" and titled "Get Started with SQL Analytics and BI on Databricks". A red box highlights the third lesson. The right sidebar is titled "ILT Lab Environment".

Your calendar should have an entry for today's Bootcamp. Select this entry, and on the syllabus, you can select the lab session "**ILT Lab Environment: Get Started with SQL Analytics and BI on Databricks**". This will create a workspace and give you the basic setup to continue. Allow the lab creation a couple of minutes.

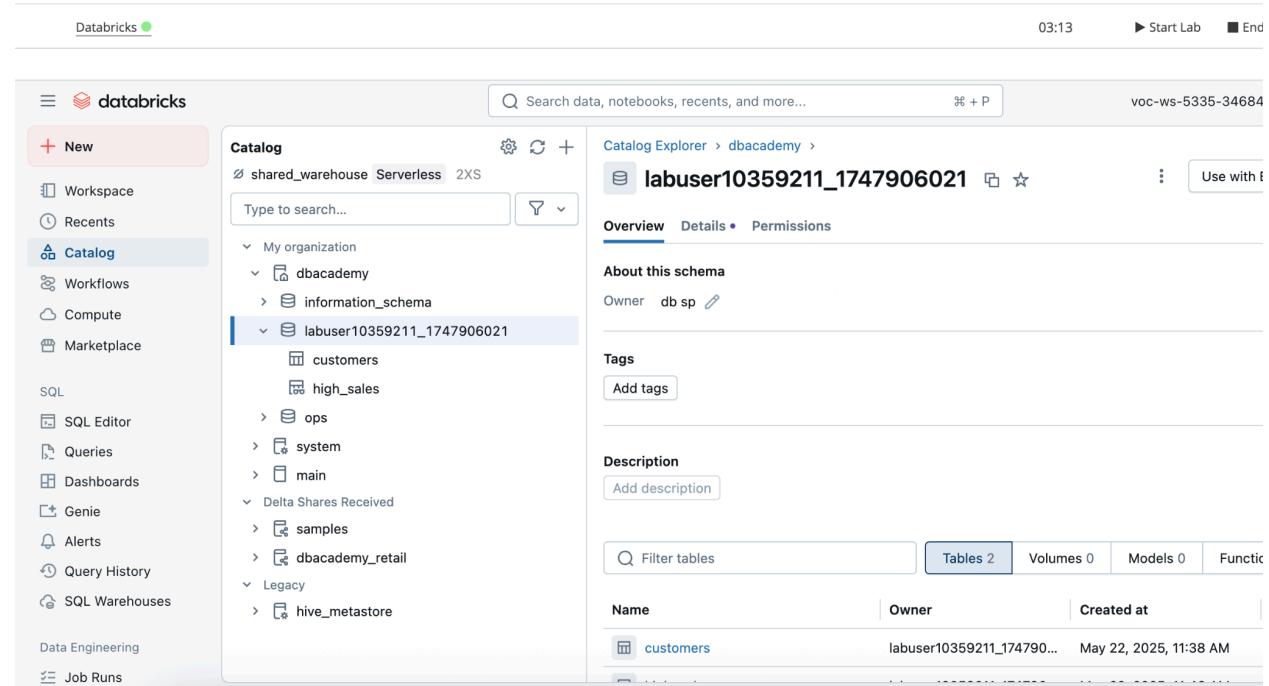
Explore the dataset using Unity Catalog

First, we'll explore the dataset that was provided in the lab through Unity Catalog and look into features like lineage and permission management.

Each one of you will have a dedicated schema created in the catalog "dbacademy" where you have the permission to create tables, views or materialized views.

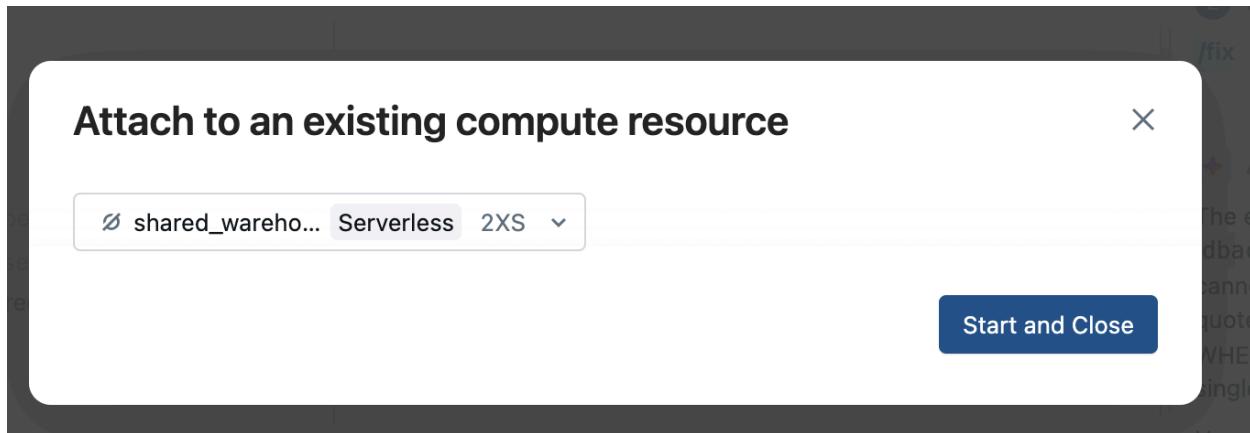
To view any data through the catalog explorer, you will need to spin up a compute. By default, this compute will be a serverless one. Navigate first to the section Catalog. You can select and start it by clicking on the shared_warehouse Serverless to start it.

Notice the startup time of the serverless warehouse.



The screenshot shows the Databricks interface with the Catalog tab selected. The left sidebar includes options like New, Workspace, Recents, Catalog (selected), Workflows, Compute, Marketplace, SQL, and Data Engineering. The Catalog pane displays a tree structure under the dbacademy schema, with the labuser10359211_1747906021 schema expanded to show its contents: customers, high_sales, ops, system, main, Delta Shares Received, samples, dbacademy_retail, Legacy, and hive_metastore. The right pane shows the Overview of the labuser10359211_1747906021 schema, which has an owner of db sp. It lists two tables (customers and high_sales), zero volumes, zero models, and zero functions. The customers table is highlighted. The top right corner shows the timestamp 03:13, a Start Lab button, and an End button.

Once you clicked on the shared_warehouse click on the start and close button.



Within the dbacademy catalog you should have a schema that is called after the username that you received. It should start with labuser followed by a number.

When clicking on the schema you should not have any tables or views yet. We will now go into the lab content to create both a table called customers and a view called high_sales.

Now we will look into another catalog called dbacademy_retail which has a schema called v01(see screenshot below).

- +
- New
- Workspace
- Recents
- Catalog
- Workflows
- Compute
- Marketplace
- SQL
- SQL Editor
- Queries
- Dashboards
- Genie
- Alerts

Catalog

shared_warehouse Serverless 2XS

Type to search... Filter

- My organization
 - > dbacademy
 - > system
 - > main
- Delta Shares Received
 - > samples
 - > dbacademy_retail
 - > information_schema
 - > v01
 - > Tables (3)
 - > Volumes (2)
 - > Legacy

Shema v01 includes 3 tables and 2 volumes containing json files.

Click on the customer's table and look at the sample data. Verify your serverless shared_warehouse is on to view the data.

Optional: Explore the metadata of a table in v01, look into the sample data and details and lineage.

The screenshot shows the Databricks interface with the Catalog tab selected. In the center, the Catalog Explorer for the 'dbacademy_retail' workspace and 'v01' schema is displayed. The 'Lineage' tab is active. A table lists the lineage paths from the 'customers' table to various dashboards:

Name	Direction	Type	Last activity
Kishan's Retail Dashboard	↓ Downstream	Dashboard	last week
Retail Dashboard	↓ Downstream	Dashboard	last week
Bo's Wonderful Dashboard	↓ Downstream	Dashboard	2 weeks ago
Retail Dashboard	↓ Downstream	Dashboard	3 weeks ago
DashBoard	↓ Downstream	Dashboard	2 weeks ago

Notebooks

Notebooks provide an easy way to interact with your data. You can run python code or SQL code and use additional functionalities like visualizations or the AI powered Databricks Assistant which helps you fix code, generate code and analyze the code.

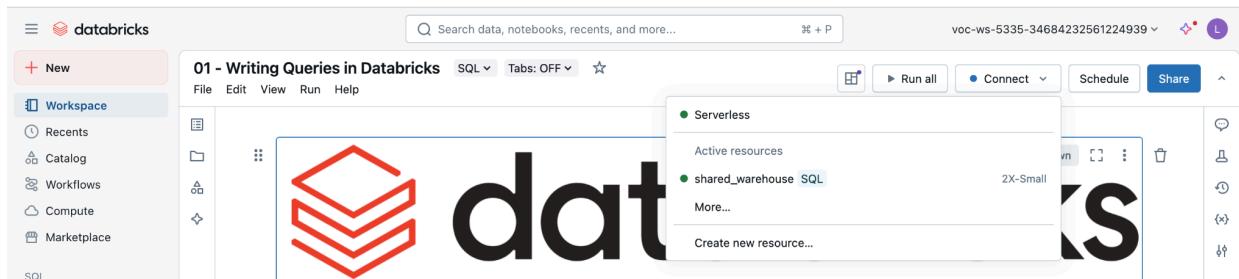
To execute the lab we will need to navigate to the Workspace tab and select Home and then the folder with the content of the training. The folder is named "get-started-with-sql-analytics-and-bi-on-databricks-4.2.0" and looks like this:

The screenshot shows the Databricks interface with the Workspace tab selected. The left sidebar shows the 'Home' workspace. The main area displays the contents of the 'get-started-with-sql-analytics-and-bi-on-databricks-4.2.0' folder:

Name	Type	Owner	Created at
Includes	Folder	db sp	May 27, 2025, 10:00 PM
M01 - Databricks Overview	Folder	db sp	May 27, 2025, 10:00 PM
M02 - Using Databricks for Data Analytics	Folder	db sp	May 27, 2025, 10:00 PM
M03 - Databricks AI-BI	Folder	db sp	May 27, 2025, 10:00 PM
AGENDA	Notebook	db sp	May 27, 2025, 10:00 PM
Version Info	Notebook	db sp	May 27, 2025, 10:00 PM

There are different labs in the folder. For the first, we'll start with "M02 -Using Databricks for Data Analytics - 01 - Writing Queries in Databricks". You can then execute the other two remaining ones afterwards.

Once the notebook opens, select the serverless compute to run the code.



You will now create a table in your schema of the catalog provided by the lab. In Notebook 02 you will explore Gen AI capabilities of the platform through Databricks Assistant writing code for you. Part 03 looks into Delta Lake features like viewing the history of the table and exploring different versions.

Workflows on Serverless

Proctor Demo

The following shows the setup of creating a job and setting up a standard cluster. Your proctor will give you a short demo now, since creating a job compute requires admin privileges on the environment. After creating the job using standard compute cluster the proctor will run it. Then you will create the same workflow using serverless compute and we can check the time it took both to finish.

The following shows the settings when creating a standard compute cluster:

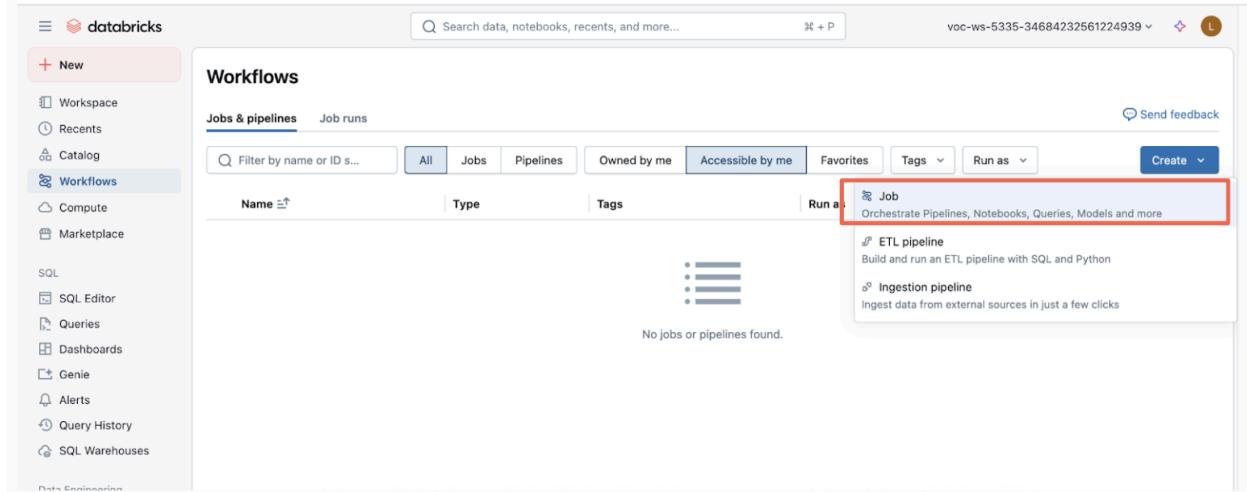
The workflow, since it's not serverless, will take some time to be run. Mainly creating the job compute will take up at least 4 minutes which makes the serverless compute option on Databricks workflows more attractive.

Below you can find the result of the workflow and its complete duration (about 9 minutes). Once you start your hands on lab using serverless compute you will see an increased duration (about 2-3 minutes).

End of proctor demo - let's get handson!

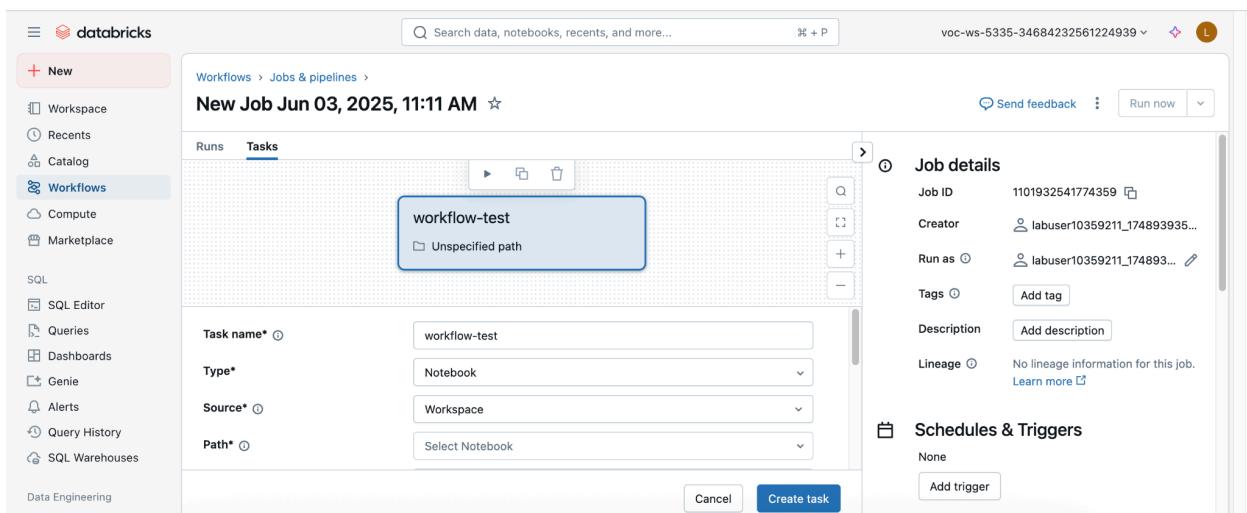
For the workflow we will reuse the Notebook 03 and attach it to be run as a workflow using serverless compute.

Go to the Workflows tab on the left side. Then select CREATE and select Job.



The screenshot shows the Databricks interface with the 'Workflows' tab selected in the sidebar. The main area displays a table with columns for 'Name', 'Type', and 'Tags'. A modal window is open over the table, titled 'Job' with the subtitle 'Orchestrate Pipelines, Notebooks, Queries, Models and more'. It contains three items: 'ETL pipeline', 'Ingestion pipeline', and 'No jobs or pipelines found.' The 'Create' button in the top right of the main area is highlighted with a red box.

This will open up a new window. Give the task a new name e.g. `workflow-test`

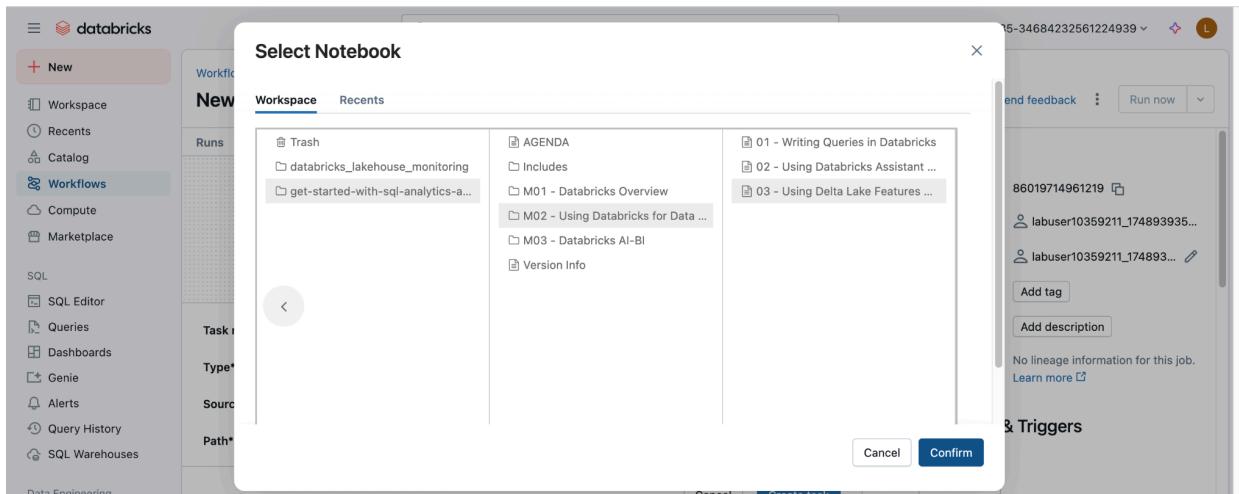


The screenshot shows the 'New Job' creation dialog. The 'Tasks' tab is selected. A task named 'workflow-test' is listed under 'Runs'. The 'Job details' section shows the following information:

- Job ID: 1101932541774359
- Creator: labuser10359211_174893935...
- Run as: labuser10359211_174893...
- Tags: Add tag
- Description: Add description
- Lineage: No lineage information for this job. Learn more

The 'Schedules & Triggers' section shows 'None' and an 'Add trigger' button.

Next, use the navigation to find the path to the notebook we want to run as a workflow. We will use the M02-03 notebook to be run using workflows.



Lastly, make sure serverless is selected as compute

New Job Jun 03, 2025, 11:11 AM ☆

Runs Tasks

Jobs Compute

- Serverless Autoscaling
- Add new job cluster
- Existing All-Purpose Compute
- Existing SQL Warehouses

Path* shared_warehouse SQL Serverless Autoscaling

Compute* Serverless

Dependent libraries + Add

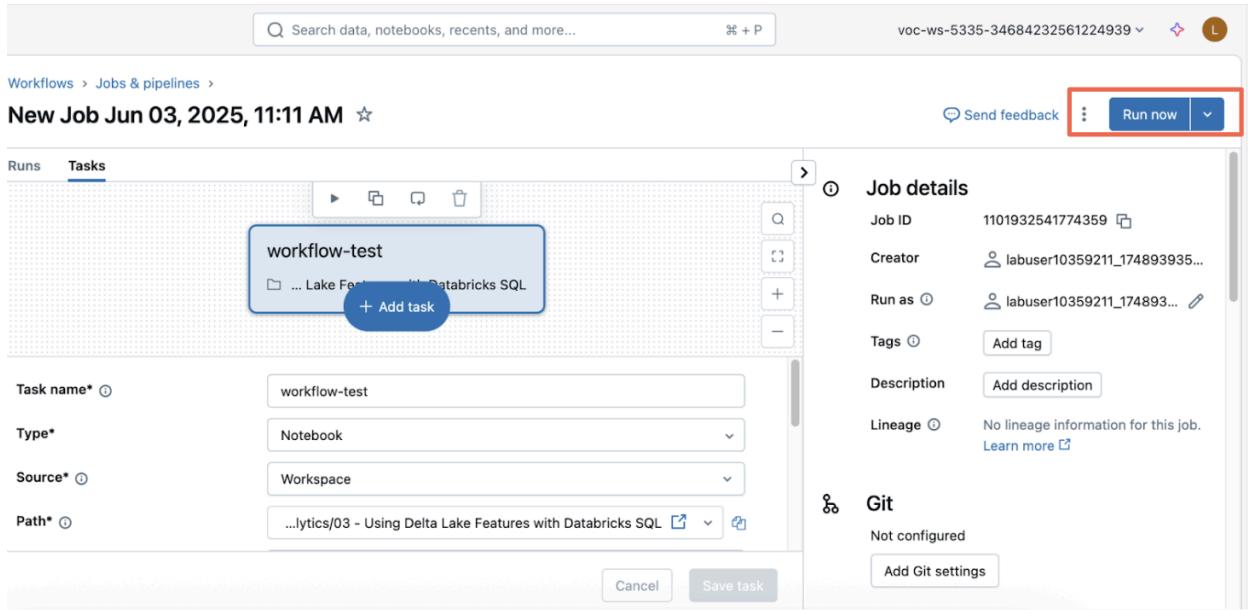
Parameters

Key	Value

UI JSON

Create task

Finally, click on Create Task and then click on Run now in the upper right corner.



The screenshot shows the Databricks UI for creating a new job. On the left, there's a grid where a task named "workflow-test" is visible. Below it, the "Tasks" tab is selected, showing fields for "Task name*", "Type*", "Source*", and "Path*". To the right, the "Job details" section is expanded, showing the Job ID (1101932541774359), Creator (labuser10359211_174893935...), Run as (labuser10359211_174893...), Tags, Description, and Lineage (No lineage information for this job). At the bottom right of the "Job details" section, there's a "Git" section with a note that it's not configured and a "Add Git settings" button. The top right of the screen has a "Run now" button, which is highlighted with a red box.

Once you run it, you can look at the updated page to see the result of the workflow.

Notice the time it takes to run the workflow: For this workflow it took around 2:30 minutes to run it. There is no provisioning of the cluster needed and no compute setup. Running workflows has been made easier and faster through serverless.

DLT on serverless

Serverless for Delta Live Tables (DLT) is a fully managed compute offered by Databricks that eliminates the need to manage clusters manually. It dynamically provisions resources as needed and automatically scales based on pipeline needs.

Follow the steps below to create a DLT pipeline using Serverless compute:

1. Download the linked .dbc file ([Link](#))
2. Navigate to your Databricks workspace and import the downloaded dbc file.

The screenshot shows the Databricks workspace interface. On the left, there's a sidebar with various navigation options like New, Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, Data Engineering, Machine Learning, and Playground. The main workspace area shows a tree view of 'Workspace' containing 'Home', 'get-started-with-sql-analytics-and-bi-on...', and 'get-started-with-sql-analytics-and-bi-on-databricks-4.2.0'. Inside the second folder, there are 'Repos', 'Shared', 'Users', 'Favorites', and 'Trash'. A search bar at the top says 'Search data, notebooks, recents, and more...'. To the right, a table lists a single item: 'get-started-with-sql-analytics-and-bi-on-databricks-4.2.0' (Type: Folder, Owner: db sp, Created at: Jun 02, 2025, 10:42 PM). Below this is an 'Import' dialog box with the title 'Import'. It has two radio buttons: 'File' (selected) and 'URL'. The target folder is set to '/Users/labuser10430623_1748896891@vocareum.com'. A preview window shows a file named 'serverless_bootcamp_dlt' with a size of 0.3 MB. At the bottom of the dialog are 'Close' and 'Import' buttons.

3. After importing the file, you will see a folder named "serverless_bootcamp_dlt" in your workspace.
4. Inside the folder, you will find two notebooks:
 - a. *00_setup_for_dlt* : used to set up the required tables.
 - b. *01_dlt_pipeline* : contains the logic for the Delta Live Tables pipeline.

The screenshot shows the Databricks workspace interface. The sidebar and search bar are identical to the previous screenshot. The main workspace area shows the 'serverless_bootcamp_dlt' folder expanded, revealing its contents: '00_setup_for_dlt' and '01_dlt_pipeline'. Both are listed as Notebooks owned by 'labuser10430623_1748896891@vocareum.com' and created on Jun 02, 2025, at 10:43 PM. The rest of the workspace interface is visible but mostly empty.

5. Open the setup notebook (*00_setup_for_dlt*).
6. Connect the notebook to Serverless compute and update the following parameters:
 - a. catalog -> Leave this as "dbacademy" (no changes needed)
 - b. Schema -> Replace the placeholder with your <labuser_id>
7. Once you've made these updates, you're ready to proceed – click **Run All**.

Databricks

00_setup_for_dlt Python Tabs: OFF

File Edit View Run Help

Creating bronze tables for DLT Pipeline from samples.tpch.orders and samples.tpch.cust

```

2
catalog = "dbacademy"
schema = "labuser10430623_1748851230"

3
# clone sample data from samples.tpch.customers and orders table
sql("create table if not exists {catalog}.{schema}.customers_raw deep clone samples.tpch.customer;")
sql("create table if not exists {catalog}.{schema}.orders_raw deep clone samples.tpch.orders;")

DataFrame[source_table_size: bigint, source_num_of_files: bigint, num_of_synced_transactions: bigint, num_removed_files: bigint, num_copied_files: bigint, removed_file_size: bigint, copied_files_size: bigint]

Checking the data loaded

5
order_raw = sql("select * from {catalog}.{schema}.orders_raw")
display(order_raw)

order_raw: pyspark.sql.connect.DataFrame = [o_orderkey: long, o_custkey: long ... 7 more fields]

```

	o_orderkey	o_custkey	o_orderstatus	o_totalprice	o_orderdate	o_orderpriority	o_clerk	o_shippriority	o_comment
1	13710944	227285	O	162169.66	1995-10-11	1-URGENT	Clerk#000000432	0	accounts.ruthie

8. Next, open notebook : [01_dlt_pipeline](#). Similar to setup notebook, locate the schema parameter and replace its value with your labuser_id.

01_dlt_pipeline Python Tabs: OFF

File Edit View Run Help

```

1
# DLT consists of three types of datasets:
# 1. Streaming Tables (Permanent/Temporary) : Used for Append Data Sources, Incremental Data
# 2. Materialized Views : Used for transformations, aggregations and computations
# 3. Views : Used for intermediate transformations, not stored in the target schema

import dlt

2
catalog = "dbacademy"
schema = "labuser10430623_1748851230"

3
# Creating a Streaming Table for the orders_raw dataset

@dlt.table(
    table_properties = {"quality": "bronze"},
    comment = "Bronze Order Table",
    name = "orders_bronze"
)
def orders_bronze():
    df = spark.readStream.table(f"{catalog}.{schema}.orders_raw")
    return df

```

9. Congratulations, your setup is complete!

10. Now, let's create a DLT pipeline. To do this, Navigate to "Pipelines" tab and then click on "ETL Pipeline".

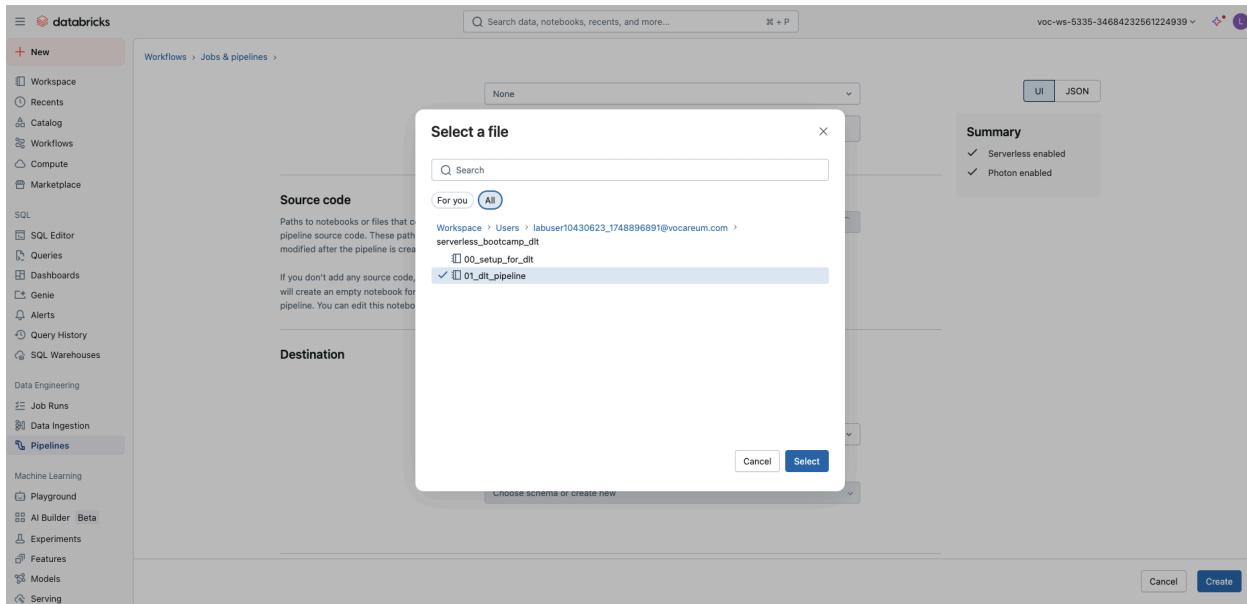
The screenshot shows the Databricks interface with the 'Workflows' section selected. The left sidebar has a 'Pipelines' item highlighted with a red box. The main area shows a table with columns: Name, Type, Tags, Run as, and a 'Tri' icon. There is a note below stating 'No pipelines found.' On the right, there is a 'Create' button and a 'Send feedback' link. A tooltip for the 'ETL pipeline' option is shown, describing it as 'Build and run an ETL pipeline with SQL and Python'.

11. Give your pipeline a meaningful name. Select **Serverless** as the compute type - this removes the overhead of manually setting up and managing clusters).

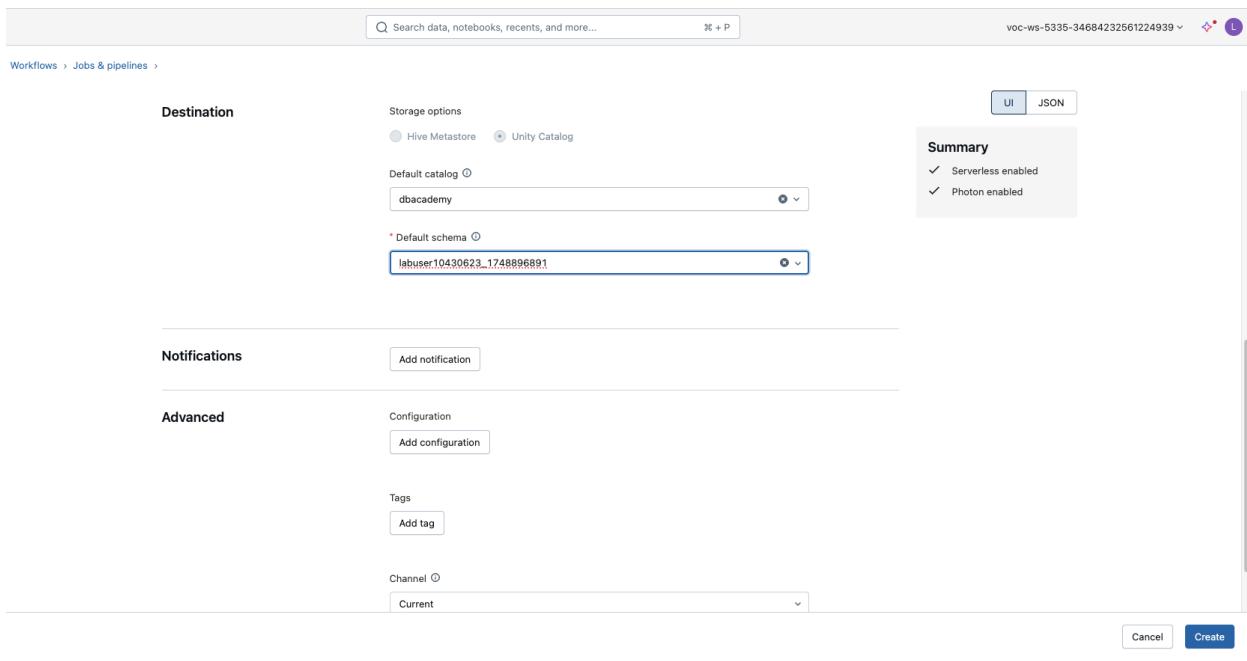
The screenshot shows the 'Create pipeline' form. In the 'General' section, the 'Pipeline name' field contains 'order_pipeline' and the 'Serverless' checkbox is checked. In the 'Summary' section, 'Serverless enabled' and 'Photon enabled' are checked. In the 'Budget' section, it says 'Serverless budget policy: None' and notes 'This pipeline does not have a budget policy associated with it'.

12. Next, In the **Source Code** section, select the **01_dlt_pipeline** notebook from your workspace.

This notebook contains the logic that defines your Delta Live Table Pipeline.



13. In the **Destination**, Set “**dbacademy**” as the Default catalog and “**<your lab user>**” as the Default schema. Once these are set, click “**Create**” to finalize the pipeline setup.



14. Congratulations, Your DLT Pipeline is ready!

15. To validate that everything is set up correctly, Click on the down arrow next to the Start button and select “**Validate**” from the dropdown menu.

The screenshot shows the Databricks interface for a pipeline named 'order_pipeline'. On the left is a sidebar with various navigation options like Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, and more. The main area is titled 'order_pipeline' and shows the pipeline's details. At the top right, there are tabs for Development, Production, Settings, Schedule, Share, and Start. Below these are buttons for 'Full refresh all' and 'Validate'. The pipeline details section includes fields for Pipeline ID (d096e679-189d-497b-9964-50cb44ecac2a), Pipeline type (ETL pipeline), Source code (a path to a notebook), Run as (labuser10430623_1748896891@vocareum.com), and Tags (None). A note at the bottom says 'A graph will be generated here once a pipeline update has started. Click "Start" to start an update.'

16. Once the validation is complete, you will see the pipeline's **DAG** displayed.

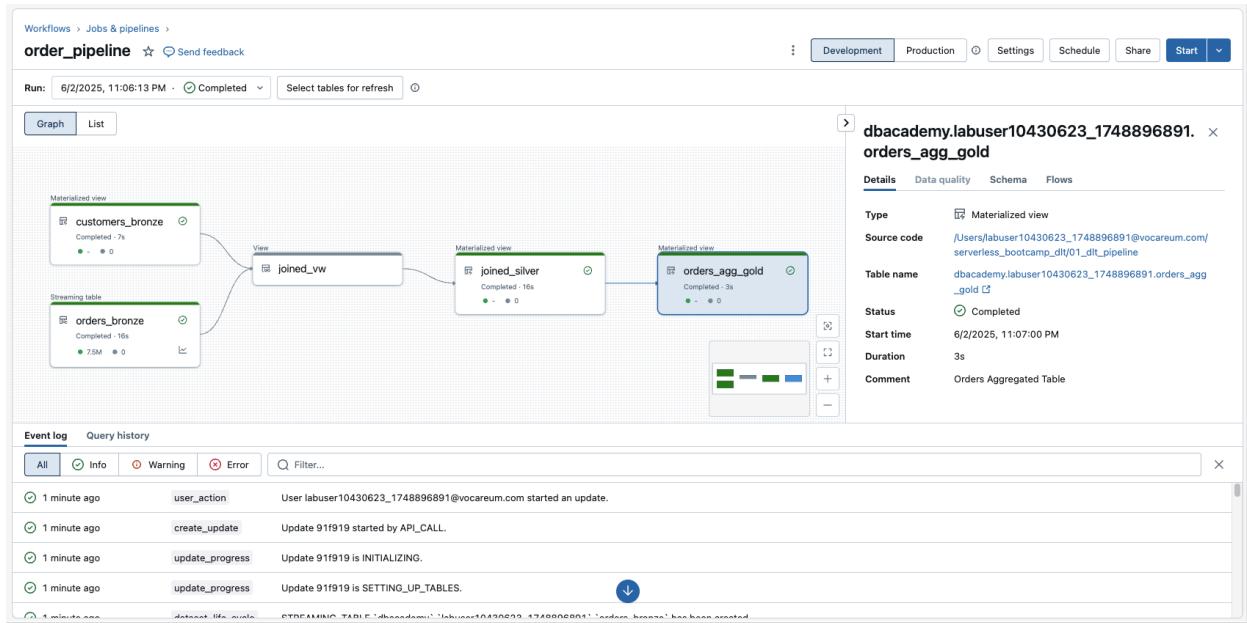
This visual representation outlines the flow and dependencies of your DLT tables as defined in the notebook.

The screenshot shows the same pipeline details page as before, but now the 'Graph' tab is selected. This displays a Data Flow Graph (DAG) for the pipeline. The nodes in the graph include 'customers_bronze' (Materialized view), 'joined_vw' (View), 'joined_silver' (Materialized view), 'orders_agg_gold' (Materialized view), and 'orders_bronze' (Streaming table). Arrows indicate the flow from 'customers_bronze' to 'joined_vw', 'joined_vw' to 'joined_silver', and 'joined_silver' to 'orders_agg_gold'. The 'orders_bronze' node is shown as a source for the 'joined_vw' view. The right side of the screen still shows the pipeline details and event log.

17. Now, go ahead and click **Start** to run your pipeline.

This will initiate the data loading process and execute the logic defined in the **01_dlt_pipeline** notebook.

Once the run is complete, you'll see the following results in the pipeline view, including table statuses, row counts, and a successful execution status.



Perfect! You've just created and executed a DLT pipeline in a matter of seconds using **Serverless** compute.

And the best part? You didn't have to wait for a cluster to spin up – everything ran seamlessly with fully managed infrastructure!

DWH on serverless

Serverless in all regards is a databricks managed compute option which takes away a lot of management and does performance optimizations in the background.

[SQL warehouses](#) are the compute resources that let you run SQL commands on data objects within Databricks SQL.

In order to explore Datawarehousing capabilities on Databricks we will navigate to the SQL Editor. A new Query window will pop up where you will be able to write SQL code. For this example, we'll create a VIEW.

The screenshot shows the Databricks interface with the SQL Editor selected. On the left, the sidebar includes options like Workspace, Recents, Catalog, Workflows, Compute, Marketplace, and SQL. Under SQL, the SQL Editor is highlighted. The main area shows a catalog tree with 'My organization' expanded, listing 'dbacademy', 'system', 'main', 'Delta Shares Received', 'samples', 'dbacademy_retail', 'Legacy', and 'hive_metastore'. A search bar at the top says 'Search data, notebooks, recents, and more...'. Below it, a query editor window titled 'New Query 2025-06-02 12:57pm' contains the SQL command: '1 SELECT or Option+Space or browse'. The results section below says 'No results available'.

When accessing and creating tables we need to follow the 3-level namespace as we are working with Unity Catalog. This means that we need to specify the table in the query using catalog.schema.table approach.

Databricks has built in [AI powered code completion](#) that helps you write your code. To see it in action let's join the customers data with the sales data and filter on products bought with prices over 500\$. Databricks Assistant Autocomplete will help you with that.

Start writing `SELECT * FROM dbacademy_retail.v01.customers as c JOIN` use the tab after autocomplete shows you the suggestion. Verify it aligns with the suggestion seen below:

This screenshot shows the same Databricks interface as above, but with the cursor placed after the 'JOIN' keyword in the query. The code completion dropdown is open, showing suggestions for 'dbacademy_retail.v01.sales' and 'dbacademy_retail.v01.sales as s'. The query editor now contains the following code:

```
1 | SELECT * from dbacademy_retail.v01.customers as c
2 | join dbacademy_retail.v01.sales as s
  on c.customer_id = s.customer_id
  where c.tax_code = 'CA'
```

To specify the filter on the price we'd need to change the where clause to :
`WHERE s.total_price > 500`

The complete query looks like this:

```
SELECT * from dbacademy_retail.v01.customers  
join dbacademy_retail.v01.sales on customers.customer_id=sales.customer_id  
where sales.total_price > 500
```

Hit Run and as you will see it will run on the serverless DWH.

Now let's create a view in our schema that was provided by the lab. For that we will use the following code - your schema name (labuser10359211_1748861767) will look different. After typing in dbacademy. You will be able to use the autocompletion to complete your schema name. Alternatively you can drag and hold the name of the schema from the catalog to the SQL Editor which will as a result copy the catalog.schema into it.

The screenshot shows the Databricks interface. On the left, the sidebar has 'New' selected under 'Workspace'. Under 'Catalog', there are sections for 'For you' and 'All'. A tree view shows 'My organization' with 'dbacademy' and 'information_schema' expanded. Below them is a schema named 'labuser10359211_1748861767' which is also expanded, showing 'No data', 'ops', 'system', 'main', 'Delta Shares Received', 'samples', 'dbacademy_retail', 'Legacy', and 'hive_metastore'. The 'dbacademy_retail' folder is highlighted with a blue selection bar. In the center, a 'New Query' window is open with the command 'Create VIEW dbacademy.labuser10359211_1748861767'. The results pane below says 'Raw results' and 'No results available'. At the top right, there are tabs for 'Run (1000)', 'dbacademy_retail', 'information_schema', and 'shared_workspace'. The status bar at the bottom indicates 'Serverless 2XS'.

```
Create VIEW dbacademy.labuser10359211_1748861767.customer2 AS  
Select * from dbacademy_retail.v01.customers Where state = "OR"
```

After successfully running the query, the view will be visible in your schema like below:

The screenshot shows the Databricks interface. On the left, the navigation sidebar includes 'New', 'Workspace', 'Recents', 'Catalog', 'Workflows', 'Compute', 'Marketplace', 'SQL', 'SQL Editor' (which is selected), 'Queries', 'Dashboards', 'Genie', 'Alerts', 'Query History', and 'SQL Warehouses'. The main area has a search bar at the top. Below it, there are two tabs: 'New Query 2025-06-02 12:57pm' and 'New Query 2025-06-02 1:06pm'. The 'New Query 2025-06-02 12:57pm' tab contains the following SQL code:

```
1 | Create VIEW dbacademy.labuser10359211_1748861767.customer2 AS
2 | Select * from dbacademy_retail.v01.customers Where state = "OR"
```

Below the queries, there's a 'Run (1000)' button, a 'Save' button, and 'Schedule' and 'Share' buttons. The 'Raw results' section shows the output of the first query. The catalog sidebar on the left shows a tree structure with 'My organization' expanded, showing 'dbacademy', 'information_schema', 'labuser10359211_1748939356' (which is expanded to show 'customer2', 'customers', 'customers_bronze', 'customers_raw', 'delta_students', 'high_sales', 'joined_silver', 'orders_agg_gold', 'orders_bronze', 'orders_raw'), 'ops', and 'system'. A tooltip 'Insert table name' is shown over the 'customer2' node.

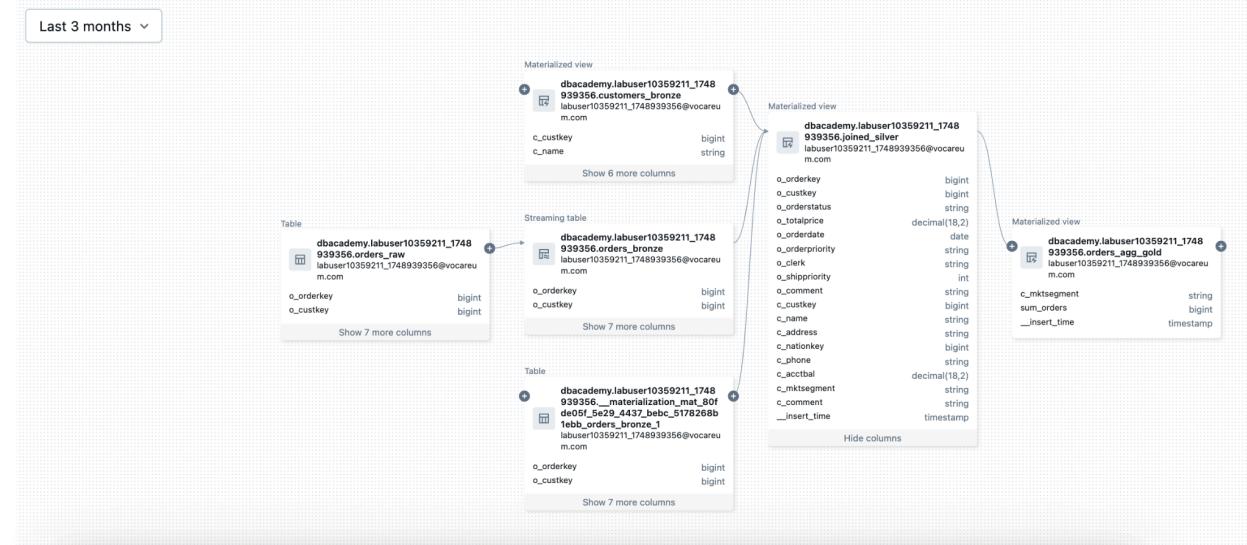
Explore newly created data

You can navigate to your catalog and explore the tables and views that you created during this hands-on lab.

The screenshot shows the Databricks Catalog interface. The sidebar on the left includes 'New', 'Workspace', 'Recents', 'Catalog' (which is selected), 'Workflows', 'Compute', 'Marketplace', 'SQL Editor', 'Queries', 'Dashboards', 'Genie', 'Alerts', 'Query History', 'SQL Warehouses', and 'Engineering'. The main area shows the 'Catalog' view for the 'shared_warehouse' Serverless 2XS workspace. The catalog tree shows 'My organization' expanded, with 'dbacademy' and 'labuser10359211_1748939356' expanded. Under 'labuser10359211_1748939356', the 'joined_silver' table is selected and highlighted with a blue bar. Other tables visible include 'customer2', 'customers', 'customers_bronze', 'customers_raw', 'delta_students', 'high_sales', 'orders_agg_gold', 'orders_bronze', and 'orders_raw'. The 'ops' and 'system' nodes are also present.

The materialized view joined_silver e.g. will show you the complete lineage now if you click on the lineage tab of the table. You can analyze the lineage on table and column.

Data Lineage for dbacademy.labuser10359211_1748939356.joined_silver



Explore Data using Genie

As a last exercise we will look into the capabilities of using natural language to explore our data. [AI/BI Genie](#) is a generative AI capability within Databricks that is tailored to your data. It can be improved through feedback and company dependent instructions. You can use AI/BI Genie in a dedicated Genie space or directly on the sampled data to explore the data ad hoc using natural language.

For the last part, we will look into AI/BI Genie from the sample data. Navigate to your catalog `dbacademy` and select your schema (`labuser_xx`). Select the `customer` table and go to [sample data](#). Click on the [Ask Genie](#) button.

The screenshot shows the Databricks Catalog Explorer interface. On the left, the Catalog sidebar lists databases like 'shared_warehouse' and 'dbacademy'. The 'customers' table under 'labuser10359211_1748939356' is selected. The main pane displays the 'Sample Data' tab of the 'customers' table, which contains 13 rows of data. The columns are: customer_id, tax_id, tax_code, customer_name, state, and city. The data includes entries such as SMITH, SHIRLEY from IN, BREMEN; STEPHENS, GERALDINE M from OR, ADDRESS; and many others from various states and cities.

	customer_id	tax_id	tax_code	customer_name	state	city
1	11123757	null	null	SMITH, SHIRLEY	IN	BREMEN
2	30585978	null	null	STEPHENS, GERALDINE M	OR	ADDRESS
3	349822	null	null	GUZMAN, CARMEN	VA	VIENNA
4	27652636	null	null	HASSETT, PATRICK J	WI	VILLAGE
5	14437343	null	null	HENTZ, DIANA L	OH	COLUMBI
6	20441596	null	null	TIRADO, MARCO A	NY	Otselic
7	5945686	null	null	SKORA, BRIAN S	MI	null
8	5385771	null	null	SLAWEK, DEAN J	PA	null
9	1427940	null	null	REAVES, LIONEL C	VA	HOT SPR
10	10457387	null	null	BONGIOVANNI, KELLY M	IN	VINCENN
11	19154815	null	null	VERTA, RAYMOND J	OH	CANTON
12	2469887	null	null	BRYANT, KATHRYN I	MA	WHITMA
13	11743397	null	null	UCZEN, MARK D	CA	Roseland

Ask the following questions:

1. How many unique customers are there?
2. What are the total units purchased by customers?
3. What is the total number of customers in each state?
4. Show me the distribution of units purchased by state and customer?
5. Create a chart out of it

Closing

We looked into different serverless compute options. We looked into serverless clusters running on Notebooks, Workflows using serverless compute and the comparison of creating a dedicated job compute, DLT and lastly Data Warehousing using serverless.

Huge Thanks for attending the Databricks Serverless Bootcamp, we hope you enjoyed the content and learned a lot, for further learning and information please visit

<https://www.databricks.com/learn>