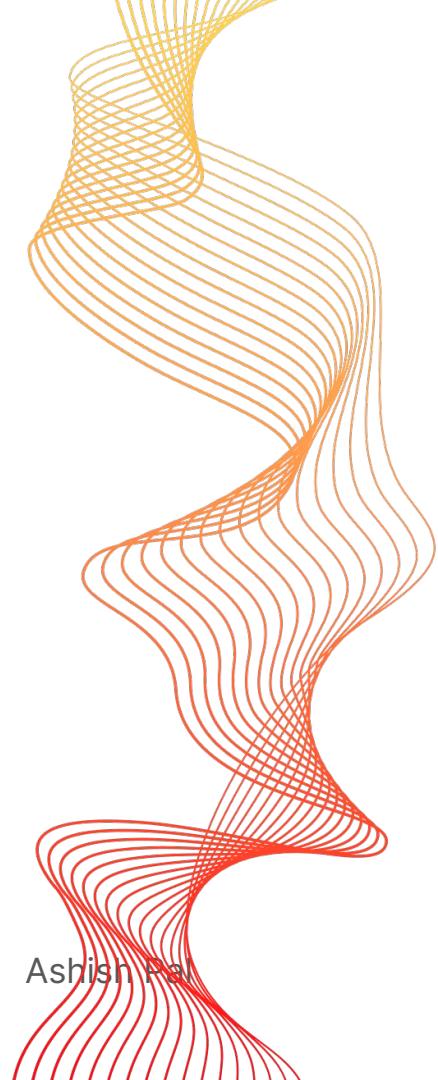




Databricks

An overview of Databricks and its features.



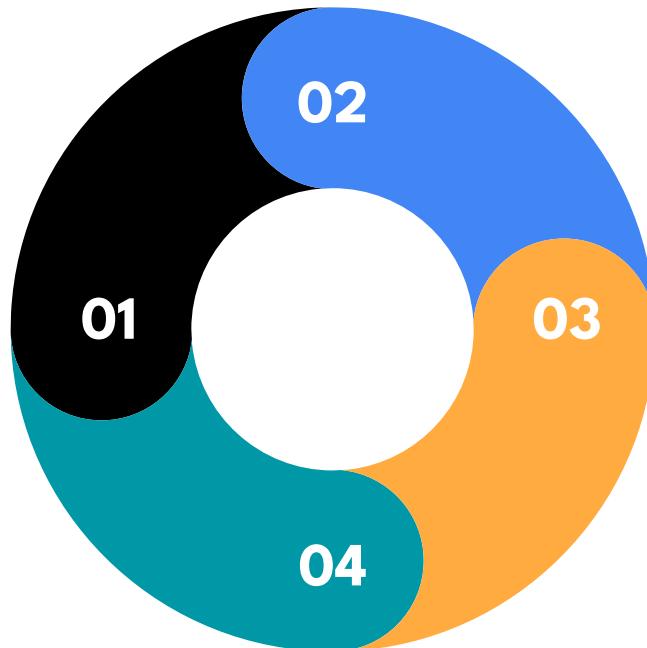
A large, abstract graphic on the right side of the slide consists of numerous thin, curved lines that form a complex, flowing pattern. The lines transition in color from orange at the top to red towards the bottom, creating a sense of depth and motion.

Ashish Patel

What is Databricks?

Databricks is a unified analytics platform.

It is built for massive scale and high-performance analytics.



Databricks provides a collaborative environment for data scientists, data engineers, and business analysts.

It supports various programming languages like Python, Scala, and R.

Ashish Pal

Key Features

Advanced analytics capabilities.



Scalable and elastic cloud infrastructure.

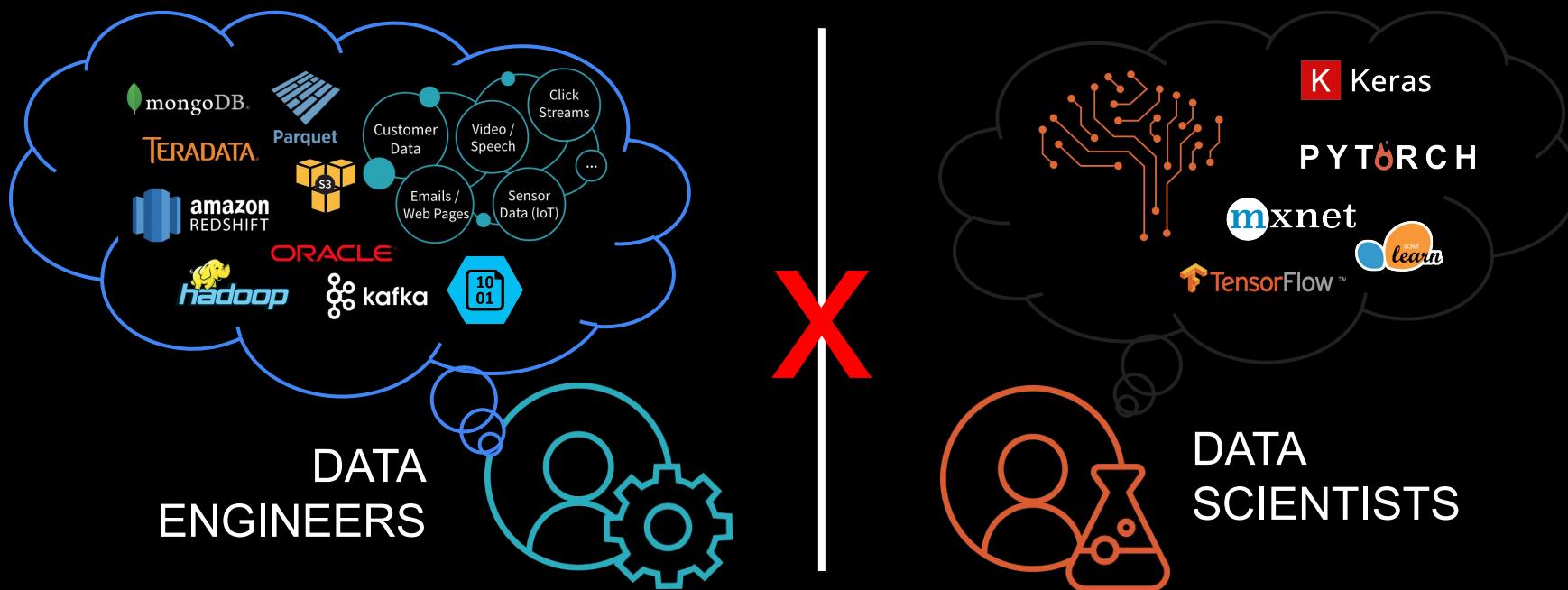
Built-in collaboration tools for teams.

Real-time data processing.

Integration with popular data sources.

Ashish Pal

Data & AI People are in Silos



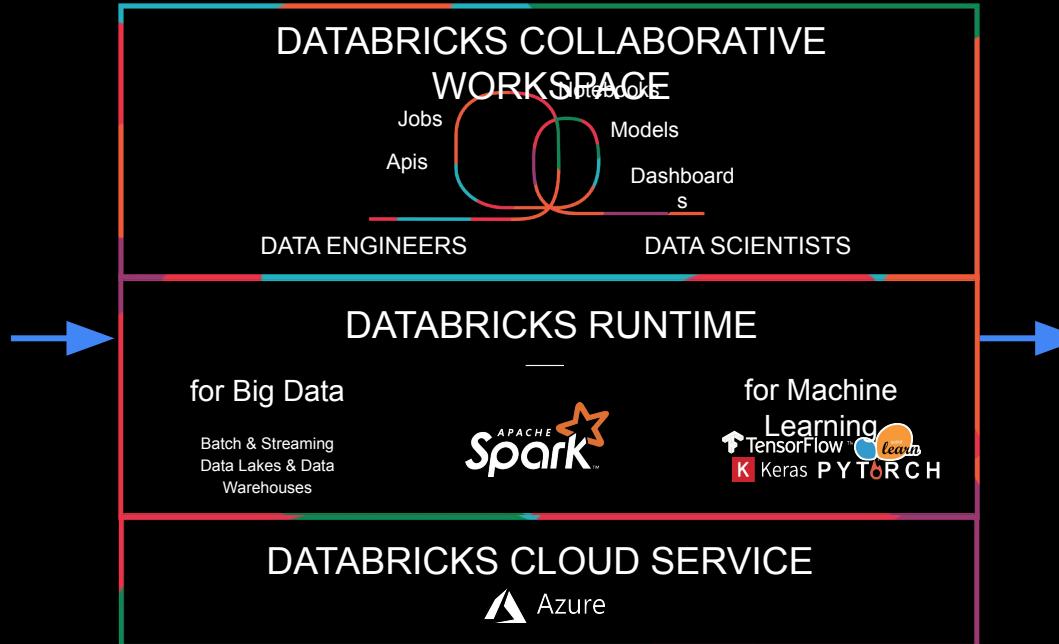
Ashish Pal



Azure Databricks

AZURE DATA SOURCES

- Blob Storage
- Data Lake Store
- SQL Data Warehouse
- Cosmos DB
- Event Hub
- IoT Hub
- Azure Data Factory



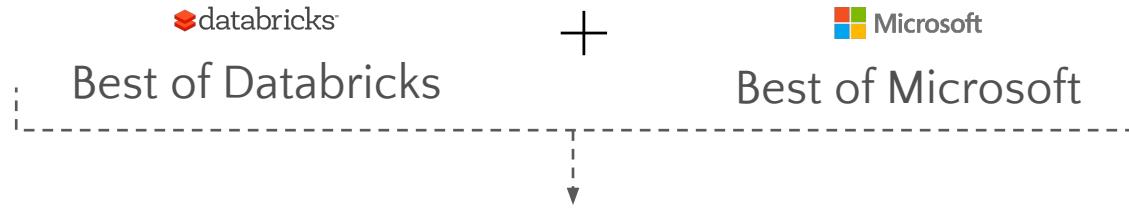
Azure Portal
One-Click setup
Unified Billing

Ashish Pal

databricks

What is Azure Databricks?

A fast, easy and collaborative Apache® Spark™ based analytics platform optimized for Azure



 Designed in collaboration with the founders of Apache Spark



One-click set up; streamlined workflows



Interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.



Native integration with Azure services (Power BI, SQL DW, Cosmos DB, Blob Storage)



Enterprise-grade Azure security (Active Directory integration, compliance, enterprise-grade SLAs)

Ashish Pal

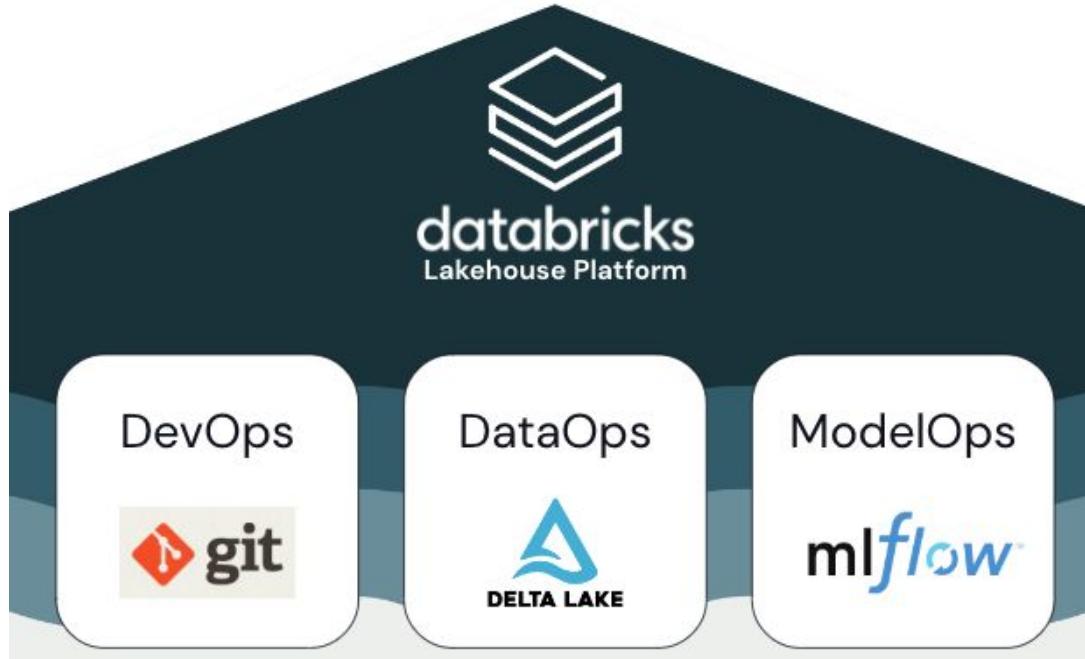
What is Azure Databricks?

Azure Databricks is a unified, open analytics platform for building, deploying, sharing, and maintaining enterprise-grade data, analytics, and AI solutions at scale. The Databricks Data Intelligence Platform integrates with cloud storage and security in your cloud account, and manages and deploys cloud infrastructure on your behalf.

The Azure Databricks workspace provides a unified interface and tools for most data tasks, including:

- Data processing workflows scheduling and management
- Generating dashboards and visualizations
- Managing security, governance, high availability, and disaster recovery
- Data discovery, annotation, and exploration
- Machine learning (ML) modeling, tracking, and model serving
- Generative AI solutions

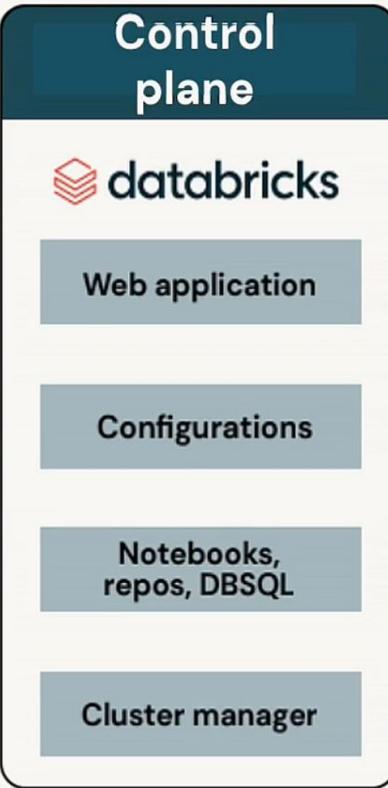
Azure Databricks MLOPS



Ashish Pal

Users

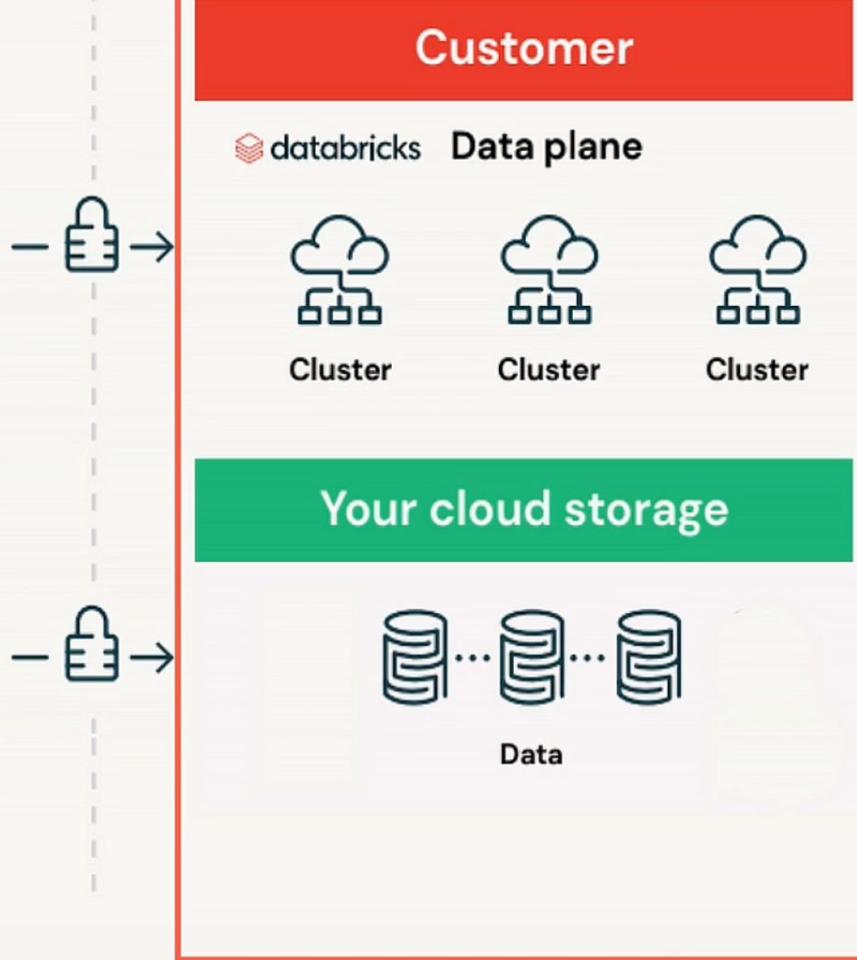
Interactive users



Data Engineers

Data Scientists

Data Analysts





What is Delta lake?

Ashish Pal

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





What is Unity Catalog?

Ashish Pal

Unity Catalog

Overview



Unified governance across clouds

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



Unified data and AI assets

Centrally share, audit, secure and manage all data types with one simple interface. 59

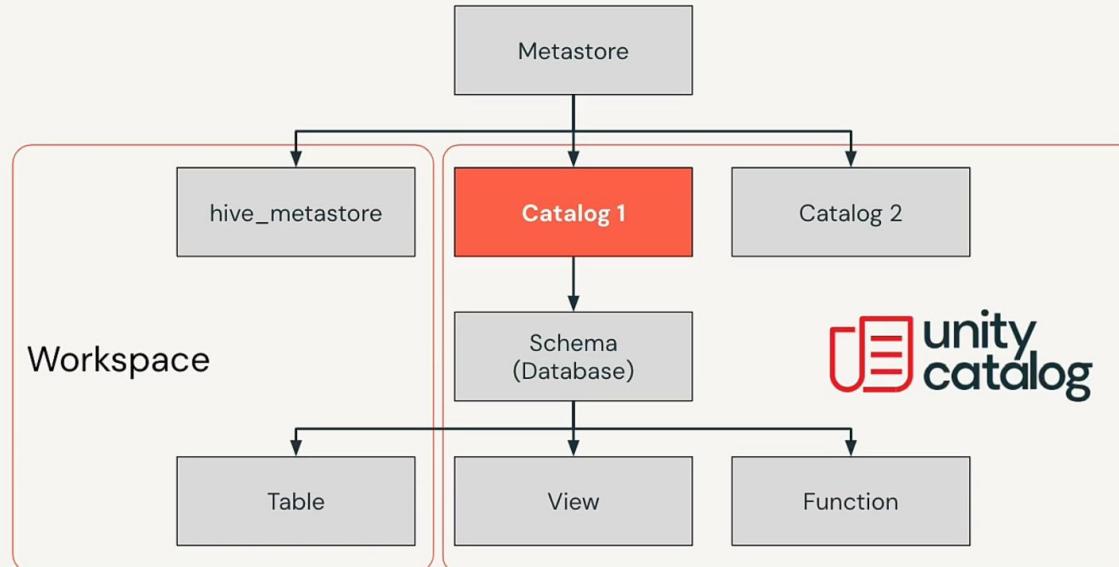


Unified existing catalogs

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

Metastore

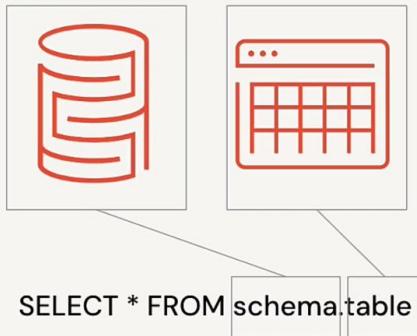
Accessing legacy Hive metastore



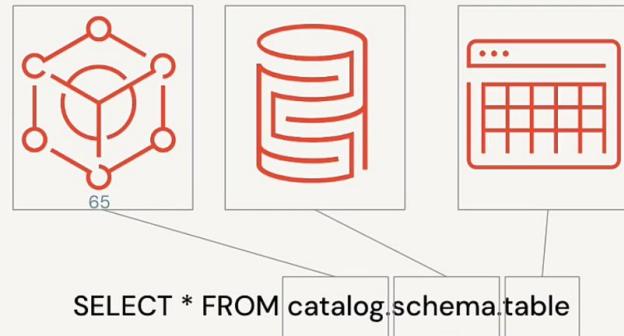
Catalog

Three-level namespace

Traditional SQL two-level
namespace



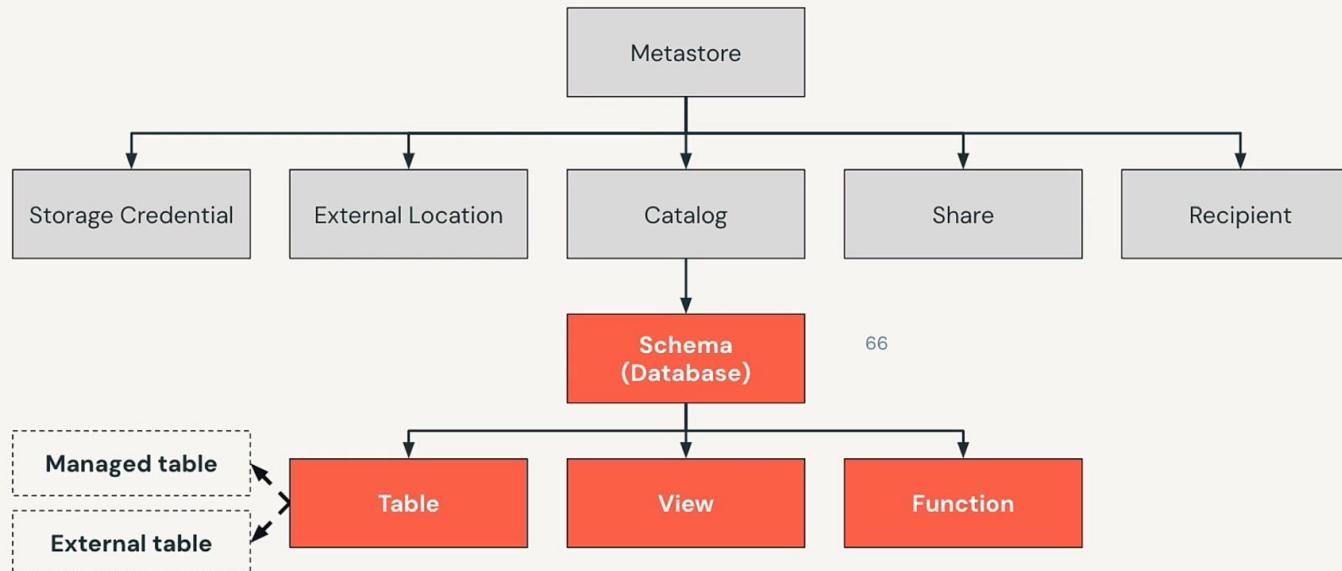
Unity Catalog three-level
namespace



Ashish Pal

Data Objects

Schema (database), tables, views, functions





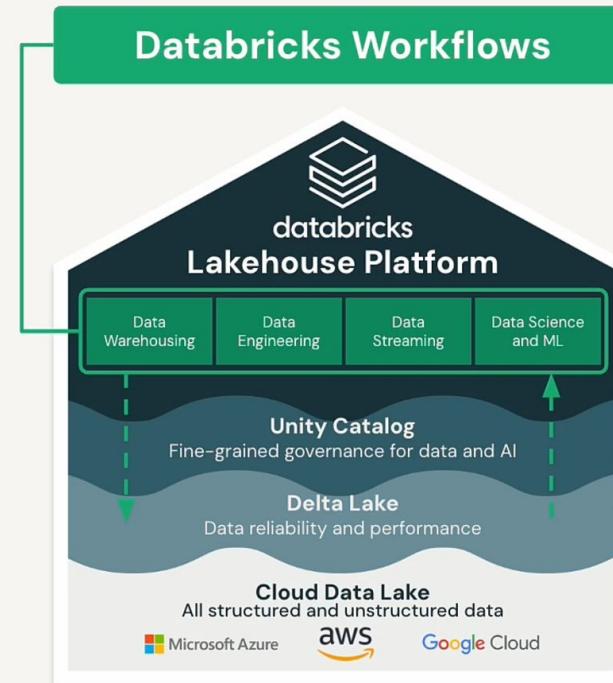
What is Workflow?

Ashish Pal

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)	80 Supported
Import Libraries	Not supported	Supported



Workflow Jobs

Use Cases

Orchestration of Dependent Jobs

Jobs running on schedule, containing dependent tasks/steps

Machine Learning Tasks

Run MLflow notebook task in a job

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

Jobs Workflows

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



Databricks Repos

Git Versioning

Native integration with
Github, Gitlab,
Bitbucket and Azure
Devops
UI-based workflows



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

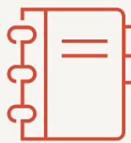
CI/CD Integration

Control Plane in Databricks

Manage customer accounts, datasets, and clusters



Databricks Web Application



Repos / Notebooks



Jobs



Cluster Management

Repos Service

Git and CI/CD Systems



Version

Review

Test

Demo



Ashish Pal



What is Cluster?

Ashish Pal

Clusters

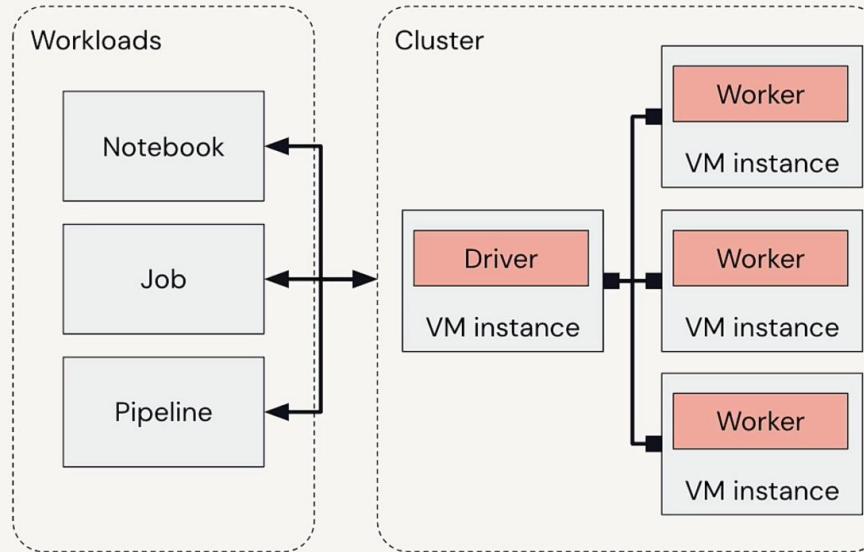
Overview

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 <i>without</i> 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





What is DB Notebooks?

Ashish Pal

Databricks Notebooks

Collaborative, reproducible, and enterprise ready

Multi-language

Use Python, SQL, Scala, and R, all in one Notebook

Collaborative

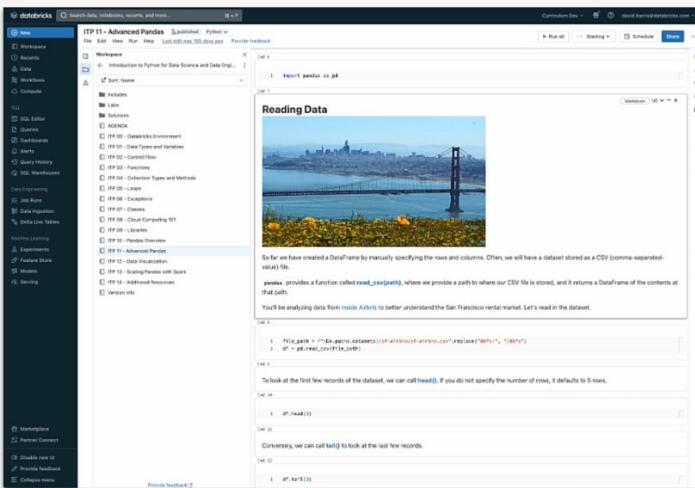
Real-time co-presence, co-editing, and commenting

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

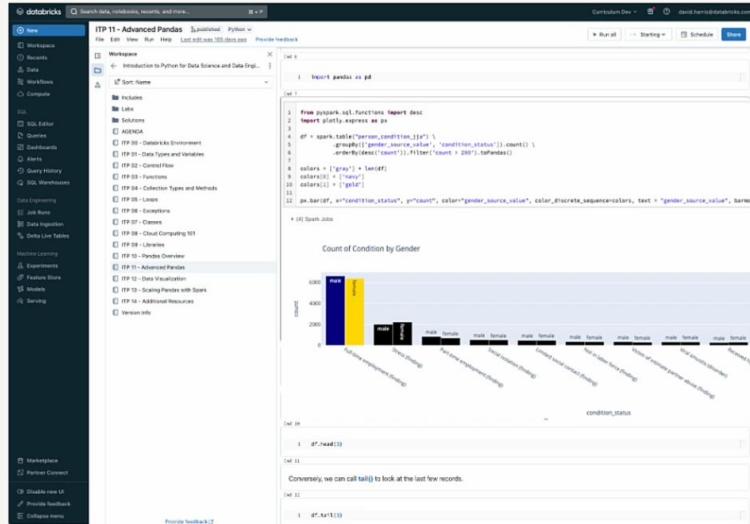
Enterprise-grade access controls, identity management, and auditability

Ashish Pal

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



The screenshot shows a Databricks Notebook titled "TPC-11 - Advanced Pandas". The notebook contains the following Python code:

```
1 import pandas as pd
2
3 from pyspark.sql.functions import desc
4 import plotly.express as px
5
6 df = spark.table("person.condition").select(
7     "gender_source_value", "condition_status"
8 ).groupby("gender_source_value", "condition_status").count()
9 .orderBy(desc("count")).filter("count > 200").toPandas()
10
11 colors = ["#FFFFE0", "#F0E68C", "#D9EAD3"]
12 colors[1] = "#F0E68C"
13 colors[2] = "#D9EAD3"
14
15 px.bar(df, x="condition_status", y="count", color="gender_source_value", color_discrete_sequence=colors, text = "gender_source_value", barmode="group")
```

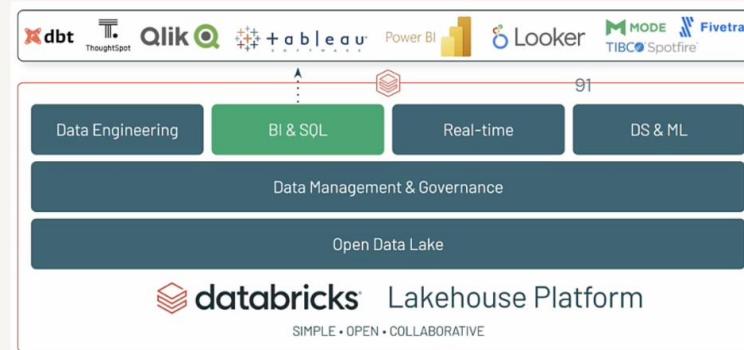
Below the code, there is a chart titled "Count of Condition by Gender" showing the count of conditions for each gender source value across different condition statuses. The chart has three bars per condition status, corresponding to the colors defined in the code.

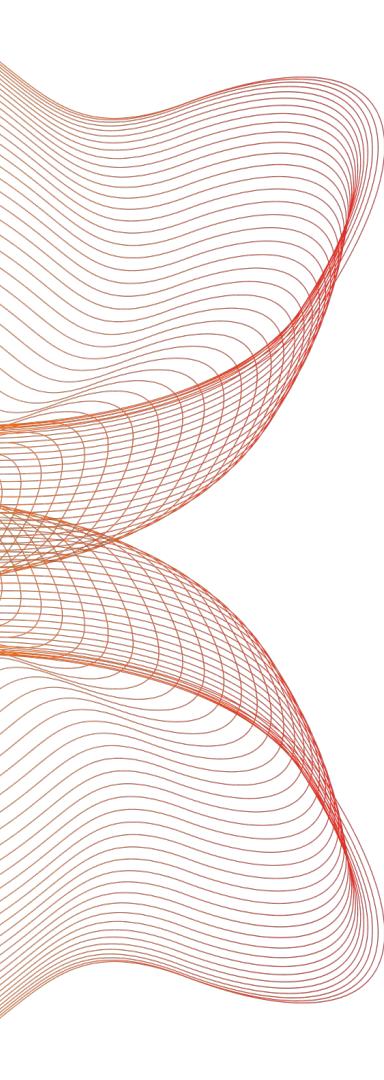


Databricks SQL

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



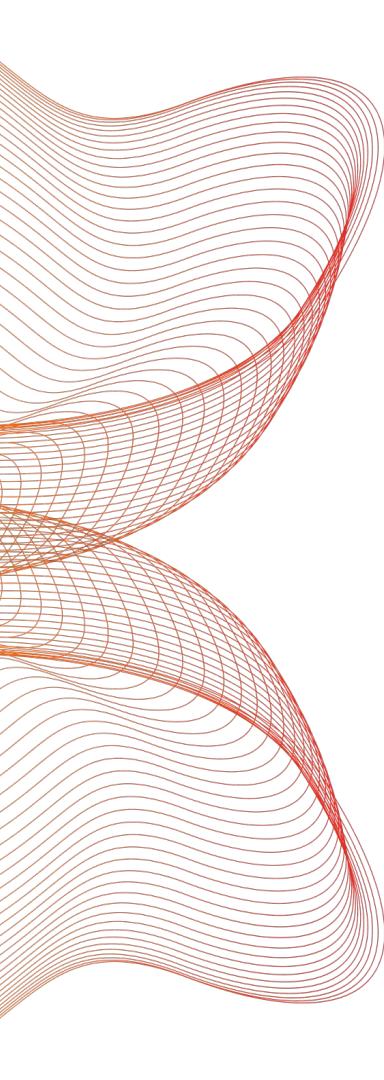


Use Cases

- Data exploration and visualization.
- Machine learning and AI.
- Predictive analytics and forecasting.
- Realtime monitoring and alerting.



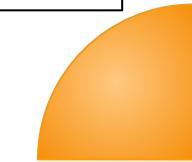
Ashish Pal



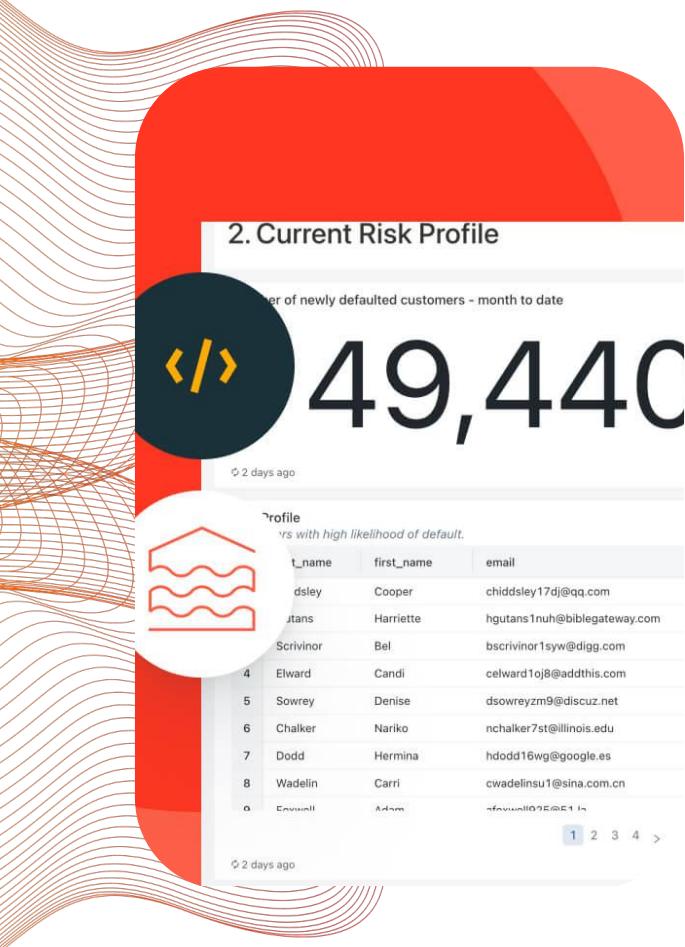
Case Study

<https://github.com/ashishpal2702/AzureDataBricksWorkshop>

- [Delta Lake creation](#)
- [ML Quick start training](#)
- [Mlflow end to end](#)



Ashish Pal



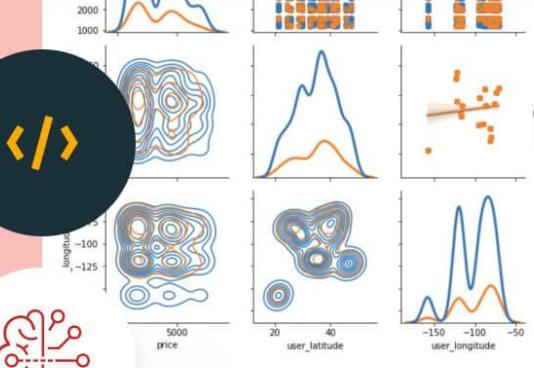
DataBricks Demo

```
pip install dbdemos
```

```
import dbdemos
```

```
dbdemos.list_demos()
```

Ashish Pal



Create Feature Store table

Our first step is to create our Feature table.

We will load data from the silver table `travel_purchase` and create features

In this first version, we'll transform the timestamp into multiple features that

In addition, we will drop the label from the table as we don't want it to leak or

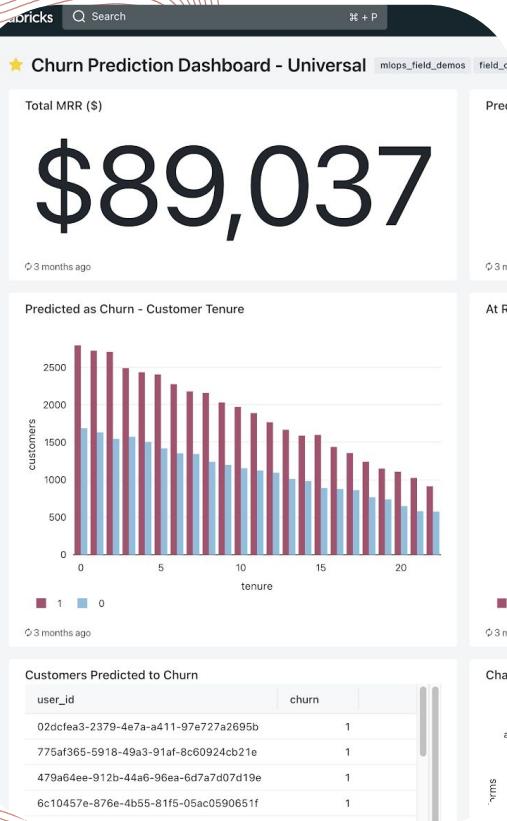
To create the feature table, we'll use the `FeatureStoreClient.create_table`

Under the hood, this will create a Delta Table to save our information.

Feature Store

```
dbdemos.install('feature-store')
```

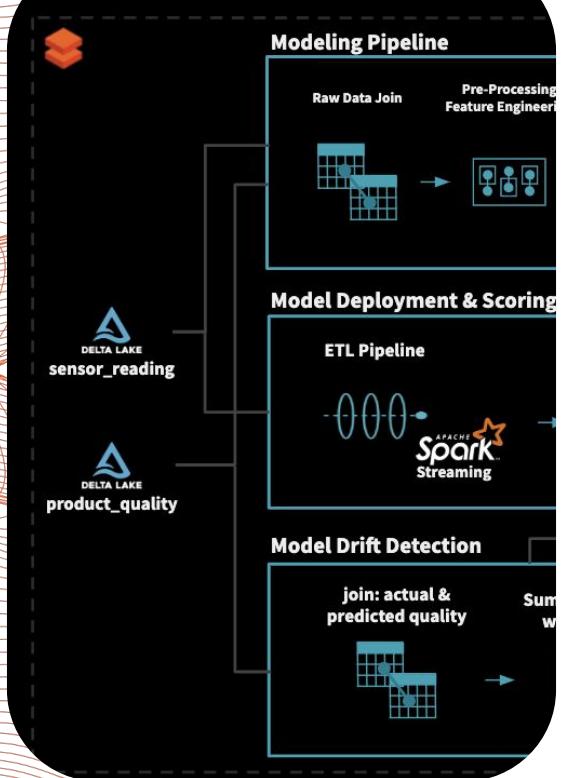
Ashish Pal



MLOps - End 2 end pipeline

```
dbdemos.install('mlops-end2end')
```

Ashish Pal



Productionizing Machine Learning: From Deployment to Drift Detection

[Blog](#)

Code Notebook -

https://joelcthomas.github.io/modeldrift/#a_glassware_quality_control.html

Ashish Pal



Thank you for your time 😊

Ashish Pal