06 Databricks Intro

https://www.databricks.com/learn/free-edition

Introduction to Databricks

Overview

- **What is Databricks?**
  - A unified data analytics platform built on Apache Spark, designed for big data processing, data engineering, data science, and machine learning.
  - Provides a collaborative workspace with tools for ETL pipelines, SQL analytics, and AI/ML workloads.
  - Integrates with cloud providers (AWS, Azure, GCP) and supports lakehouse architectures (combining data lake scalability with data warehouse reliability).
- **Purpose**: Simplifies big data workflows by offering managed compute, storage, and orchestration tools, enabling users to focus on data processing rather than infrastructure management.
- **Free Edition** (formerly Community Edition):
  - Fully free, no credit card or cloud account required.
  - Production-ready workspace with most features of paid Databricks, unlike the limited Community Edition.
  - Ideal for learning PySpark, Databricks features, and preparing for data engineering roles.
  - Supports notebooks, SQL queries, pipelines, and more.

Getting Started with Databricks Free Edition

- **Accessing Databricks Free Edition**:
  - Open a browser (e.g., Chrome, Edge) and search for "Databricks Free Edition."
  - Click the first link (e.g., "Try Databricks for Free") or navigate to databricks.com/try-databricks.
  - Look for "Looking for Databricks Free Edition? Click here" to avoid the 14-day paid trial.
  - Sign up with any email (Gmail, Outlook, etc.), no business or student account needed.
  - After signup, log in using the same email and select your username (backed by AWS) to access the workspace.
- **Workspace UI**:
  - Modern, user-friendly interface with enhanced features compared to Community Edition.
  - Key sections: Workspace, Catalog, Workflows, Compute, Marketplace, SQL, Data Engineering, AI/ML.
  - Enable all preview features (under Profile > Previews) to access beta and generally available (GA) functionalities for consistency.

Databricks Workspace

- **Purpose**: A centralized repository for managing development resources (notebooks, SQL queries, Python files, pipelines).
- **Structure**:

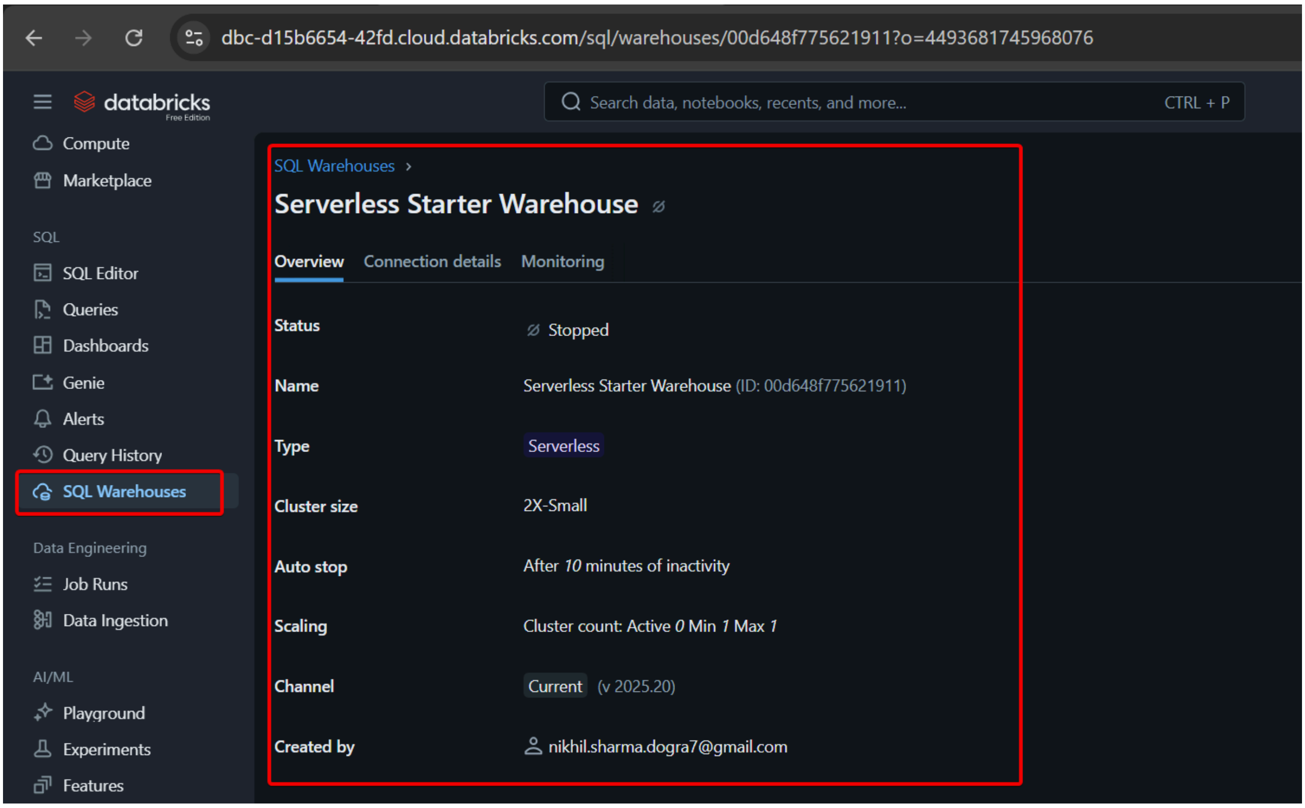
- Organized as folders and subfolders (e.g., "Databricks Bootcamp" folder).
- Create folders via Workspace > Create > Folder.
- Import resources (e.g., `.dbc` archives) for pre-built notebooks or reference code.

- **Key Actions**:
  - Create subfolders for organizing notebooks, SQL files, and pipelines.
  - Import `.dbc` (Databricks Archive) files for hierarchical folder structures and notebooks.
  - Manually upload `.py` or `.sql` files from reference resources if needed.
  - Recommendation: Write your own code but refer to provided resources if stuck.

Key Features

1. Compute

- **Definition**: Compute resources (clusters or warehouses) power Databricks workloads (Spark jobs, SQL queries, etc.).
- **Types of Compute**:
  - **All-Purpose Compute** (Legacy):
    - For development, exploration, notebooks, and ad-hoc analysis.
    - Supports multiple users, notebooks, dashboards, APIs.
    - Always running, higher cost, 3–5 minute startup time.
    - Use case: Testing pipelines, data exploration.
  - **Job Compute**:
    - For production ETL jobs or scheduled pipelines.
    - Auto-created during job configuration, terminates after job completion.
    - Cost-effective (only runs during jobs).
    - Not for development; used in production environments.
  - **Pools**:
    - Pre-warmed clusters to reduce startup time (3–5 minutes for all-purpose).
    - Set minimum/maximum machines for availability and scaling.
    - Used for both all-purpose and job compute.
    - Benefit: Cost savings by avoiding cold starts.
  - **SQL Warehouse** (formerly SQL Endpoints):
    - Optimized for SQL workloads and BI tools (Power BI, Tableau).
    - Features: Query caching, auto-stop (e.g., after 10 minutes of inactivity), scalable cluster sizes (2X-Small default in Free Edition).
    - Use case: Sharing lakehouse/warehouse data with analysts for reporting.
    - Connection details (hostname, HTTP path, JDBC URL) for BI tool integration.



Serverless Starter Warehouse

Status: Stopped

Name: Serverless Starter Warehouse (ID: 00d648f775621911)

Type: Serverless

Cluster size: 2X-Small

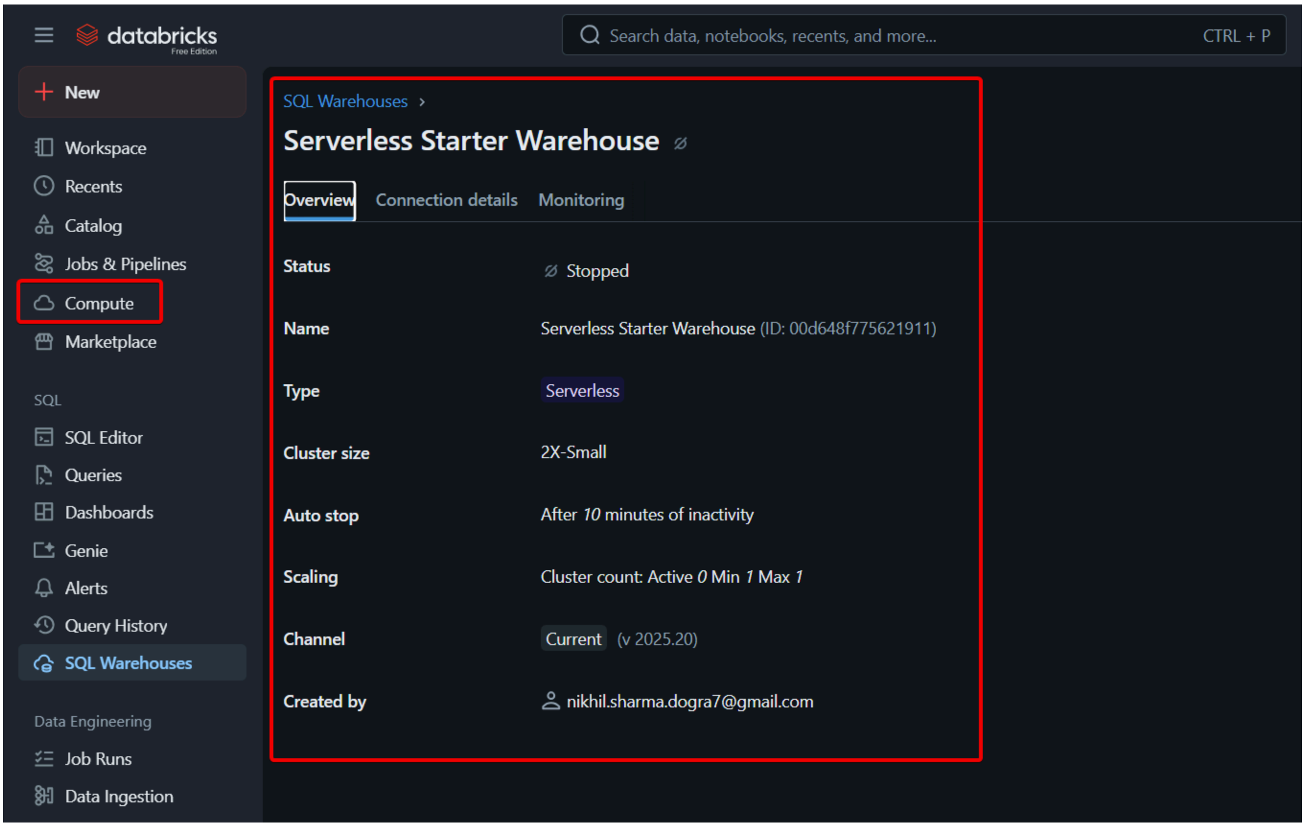
Auto stop: After *10* minutes of inactivity

Scaling: Cluster count: Active *0* Min *1* Max *1*

Channel: Current (v 2025.20)

Created by: nikhil.sharma.dogra7@gmail.com

- **Serverless Compute** (Recommended):
  - Auto-scales (up/down) based on workload; no manual instance management.
  - Near-instant startup (seconds vs. minutes for all-purpose).
  - Supports ad-hoc analysis, job scheduling, and Delta Live Tables (Lake Flow).
  - Ideal for unpredictable workloads; cost-effective and hands-off.
  - Automatically available in Free Edition; no need to create manually.
- **Free Edition Notes:**
  - Only SQL Warehouse and Serverless Compute are available (all-purpose, job, and pools are paid features).
  - Databricks recommends Serverless Compute + SQL Warehouse for simplicity and performance.



SQL Warehouses ›

Serverless Starter Warehouse

Overview | Connection details | Monitoring

Status	Stopped
Name	Serverless Starter Warehouse (ID: 00d648f775621911)
Type	Serverless
Cluster size	2X-Small
Auto stop	After *10* minutes of inactivity
Scaling	Cluster count: Active *0* Min *1* Max *1*
Channel	Current (v 2025.20)
Created by	nikhil.sharma.dogra7@gmail.com

2. Catalog (Unity Catalog)

- **Definition:** A governance layer for managing data assets (tables, schemas, databases) across the lakehouse.
- **Purpose:**
  - Centralized metadata management, lineage tracking, and access control.
  - Supports Medallion Architecture (Bronze, Silver, Gold layers).
  - Enables secure data sharing and querying across clouds.
- **Key Features:**
  - Hierarchical organization: Catalogs > Schemas > Tables.
  - Integrates with Delta Lake for reliable, ACID-compliant storage.
  - Used for managing data in lakehouses and warehouses.
- **Learning Focus:** Master Unity Catalog for data engineering interviews and real-world governance.

3. Workflows

- **Definition:** Orchestration hub for ETL jobs, pipelines, and task automation.
- **Features:**
  - Supports control flow (if/else), parameterization, and dynamic task values.
  - Configures jobs for production-ready pipelines.
  - Integrates with Delta Live Tables (now Lake Flow) for declarative pipelines.
- **Use Case:** Automating data ingestion, transformation, and scheduling.

4. Marketplace

- **Purpose:** Connects Databricks with external tools/services to enhance functionality.
- **Examples:**

  - **DBT (Data Build Tool):** For SQL-based transformations.
  - **Fivetran**: For data ingestion from multiple sources.
- **Benefit**: Simplifies integration with third-party tools to build robust solutions.

5. SQL

- **Purpose**: Data warehousing and analytics hub.
- **Features:**
  - SQL Editor for queries, dashboards, and alerts.
  - Query history and monitoring for performance insights.
  - SQL Warehouse for optimized query execution.
- **Use Case:** Building reports, dashboards, and analytics for business users.

6. Data Engineering

- **Focus Areas:**
  - Job runs, data ingestion pipelines, and transformations.
  - **Lake Flow (Declarative Pipelines):**
    - Evolution of Delta Live Tables (DLT), now part of Apache Spark.
    - Simplifies pipeline development with declarative syntax.
    - New coding platform (Lake Flow Editor) for streamlined ETL.
    - Supports streaming and batch processing.
- **Why Important?:** Revolutionary for data engineers; simplifies complex ETL workflows.

7. AI/ML

- **Overview**: Tools for building, deploying, and managing ML models and agents.
- **Focus**: Secondary for data engineering; primary for data scientists.
- **Use Case:** Model training, feature engineering, and AI-driven analytics.

Practical Setup

- **Creating a Folder**:
  - Go to Workspace > Create > Folder (e.g., "Databricks Bootcamp").
  - Use for organizing notebooks, SQL files, and pipelines.
- **Importing Resources:**
  - Import `.dbc` archives via Workspace > Import > Browse.
  - Creates hierarchical folder structure with notebooks.
  - Manually upload `.py` or `.sql` files from reference resources.
- **Using Serverless Compute:**
  - Automatically available in Free Edition; no setup needed.
  - Select when creating notebooks for instant execution.
- **SQL Warehouse:**
  - Auto-created in Free Edition (2X-Small, auto-stop after 10 minutes).
  - Use for SQL queries; monitor via the Monitoring tab for query performance.
- **Unity Catalog:**

  - Explore via Catalog section; create schemas/tables for data management.
  - Aligns with Medallion Architecture for structured data organization.

Best Practices

- **Enable Previews:** Turn on all preview features to access the latest functionalities.
- **Use Serverless Compute:** Preferred for most workloads due to auto-scaling and instant startup.
- **Organize Workspace:** Create clear folder structures (e.g., by project or module).
- **Leverage Unity Catalog:** Centralize governance for data assets and lineage.
- **Learn Lake Flow:** Master declarative pipelines for modern ETL workflows.
- **Monitor SQL Warehouse:** Use monitoring tools to optimize query performance.
- **Refer to Resources:** Use provided `.dbc` archives and reference files to troubleshoot errors.

Why Learn Databricks?

- **Industry Relevance:** Widely adopted for data engineering, analytics, and ML (e.g., by Microsoft Fabric, AWS, Azure).
- **Career Benefits:** Essential for data engineering roles; prepares for interviews with hands-on skills.
- **Free Edition Advantage:** Full-featured workspace for learning without cost barriers.
- **Recent Advancements:**
  - Lake Flow (Declarative Pipelines) revolutionizes ETL.
  - Unity Catalog enhances governance and scalability.
  - Serverless Compute simplifies resource management.

References

- Databricks Free Edition Signup
- Databricks Documentation
- Microsoft Fabric and Databricks Integration
- Delta Live Tables (Lake Flow)
- Unity Catalog