

AgentBricks Workshop

| | |
|---|-----------|
| Introduction | 1 |
| Prerequisites | 2 |
| Access to Databricks workspace | 2 |
| Setting up Git Integration in your workspace | 2 |
| Clone & Fork Github repo (optional) | 3 |
| Launch your workspace in Vocareum | 4 |
| Create the Repo in your workspace | 6 |
| Start Lab 1 - Basic Knowledge Assistant | 7 |
| Lab Setup | 8 |
| Explore the data | 8 |
| STEP 1: Navigate to the Catalog Explorer | 9 |
| STEP 2: Explore the data objects | 9 |
| Follow the instructions in the notebook | 10 |
| Start Lab 2: Building Multi Agent Systems with Agent Bricks (including Knowledge Assistant) | 11 |
| Lab Setup | 12 |
| Explore the data | 12 |
| Follow the instructions in the notebook | 15 |
| Backup environment and lab (if Vocareum doesn't work): Databricks Free Edition | 21 |
| Create the Repo in your workspace | 21 |
| Start the Lab | 21 |
| Agents Workshop | 22 |
| Agent Bricks Workshop (Once Agents are available in free edition, in 2026, in the meantime you can use other workshops under tmm, like Agents Workshop above) | 23 |

Introduction

This document serves as a comprehensive **User Guide** for the **AgentBricks Workshop**. It is designed to walk participants through the entire setup process for the workshop environment, primarily leveraging a Databricks workspace provided through the Databricks Academy via **Vocareum**.

The guide covers essential prerequisites, including:

- Access to a Databricks workspace and setting up Git integration.
- Instructions for cloning and forking the main GitHub repository.
- Detailed steps for launching your workspace in Vocareum and creating the necessary repository.

Following the setup, the guide provides an outline for two main hands-on labs:

- Lab 1: Basic Knowledge Assistant
- Lab 2: Building Multi Agent Systems with Agent Bricks (including Knowledge Assistant)

Prerequisites

During the workshop, participants will have access to a Databricks workspace spin up through the Databricks Academy. The following prerequisites would only be needed if you would like to follow this guide in your own environment.

Access to Databricks workspace

Please refer to below if you don't have a Databricks Workspace or [here](#) for Lighthouse access later in the year.

- [Get started: Account and workspace setup | Databricks on AWS](#)
- [Get started: Account Get started with Databricks and workspace setup - Azure Databricks | Microsoft Learn](#)
- [Get started: Account and workspace setup | Databricks on Google Cloud](#)

Setting up Git Integration in your workspace

Here is how you set up a Github integration by clicking your profile at the top right corner in your Databricks Workspace , and then go to **Settings** -> **Linked accounts** -> **Git provider** -> Personal access token, input your Token and then click Save as below.

For more detail and on how to create a Github token, please refer to [Configure Git credentials & connect a remote repo to Databricks](#)

Settings

Workspace admin

Appearance

Identity and access

Security

Compute

Development

Notifications

Advanced

User

Profile

Preferences

Developer

Linked accounts

Notifications

Linked accounts

Connect your Databricks account to other services

Git integration

With co-versioned repo

Databricks Git folders and Repos allow you to clone a remote Git repo, which you can specify when you add a Git folder. [Learn more](#)

With individual notebooks

Although we recommended using co-versioned repo for Git integration, Databricks supports individual notebook version control integration with [GitHub](#), [Bitbucket Cloud](#), or [Azure DevOps Services](#) (using AAD authentication only).

Set your Git provider and credentials

You can also set your Git provider credentials via API. [Learn more](#)

Git provider

GitHub

Link Git account
Simple setup in a few clicks to link your GitHub account. [Learn more](#)

Personal access token
Use an existing personal access token to authenticate Git requests.

Git provider username or email

dbsys21

Token
To generate a GitHub personal access token, follow the [GitHub documentation](#). The token must have the "repo" and "workflow" (if your repository has GitHub Action workflows) scopes.

.....

Save **Cancel**

Clone & Fork Github repo (optional)

In your github repository, please fork the [main repo](#), so that you could work with the forked repo when required. Here is how you could fork the repo in Github.

miguelpera-db / agentbricks-workshop

Type to search

Code Issues Pull requests Actions Projects Security Insights

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner * MiguelPeralvo / **Repository name *** agentbricks-workshop agentbricks-workshop is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description agentbricks-workshop 20 / 350 characters

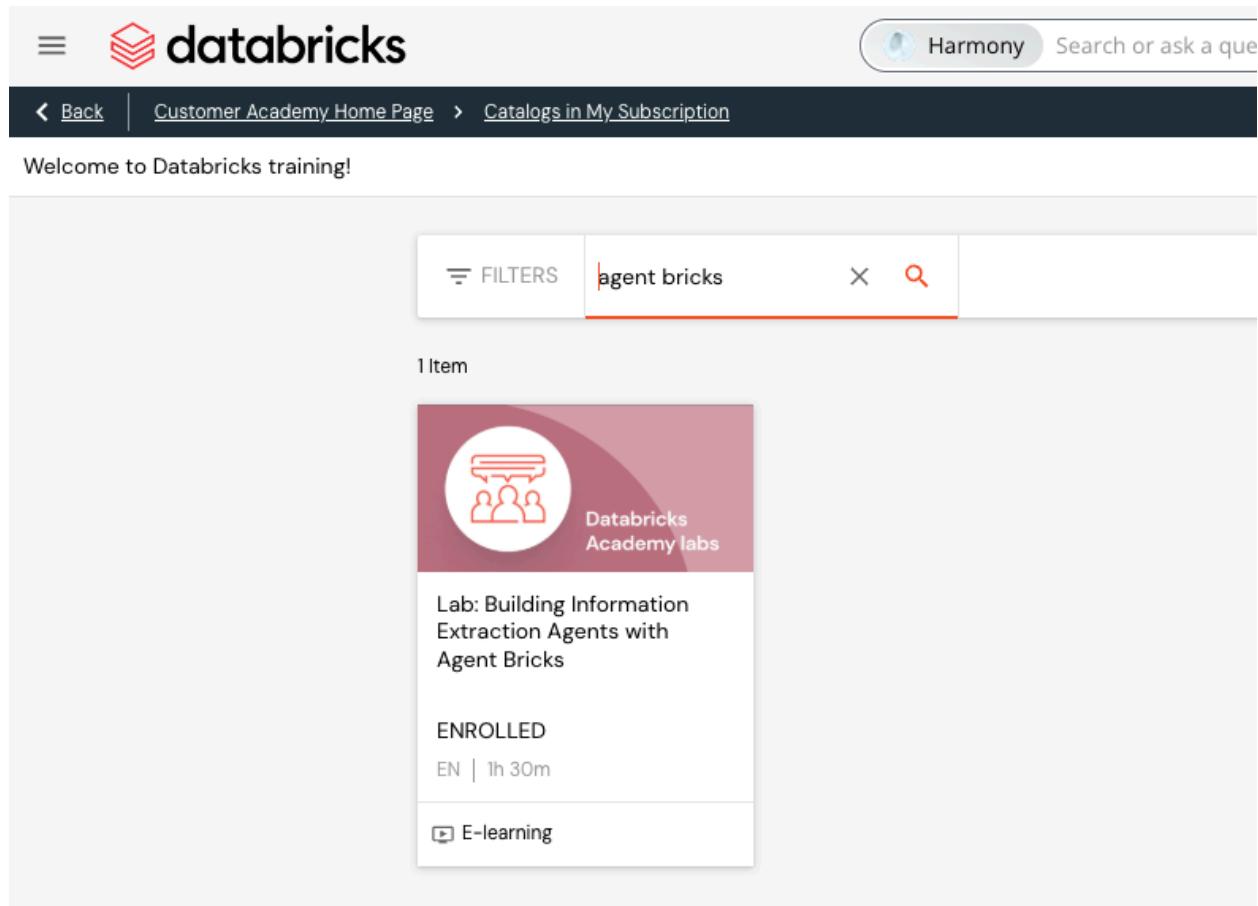
Copy the main branch only Contribute back to miguelpera-db/agentbricks-workshop by adding your own branch. [Learn more](#).

(1) You are creating a fork in your personal account.

Create fork

Launch your workspace in Vocareum

Browse the “Catalogs in my subscription” and look for the course “agent bricks” as below:



The screenshot shows the Databricks Customer Academy Home Page. At the top, there is a navigation bar with a back button, a customer icon, the text "databricks", and a search bar labeled "Search or ask a que". Below the navigation bar, the page displays a search result for "agent bricks". The search bar has "agent bricks" typed into it. A single item is listed under the heading "1 Item". The item is titled "Databricks Academy labs" and features a circular icon with three people and a speech bubble. The description of the item is "Lab: Building Information Extraction Agents with Agent Bricks". It is marked as "ENROLLED" and "EN | 1h 30m". Below the item, there is a link labeled "E-learning".

Or alternatively follow this link:

<https://customer-academy.databricks.com/learn/courses/4599/lab-building-information-extraction-agents-with-agent-bricks/lessons>

Then go to task “Lab: Building Information Extraction Agents with Agent Bricks” - you will get a new tab with your Vocareum workspace after a few minutes.

[Back](#)[Customer Academy Home Page](#)[My Courses and Learning Plans](#)[Lab: Building Information Extraction Agents with Agent Bricks](#)

Welcome to Databricks training!

Lab: Building Information Extraction Agents with Agent Bricks

 E-learning • English •  Course completed

Syllabus

2 Lessons • 1hr 30min



Lab: Building Information Extraction Agents with Agent Bricks



Completed

Accessing Vocareum Labs in Your Training Slides

This content walks you through the process on how to launch your Vocareum lab environment from Databricks Academy and how to enable pop-ups from Databricks Academy.



Completed

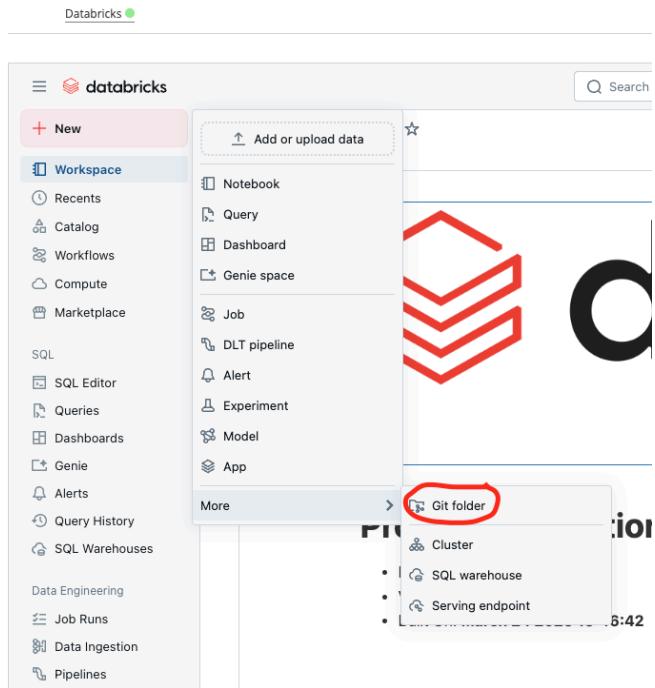
Lab: Building Information Extraction Agents with Agent Bricks

LTI

The workspace will last for 2 hours, but if you need it you can relaunch the workspace by clicking on "Retake the lesson".

Create the Repo in your workspace

In your workspace, navigate to **+ New -> Git Folder** as below.

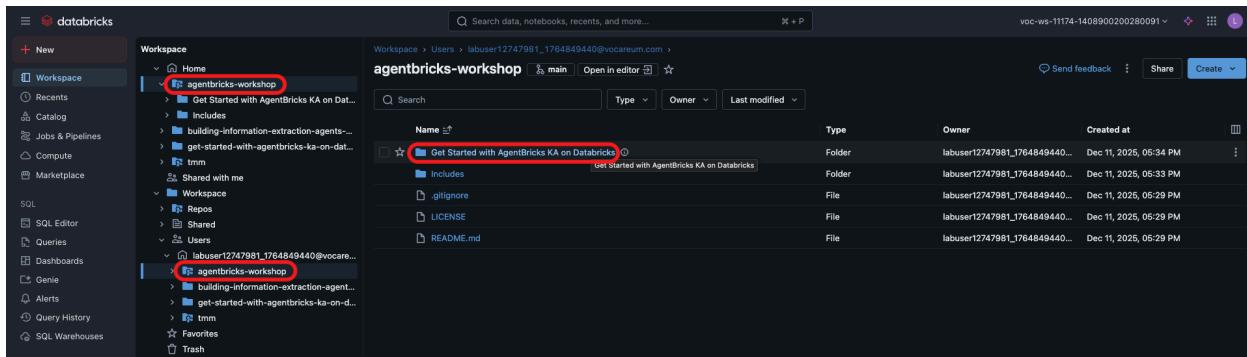


Once you click Git Folder, you will be directed to the screen below, simply input the **Git repository URL**: <https://github.com/miguelpera-db/agentbricks-workshop>, and click **Create Git Folder**.

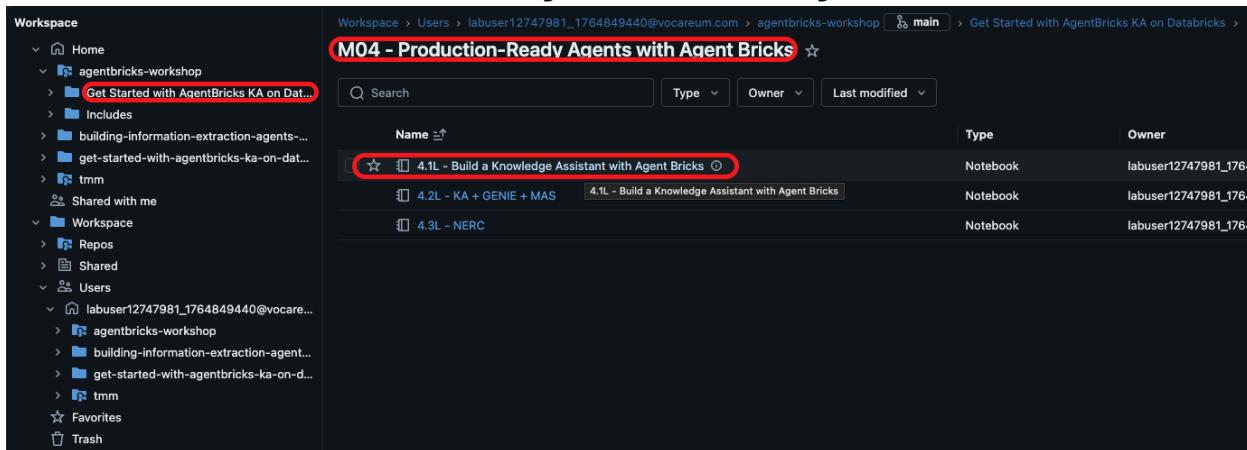
A screenshot of the 'Create Git folder' dialog box. It has fields for 'Git repository URL' (containing 'https://github.com/miguelpera-db/agentbricks-workshop') and 'Git provider' (set to GitHub). Below these are fields for 'Git folder name' (containing 'agentbricks-workshop') and 'Sparse checkout mode' (unchecked). At the bottom are 'Cancel' and 'Create Git folder' buttons.

Start Lab 1 - Basic Knowledge Assistant

Once the repo is created, navigate to the repo **agentbricks-workshop** (both under your home or User in the Users section), directory "**Get Started with AgentBricks KA on Databricks**".



Now go to the subdirectory "M04 - Production-Ready Agents with Agent Bricks", and click on the notebook **4.1L - Build a Knowledge Assistant with Agent Bricks**.



You will be navigated to the the notebook contents below:



Lab Setup

In the notebook, run setup by executing the following two commands in your notebook cells.

```
%run "../../Includes/Workspace-Setup"
```

```
%run "../../Includes/Classroom-Setup-4.1L"
```

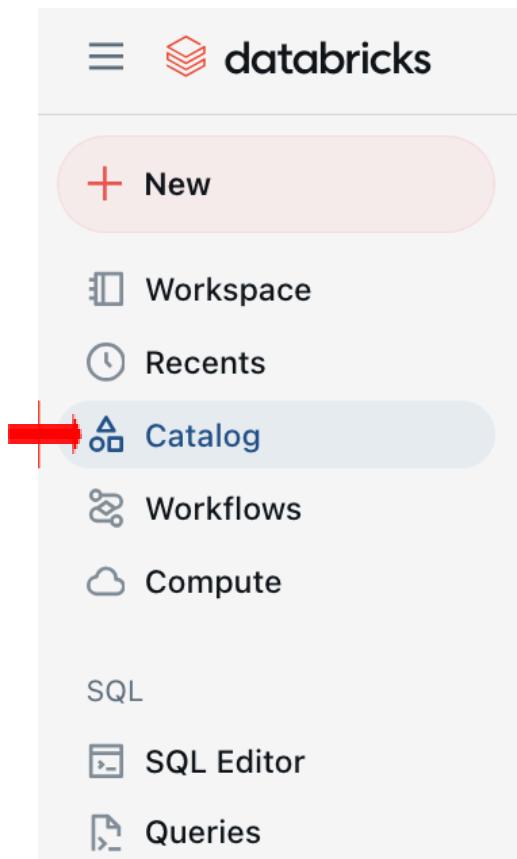
Explore the data

The [Catalog Explorer](#) (Catalog) provides a space to explore and manage data, schemas (databases), tables, permissions, and models.

The data explorer is the main UI for the [Unity Catalog object model](#). Here, you can view schema details, preview sample data, see table details and properties, and explore lineage.

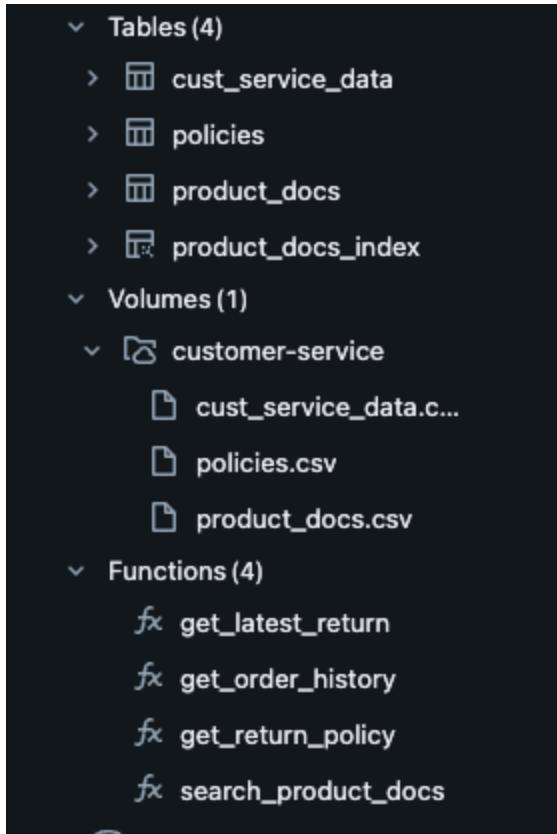
Admins can change owners and grant and revoke permissions.

STEP 1: Navigate to the Catalog Explorer



STEP 2: Explore the data objects

In the Catalog page, you will be able to see the catalogs you have access to. In your vocareum environment, you can only write into schema **labuser{nnn}-{nnn} under catalog dbacademy**. Once the setup script is complete, you should see, at a minimum, the following objects under your schema:



Follow the instructions in the notebook

Start with section A:

A. Create the Knowledge Assistant

Please follow along, but this part of the lab will be performed by you so you can query the Agent Brick deployed within this Workspace.

1. In the left nav, go to AI / ML → Agents → Agent Bricks → Knowledge Assistant. (Or search "Knowledge Assistant" in the workspace search.)
2. On **Configure**, fill in:
 - **Name:** my-product-agent
 - **Description:** An agent used to answer questions regarding customer product information.
3. Add **knowledge source** (the core of your RAG):
 - **Type:** Vector Search Index
 - **Source:** product_docs_index
 - **Doc URI Column:** product_id
 - **Text Column:** indexed_doc
 - **Describe the content:** Product documentation
4. (Optional) **Instructions:** Style/guardrails for answers (tone, citation behavior, etc.).
5. Legacy (skipped)

A1. WAITING FOR THE KNOWLEDGE ASSISTANT

6. The right-side panel will show build/sync progress and, once ready, links to the **deployed endpoint**, **experiment**, and **synced sources**. (Initial build/sync can take a while.)

| Please allow for 5-10 minutes for the agent to be deployed.

A2. Using the Knowledge Assistant

7. Navigate to the Playground using the left menu.
8. Using the model name dropdown menu at the top, select your model. You can search for `my-product-agent`.
9. Start querying! Here are some sample questions you can ask
 - "Can you tell me about the BlendMaster Elite 4000?"

Note the citations presented as a part of the output. This brings in the power of Agent Bricks; your agent is connected to your data and completely governed by Unity Catalog. It's in this way that your agent lives alongside your data.

A3. Explore the Dataset and Test your Brick

You can reference different product names by running the cell below.

```
%sql  
select product_name from product_docs where product_name is not null
```

Continue with the B, C and D sections if you have enough time.

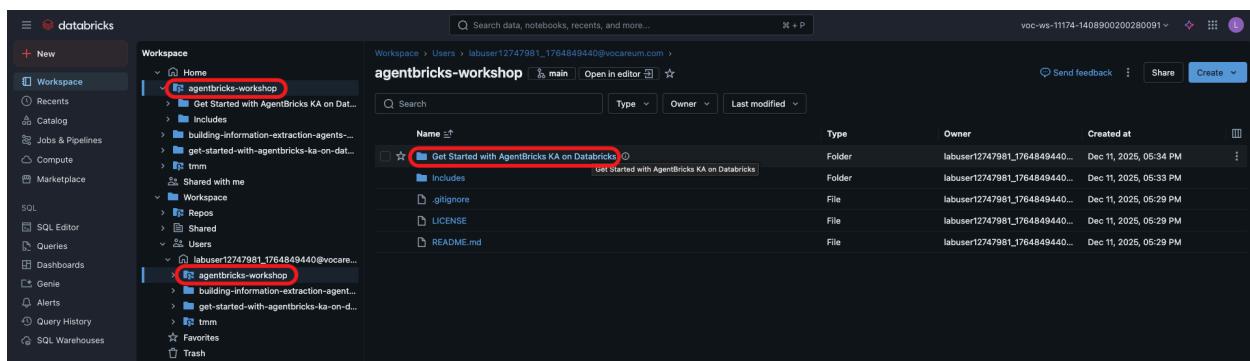
And finally remember to clean up:

Clean Up

If you *did* create an agent using the instructions above, please navigate to your agent using the **Agents** menu on the left side of the workspace, select the three vertical dots, and select **Delete**. This helps preserve resources for other users in this lab environment.

Start Lab 2: Building Multi Agent Systems with Agent Bricks (including Knowledge Assistant)

Navigate again to the repo **agentbricks-workshop** (both under your home or User in the Users section), directory "**Get Started with AgentBricks KA on Databricks**".



Now go to the subdirectory "M04 - Production-Ready Agents with Agent Bricks", and click on the notebook **4.2L - KA + GENIE + MAS**

The screenshot shows a Databricks workspace interface. On the left, there's a sidebar with options like Home, agentbricks-workshop, Get Started with AgentBricks KA on Dat..., Includes, building-information-extraction-agents..., get-started-with-agentbricks-ka-on-dat..., tmr, Shared with me, Workspace, Favorites, and Trash. The main area is titled 'M04 - Production-Ready Agents with Agent Bricks'. It contains a table with three rows:

| Name | Type | Owner | Created at |
|--|----------|-----------------------|-------------------------|
| 4.1L - Build a Knowledge Assistant with Agent Bricks | Notebook | labuser12747981_17... | Dec 11, 2025, 05:31 ... |
| 4.2L - KA + GENIE + MAS | Notebook | labuser12747981_17... | Dec 11, 2025, 05:31 ... |
| 4.3L - NERC | Notebook | labuser12747981_17... | Dec 11, 2025, 05:31 ... |

You will be navigated to the the notebook contents below:

The screenshot shows a Databricks notebook titled '4.2L - KA + GENIE + MAS'. The notebook content includes:

- The Databricks logo with the text 'databricks Academy'.
- The title 'Agent Bricks Lab: Build, Orchestrate, and Improve Multi-Agent Systems'.
- A description: 'In this lab, you'll learn how to create and refine AI agents using Databricks Agent Bricks. You'll start by building a Knowledge Assistant grounded in company product docs and historical support tickets, then expand it with a Genie-powered structured data agent. Finally, you'll orchestrate them together with a Multi-Agent Supervisor and guide the system to produce better, user-friendly responses.'
- A code cell with the command '%run ".../.../Includes/bricks_setup"' and a timestamp '2 days ago (4m)'.

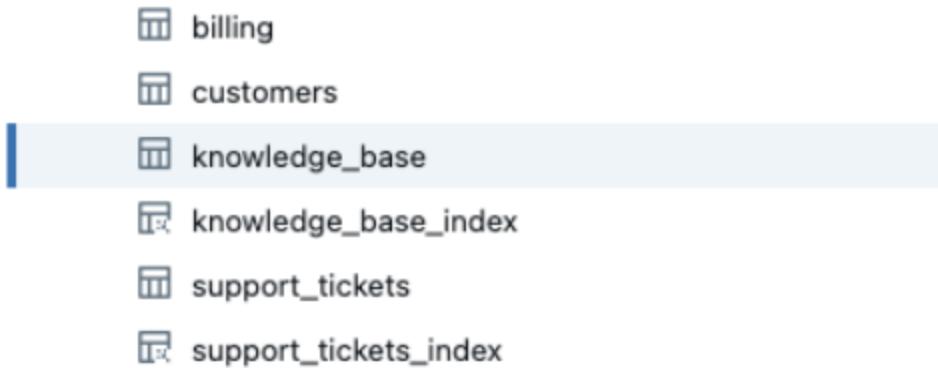
Lab Setup

In the notebook, run setup by executing the following two commands in your notebook cells.

```
%run ".../.../Includes/bricks_setup"
```

Explore the data

Once the setup script is complete, you should see, at a minimum, the following objects under your schema:



If you take a look at the knowledge_base table, for example, you should see the following:

The screenshot shows the 'knowledge_base' table details page. The table has 17 rows of data. The columns are:

| # | kb_id | content_type | category | subcategory | title |
|----|---------|--------------|-----------|--|---|
| 1 | KB-1001 | Policy | Billing | Unexpected charges explanation | Unexpected Charges Explanation Policy |
| 2 | KB-1002 | FAQ | Billing | Paper bill vs. electronic billing | What's the difference between paper bills and electronic billing? |
| 3 | KB-1003 | Policy | Account | Changing account details | Account Management Policy: Changing Account Details |
| 4 | KB-1004 | FAQ | Billing | Bill payment options | What are my bill payment options? |
| 5 | KB-1005 | FAQ | Billing | Prepaid vs. postpaid billing differences | What's the difference between prepaid and postpaid billing? |
| 6 | KB-1006 | Procedure | Account | Account security | Account Security Management Procedure |
| 7 | KB-1007 | Policy | Technical | International roaming setup | International Roaming Setup Policy |
| 8 | KB-1008 | FAQ | Account | Multi-line account management | How Do I Manage Multiple Lines on My Account? |
| 9 | KB-1009 | FAQ | Billing | Billing error resolution | How Do I Resolve an Error on My Bill? |
| 10 | KB-1010 | FAQ | Technical | Bluetooth connectivity problems | Bluetooth Connectivity Problems |
| 11 | KB-1011 | Policy | Billing | Payment methods | Payment Methods Policy |
| 12 | KB-1012 | Policy | Technical | 5g connection troubleshooting | 5G Connection Troubleshooting Policy |
| 13 | KB-1013 | FAQ | Billing | Payment due dates | When is my payment due? |
| 14 | KB-1014 | Procedure | Account | Port-out requirements | Port-Out Management Procedure |
| 15 | KB-1015 | Procedure | Technical | Device blacklist removal | Technical Procedure: Device Blacklist Removal |
| 16 | KB-1016 | Guide | Account | Switching plans | User Guide: How to Switch Your Telecom Plan |
| 17 | KB-1017 | FAQ | Technical | 5g connection troubleshooting | 5G Connection Troubleshooting |

You should also be able to see any vector search index that has been created on top of that table in the table lineage:

Screenshot of the Catalog Explorer interface showing the lineage of the 'knowledge_base' asset.

The left sidebar shows the navigation path: Catalog > agent_XS Serverless > knowledge_base.

The main view displays the 'Lineage' tab, which lists the assets connected to the 'knowledge_base'. The results are filtered by 'Up and Downstream' direction and 'Last 3 months' time period.

Key columns in the lineage table include:

- Name
- Direction
- Type
- Last activity

The lineage graph shows the following connections:

- knowledge_base (Downstream) -> 4.2L - KA + GENIE + MAS (Upstream)
- knowledge_base (Downstream) -> 4bb52203-5b14-445d-85c8-dd1dde27dc24 (Downstream)
- knowledge_base (Downstream) -> kbi (Downstream)
- knowledge_base (Upstream) -> 4.2L - KA + GENIE + MAS (Upstream)
- knowledge_base (Upstream) -> 4.1L - Build a Knowledge Assistant with Agent Bricks (Upstream)
- knowledge_base (Upstream) -> t454680d-a3fa-4778-83c2-a1da59adc8aa (Downstream)
- knowledge_base_index (Downstream) -> knowledge_base (Upstream)
- bricks_setup (Upstream) -> knowledge_base (Upstream)

A red box highlights the 'See lineage graph' button in the top right corner of the lineage table.

Vector search index

| | |
|------------------------|--------|
| kb_id | string |
| content_type | string |
| category | string |
| subcategory | string |
| title | string |
| content | string |
| tags | string |
| last_updated | string |
| formatted_content | string |
| __db_formatted_cont... | array |

Table

dbacademy.labuser12747981_1764
849440.knowledge_base
labuser12747981_1764849440@vocare
um.com

| | |
|-------------------|--------|
| kb_id | string |
| content_type | string |
| category | string |
| subcategory | string |
| title | string |
| content | string |
| tags | string |
| last_updated | string |
| formatted_content | string |

Hide columns

Vector search index

| | |
|------------------------|--------|
| kb_id | string |
| content_type | string |
| category | string |
| subcategory | string |
| title | string |
| content | string |
| tags | string |
| last_updated | string |
| formatted_content | string |
| __db_formatted_cont... | array |

dbacademy.labuser12747981_1764
849440.knowledge_base_index
labuser12747981_1764849440@vocare
um.com

Hide columns

Follow the instructions in the notebook

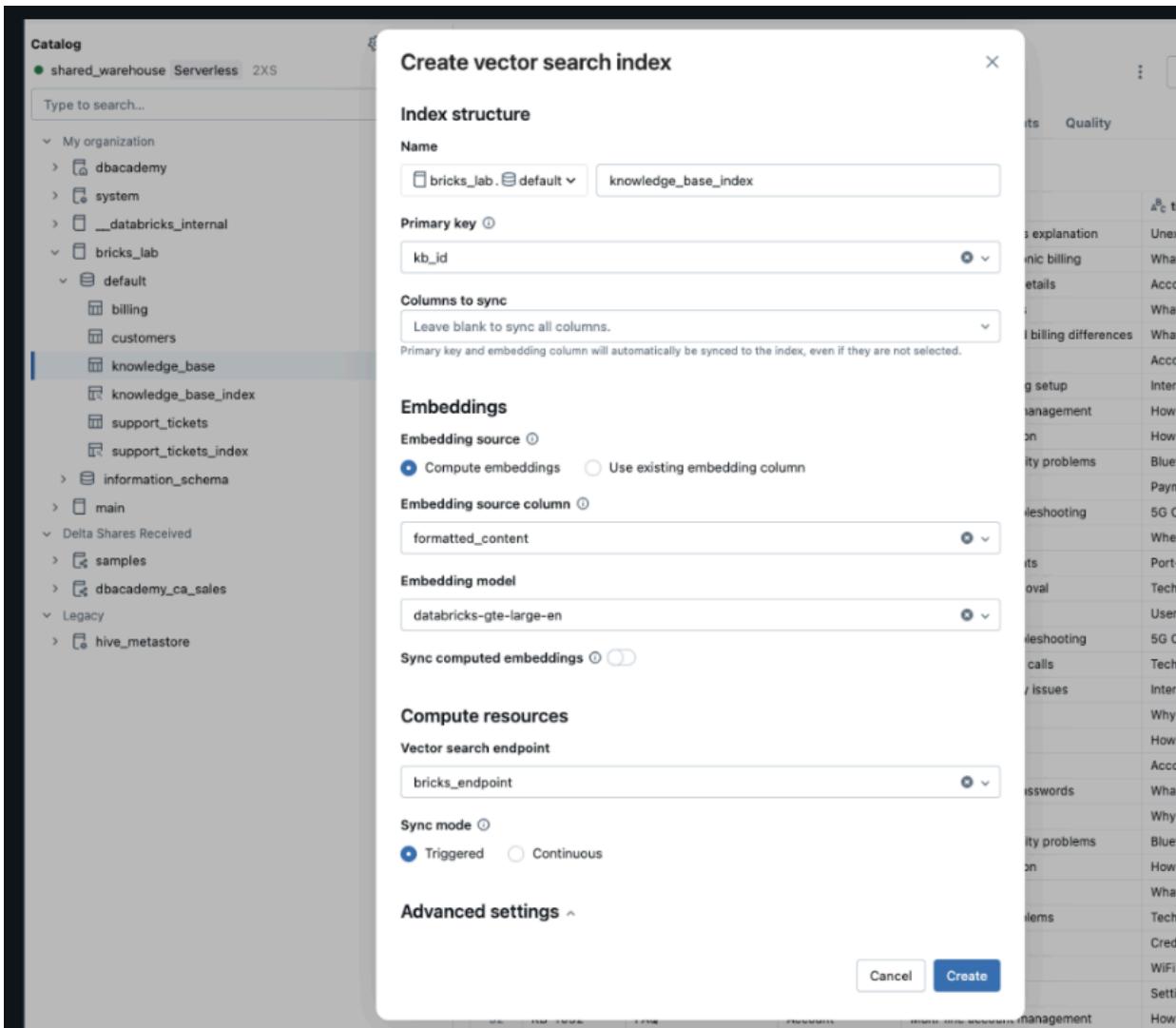
Part 1: Build Your First Knowledge Assistant

1.1 Create a Vector Search Index

- **Why Vector Search?**
 - Provides efficient retrieval of relevant chunks of data for grounding LLM responses.
 - Two common types:
 - **Triggered updates** (for static knowledge bases like FAQs/policies).
 - **Continuous updates** (for dynamic sources like support tickets).
- **Demo:** Indexes are pre-built for this lab, but you'll see how easy it is to create one.

For the Knowledge Assistant sections, you may want to use this text:

- **Overview:** Knowledge Assistant turns your documents - like PDFs, word files, and more-into a high-quality Q&A chatbot. It uses advanced AI to deliver accurate, reliable answers from your content with no coding or complex setup required. Ideal for product documentation, customer support knowledge bases, regulatory documents, and more. Agent Bricks may use endpoints hosted on Databricks Inc.
- **Basic Info:**
 - **Name:** BricksLab-TechnicalSupport
 - **Description:** This agent provides customer support for a Telecommunications company and references multiple knowledge sources.



1.2 Build the Knowledge Assistant Agent

- Navigate to "Agents" in the UI.
- Create a new Knowledge Assistant using the two pre-built vector search indices:
 - Knowledge Base – Company details, FAQs, policies, and procedures.
 - Support Tickets – Historical tickets and their resolutions.
- Example setup:
 - Name: [your_initials]-BricksLab-TechnicalSupport
 - Description: Provides telco product support using company docs and historical tickets.

The screenshot shows the 'Configure Knowledge Sources' dialog for the 'BricksLab-TechnicalSupport' agent. It lists two sources: 'knowledge_base_index' (Type: Vector Search Index, Doc Uri Column: 'kb_id', Text Column: 'formatted_content') and 'support_tickets_index' (Type: Vector Search Index, Doc Uri Column: 'ticket_id', Text Column: 'formatted_content'). Below the dialog, the 'Basic Info' section of the agent configuration is visible, showing the agent's name, description, and a note about providing customer support for a telecommunications company.

1.2 Build the Knowledge Assistant Agent

- Navigate to "Agents" in the UI.
- Create a new Knowledge Assistant using the two pre-built vector search indices:
 - Knowledge Base – Company details, FAQs, policies, and procedures.
 - Support Tickets – Historical tickets and their resolutions.
- Example setup:
 - Name: [your_initials]-BricksLab-TechnicalSupport
 - Description: Provides telco product support using company docs and historical tickets.

BricksLab-TechnicalSupport

Endpoint: MLflow experiment: ~

Overview
Knowledge Assistant turns your documents—like PDFs, Word files, and more—into a high-quality Q&A chatbot. It uses advanced AI to deliver accurate, reliable answers from your content with no coding or complex setup required. Ideal for product documentation, customer support knowledge bases, regulatory documents, and more. Agent Bricks may use endpoints hosted on Databricks ML. Documentation & License

Basic Info

Name: BricksLab-TechnicalSupport
Description: This agent provides customer support for a Telecommunications company and references multiple knowledge sources.

Configure Knowledge Sources

Select up to 10 different knowledge sources to use in your knowledge assistant.

knowledge_base_index

Type: Vector Search Index
Source: databricks default:knowledge_base_index
Text Column: formatted_content

Base URL Column: knowledge_base containing company details such as policies, frequently asked questions, and procedures.

support_tickets_index

Type: Vector Search Index
Source: databricks default:support_tickets_index
Text Column: formatted_content

Base URL Column: support_tickets containing historical tickets and resolutions of past customer issues.

+ Add

Optional >

Create Agent

1.3 Test the Knowledge Assistant

- Sample Question: *How do I know if my 5G is working?*
- Observe the verbose answer with citations and traces.
- Explore how the assistant grounds responses in living company data.

BricksLab-TechnicalSupport

Endpoint: ka-820db4fb-endpoint MLflow experiment: ka-820db4fb-dev-experiment

Basic Info

Name: BricksLab-TechnicalSupport
Description: This agent provides customer support for a Telecommunications company and references multiple knowledge sources.

Configure Knowledge Sources

Select up to 10 different knowledge sources to use in your knowledge assistant.

knowledge_base_index
Vector Search Index: knowledge_base_index
Last sync: Sep 12, 2025, 02:43:02 PM
Description: Knowledge base containing company details such as policies, frequently asked questions, and procedures.

support_tickets_index
Vector Search Index: support_tickets_index
Last sync: Sep 12, 2025, 02:43:02 PM
Description: Knowledge base containing historical tickets and resolutions of past customer issues.

Test your Agent

ka-820db4fb-endpoint

View thoughts

To check if your 5G is working, follow these steps:

1. Look for the 5G icon:
 - On your phone's status bar (usually at the top of the screen), you should see a "5G", "5G UW", "5G UC", or similar icon when your device is connected to a 5G network.[\[1\]](#)[\[2\]](#)
2. Check Your Device Settings:
 - iPhone: Go to Settings > Cellular > Cellular Data Options > Voice & Data.[\[3\]](#)[\[4\]](#)
Android: Go to Settings > Connections > Mobile Networks > Network Mode (or similar).[\[5\]](#)[\[6\]](#)[\[7\]](#) Ensure a mode including 5G is selected (e.g., "5G/LTE/3G/2G").[\[8\]](#)[\[9\]](#)[\[10\]](#)[\[11\]](#)
3. Verify 5G Coverage:
 - Use your carrier's app or coverage map to confirm that 5G service is available in your current location.[\[12\]](#)[\[13\]](#)[\[14\]](#)[\[15\]](#)[\[16\]](#) 5G coverage can vary, especially indoors.[\[16\]](#)[\[17\]](#)
4. Check Your Plan and Device Compatibility:
 - Make sure your mobile plan includes 5G access and your device supports 5G.[\[18\]](#)[\[19\]](#)[\[20\]](#) For iPhones, models 12 and newer support 5G.[\[4\]](#)[\[4\]](#)[\[21\]](#) For Android, check your device specifications.[\[8\]](#)[\[9\]](#)[\[22\]](#)
5. Test Your Connection:
 - Try using data-heavy apps or run a speed test.[\[23\]](#)[\[24\]](#) If you see fast speeds and the 5G icon, your 5G is working.
6. Troubleshooting if 5G Isn't Working:
 - Check for software updates on your device and carrier.[\[25\]](#)

Start typing ...

Part 2: Expand with Genie for Structured Data

2.1 Query Structured Data with Genie

- **Genie spaces** allow natural language queries over structured data (SQL tables).
- Genie is pre-configured to access **billing** and **customer** tables.
- **Try It Out:**
 - Ask “*What is the average total bill?*”
 - Guide Genie with examples/instructions to better align with your data structures.

| | <code>billing_id</code> | <code>subscription_id</code> | <code>customer_id</code> | <code>billing_date</code> | <code>due_date</code> | Genie space |
|----|-------------------------|------------------------------|--------------------------|---------------------------|-----------------------|---------------------|
| 1 | BILL-1234589394 | SUB-10001608 | CUS-11094 | 2025-04-01 | 2025-04-1 | Vector search index |
| 2 | BILL-1234589395 | SUB-10001608 | CUS-11094 | 2025-05-01 | 2025-05-1 | Synced table |
| 3 | BILL-1234589396 | SUB-10001608 | CUS-11094 | 2025-06-01 | 2025-06-1 | Online table |
| 4 | BILL-1234589397 | SUB-10001609 | CUS-11095 | 2024-05-01 | 2024-05-1 | Metric view |
| 5 | BILL-1234589398 | SUB-10001609 | CUS-11095 | 2024-06-01 | 2024-06-1 | |
| 6 | BILL-1234589399 | SUB-10001609 | CUS-11095 | 2024-07-01 | 2024-07-1 | |
| 7 | BILL-1234589400 | SUB-10001609 | CUS-11095 | 2024-08-01 | 2024-08-15 | 26.37 |
| 8 | BILL-1234589401 | SUB-10001609 | CUS-11095 | 2024-09-01 | 2024-09-15 | 26.37 |
| 9 | BILL-1234589402 | SUB-10001609 | CUS-11095 | 2024-10-01 | 2024-10-15 | 26.37 |
| 10 | BILL-1234589403 | SUB-10001609 | CUS-11095 | 2024-11-01 | 2024-11-15 | 26.37 |
| 11 | BILL-1234589404 | SUB-10001609 | CUS-11095 | 2024-12-01 | 2024-12-15 | 26.37 |
| 12 | BILL-1234589405 | SUB-10001609 | CUS-11095 | 2025-01-01 | 2025-01-15 | 26.37 |
| 13 | BILL-1234589406 | SUB-10001609 | CUS-11095 | 2025-02-01 | 2025-02-15 | 26.37 |
| 14 | BILL-1234589407 | SUB-10001609 | CUS-11095 | 2025-03-01 | 2025-03-15 | 26.37 |
| 15 | BILL-1234589408 | SUB-10001609 | CUS-11095 | 2025-04-01 | 2025-04-15 | 26.37 |
| 16 | BILL-1234589409 | SUB-10001609 | CUS-11095 | 2025-05-01 | 2025-05-15 | 26.37 |
| 17 | BILL-1234589410 | SUB-10001609 | CUS-11095 | 2025-06-01 | 2025-06-15 | 26.37 |
| 18 | BILL-1234589411 | SUB-10001610 | CUS-11096 | 2025-03-01 | 2025-03-15 | 84.5 |
| 19 | BILL-1234589412 | SUB-10001610 | CUS-11096 | 2025-04-01 | 2025-04-15 | 84.5 |

The screenshot shows the Genie interface. On the left, there's a sidebar with a search bar labeled "Filter spaces" and a dropdown menu with "All" selected. Below it is a table with columns "Name" and "Description". A single row is listed: "BricksLab" with a small icon and the text "BricksLab" below it. A red box highlights this row. To the right of the table is a list of generated questions under the heading "BricksLab". The questions are:

- Explain the data set
- What are the distinct payment methods used by customers?
- What is the average total amount billed to customers?
- How many new customers have registered each month?

At the top right of the interface, there are buttons for "+ New chat", "History", "Data", "Monitoring", and "Share". On the far right, there's a sidebar titled "Data Instructions" with tabs for "Data" and "Instructions". Under "Data", there are two tables: "billing" and "customers", both of which are "Table" type.

2.2 Treat Genie as an Agent

- Genie rooms can be registered as agents, enabling them to participate in multi-agent workflows.
- Use Genie when customer-specific or billing data is required.

The screenshot shows the Genie interface. On the left, there's a sidebar with a search bar labeled "What is the average total amount billed to customers?". Below it is a table with one row and two columns. The first column is labeled "1.2 avg_total_amount" and the second column contains the value "131.31". At the bottom of this section, there are buttons for "Is this correct?", "Yes", "Fix it", and "Request review".

To the right, there's a sidebar titled "Data Instructions" with tabs for "Text", "Joins", and "SQL Queries". The "SQL Queries" tab is selected. It has a section titled "SQL queries & functions" with the subtext "Example queries that Genie can learn from." Below this, there's a table with columns "Name" and "Type". Two entries are listed:

| Name | Type |
|---|-------|
| answers questions related to user's account informi | Query |
| billing history questions | Query |

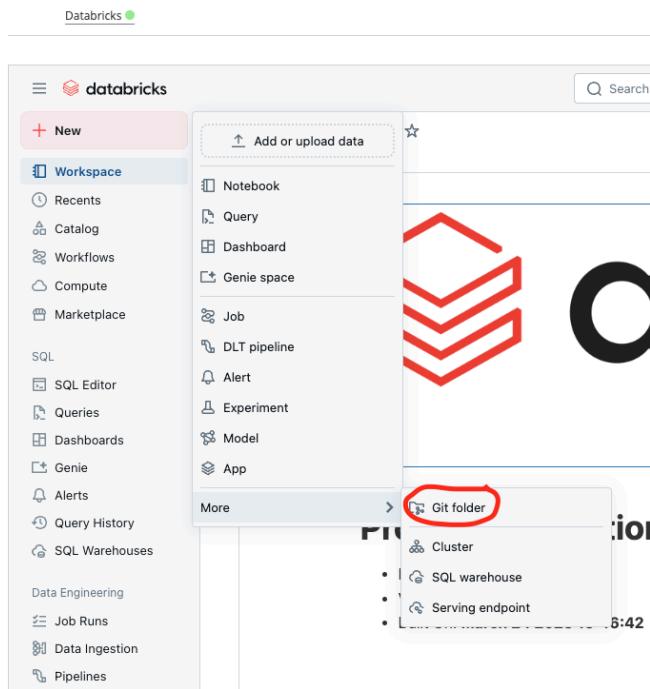
Continue with the rest of the sections, following the instructions.

Backup environment and lab (if Vocareum doesn't work): Databricks Free Edition

Create an Databricks Free Edition account with your corporate account (or personal email if your corporate account gets blocked): <https://www.databricks.com/learn/free-edition>

Create the Repo in your workspace

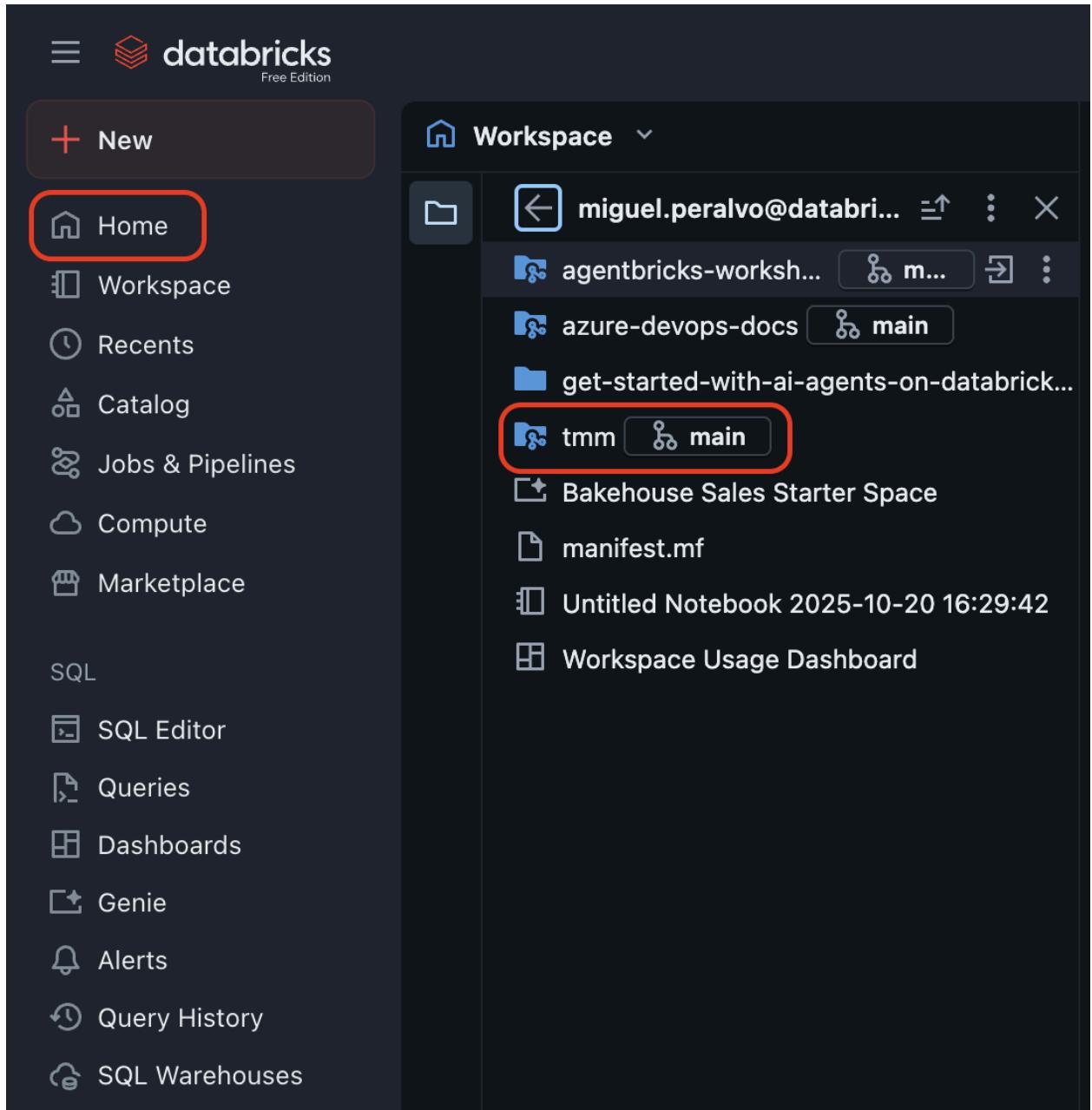
In your workspace, navigate to **+ New -> Git Folder** as below.



Once you click Git Folder, you will be directed to the screen below, simply input the **Git repository URL** <https://github.com/databricks/tmm>, and click **Create Git Folder**.

Start the Lab

Once the repo is created, navigate to the repo **tmm**:

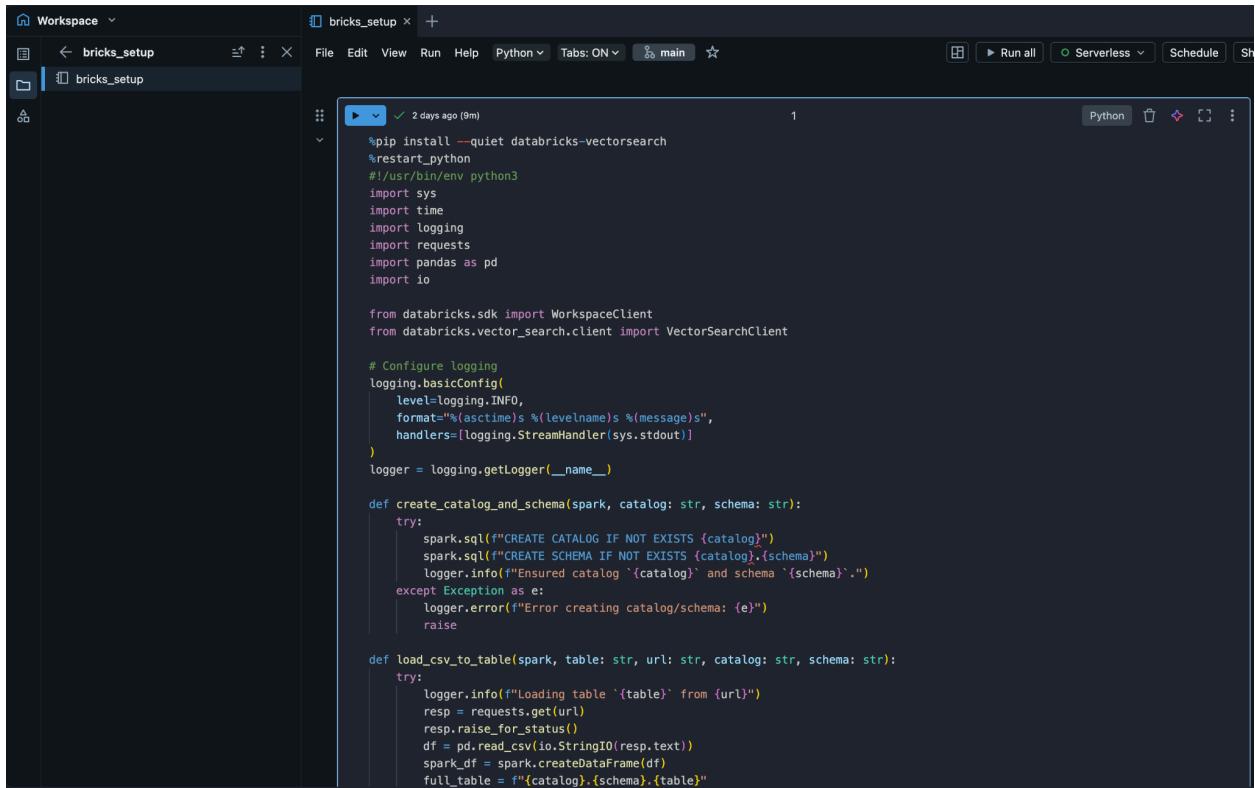


Agents Workshop

Navigate to **agents-workshop/README.md** and follow the instructions

Agent Bricks Workshop (Once Agents are available in free edition, in 2026, in the meantime you can use other workshops under tmm, like Agents Workshop above)

Navigate to **bricks-workshop/bricks_setup/bricks_setup**:



```
*pip install --quiet databricks-vectorsearch
%restart_python
#!/usr/bin/env python3
import sys
import time
import logging
import requests
import pandas as pd
import io

from databricks.sdk import WorkspaceClient
from databricks.vector_search.client import VectorSearchClient

# Configure logging
logging.basicConfig(
    level=logging.INFO,
    format=f"%(asctime)s %(levelname)s %(message)s",
    handlers=[logging.StreamHandler(sys.stdout)]
)
logger = logging.getLogger(__name__)

def create_catalog_and_schema(spark, catalog: str, schema: str):
    try:
        spark.sql(f"CREATE CATALOG IF NOT EXISTS {catalog}")
        spark.sql(f"CREATE SCHEMA IF NOT EXISTS {catalog}.{schema}")
        logger.info(f"Ensured catalog '{catalog}' and schema '{schema}'")
    except Exception as e:
        logger.error(f"Error creating catalog/schema: {e}")
        raise

def load_csv_to_table(spark, table: str, url: str, catalog: str, schema: str):
    try:
        logger.info(f"Loading table '{table}' from {url}")
        resp = requests.get(url)
        resp.raise_for_status()
        df = pd.read_csv(io.StringIO(resp.text))
        spark_df = spark.createDataFrame(df)
        full_table = f"{catalog}.{schema}.{table}"
    
```

Now run it.

And go back to **bricks-workshop/README.md**, following the instructions:

The screenshot shows a Databricks workspace interface. On the left, there's a sidebar with a 'Workspace' dropdown and a tree view of a folder structure under 'bricks-workshop'. The tree includes 'bricks_lab', 'bricks_setup', 'data', 'README.md', and 'setup.py'. The main area is a notebook titled 'bricks_setup' with the tab 'main' selected. The notebook content starts with a header:

```
B I <> ⇝ H1 H2 H3 ━━  
1 # Agent Bricks Lab: Build, Orchestrate, and Improve Multi-Agent Systems  
2  
3 In this lab, you'll learn how to create and refine AI agents using **Databricks Agent Bricks**. You'll start by building a **Knowledge Assistant** grounded in company product docs and historical support tickets, then expand it with a **Genie-powered structured data agent**.
```

Agent Bricks Lab: Build, Orchestrate, and Improve Multi-Agent Systems

In this lab, you'll learn how to create and refine AI agents using **Databricks Agent Bricks**. You'll start by building a **Knowledge Assistant** grounded in company product docs and historical support tickets, then expand it with a **Genie-powered structured data agent**. Finally, you'll orchestrate them together with a **Multi-Agent Supervisor** and guide the system to produce better, user-friendly responses.

Part 1: Build Your First Knowledge Assistant

1.1 Create a Vector Search Index

- Why Vector Search?
 - Provides efficient retrieval of relevant chunks of data for grounding LLM responses.
 - Two common types:
 - Triggered updates (for static knowledge bases like FAQs/policies).
 - Continuous updates (for dynamic sources like support tickets).
- Demo: Indexes are pre-built for this lab, but you'll see how easy it is to create one.

1.2 Build the Knowledge Assistant Agent

- Navigate to "Agents" in the UI.
- Create a new **Knowledge Assistant** using the two pre-built vector search indices:
 - Knowledge Base – Company details, FAQs, policies, and procedures.
 - Support Tickets – Historical tickets and their resolutions.
- Example setup:

This notebook with pictures can help you out:

<https://github.com/miquelpera-db/agentbricks-workshop/blob/main/Get%20Started%20with%20AgentBricks%20KA%20on%20Databricks/M04%20-%20Production-Ready%20Agents%20with%20Agent%20Bricks/4.2L%20-%20KA%20%2B%20GENIE%20%2B%20MAS.ipynb>

Part 1: Build Your First Knowledge Assistant

1.1 Create a Vector Search Index

- **Why Vector Search?**
 - Provides efficient retrieval of relevant chunks of data for grounding LLM responses.
 - Two common types:
 - **Triggered updates** (for static knowledge bases like FAQs/policies).
 - **Continuous updates** (for dynamic sources like support tickets).
- **Demo:** Indexes are pre-built for this lab, but you'll see how easy it is to create one.

The screenshot shows the Databricks Catalog interface. On the left, there's a sidebar with a tree view of databases and tables. The 'knowledge_base' table under the 'bricks_lab' database is selected. A modal window titled 'Create vector search index' is open over the catalog. The modal has several sections: 'Index structure' (Name: 'bricks_lab.knowledge_base_index', Primary key: 'kb_id'), 'Columns to sync' (Leave blank to sync all columns), 'Embeddings' (Embedding source: 'Compute embeddings', Embedding source column: 'formatted_content', Embedding model: 'databricks-gte-large-en'), 'Compute resources' (Vector search endpoint: 'bricks_endpoint', Sync mode: 'Triggered'), and 'Advanced settings'. At the bottom right of the modal are 'Cancel' and 'Create' buttons. The background of the catalog shows a list of tables and their details.