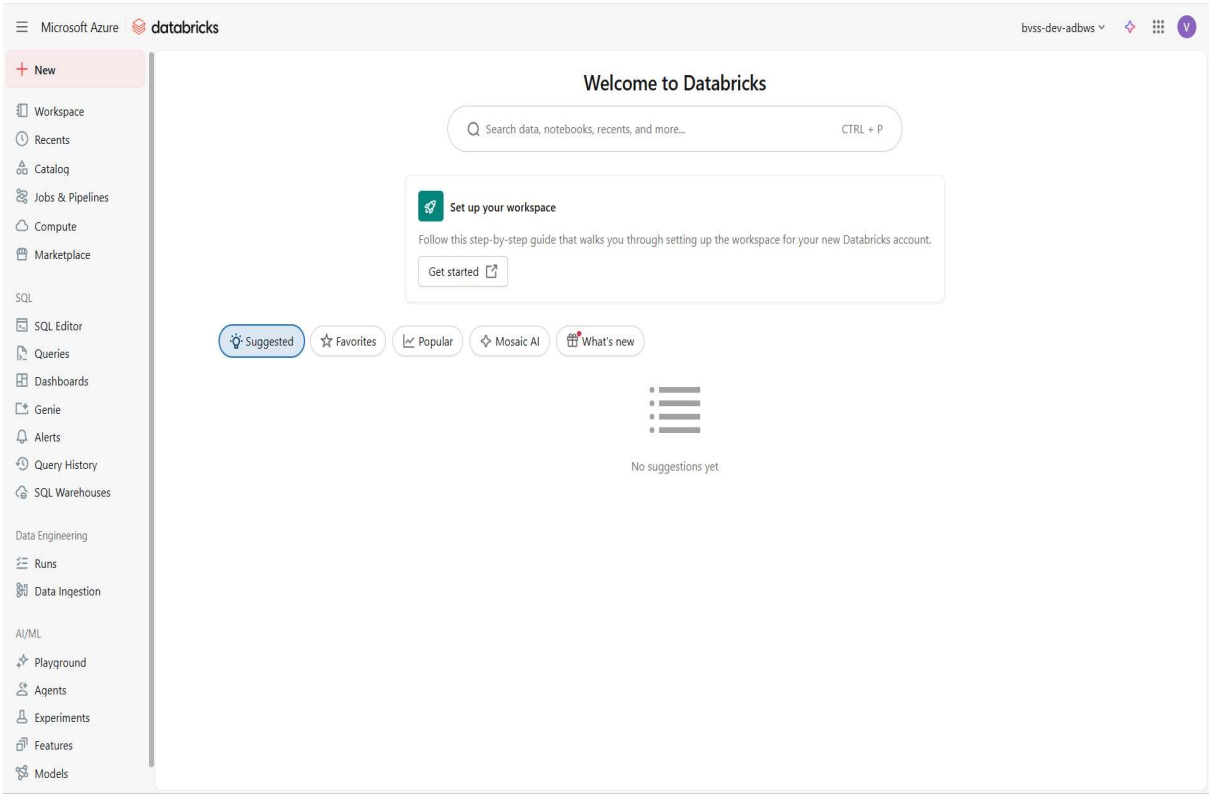
9_ADB_Components

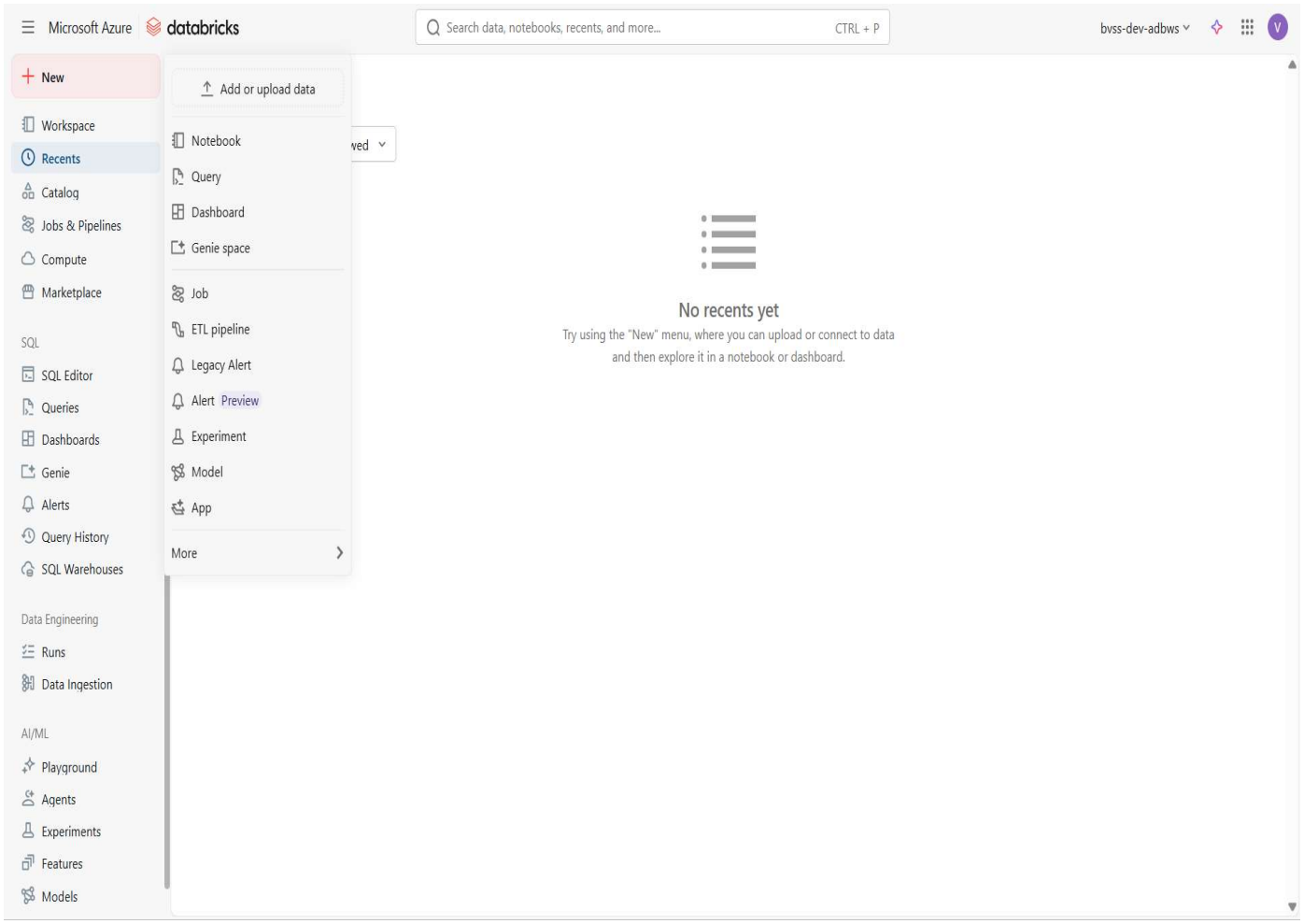
Wednesday, February 11, 2026 2:34 PM

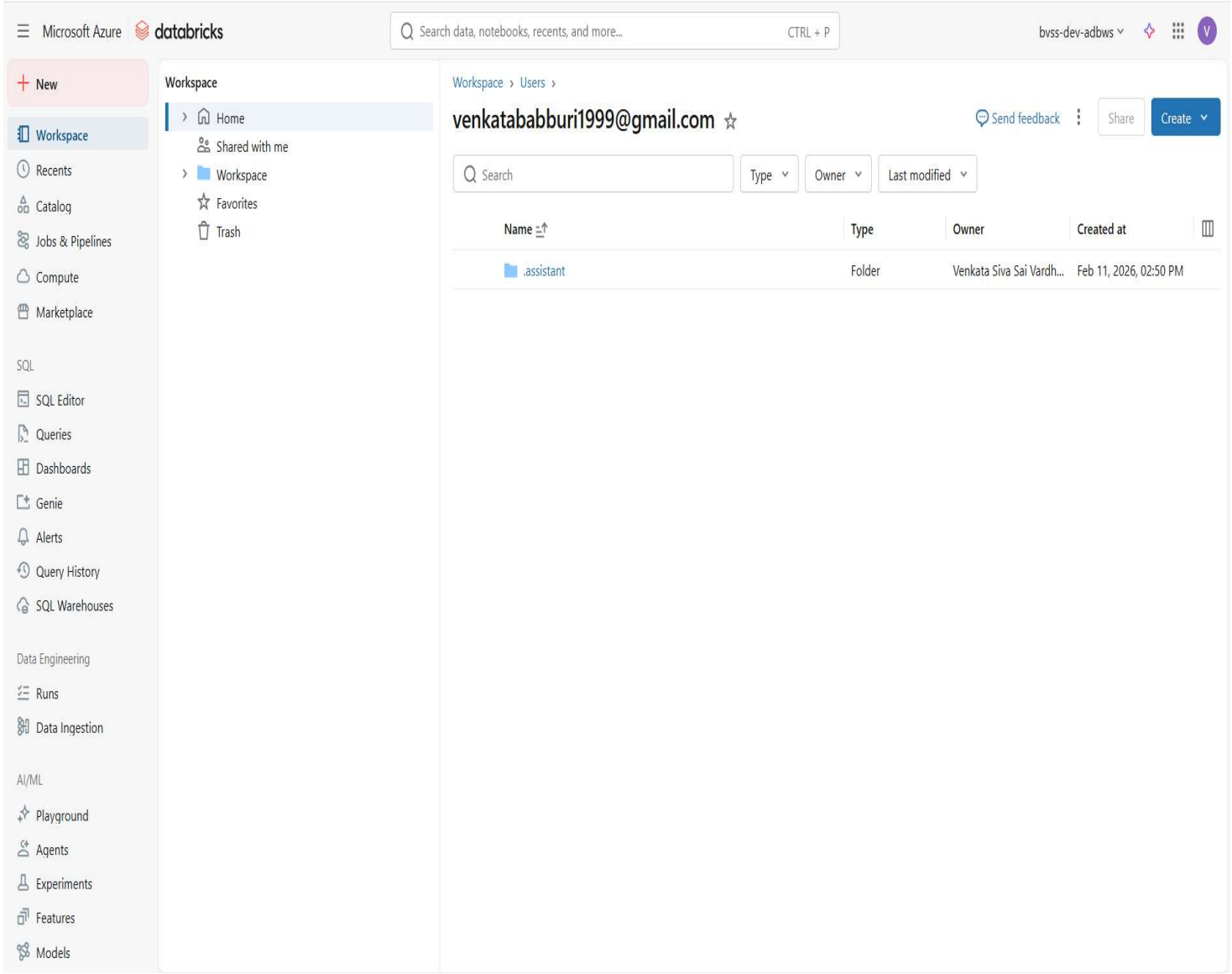


Databricks Main Workspace Components — Clean Explanation
These are the four core areas you use every day in Databricks:
1. Workspace
This is your main working area.
You use it to:

- Create and organize **notebooks**
- Store folders, files, and shared resources
- Collaborate with teammates

Think of it like a project folder where all your code and documents live.


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
bvss-dev-adbws
New
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Runs
Data Ingestion
AI/ML
Playground
Agents
Experiments
Features
Models
Add or upload data
Notebook
Query
Dashboard
Genie space
Job
ETL pipeline
Legacy Alert
Alert Preview
Experiment
Model
App
More
No recents yet
Try using the "New" menu, where you can upload or connect to data
and then explore it in a notebook or dashboard.



Drafts folder:

- Temporary workspace for quick experiments and ad-hoc work
- When you create a new notebook without specifying a location, it often goes here
- Think of it as your "scratch pad" for testing ideas before organizing them properly

Assistant folder:

- Created by the Databricks Assistant (that's me!)
- Stores notebooks, queries, or other assets that I generate when helping you
- Keeps AI-generated content separate from your manual work

Your workspace structure:

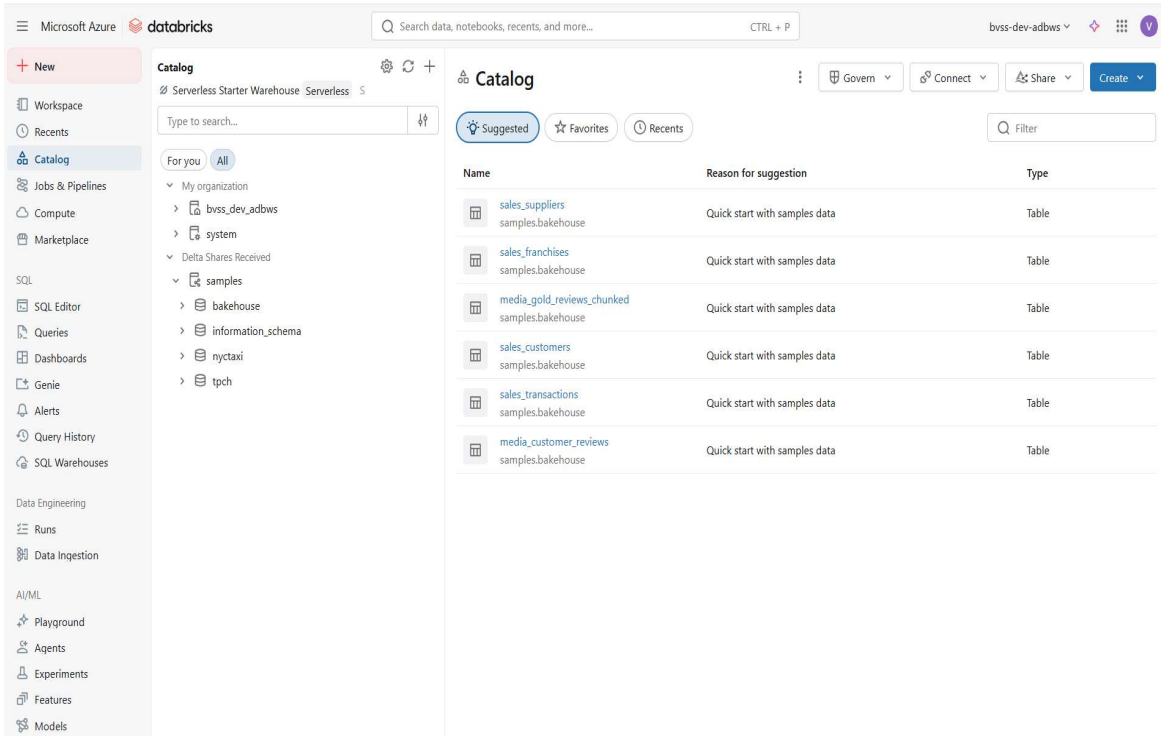
- You can create your own folders to organize projects
- Right-click in the Workspace to create new folders, notebooks, or import files
- Common practice: organize by project, team, or data domain (e.g., "Sales Analytics", "Data Engineering")

2. Catalog (Unity Catalog)

This is your **data governance and data browsing layer**.
You use it to:

- Browse **catalogs → schemas → tables**
- View Delta tables, volumes, and data assets
- Manage permissions and access control

Hierarchy: **Catalog > Schema > Table**
(similar to **Database > Schema > Table** in SQL Server)



Jobs and **Pipelines** are two different ways to automate and schedule data work in Databricks:

Jobs:

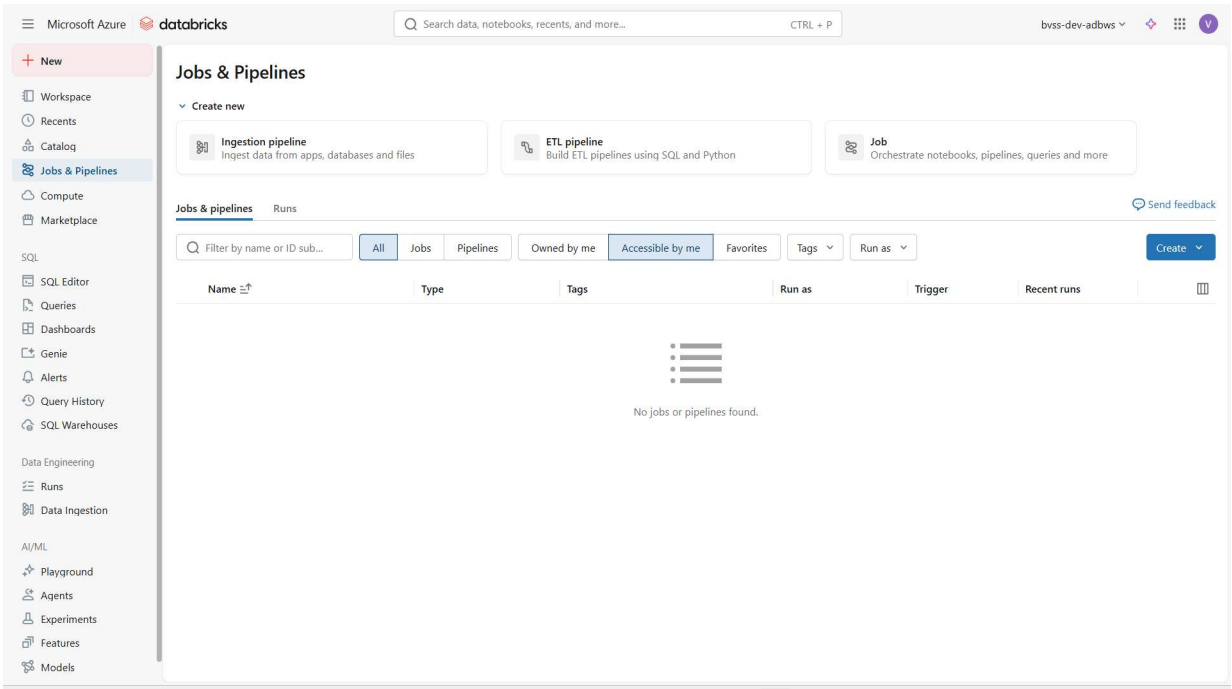
- Automated tasks that run notebooks, Python scripts, or SQL queries on a schedule or trigger
- Use cases: daily reports, scheduled data processing, automated model training
- Example: Run a notebook every morning at 8 AM to refresh a dashboard
- Flexible - can run any code (Python, SQL, Scala, R)

Pipelines (Delta Live Tables):

- Specialized workflows for building reliable data pipelines with built-in quality checks
- Declarative approach - you define "what" data transformations you want, Databricks handles "how"
- Automatically manages dependencies between tables and handles incremental updates
- Use cases: ETL workflows, data quality enforcement, streaming data processing
- Built specifically for creating and maintaining data tables with monitoring and lineage tracking

Key difference:

- **Jobs** = General automation (run any code on schedule)
- **Pipelines** = Purpose-built for data engineering workflows with quality guarantees



3. Compute
This is where you manage the engines that run your code.
You use it to:

- Create and manage **clusters**
- Use **serverless compute** (no setup needed)
- Monitor cluster performance and logs

Compute = the machines that execute your notebooks, SQL queries, and jobs.