databricks
Academy

Get Started with Databricks for Data Engineering

Course learning objectives

Things you'll be able to do after completing this course

- Explain fundamental concepts about using the Databricks Lakehouse Platform for new users responsible for data engineering workflows.
- Describe how to work in the Databricks Lakehouse Platform
- Perform basic notebook tasks using the Databricks Lakehouse Platform.
- Manage Delta tables using the Databricks Lakehouse Platform.
- Describe features available through the Databricks Lakehouse Platform to secure and govern data.
- Use Workflow Jobs within the Databricks Lakehouse Platform to automate a basic data engineering workflow. Automate a basic data engineering workflow using Workflow Jobs

Prerequisites

Things you should already know or be able to do before taking this course

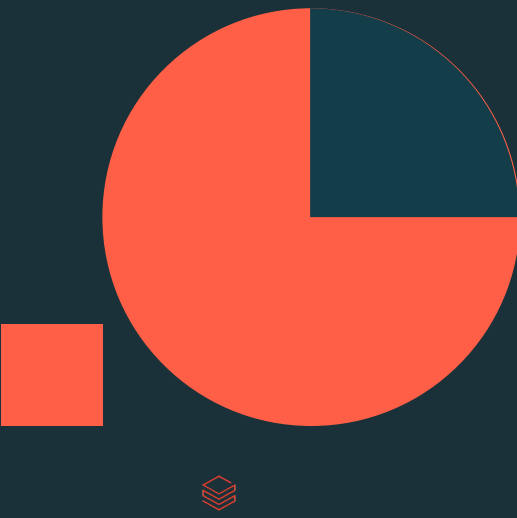
- Basic knowledge of data engineering topics such as extraction, cleaning (and other transformations), and loading

Technical Requirements

Things to keep in mind before attempting to

- This course has been tested in DBR 13.2
- Not all notebooks will run in Community Edition
- You will need a Github account (or, you can just watch the demo)

Module 1



Module 1 - Agenda

Lesson Name	Lesson Name
Lecture: Databricks Fundamentals	Lecture: Introduction to Compute Resources
Demo: The Lakehouse Platform	Demo: Working with Compute Resources
Lecture: Introduction to Repos on Databricks	Lecture: Databricks Notebooks
Demo: Working with Repos on Databricks	Demo: Working with Notebooks

Databricks Fundamentals

Learning objectives

Things you'll be able to do after completing this lesson

- Identify Databricks as *the* Lakehouse Platform
- Connect common data personas to the core services of the Databricks Lakehouse Platform.

APACHE Spark™

DELTA LAKE

mlflow™

Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics

Michael Armbrust[1], Ali Ghodsi[1,2], Reynold Xin[1], Matei Zaharia[1,3]
[1]Databricks, [2]UC Berkeley, [3]Stanford University

Abstract

This paper argues that the data warehouse architecture as we know it today will wither in the coming years and be replaced by a new architectural pattern, the Lakehouse, which will (i) be based on open direct-access data formats, such as Apache Parquet, (ii) have first-class support for machine learning and data science, and (iii) offer state-of-the-art performance. Lakehouses can help address several major challenges with data warehouses, including data staleness, reliability, total cost of ownership, data lock-in, and limited use-case support. We discuss how the industry is already moving toward Lakehouses and how this shift may affect work in data management. We also report results from a Lakehouse system using Parquet that is competitive with popular cloud data warehouses on TPC-DS.

1 Introduction

This paper argues that the data warehouse architecture as we know it today will wane in the coming years and be replaced by a new architectural pattern, which we refer to as the Lakehouse, characterized by (i) open direct-access data formats, such as Apache Parquet and ORC, (ii) first-class support for machine learning and data science workloads, and (iii) state-of-the-art performance.

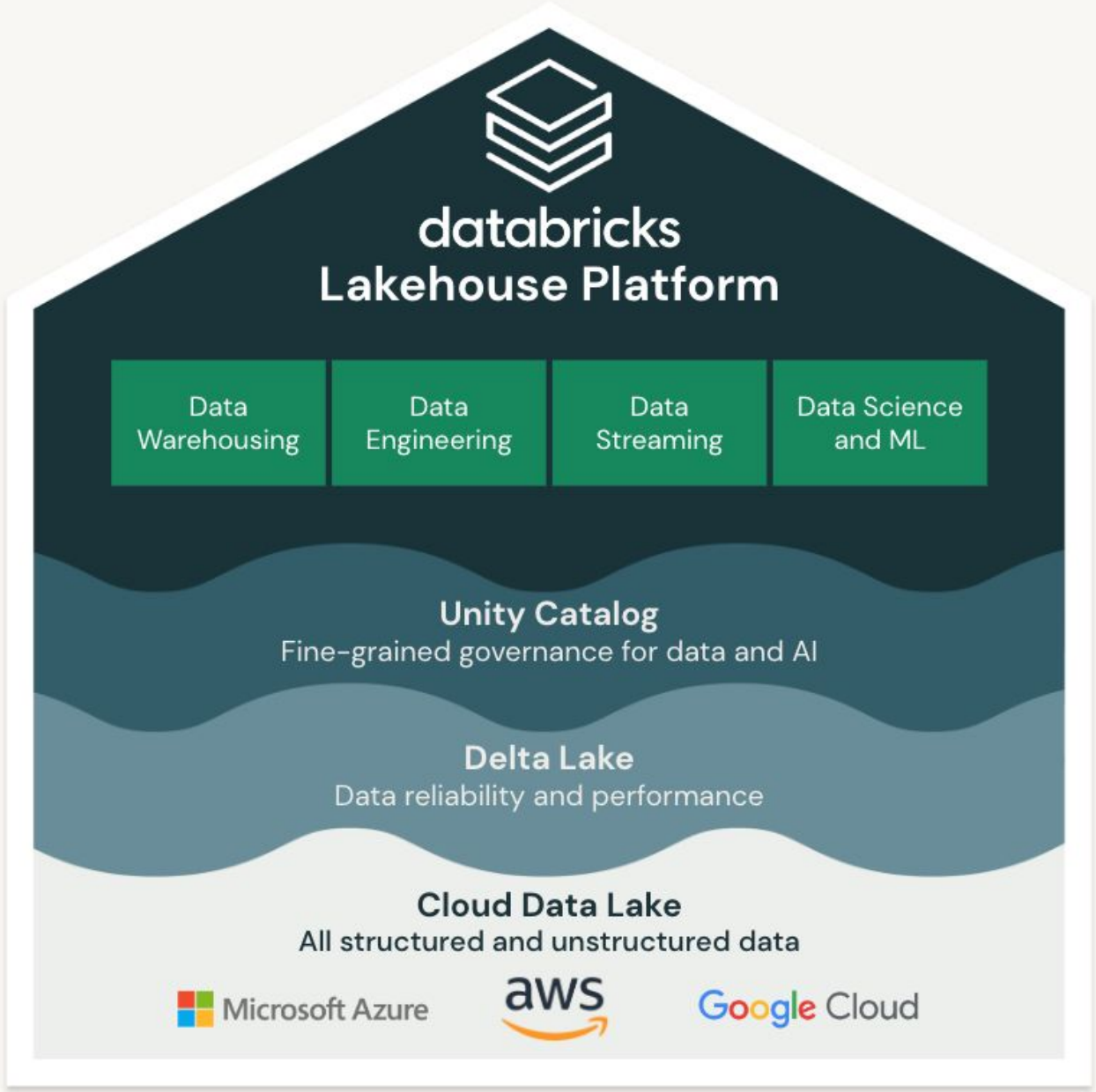
The history of data warehousing started with helping business leaders get analytical insights by collecting data from operational databases into centralized warehouses, which then could be used for decision support and business intelligence (BI). Data in these warehouses would be written with schema-on-write, which ensured that the data model was optimized for downstream BI consumption. quality and governance downstream. In this architecture, a small subset of data in the lake would later be ETLed to a downstream data warehouse (such as Teradata) for the most important decision support and BI applications. The use of open formats also made data lake data directly accessible to a wide range of other analytics engines, such as machine learning systems [30, 37, 42].

From 2015 onwards, cloud data lakes, such as S3, ADLS and GCS, started replacing HDFS. They have superior durability (often >10 nines), geo-replication, and most importantly, extremely low cost with the possibility of automatic, even cheaper, archival storage, e.g., AWS Glacier. The rest of the architecture is largely the same in the cloud as in the second generation systems, with a downstream data warehouse such as Redshift or Snowflake. This two-tier data lake + warehouse architecture is now dominant in the industry in our experience (used at virtually all Fortune 500 enterprises).

This brings us to the challenges with current data architectures. While the cloud data lake and warehouse architecture is ostensibly cheap due to separate storage (e.g., S3) and compute (e.g., Redshift), a two-tier architecture is highly complex for users. In the first generation platforms, all data was ETLed from operational data systems directly into a warehouse. In today's architectures, data is first ETLed into lakes, and then again ELTed into warehouses, creating complexity, delays, and new failure modes. Moreover, enterprise use cases now include advanced analytics such as machine learning, for which *neither* data lakes nor warehouses are ideal. Specifically, today's data architectures commonly suffer from four problems:

Reliability. Keeping the data lake and warehouse consistent is difficult and costly. Continuous engineering is required to ETL data between the two systems and make it available to high-performance

Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics. M. Armbrust, A. Ghodsi, R. Xin, M. Zaharia. 11th Annual Conference on Innovative Data Systems Research (CIDR '21), January 11–15, 2021, Online.



Databricks Lakehouse Platform

Simple

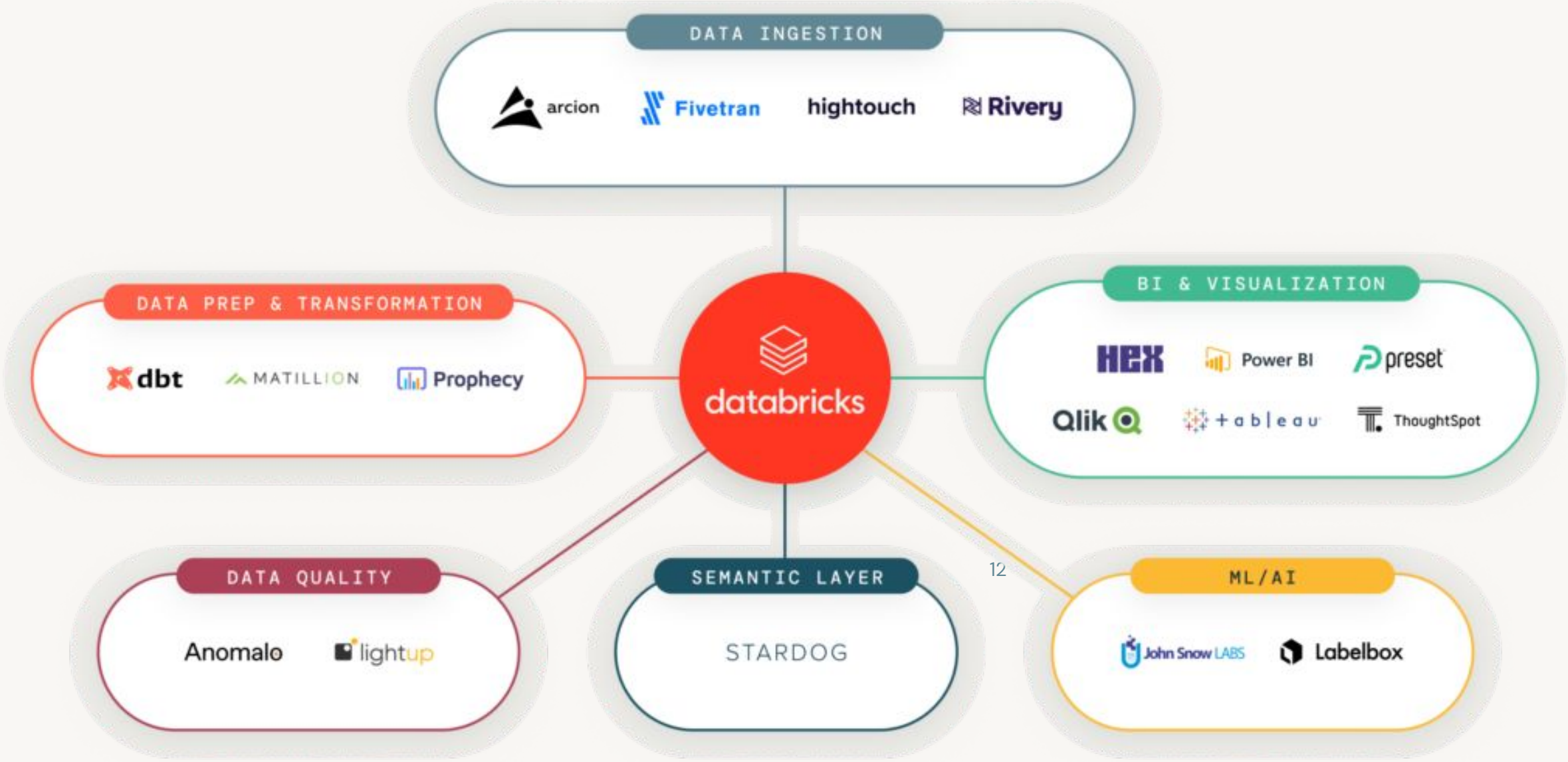
Unify your data warehousing and AI use cases on a single platform

Open

Built on open source and open standards

Multicloud

One consistent data platform across clouds



The lakehouse is for all data practitioners

Data Engineers

Delta Live Tables

Delta Lake

Unity Catalog

Data Analysts

Databricks SQL

Visualizations

Machine Learning Practitioners

ML Flow

Feature Store

Demo:
The Lakehouse Platform

Learning objectives

Things you'll be able to do after completing this lesson

- Describe the navigational interface to the Databricks Lakehouse Platform, including the organization of services and capabilities on the left-side navigation bar and the availability of settings in the top-right corner.
- Describe the Workspace as the solution for organizing assets within Databricks.

Learning objectives

Things you'll be able to do after completing this lesson

- Identify that many different types of assets can be accessed from and organized within the Workspace, including notebooks and files.
- Navigate throughout the Workspace.
- Perform the actions available in the Workspace.

Demo

High-level steps

Overview of the UI

- Landing page
- Navigation
- Top-right menu

Workspace menu features

- Local to the workspace

Introduction to Repos on Databricks

Learning objectives

Things you'll be able to do after completing this lesson

- Describe Repos as a capability centered around continuous integration of assets in Databricks and external Git repositories.

Databricks Repos

Git Versioning

Native integration with Github, Gitlab, Bitbucket and Azure Devops

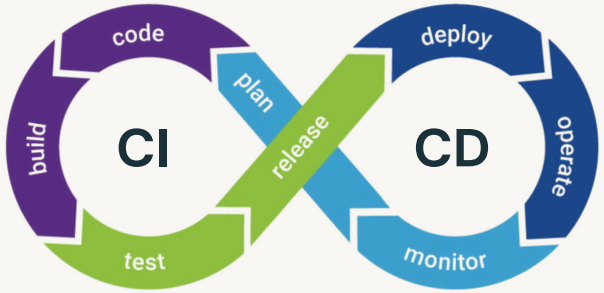
UI-based workflows

Azure DevOps GitHub GitLab Bitbucket

CI/CD Integration

API surface to integrate with automation

Simplifies the dev/staging/prod multi-workspace story



Enterprise ready

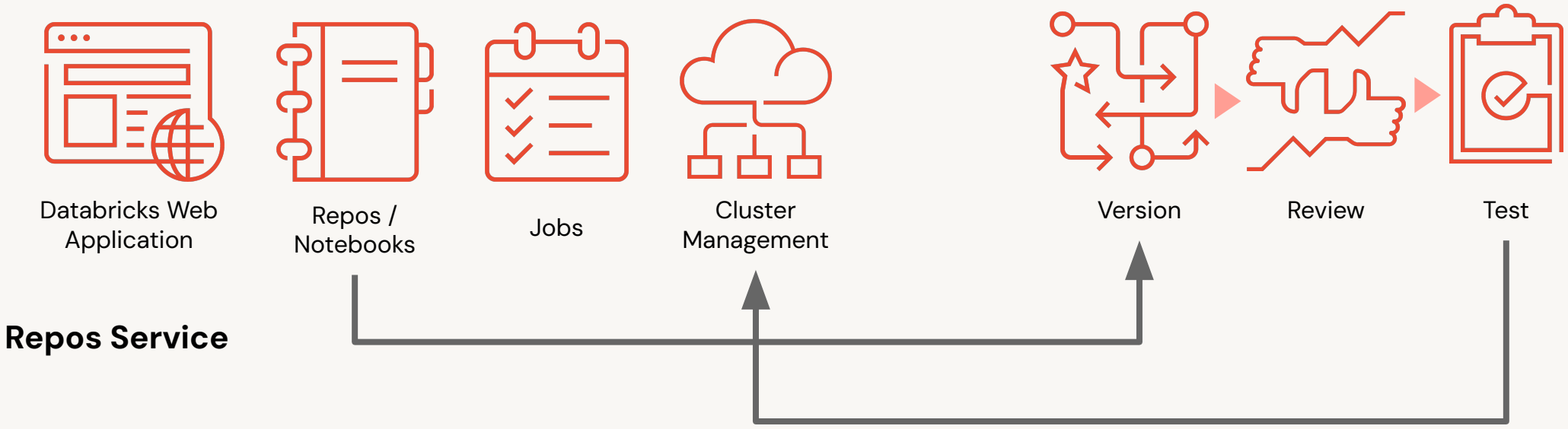
Allow lists to avoid exfiltration

Secret detection to avoid leaking keys

Databricks Repos
CI/CD Integration

Control Plane in Databricks
Manage customer accounts, datasets, and clusters

Git and CI/CD Systems



Demo: Working with Repos on Databricks

Learning objectives

Things you'll be able to do after completing this lesson

- Add a repo from an existing Git repository.
- Describe how to compare, pull, and push changes between Databricks and a Git repository.
- Create a notebook
- Change the name of a notebook

Demo

High-level steps

Repos

- Cloning repos
- Pulling from repos
- CI operations
- Compare and push

Introduction to Compute Resources

Learning objectives

Things you'll be able to do after completing this lesson

- Describe the basic cloud-based compute structure of Databricks.
- Compare and contrast clusters and warehouses.
- Describe the high-level configuration options in a cluster.
- Describe the high-level configuration options in a warehouse.
- Describe the benefits of using the available serverless compute features.

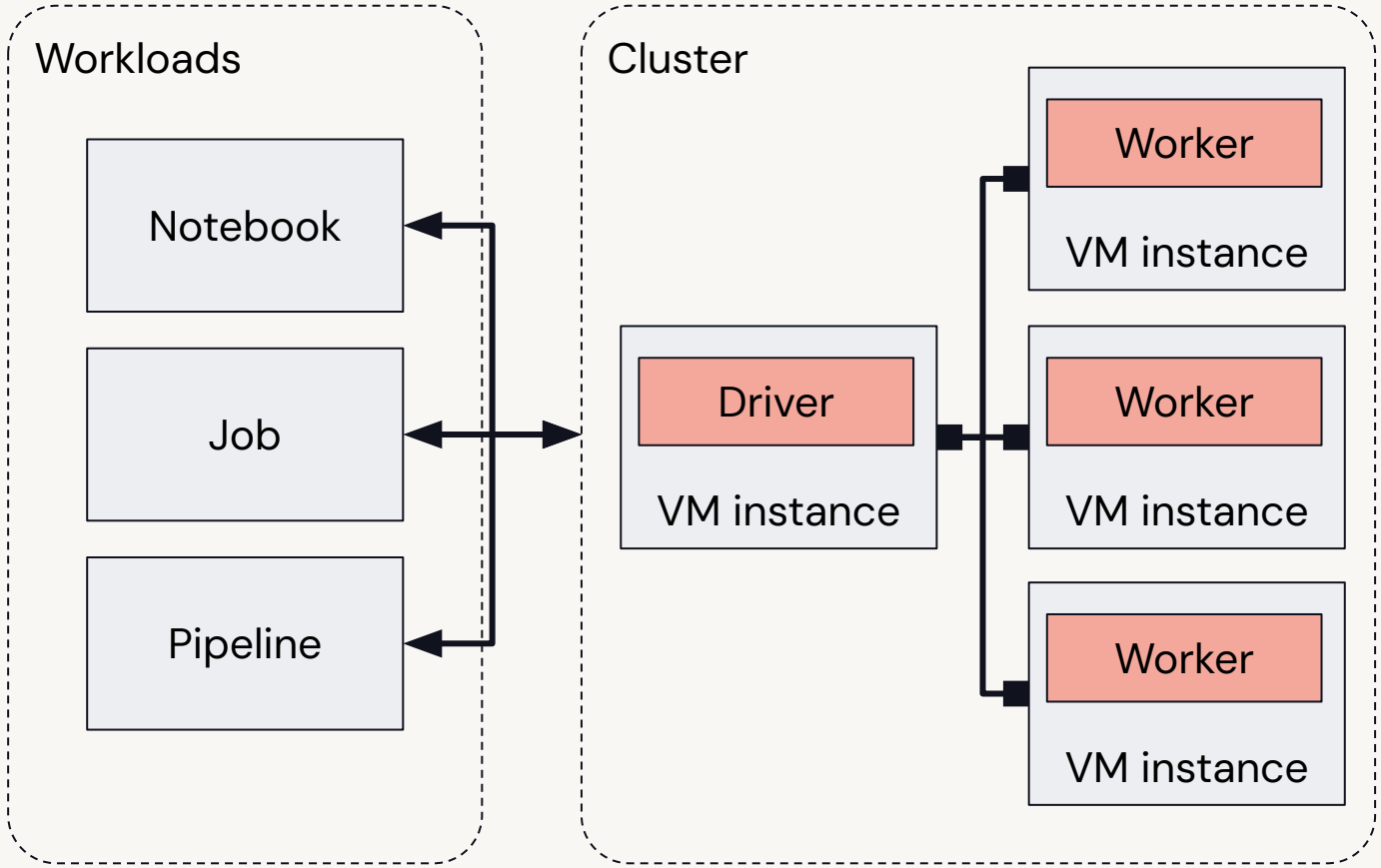
Clusters
Overview

Collection of VM instances

Distributes workloads across workers

Two main types:

1. **All-purpose** clusters for interactive development
2. **Job** clusters for automating workloads



Cluster Types

All-purpose Clusters

Analyze data collaboratively using **interactive** notebooks

Job Clusters

Run **automated** jobs

The Databricks job scheduler creates job clusters when running jobs

Cluster Mode

Single node

Low-cost single-instance cluster catering to single-node machine learning workloads and lightweight exploratory analysis

Standard (Multi Node)

Default mode for workloads developed in any supported language (requires at least two VM instances)

Databricks Runtime Version

Standard

Apache Spark and many other components and updates to provide an optimized big data analytics experiences

Photon

An optional add-on to optimize Spark queries (e.g. SQL, DataFrame)

Machine learning

Adds popular machine learning libraries like TensorFlow, Keras, PyTorch, and XGBoost.

Access Mode

Access mode dropdown	Visible to user	Unity Catalog support	Supported languages
Single user	Always	Yes	Python, SQL, Scala, R
Shared	Always (Premium plan required)	Yes	Python (DBR 11.1+), SQL
No isolation shared	Can be hidden by enforcing user isolation in the admin console or configuring account-level settings	No	Python, SQL, Scala, R
Custom	Only shown for existing clusters *without* access modes (i.e. legacy cluster modes, Standard or High Concurrency); not an option for creating new clusters.	No	Python, SQL, Scala, R

Cluster Policies

Cluster policies can help to achieve the following:

- Standardize cluster configurations
- Provide predefined configurations targeting specific use cases
- Simplify the user experience
- Prevent excessive use and control cost
- Enforce correct tagging

Cluster Access Control

	No Permissions	Can Attach To	Can Restart	Can Manage
Attach notebook		✓	✓	✓
View Spark UI, cluster metrics, driver logs		✓	✓	✓
Start, restart, terminate			✓	✓
Edit				✓
Attach library				✓
Resize				✓
Change permissions				✓

Demo: Working with Compute Resources

Learning objectives

Things you'll be able to do after completing this lesson

- Launch a new cluster.
- Launch a new warehouse.

Demo

High-level steps

Clusters

- Configure and launch a cluster

Warehouses

- Configure and launch a warehouse

Knowledge check

Think about this question and volunteer an answer

<Add question stem>

- Add option 1
- Add option 2
- Add option 3
- Add option 4

Knowledge check

Think about this question and volunteer an answer

<Add question stem>

- Add option 1
- Add option 2
- Add option 3
- Add option 4

Knowledge check

Think about this question and volunteer an answer

<Add question stem>

- Add option 1
- Add option 2
- Add option 3
- Add option 4

Databricks Notebooks

Learning objectives

Things you'll be able to do after completing this lesson

- Describe Databricks Notebooks as the most common interface for data engineers when working with Databricks.
- Recognize common use cases for data engineers when working with Notebooks.
- Explore basic visualization capabilities in Databricks Notebooks.

Databricks Notebooks

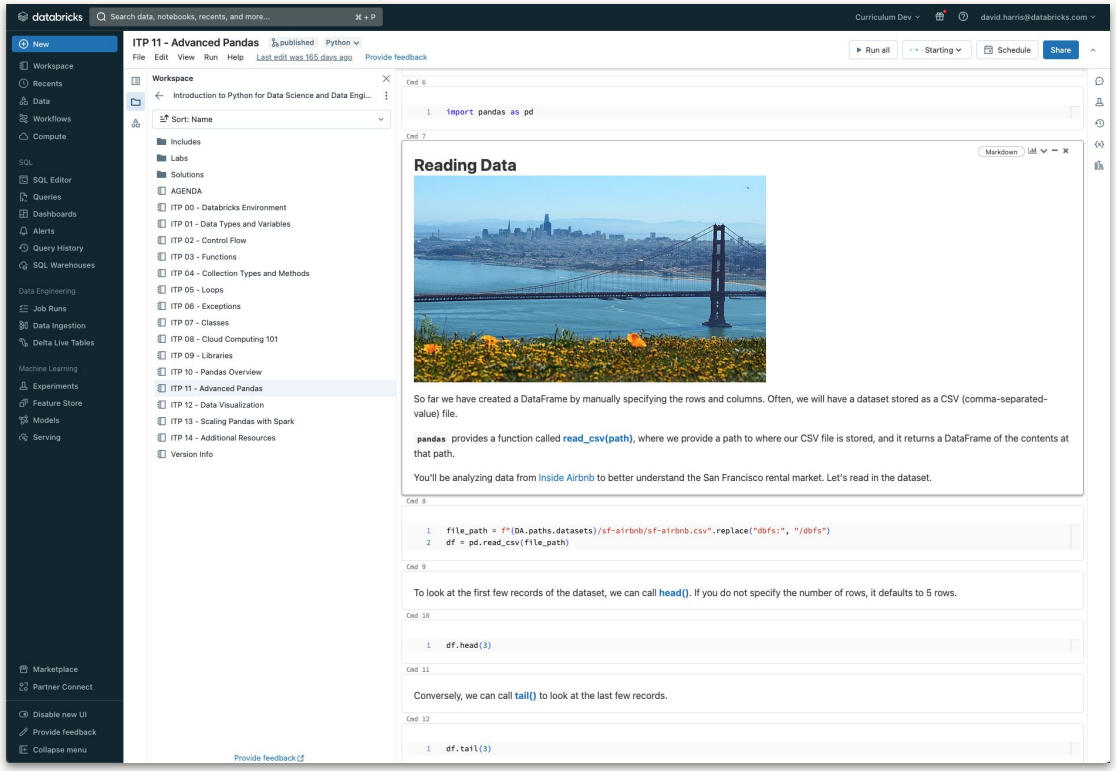
Collaborative, reproducible, and enterprise ready

Multi-language
Use Python, SQL, Scala, and R, all in one Notebook

Collaborative
Real-time co-presence, co-editing, and commenting

Ideal for exploration
Explore, visualize, and summarize data with built-in charts and data profiles

Adaptable
Install standard libraries and use local modules



Reproducible
Automatically track version history, and use git version control with Repos

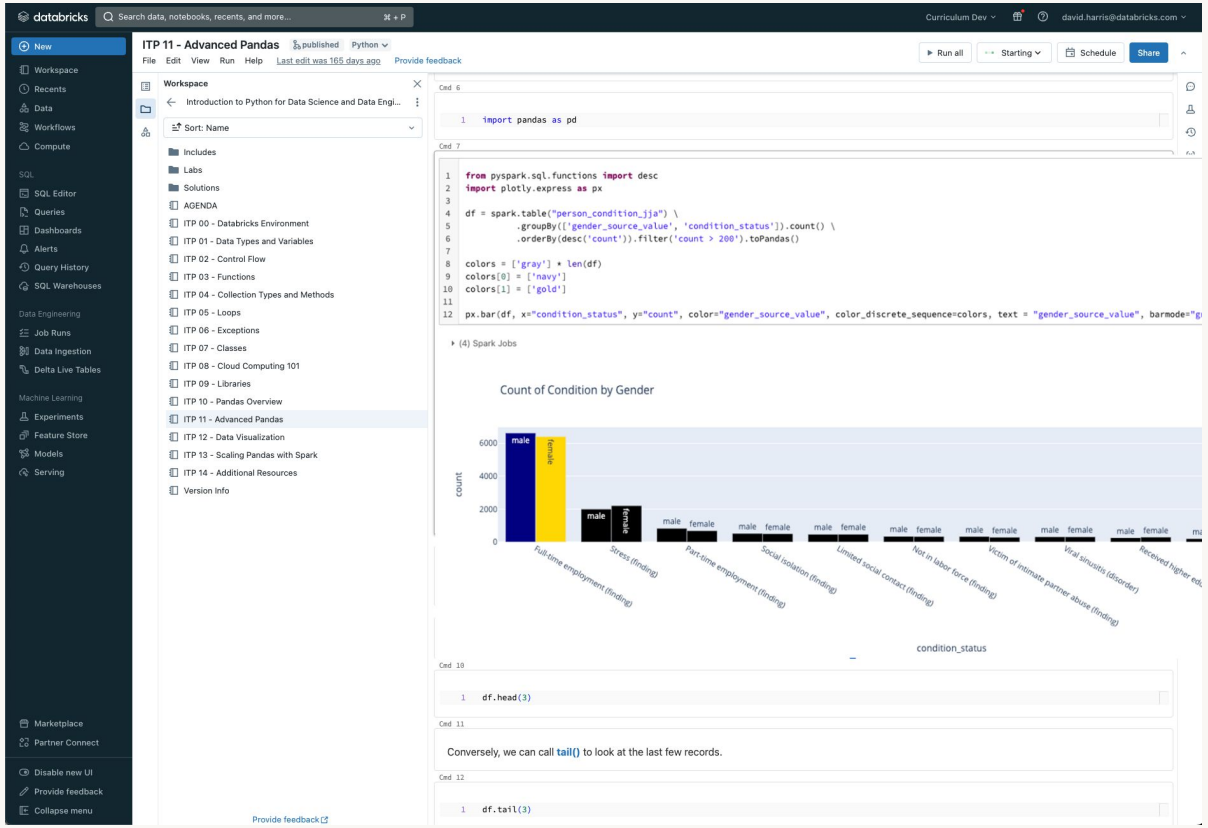
Get to production faster
Quickly schedule notebooks as jobs or create dashboards from their results, all in the Notebook

Enterprise-ready
Enterprise-grade access controls, identity management, and auditability

Databricks Notebooks

Easily develop standard or custom visualizations

- Create visualizations based on query results or dataframes
- Can use SQL, Python, or Scala
  - Use SQL for out-of-the-box, standard visualizations
  - Use Python and Scala for custom visualizations
- Stitch together custom and standard visuals
- Can be used on existing tables or can write model results to a table for model monitoring



Demo: Working with Notebooks

Learning objectives

Things you'll be able to do after completing this lesson

- Write code in, and run a Notebook.
- Write markdown-based notes in a Notebook.
- Run code using multiple languages within the same Notebook.

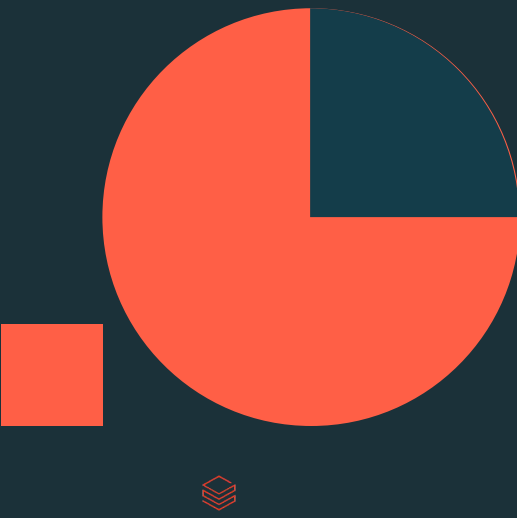
Demo

High-level steps

Notebooks

- About notebooks
- Writing code and markdown
- Magic commands
- Creating a visualization

Module 2



Module 2 - Agenda

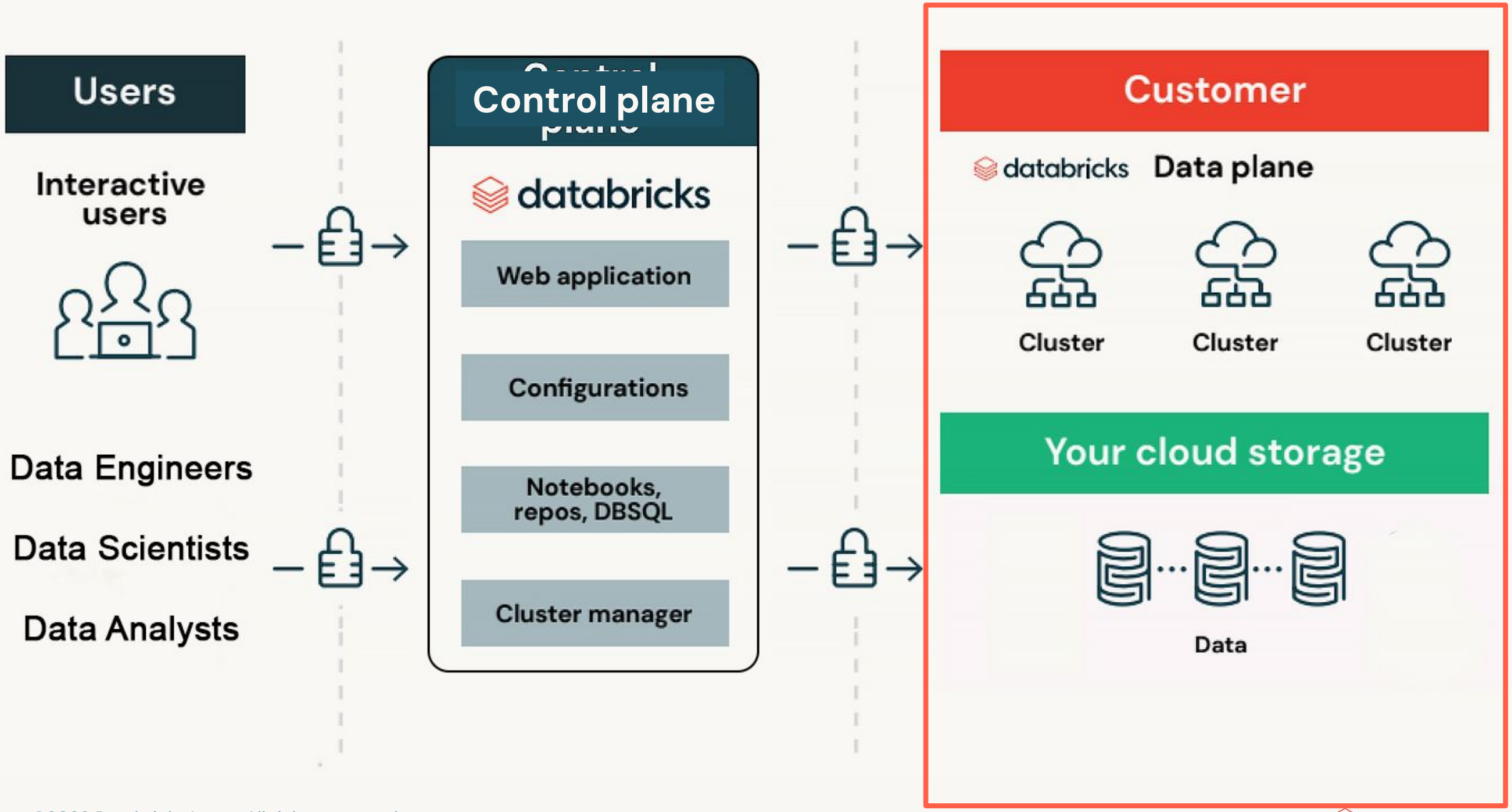
Lesson Name	Lesson Name
2.1 - Lecture: Data Storage and Delta Lake	2.6 - Demo: Using Workflow Jobs
2.2 - Lecture: Unity Catalog	2.7 - Lecture: Databricks SQL for Data Engineers
2.3 - Demo: Data Management	2.8 - Demo: Using Databricks SQL
2.4 - Demo: Data Governance and Security	2.9 - Comprehensive Lab
2.5 - Lecture: Introduction to Workflow Jobs	

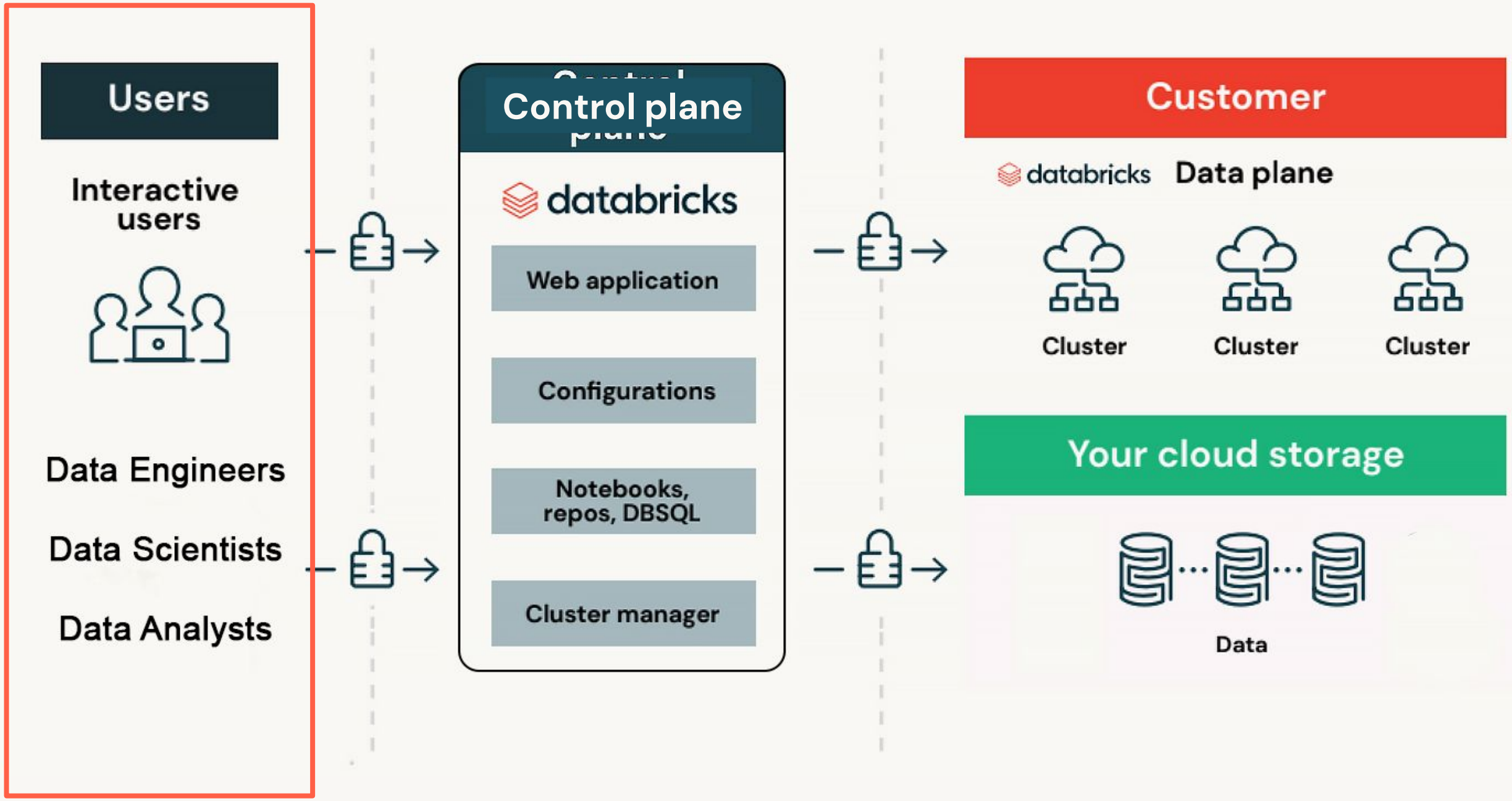
Data Storage and Delta Lake

Learning objectives

Things you'll be able to do after completing this lesson

- Describe that data is stored in cloud object storage locations and accessed via Databricks.
- Explain the benefits of data storage in the data lakehouse architecture across roles and Databricks services.
- Identify Delta Lake as the optimized storage layer that provides the foundation for data storage for the data lakehouse.
- Identify that all tables in Databricks are Delta tables by default.
- Identify that Delta Lake has a series of built-in and easy optimizations to improve performance.





What is Delta Lake?

Delta Lake is the default format for tables created in Databricks

```
CREATE TABLE foo
USING DELTA
```

```
df.write
  .format("delta")
```

Delta Lake is an open-source project that enables building a data lakehouse on top of existing cloud storage

Delta Lake brings ACID to object storage

Atomicity means all transactions either succeed or fail completely

Consistency guarantees relate to how a given state of the data is observed by simultaneous operations

Isolation refers to how simultaneous operations conflict with one another. The isolation guarantees that Delta Lake provides do differ from other systems

Durability means that committed changes are permanent

Problems solved by ACID

- Hard to append data
- Modification of existing data difficult
- Jobs failing mid way
- Real-time operations hard
- Costly to keep historical data versions

Unity Catalog

Learning objectives

Things you'll be able to do after completing this lesson

- Describe Unity Catalog as a centralized governance solution in Databricks.
- Explain the three-tier namespace and its levels.

Unity Catalog
Overview

Unified governance across clouds

Fine-grained governance for data lakes across clouds - based on open standard ANSI SQL.

1

Unified data and AI assets

Centrally share, audit, secure and manage all data types with one simple interface.

2

Unified existing catalogs

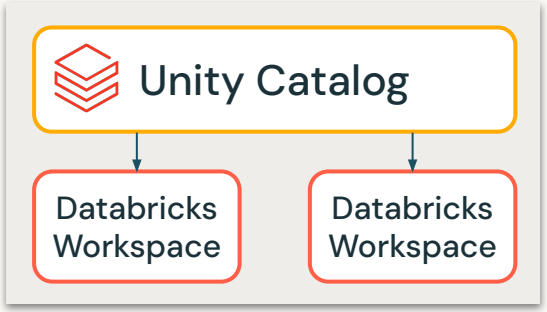
Works in concert with existing data, storage, and catalogs - no hard migration required.

3

Unity Catalog

Key Capabilities

- Centralized metadata and user management
- Centralized data access controls
- Data access auditing
- Data lineage
- Data search and discovery
- Secure data sharing with Delta Sharing



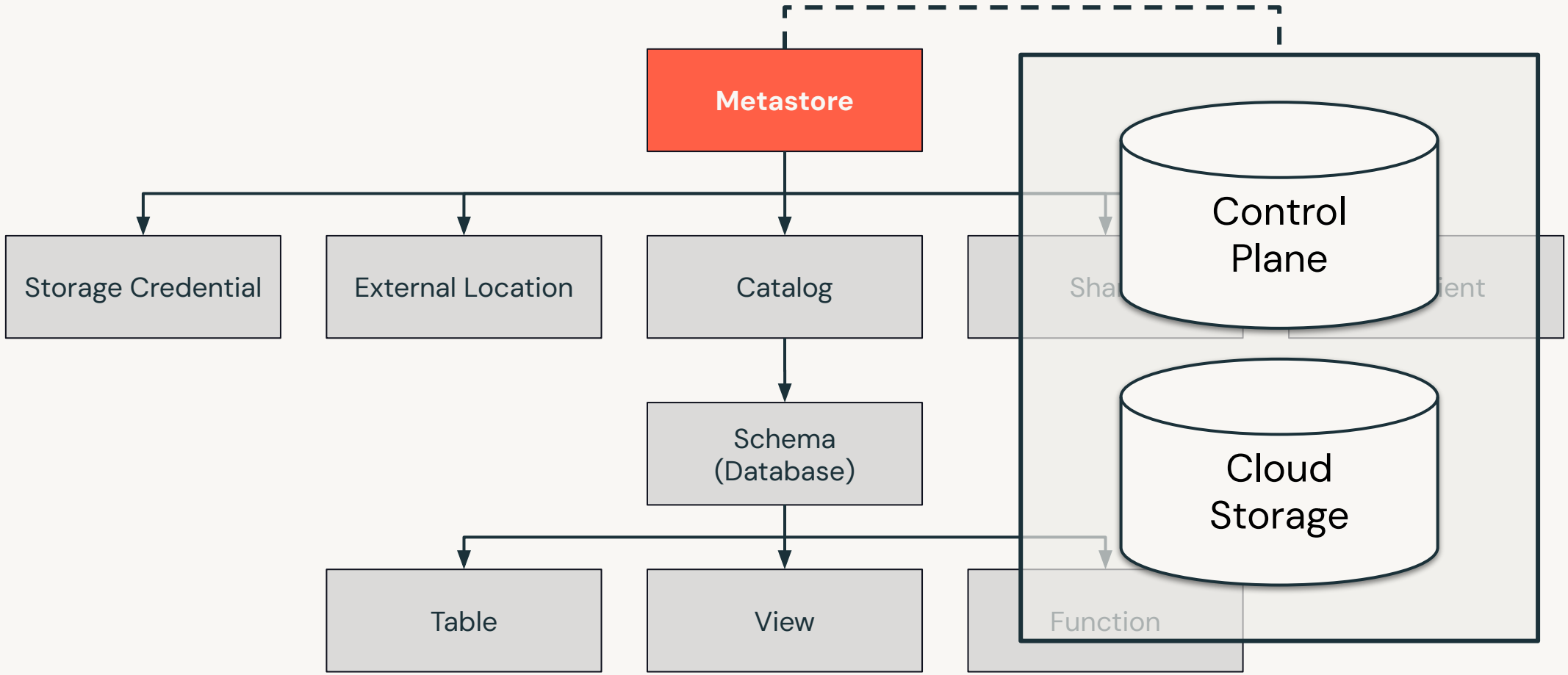

```
GRANT ... ON ... TO ...
REVOKE ... ON ... FROM ...
```

Catalogs, Databases (schemas), Tables, Views, Storage credentials, External locations

Unity Catalog Key Concepts

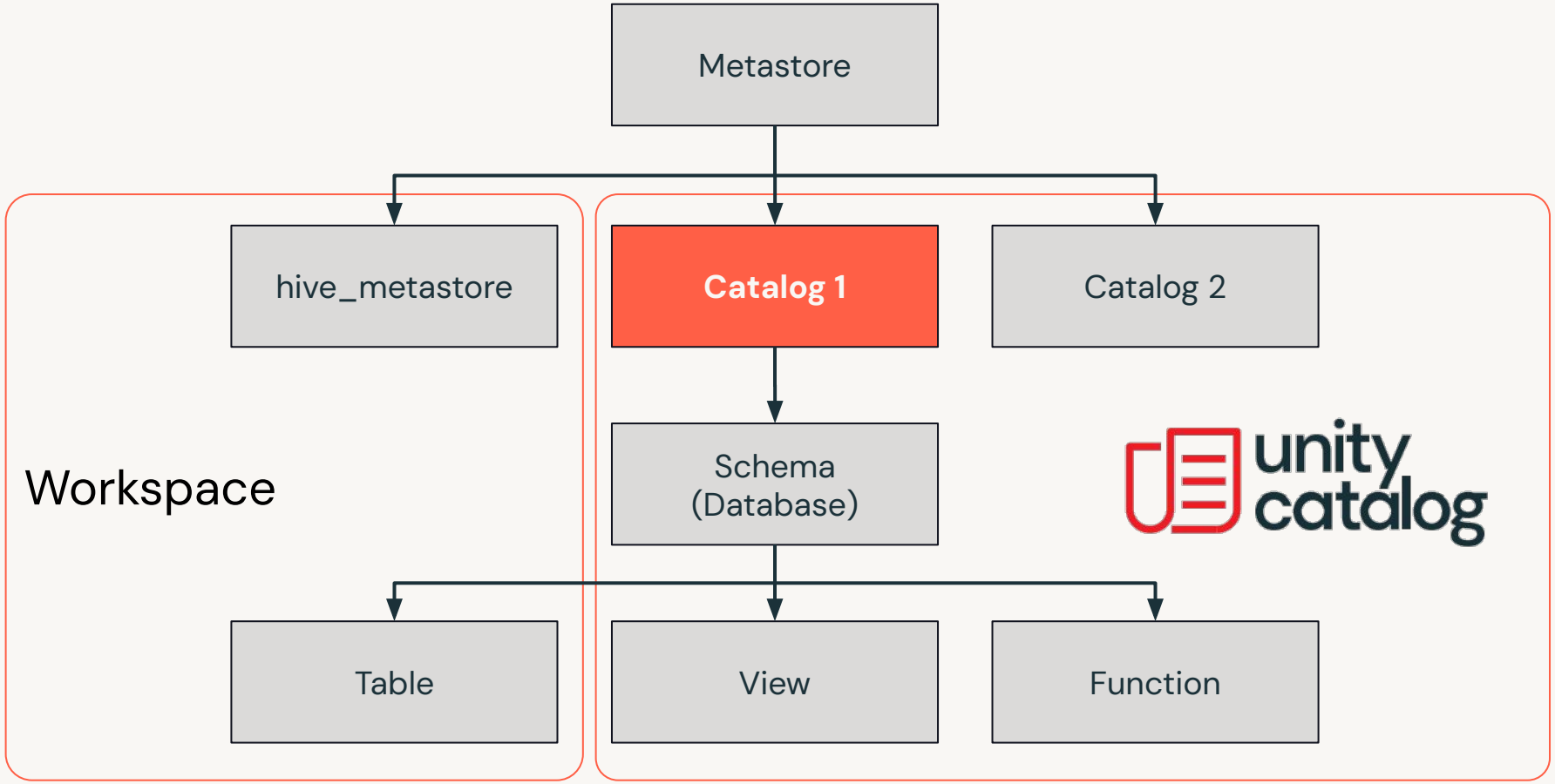
Metastore

Unity Catalog metastore elements



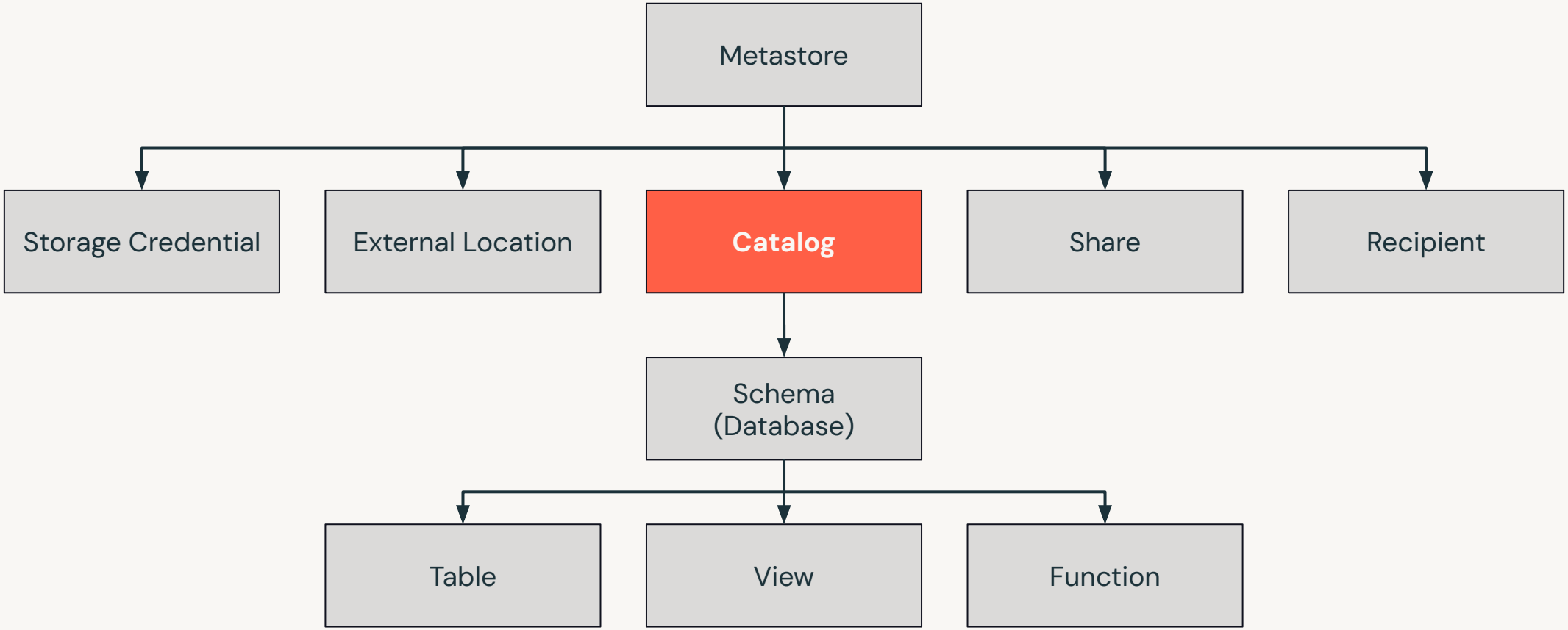
Metastore

Accessing legacy Hive metastore



Catalog

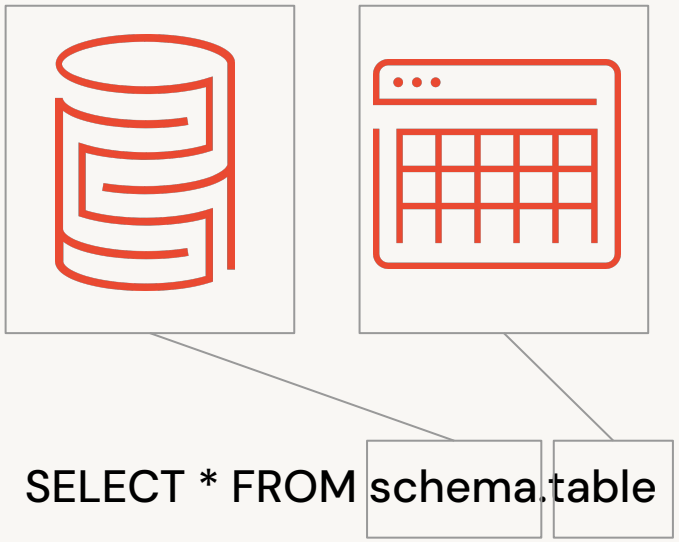
Top-level container for data objects



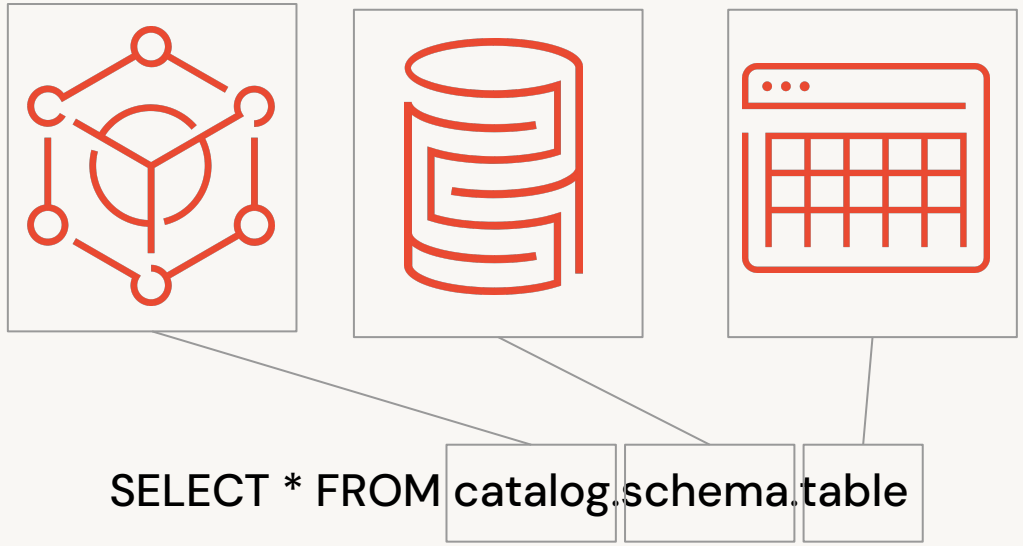
Catalog

Three-level namespace

Traditional SQL two-level namespace

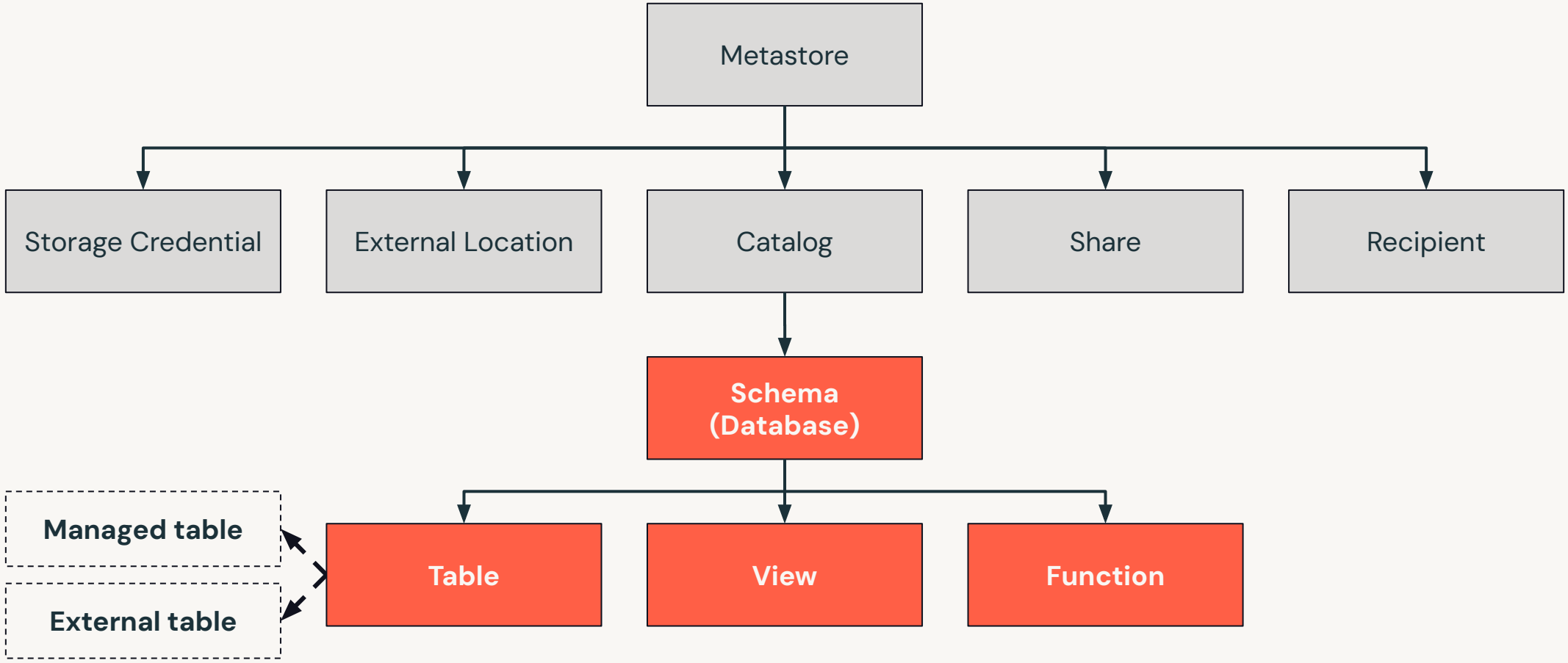


Unity Catalog three-level namespace



Data Objects

Schema (database), tables, views, functions



Demo: Data Management

Learning objectives

Things you'll be able to do after completing this lesson

- Create a new schema in an existing catalog.
- Create a new managed Delta table from an existing cloud file using SQL.
- Create a new managed Delta table from an existing Delta table using SQL.
- Drop a managed Delta table that is no longer needed.

Demo

High-level steps

Working with data

- About Delta tables
- About Unity Catalog
- Catalogs and schemas
- Creating tables

Knowledge check

Think about this question and volunteer an answer

<Add question stem>

- Add option 1
- Add option 2
- Add option 3
- Add option 4

Knowledge check

Think about this question and volunteer an answer

<Add question stem>

- Add option 1
- Add option 2
- Add option 3
- Add option 4

Knowledge check

Think about this question and volunteer an answer

<Add question stem>

- Add option 1
- Add option 2
- Add option 3
- Add option 4

Demo: Data Governance and Security

Learning objectives

Things you'll be able to do after completing this lesson

- Grant another user the appropriate access to SELECT a table.
- Grant a group the appropriate access to SELECT a table.
- Revoke another user's access to a table.

Demo

High-level steps

Governance & security

- Users and groups
- Granting and revoking access

Introduction to Workflow Jobs

Learning objectives

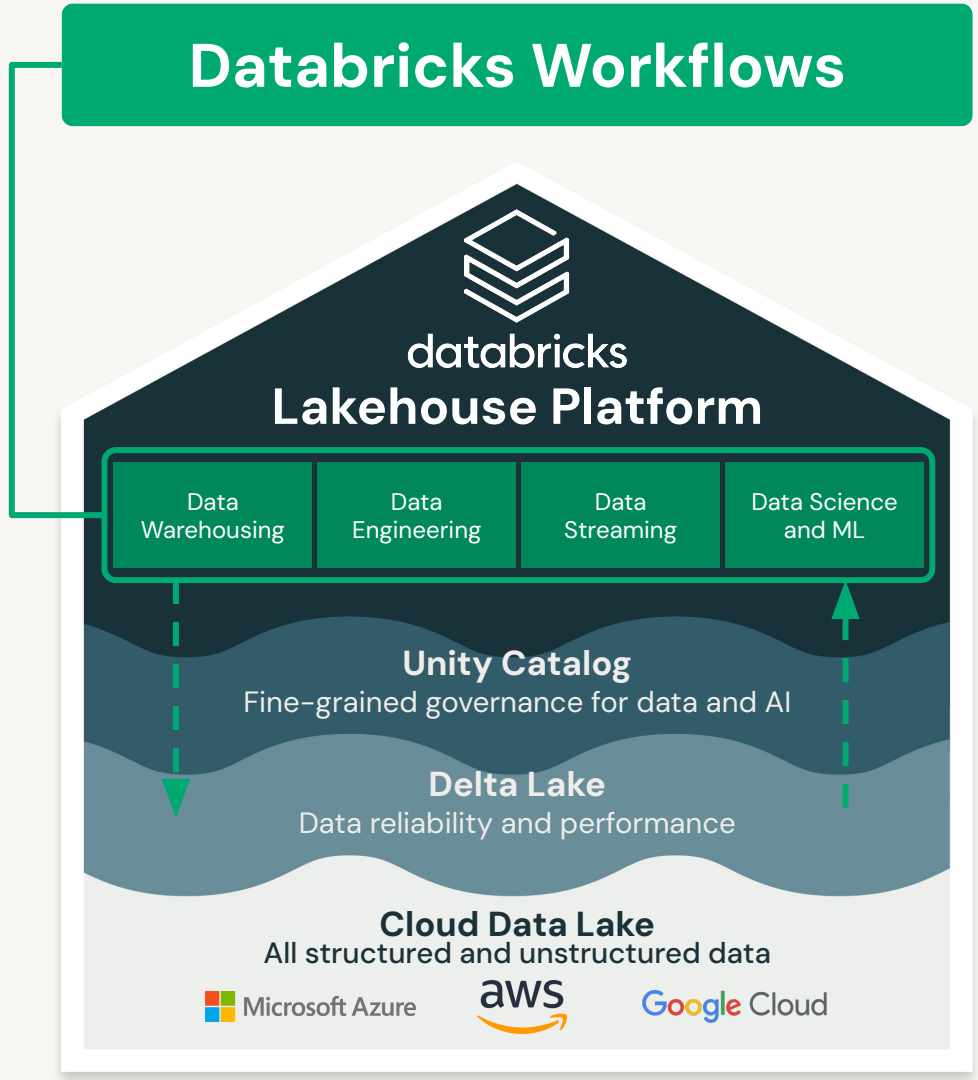
Things you'll be able to do after completing this lesson

- Describe Workflows as a capability to productionize data workflows.
- Describe Jobs as a simple solution to schedule and automate one or more tasks.
- Recognize the types of assets that are able to be automated with Jobs.
- Describe Delta Live Tables as a solution for building and running robust data pipelines.

Databricks Workflows

Workflows is a **fully-managed cloud-based general-purpose task orchestration service** for the entire Lakehouse.

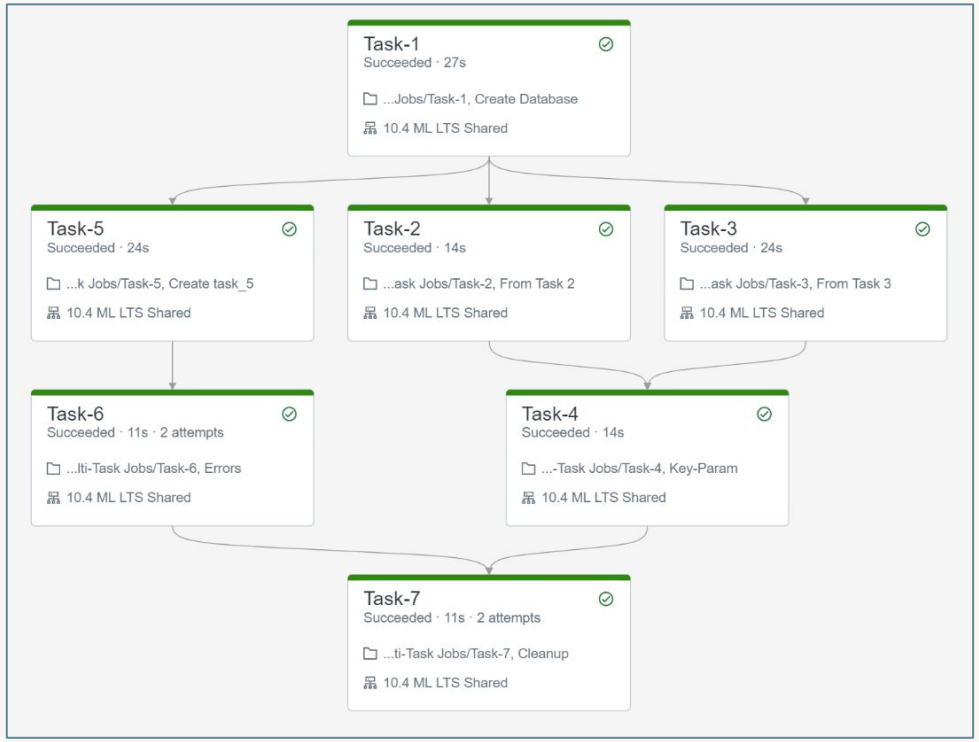
Workflows is a service for data engineers, data scientists and analysts to build reliable data, analytics and AI workflows on any cloud.



Databricks Workflows

Databricks has two main task orchestration services:

- **Workflow Jobs (Workflows)**
  - Workflows for every job
- **Delta Live Tables (DLT)**
  - Automated data pipelines for Delta Lake



Note: DLT pipeline can be a task in a workflow

DLT versus Workflow Jobs

Considerations

	Delta Live Tables	Workflow Jobs
Source	Notebooks only	JARs, notebooks, DLT, application written in Scala, Java, Python
Dependencies	Automatically determined	Manually set
Cluster	Self-provisioned	Self-provisioned or existing
Timeouts and Retries	Timeouts not supported Retries handled automatically (in production mode)	Supported
Import Libraries	Not supported	Supported

Workflow Jobs

Use Cases

Orchestration of Dependent Jobs

Jobs running on schedule, containing dependent tasks/steps

Jobs Workflows

Machine Learning Tasks

Run MLflow notebook task in a job

Jobs Workflows

Arbitrary Code, External API Calls, Custom Tasks

Run tasks in a job which can contain Jar file, Spark Submit, Python Script, SQL task, dbt

Jobs Workflows

How to Leverage Workflows

- Allows you to build simple ETL/ML task orchestration
- Reduces infrastructure overhead
- Easily integrate with external tools
- Enables non-engineers to build their own workflows using simple UI
- Cloud-provider independent
- Enables re-using clusters to reduce cost and startup time

Demo: Using Workflow Jobs

Learning objectives

Things you'll be able to do after completing this lesson

- Automate the running of a single notebook using Jobs.
- Review the results of a completed single-notebook Job.

Demo

High-level steps

Workflows

- About workflows
- Creating and running jobs
- Reviewing job results

Databricks SQL for Data Engineers

Learning objectives

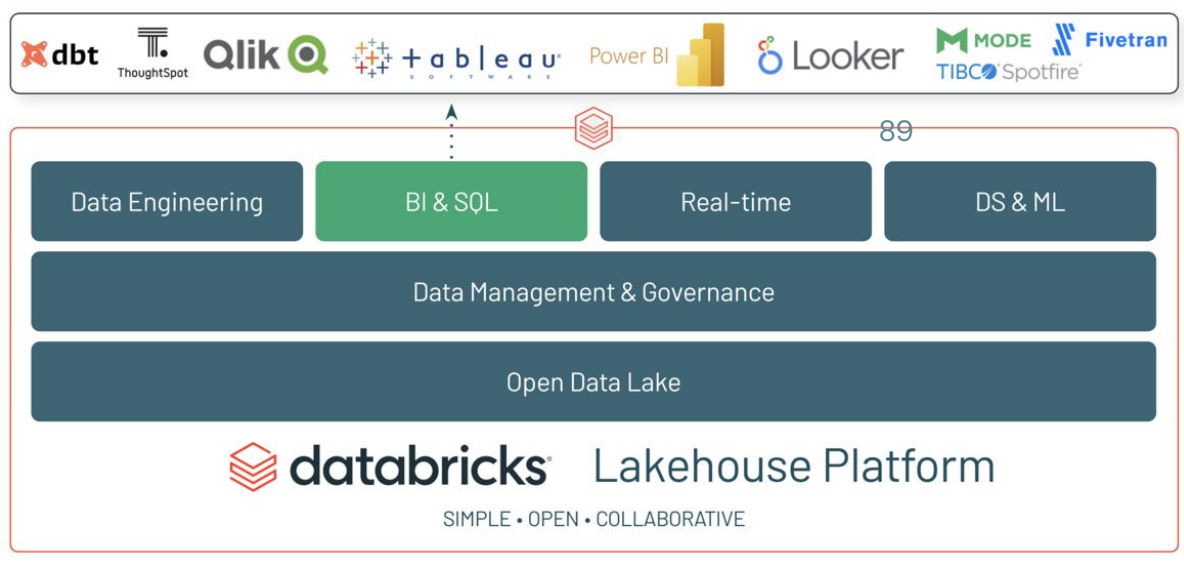
Things you'll be able to do after completing this lesson

- Describe Databricks SQL as a data warehousing solution for analysts and engineers working with Databricks.
- Recognize common use cases for data engineers when working with Databricks SQL.
- Describe the visualization and dashboarding capabilities of Databricks SQL.

Databricks SQL

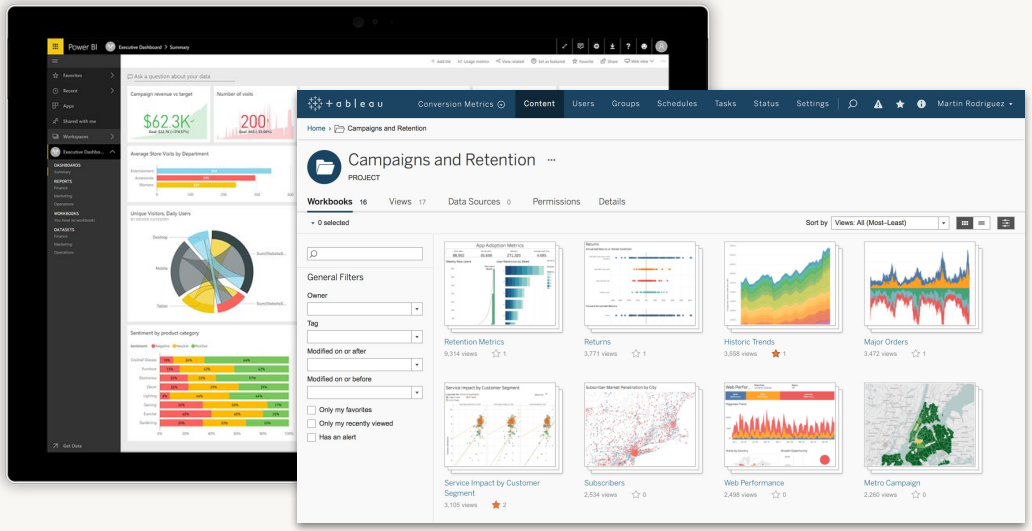
Delivering analytics on the freshest data with data warehouse performance and data lake economics

- Better price/performance than other cloud data warehouses
- Simplify discovery and sharing of new insights
- Connect to familiar BI tools, like Tableau or Power BI
- Simplified administration and governance



Broad integration with BI tools

Connect your preferred BI tools with optimized connectors that provide fast performance, low latency, and high user conconcurrency to your data lake for your existing BI tools.

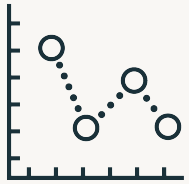


Coming soon:

Use Cases

Visualize Data to Discover Issues

It can be difficult to find issues in data without creating a visualization. Databricks SQL provides the opportunity to produce a wide variety of visualization types that will allow you to examine data and find issues that can be addressed quickly. The ease with which a query can be run, a visualization can be produced, and a dashboard created, as needed, makes Databricks SQL an awesome solution.



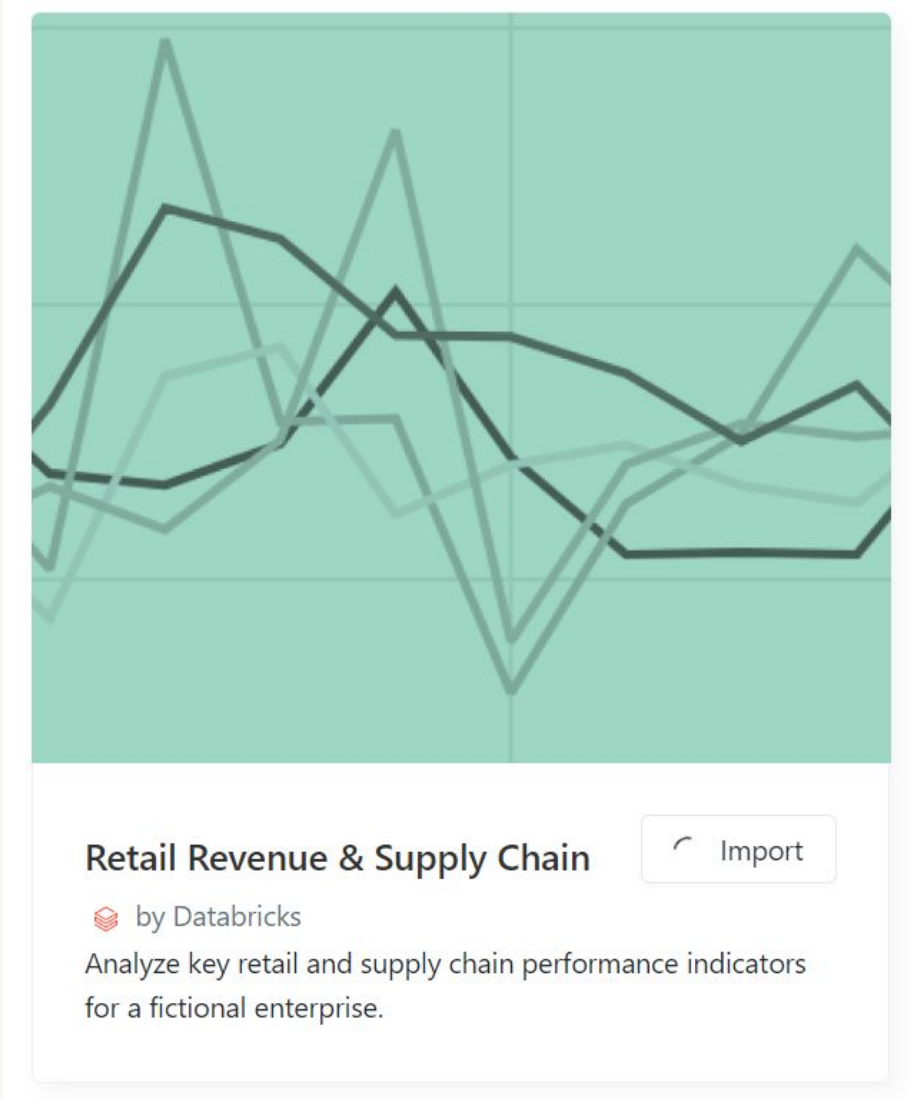
Collaboratively explore the latest and freshest data

Respond to business needs faster with a self-served experience designed for every analysts in your organization. Databricks SQL Analytics provides a simple and secure access to data, ability to create or reuse SQL queries to analyze the data that sits directly on your data lake, and quickly mock-up and iterate on visualizations and dashboards that fit best the business.

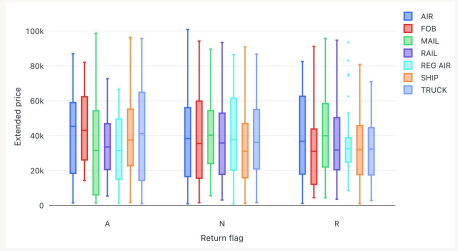
Build data-enhanced applications

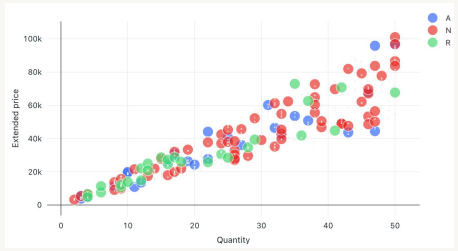
Build rich and custom data enhanced applications for your own organization or your customers. Benefit from the ease of connectivity, management, and better price / performance of Databricks SQL Analytics to simplify development of data-enhanced applications at scale, all served from your data lake.

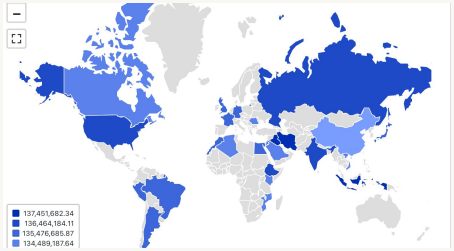
Import Existing Dashboard

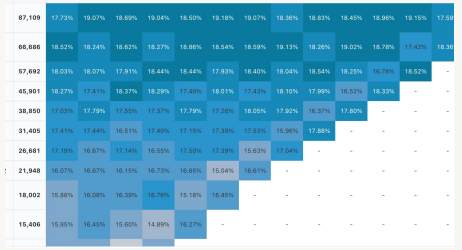


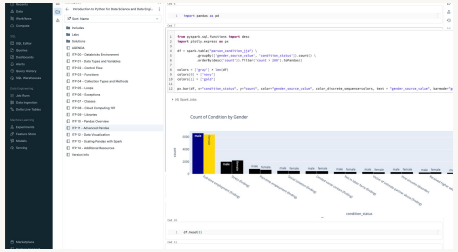
Visualization Types

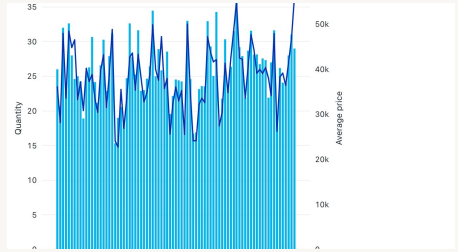


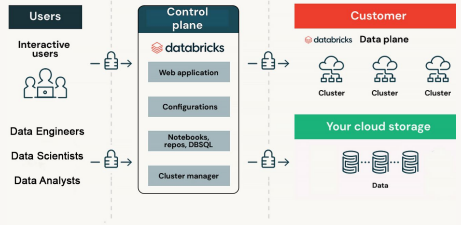


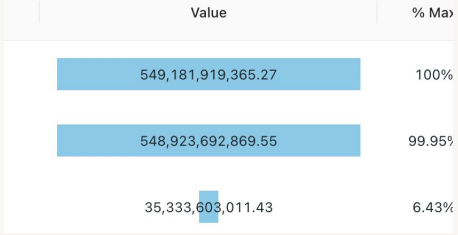


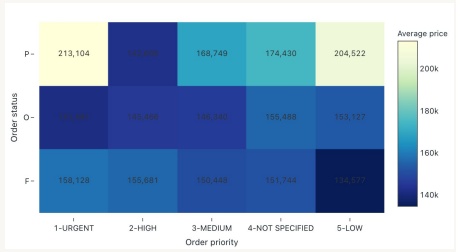


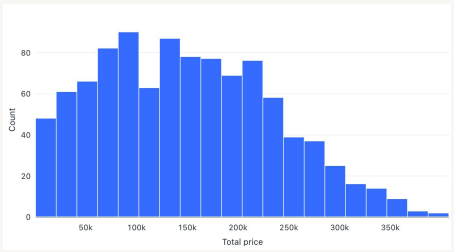


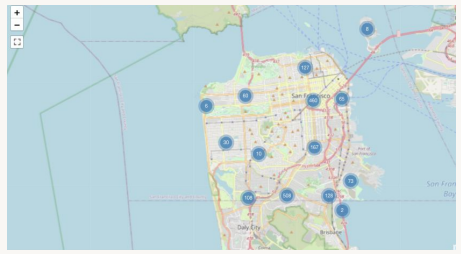


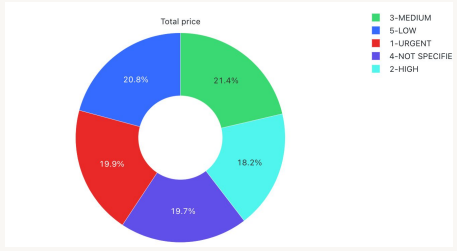


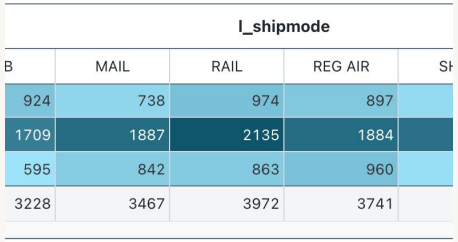


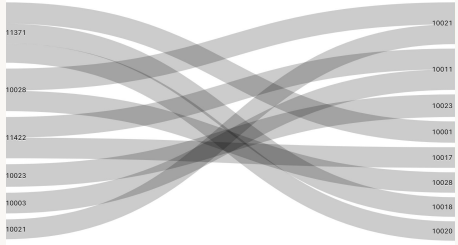


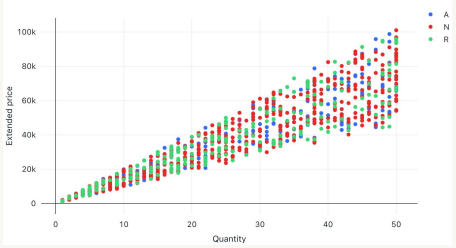


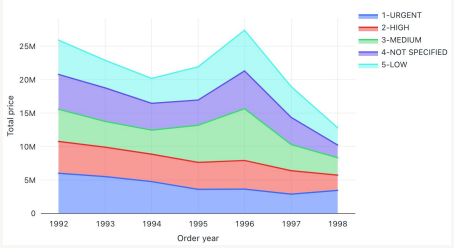


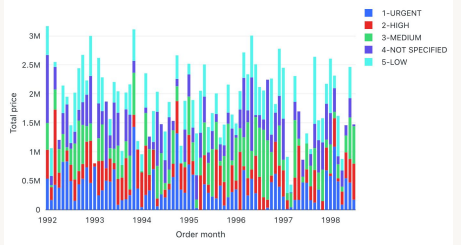


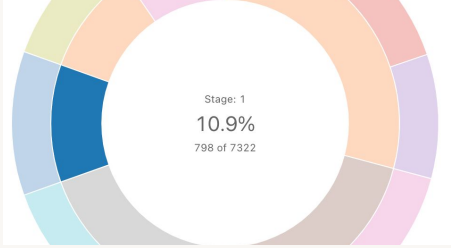












Using Databricks SQL

Learning objectives

Things you'll be able to do after completing this lesson

- Open the SQL Editor.
- Write and run a Query.

Demo

High-level steps

Databricks SQL

- About Databricks SQL
- Creating and running queries
- Creating visualizations

Knowledge check

Think about this question and volunteer an answer

<Add question stem>

- Add option 1
- Add option 2
- Add option 3
- Add option 4

Knowledge check

Think about this question and volunteer an answer

<Add question stem>

- Add option 1
- Add option 2
- Add option 3
- Add option 4

Knowledge check

Think about this question and volunteer an answer

<Add question stem>

- Add option 1
- Add option 2
- Add option 3
- Add option 4

Comprehensive Lab

Learning objectives

Things you'll be able to do after completing this lesson

- Demonstrate how to create a complete data engineering workflow in the Databricks Lakehouse Platform

Lab

High-level steps

Demonstrate your skills

- Work with clusters
- Create a table and grant access
- Write code in a notebook that ingests, cleans, and loads data
- Create and run a job
- Create an run a query
- Create a visualization

databricks
Academy

Course Summary and Next Steps

Summary and next steps

What did we cover? What should you do next?

Session summary

- Introduction to the Databricks Lakehouse Platform
- Basic skills for running a data engineering workflow

Helpful resources

- Data Engineering Docs

Next Steps

- Take the course, "Data Engineering on Databricks," for a more comprehensive look
- Take a look at our Certification in Data Engineering