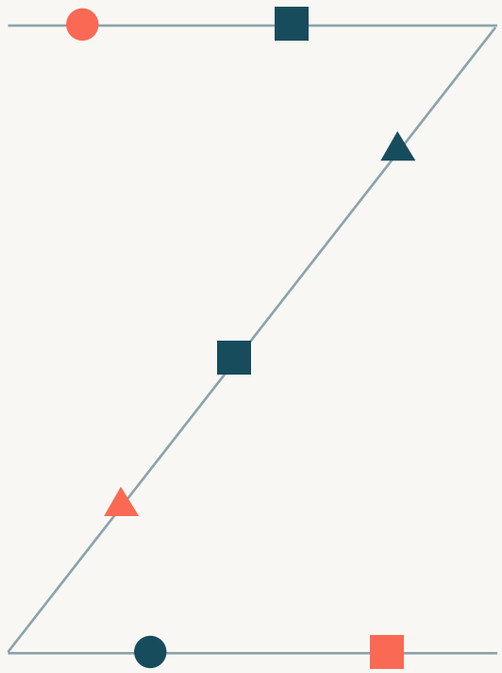
TECHNICAL GUIDE

Solving Common Data Challenges

Startups and Digital Native Businesses

databricks

Table of Contents



INTRODUCTION

This guide shares how the lakehouse architecture can increase productivity and cost-efficiently support all your data, analytics and AI workloads, and flexibly scale with the pace of growth for your company. Read the entire guide or dive straight into a specific challenge.

With the advent of cloud infrastructure, a new generation of startups has rapidly built and scaled their businesses. The use of cloud infrastructure, once seen as innovative, has now become table stakes. The differentiator for the fastest-moving startups and digital natives now comes from the effective use of data at scale, primarily analytics and AI. Digital natives — defined as fast-moving, lean, and technically savvy, born-in-the-cloud organizations — are beginning to focus on new data-driven use cases such as real-time machine learning and personalized customer experiences.

To pursue these new data-intensive use cases and initiatives, organizations must look beyond the technologies that delivered them to this point in time. Over time, these technologies, such as transactional databases, streaming/batch pipelines and first-generation analytics engines, have led to brittle

This guide examines some of the biggest data challenges and solutions for startups and for scaling digital native businesses that have reached the point where an end-to-end modern data platform is a smart investment. Some key considerations include: systems that are not cost-efficient and require time-consuming administration and engineering toil. In addition to growing maintenance needs, data is often stored in disparate locations and formats, with little or no governance, making real-time use cases, analytics and AI difficult or impossible.

1 **Consolidating on a unified data platform**

As mentioned above, siloed data storage and management add administrative and financial cost. You can benefit significantly when you unify your data in one location with a flexible architecture that scales with your needs and delivers performance for future success. For this, you will want an open platform that supports all your data including batch and streaming workloads, data analytics and machine learning. With data unification, you create a more efficient, integrated approach to ingesting, cleaning and organizing your data. You also need automation to make data analysis easier for the nontechnical users in the company. But broader data access also means more focus on security, privacy, compliance and access control, which can create overhead for a growing.

2 **Scaling up capacity and increasing performance and usability of the data solutions**

Data teams at growing digital native organizations find it time intensive and costly to handle the growing volume and velocity of their data being ingested from multiple sources, across multiple clouds. You now need a unified and simplified platform that can instantly scale up capacity and deliver more computing power on demand to free up your data teams to produce outputs more quickly. This lowers the total cost for the overall infrastructure by eliminating redundant licensing, infrastructure and administration costs.

3 **Building effective machine learning operations**

For data teams beginning their machine learning journeys, the challenge of training data models can increase in management complexity. Many teams with disparate coding needs for the entire model lifecycle suffer inefficiencies from transferring data and code across many separate services. To build and manage effective ML operations, consider an end-to-end MLOps environment that brings all data together in one place and incorporates managed services for experiment tracking, model training, feature development and feature and model serving.

01

CHALLENGE:

Create a unified data architecture for data quality, governance and efficiency

CHALLENGE 01

Create a unified data architecture for data quality, governance and efficiency

As cloud-born companies grow, data volumes rapidly increase, leading to new challenges and use cases. Among the challenges:

1. Application stacks optimized for transaction use cases aren't able to handle the volume, velocity and variety of data that modern data teams require. For example, this leads to query performance issues as data volume grows.

2. Data silos develop as each team within an organization chooses different ETL/ELT and storage solutions for their needs. As the organization grows and changes, these pipelines and storage solutions become brittle, hard to maintain and nearly impossible to integrate.

3. These data silos lead to discoverability, integration and access issues, which prevent teams from leveraging the full value of the organization's available data.

4. Data governance is hard. Disparate ETL/ELT and storage solutions lead to governance, compliance, auditability and access control challenges, which expose organizations to tremendous risk.

For all the reasons above, the most consistent advice from successful data practitioners is to create a "single source of truth" by unifying all data on a single platform. With the Databricks Lakehouse Platform, you can unify all your data on one platform, reducing data infrastructure costs and compute. You don't need excess data copies and you can retire expensive legacy infrastructure.

The Databricks Lakehouse Platform provides a unified set of tools for building, deploying, sharing and maintaining data solutions at scale. It integrates with cloud storage and the security in your cloud account, manages and deploys cloud infrastructure on your behalf. Your data practitioners no longer need separate storage systems for their data. And you don't have to rely on your cloud provider for security. The lakehouse has its own robust security built into the platform.

CUSTOMER STORY: GRAMMARLY

Helping 30 million people and 50,000 teams communicate more effectively

While its business is based on analytics, Grammarly for many years relied on a homegrown analytics platform to drive its AI writing assistant to help users improve multiple aspects of written communications. As teams developed their own requirements, data silos inevitably emerged as different business areas implemented analytics tools individually.

"Every team decided to solve their analytics needs in the best way they saw fit," said Chris Locklin, Engineering Manager, Data Platforms, at Grammarly. "That created challenges in consistency and knowing which data set was correct."

To better scale and improve data storage and query capabilities, Grammarly brought all its analytical data into the Databricks Lakehouse Platform and created a central hub for all data producers and consumers across the company.

Grammarly had several goals with the lakehouse, including better access control, security, ingestion flexibility, reducing costs and fueling collaboration. "Access control in a distributed file system is difficult, and it only gets more complicated as you ingest more data sources," said Locklin. To manage access control, enable end-to-end observability and monitor data quality, Grammarly relies on the data lineage capabilities within Unity Catalog. "Data lineage allows us to effectively monitor usage of our data and ensure it upholds the standards we set as a data platform team," said Locklin. "Lineage is the last crucial piece for access control."

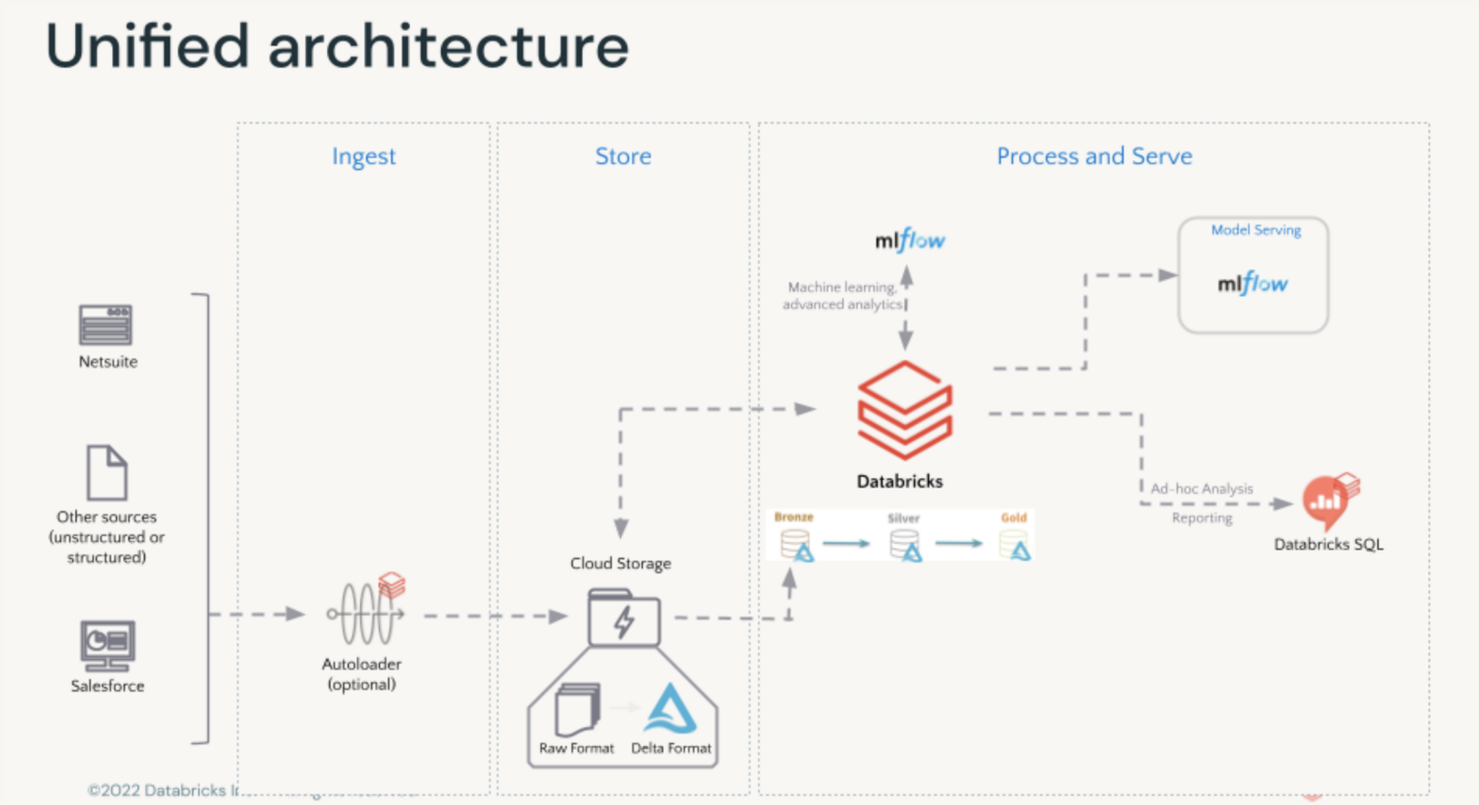
Data analysts within Grammarly now have a consolidated interface for analytics, which leads to a single source of truth and confidence in the accuracy and availability of all data managed by the data platform team. Having a consistent data source across the company also resulted in greater speed and efficiency and reduced costs. Data practitioners experienced 110% faster querying at 10% of the cost to ingest compared to a data warehouse. Grammarly can now make its 5 billion daily events available for analytics in under 15 minutes rather than 4 hours. Migrating off its rigid legacy infrastructure gave Grammarly the flexibility to do more and the confidence that the platform will evolve with its needs. Grammarly is now able to sustain a flexible, scalable and highly secure analytics platform that helps 30 million people and 50,000 teams worldwide write more effectively every day.

Read the full story here.

How to unify the data infrastructure with Databricks

The Databricks Lakehouse Platform architecture is composed of two primary parts:

- The infrastructure to deploy, configure and manage the platform and services
- the customer-owned infrastructure managed in collaboration by Databricks and the customer.



The lakehouse handles all varieties of data (structured, semi-structured, unstructured), as well as all velocities of data (streaming, batch or somewhere in the middle).

Sign up for a free trial account with the instructions on the get started page.

databricks

You can build a Databricks workspace by configuring secure integrations between the Databricks platform and your cloud account, and then Databricks deploys temporary Apache Spark™/Photon clusters using cloud resources in your account to process and store data in object storage and other integrated services you control. Here are three steps to get started with the Databricks Lakehouse Platform:

1 Understand the architecture

The lakehouse provides a unified architecture, meaning that all data is stored in the same accessible place. The diagram shows how data comes in from sources like a customer relationship management (CRM) system, an enterprise resource planning (ERP) system, websites or unstructured customer emails.

2 Optimize the storage layer

All data is stored in cloud storage while Databricks provides tooling to assist with ingestion, such as Auto Loader, and we recommend open-source Delta Lake as the storage format of choice. Delta optimized storage layer that provides the foundation for storing data and tables in the Databricks Lakehouse Platform. Having all your data in the same optimized, open storage keeps all your use cases in the same place, thus enabling collaboration and removing software tool overhead.

The Databricks Lakehouse organizes data stored with Delta Lake in cloud object storage with familiar concepts like database, tables and views. Delta Lake extends Parquet data files with a file-based transaction log for ACID transactions and scalable metadata handling. Delta Lake is fully compatible with Apache Spark APIs, and was developed for tight integration with Structured Streaming, allowing you to easily use a single copy of data for both batch and streaming operations to provide incremental processing at scale.This model combines many of the benefits of a data warehouse with the scalability and flexibility of a data lake.

To learn more about the optimized storage layer that provides the foundation for storing data and tables in the Databricks Lakehouse Platform, see Getting started with Delta Lake.

The first step in unifying your data architecture is setting up how data is to be accessed and used across the organization. We'll discuss this as a series of steps:

1. Set up governance with Unity Catalog
2. Grant secure access to the data
3. Capture audit logs
4. View data lineage
5. Set up data sharing

"Delta Lake provides us with a single source of truth for all of our data," said Stone. "Now our data engineers are able to build reliable data pipelines that thread the needle on key topics, such as inventory management, allowing us to identify in near real-time what our trends are so we can figure out how to effectively move inventory."

–Jake Stone, Senior Manager, Business Analytics at ButcherBox

Learn more

3

Configure unified governance

Databricks recommends using catalogs to provide an easily searchable inventory of data, notebooks, dashboards and models. Often this means that catalogs can correspond to software development environment scope, team or business unit. Unity Catalog manages how data is secured, accessed and shared. Unity Catalog offers a single place to administer data access policies that apply across all workspace and personas and automatically captures user-level audit logs that record access to your data.

Data stewards can securely grant access to a broad set of users to discover and analyze data at scale. These users can use a variety of languages and tools, including SQL and Python, to create derivative data sets, models and dashboards that can be shared across teams.

To set up Unity Catalog for your organization, you do the following:

1. Configure an S3 bucket and IAM role that Unity Catalog can use to store and access data in your AWS account.
2. Create a metastore for each region in which your organization operates, and attach workspaces to the metastore. Each workspace will have the same view of the data you manage in Unity Catalog.
3. If you have a new account, add users, groups and service principals to your Databricks account.
4. Next, create and grant access to catalogs, schemas and tables.

For complete setup instructions, see Get started using Unity Catalog.

How Unity Catalog works

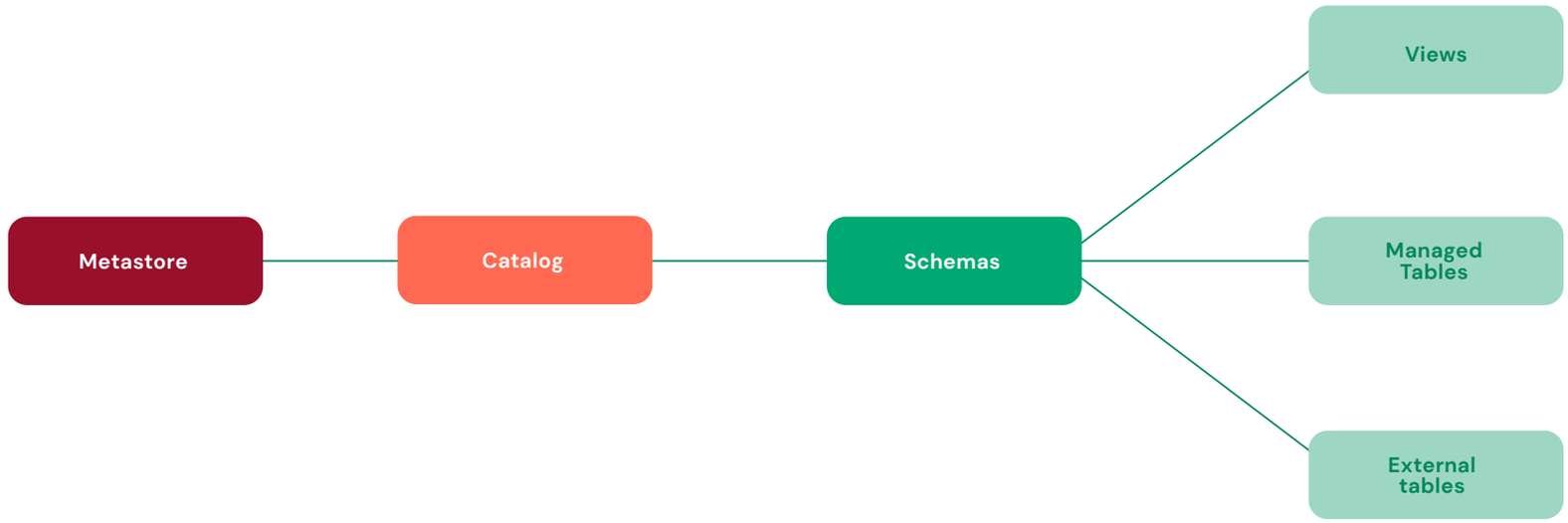
You will notice that the hierarchy of primary data objects in Unity Catalog flows from metastore to table:

Metastore is the top-level container for metadata. Each metastore exposes a three-level namespace (catalog.schema.table) that organizes your data.

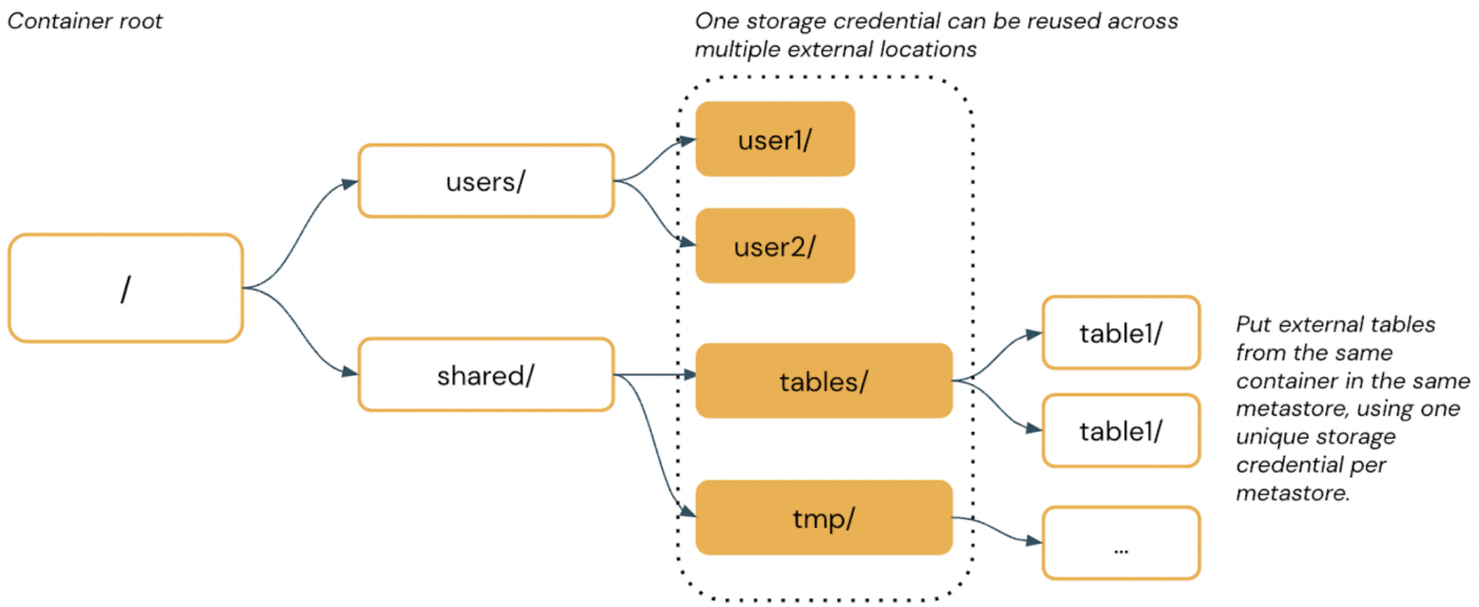
Catalog is the first layer of the object hierarchy, used to organize your data assets.

Schemas, also known as databases, are the second layer of the object hierarchy and contain tables and views.

Table is the lowest level in the object hierarchy, and tables can be external (stored in external locations in your cloud storage of choice) or managed (stored in a storage container in your cloud storage that you create expressly for Databricks). You can also create read-only **Views** from tables.



The diagram below represents the file system hierarchy of a single storage bucket:



Databricks Data Explorer is the main user interface for many Unity Catalog features. Use Data Explorer to view schema details, preview sample data, and see table details and properties. Administrators can view and change owners. Admins and data object owners can grant and revoke permissions through this interface.

Set up secure access

In Unity Catalog, data is secure by default. Initially, users have no access to data in a metastore. Access can be granted by either a metastore admin, the owner of an object, or the owner of the catalog or schema that contains the object. Securable objects in Unity Catalog are hierarchical and privileges are inherited downward.

Unity Catalog's security model is based on standard ANSI SQL and allows administrators to grant permissions in their existing data lake using familiar syntax, at the level of catalogs, databases (schema), tables and views. Privileges and metastores are shared across workspaces, allowing administrators to set secure permissions once against groups synced from identity providers and know that end users only have access to the proper data in any Databricks workspace they enter.

Unity Catalog uses the identities in the Databricks account to resolve users, service principals, and groups and to enforce permissions. To configure identities in the account, follow the instructions in Manage users, service principals, and groups. Refer to those users, service principals, and groups when you create access-control policies in Unity Catalog.

Unity Catalog users, service principals, and groups must also be added to workspaces to access Unity Catalog data in a notebook, a Databricks SQL query, Data Explorer or a REST API command. The assignment of users, service principals, and groups to workspaces is called identity federation. All workspaces attached to a Unity Catalog metastore are enabled for identity federation.

Securable objects in Unity Catalog are hierarchical, meaning that granting a privilege on a catalog or schema automatically grants the privilege to all current and future objects within the catalog or schema. For more on granting privileges, see the Inheritance model.
A common scenario is to set up a schema per team where only that team has USE SCHEMA and CREATE on the schema. This means that any tables produced by team members can only be shared within the team.
Data Explorer uses the privileges configured by Unity Catalog administrators to ensure that users are only able to see catalogs, databases, tables and views that they have permission to query.

CUSTOMER STORY: BUTCHERBOX

How Butcherbox Uses Data Insights to Provide Quality Food Tailored to Each Customer's Unique Taste

As a young e-commerce company, ButcherBox has to be nimble as its customers' needs change, which means it is constantly considering behavioral patterns, distribution center efficiency, a growing list of marketing and communication channels, and order processing systems.

The meat and seafood subscription company collects data on hundreds of thousands of subscribers. It deployed the Databricks Lakehouse Platform to gain visibility across its diverse range of data systems and enable its analytics team to securely view and export data in the formats needed.

With so much data feeding in from different sources — from email systems to its website — the data team at ButcherBox quickly discovered that data silos were a significant problem because they blocked complete visibility into critical insights needed to make strategic and marketing decisions.

"We knew we needed to migrate from our legacy data warehouse environment to a data analytics platform that would unify our data and make it easily accessible for quick analysis to improve supply chain operations, forecast demand and, most importantly, keep up with our growing customer base," explained Jake Stone, Senior Manager, Business Analytics, at ButcherBox.

The platform allows analysts to share builds and iterate on a project without getting into the code. Querying a table of 18 billion rows would have been problematic with a traditional platform. With Databricks, ButcherBox can do it in three minutes.

"Delta Lake provides us with a single source of truth for all of our data," said Stone. "Now our data engineers are able to build reliable data pipelines that thread the needle on key topics such as inventory management, allowing us to identify in near real-time what our trends are so we can figure out how to effectively move inventory."

Read the full story here.

Capture audit logs

Unity Catalog captures an audit log of actions performed against the metastore. To access audit logs for Unity Catalog events, you must enable and configure audit logs for your account. Audit logs for each workspace and account-level activities are delivered to your account. See how to configure audit logs and create a dashboard to analyze audit log data.

View data lineage

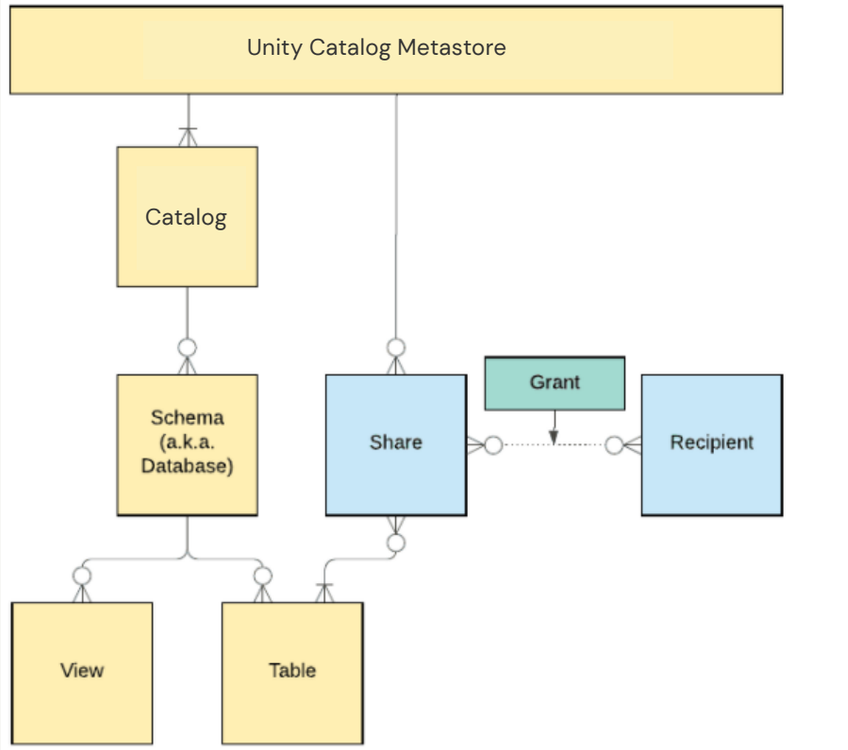
You can use Unity Catalog to capture runtime data lineage across queries in any language executed on a Databricks cluster or SQL warehouse. Lineage can be visualized in Data Explorer in near real-time and retrieved with the Databricks REST API. Lineage is aggregated across all workspaces attached to Unity Catalog and captured down to the column level, and includes notebooks, workflows and dashboards related to the query. To understand the requirements and how to capture lineage data, see Capture and view data lineage with Unity Catalog.

Set up secure data sharing

Databricks uses an open protocol called Delta Sharing to share data with other entities regardless of their computing platforms. Delta Sharing is integrated with Unity Catalog. Your data must be registered with Unity Catalog to manage, govern, audit and track usage of the shared data on the Lakehouse Platform. The primary concepts of Delta Sharing are shares (read-only collections of tables and table partitions to be shared) and recipients (objects that associate an organization with a credential or secure sharing identifier).

As a data provider, you generate a token and share it securely with the recipient. They use the token to authenticate and get read access to the tables you've included in the shares you've given them access to. Recipients access the shared data in read-only format. Whenever the data provider updates data tables in their own Databricks account, the updates appear in near real-time in the recipient's system.

Data providers can use Databricks audit logging to monitor the creation and modification of shares, and recipients can monitor recipient activity on shares. Data recipients who use shared data in a Databricks account can use Databricks audit logging to understand who is accessing which data.



Key Takeaways

- With the Databricks Lakehouse Platform, you can unify and simplify all your data on one platform to better scale and improve data storage and query capabilities
- The lakehouse helps reduce data infrastructure and compute costs. You don't need excess data copies and can retire expensive legacy infrastructure.
- Leverage Delta Lake as the open format storage layer to deliver reliability, security and performance on your data lake — for both streaming and batch operations — replacing data silos with a single home for structured, semi-structured and unstructured data
- With Unity Catalog you can centralize governance for all data and AI assets including files, tables, machine learning models and dashboards in your lakehouse on any cloud
- The Databricks Lakehouse Platform is open source with multicloud flexibility so that you can use your data however and wherever you want — no vendor lock-in

Resources:

- Databricks documentation
- Getting Started With Delta Lake
- Webinar: Deep Dive Into Lakehouse With Delta Lake
- Big Book of Data Engineering Use Cases
- 10 Powerful Features to Simplify Semi-structured Data Management in the Databricks Lakehouse

02

CHALLENGE:

Build your data architecture to support scale and performance

CHALLENGE 02

Build your data architecture to support scale and performance

As modern digital native companies mature, data volumes grow and new use cases develop. This inevitably leads to the increasing complexity of data architecture as new storage and access patterns emerge. Data growth can come suddenly and unexpectedly, when it does, the existing architecture needs to sustain performance, all the while being cost-effective. The relational databases and traditional data warehouses that met the needs of the businesses once upon a time are now creating limitations for new real-time use cases and large-scale data analytics pipelines.

Here are some common challenges around managing data and performance at scale:

1. **Volume and velocity** — Exponentially increasing data sources, and the speed at which they capture and create data.
2. **Latency requirements** — The demands of downstream applications and users have evolved (people want data and the results from the data faster).
3. **Data storage** — Storing data in the wrong format is slow to access, query and is expensive at scale.
4. **Data format** — Supporting structured, semi-structured and unstructured data formats is now a requirement. Most data storage solutions are designed to handle only one type of data, requiring multiple products to be stitched together.
5. **Governance** — Cataloging, auditing, securing and reporting on data is burdensome at scale when using old systems not built with data access controls and compliance in mind.
6. **Multicloud** is really hard.

Lakehouse solves scale and performance challenges

The solution for growing digital companies is a unified and simplified platform that can instantly scale up capacity to deliver more computing power on demand, freeing up teams to go after the much-needed data and produce outputs more quickly. With a lakehouse, they can replace their data silos with a single home for their structured, semi-structured and unstructured data. Users and applications throughout the enterprise environment can connect to the same single copy of the data to drive diverse workloads.

The lakehouse architecture is cost-efficient for scaling, lowering the total cost of ownership for the overall infrastructure by consolidating all data estate and use cases onto a single platform and eliminating redundant licensing, infrastructure and administration costs. Unlike other warehouse options that can only scale horizontally, the Databricks Lakehouse can scale horizontally and vertically based on workload demands.

With the Databricks Lakehouse, you can optimize the compute costs on a platform that is 2.7x faster and 12x more performant than Snowflake, according to research by the Barcelona Supercomputing Center. And your data teams are more productive by focusing on more strategic initiatives versus managing multiple data solutions.

CUSTOMER STORY: RIVIAN

Driving into the future of electric transportation

With more than 11,000 electric adventure vehicles (EAVs) on the road generating multiple terabytes of IoT data per day, Rivian is using data insights and machine learning to improve vehicle health and performance. However, with legacy cloud tooling, it struggled to scale pipelines cost-effectively and spent significant resources on maintenance.

Before Rivian even shipped its first EAV, it was already up against data visibility and tooling limitations that decreased output, prevented collaboration and increased operational costs. Rivian chose to modernize its data infrastructure on the Databricks Lakehouse Platform, giving it the ability to unify all its data into a common view for downstream analytics and machine learning. Now, unique data teams have a range of accessible tools to deliver actionable insights for different use cases, from predictive maintenance to smarter product development.

“Today we have various teams, both technical and business, using Databricks Lakehouse to explore our data, build performant data pipelines, and extract actionable business and product insights via visual dashboards,” said Wassym Bensaid, Vice President of Software Development at Rivian.

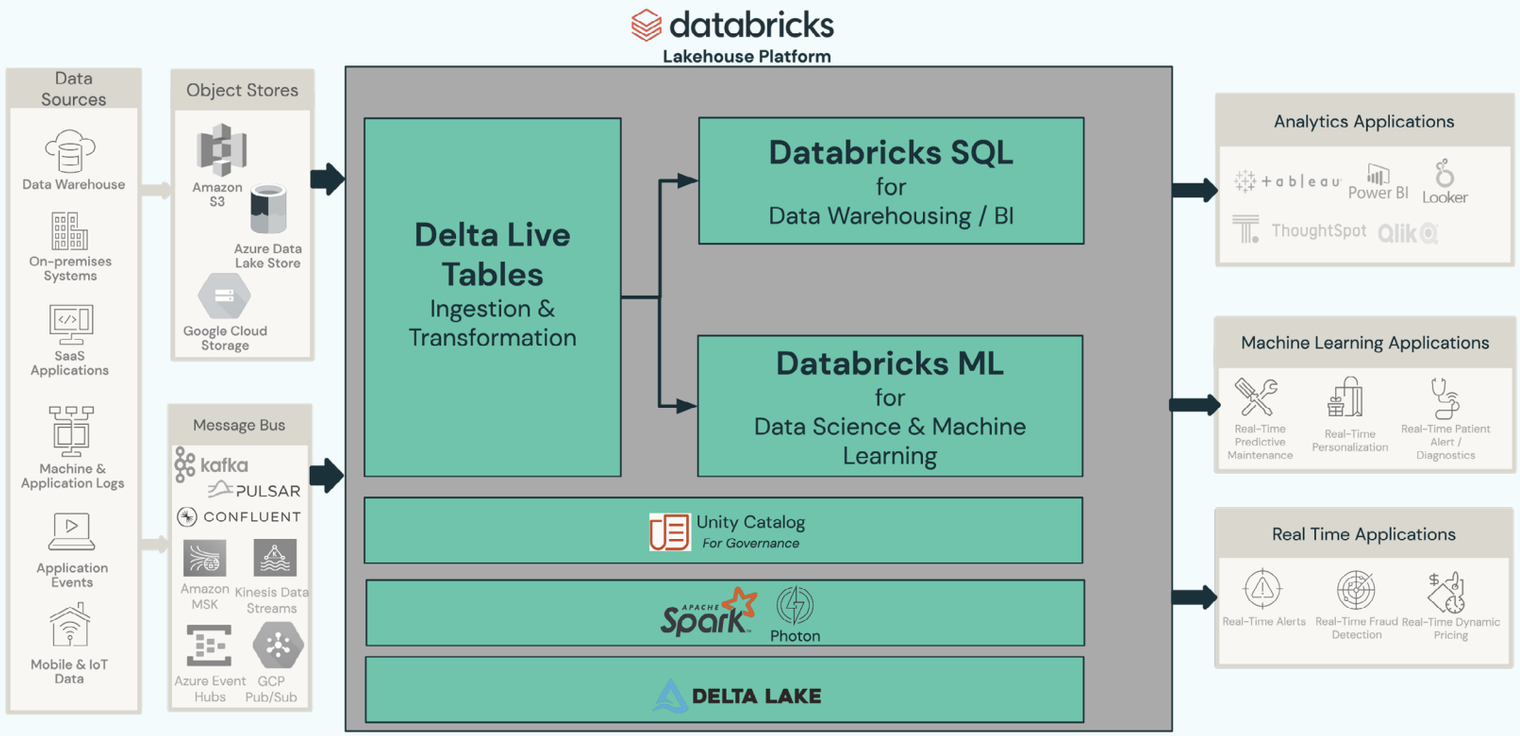
For instance, Rivian’s ADAS (advanced driver-assistance systems) Team can now easily prepare telemetric accelerometer data to understand all EAV motions. This core recording data includes information about pitch, roll, speed, suspension and airbag activity to help Rivian understand vehicle performance, driving patterns and connected car system predictability. Based on these key performance metrics, Rivian can improve the accuracy of smart features and the control that drivers have over them. By leveraging the Databricks Lakehouse Platform, Rivian has seen a 30%–50% increase in runtime performance, which has led to faster insights and model performance.

Read the full story here.

How to ensure scalability and performance with Databricks

The Databricks Lakehouse Platform is built for ensuring scalability and performance for your data architecture based on the following features and capabilities:

- A simplified and cost-efficient architecture that increases productivity
- A platform that ensures reliable, high performing ETL workloads — for streaming and batch data — while Databricks automatically manages your infrastructure
- The ability to ingest, transform and query all your data in one place, and scale on demand with serverless compute
- Enables real-time data access for all data, analytics and AI use cases



The following section will provide a short series of steps for understanding the key components of the Databricks Lakehouse Platform.

Step 1

Get a trial Databricks account

Start your 14-day free trial with Databricks on AWS in a few easy steps.
Get started with a free trial and setup. During the 14-day free trial, all Databricks usage is free, but Databricks uses compute and S3 storage resources in your cloud provider account.

Step 2

Understand the common Delta Lake operations

The Databricks Lakehouse Platform simplifies the entire data lifecycle, from data ingestion to monitoring and governance, and it starts with Delta Lake, a fully open-source storage system based on the Delta format providing reliability through ACID transactions and scalable metadata handling. Large quantities of raw files in blob storage can be converted to Delta to organize and store the data cheaply. This allows for flexibility of data movement while being performant and less expensive.

Get acquainted with the Delta Lake storage format and learn how to create, manage and query tables. With support for ACID transactions and schema enforcement, Delta Lake provides the reliability that traditional data lakes lack. This enables you to scale reliable data insights throughout the organization and run analytics and other data projects directly on your data lake — for up to 50x faster time-to-insight.

Delta Lake transactions use log files stored alongside data files to provide ACID guarantees at a table level. Because the data and log files backing Delta Lake tables live together in cloud object storage, reading and writing data can occur simultaneously without risk of many queries resulting in performance degradation or deadlock for business-critical workloads.

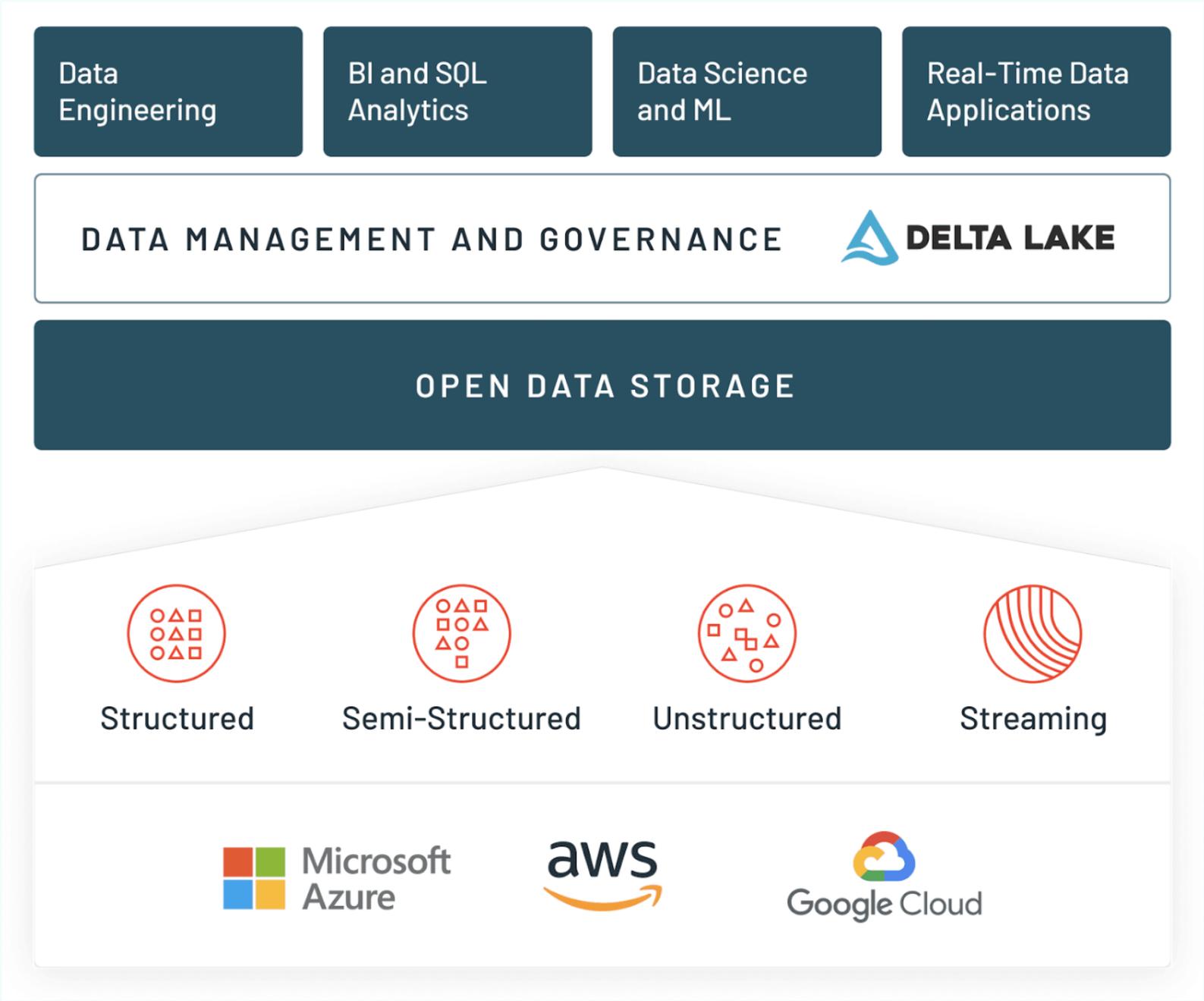
This means that users and applications throughout the enterprise environment can connect to the same single copy of the data to drive diverse workloads, with all viewers guaranteed to receive the most current version of the data at the time their query executes. With performance features like indexing, Delta Lake customers have seen ETL workloads execute up to 48x faster.

All data in Delta Lake is stored in open Apache Parquet format, allowing data to be read by any compatible reader. APIs are open and compatible with Apache Spark, so you have access to a vast open-source ecosystem to avoid data lock-in from proprietary formats and conversions, which have embedded and added costs.

"By leveraging Databricks and Delta Lake, we have already been able to democratize data at scale while lowering the cost of running production workloads by 60%, saving us millions of dollars."

— Steve Pulec, Chief Technology Officer, YipitData

Learn more



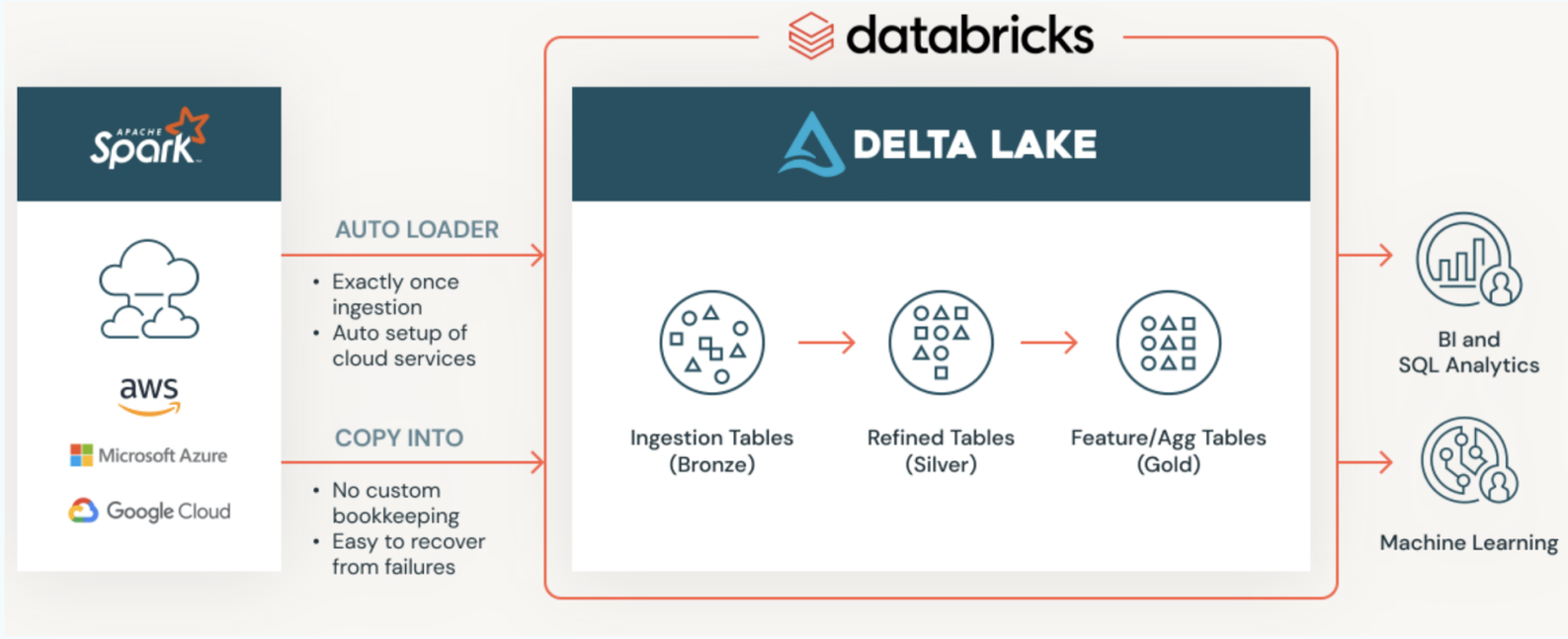
Step 3

Ingest data efficiently at scale

With a Lakehouse Platform, data teams can ingest data from hundreds of data sources for analytics, AI and streaming applications into one place.

Databricks recommends Auto Loader for incremental data ingestion. To ingest any file that can land in a data lake, Auto Loader incrementally and automatically processes new data files as they arrive in cloud storage in scheduled or continuous jobs. Auto Loader scales to support near real-time ingestion of millions of files per hour.

For pushing data in Delta Lake, the SQL command COPY INTO allows you to perform batch file ingestion into Delta Lake. COPY INTO is best used when the input directory contains thousands of files or fewer, and the user prefers SQL. COPY INTO can be used over JDBC to push data into Delta Lake at your convenience.



Step 4

Leverage production-ready tools to automate ETL pipelines

Once the raw data is ingested, Databricks provides a suite of production-ready tools that allow data professionals to quickly develop and deploy extract, transform and load (ETL) pipelines. Databricks SQL allows analysts to run SQL queries against the same tables used in production ETL workloads, allowing for real-time business intelligence at scale.

With your trial account, it's time to develop and deploy your first extract, transform and load (ETL) pipelines for data orchestration and learn how easy it is to create a cluster, create a Databricks notebook, configure Auto Loader for ingestion into Delta Lake, process and interact with the data, and schedule a job.

Databricks supports workloads in SQL, Python, Scala and R, allowing users with diverse skill sets and technical backgrounds to leverage their knowledge to derive analytic insights. You can use all languages supported by Databricks to define production jobs, and notebooks can leverage a combination of languages.

This means that you can promote queries written by SQL analysts for last-mile ETL into production data engineering code with almost no effort. Queries and workloads defined by personas across the organization leverage the same data sets, so there's no need to reconcile field names or make sure dashboards are up to date before sharing code and results with other teams.

With Delta Live Tables (DLT), data professionals have a framework that uses a simple declarative approach to build ETL and ML pipelines on batch or streaming data while automating operational complexities such as infrastructure management, task orchestration, error handling and recovery, retries, and performance optimization.

Delta Live Tables extends functionality in Apache Spark Structured Streaming and allows you to write just a few lines of declarative Python or SQL to deploy a production-quality data pipeline with:

- Autoscaling compute infrastructure for cost savings
- Data quality checks with expectations
- Automatic schema evolution handling
- Monitoring via metrics in the event log

With DLT, engineers can also treat their data as code and apply software engineering best practices like testing, monitoring and documentation to deploy reliable pipelines at scale. You can easily define end-to-end data pipelines in SQL or Python and automatically maintain all data dependencies across the pipeline and reuse ETL pipelines with environment-independent data management.

CUSTOMER STORY: ABNORMAL SECURITY

Stopping sophisticated ransomware in its tracks

The increase in email phishing and ransomware attacks requires the type of protection that can scale and evolve to meet the challenges of modern cyberattacks. Abnormal Security, a cloud-native email security provider, knew that scalability would become a major focus to stay ahead of attack strategies with frequent product updates.

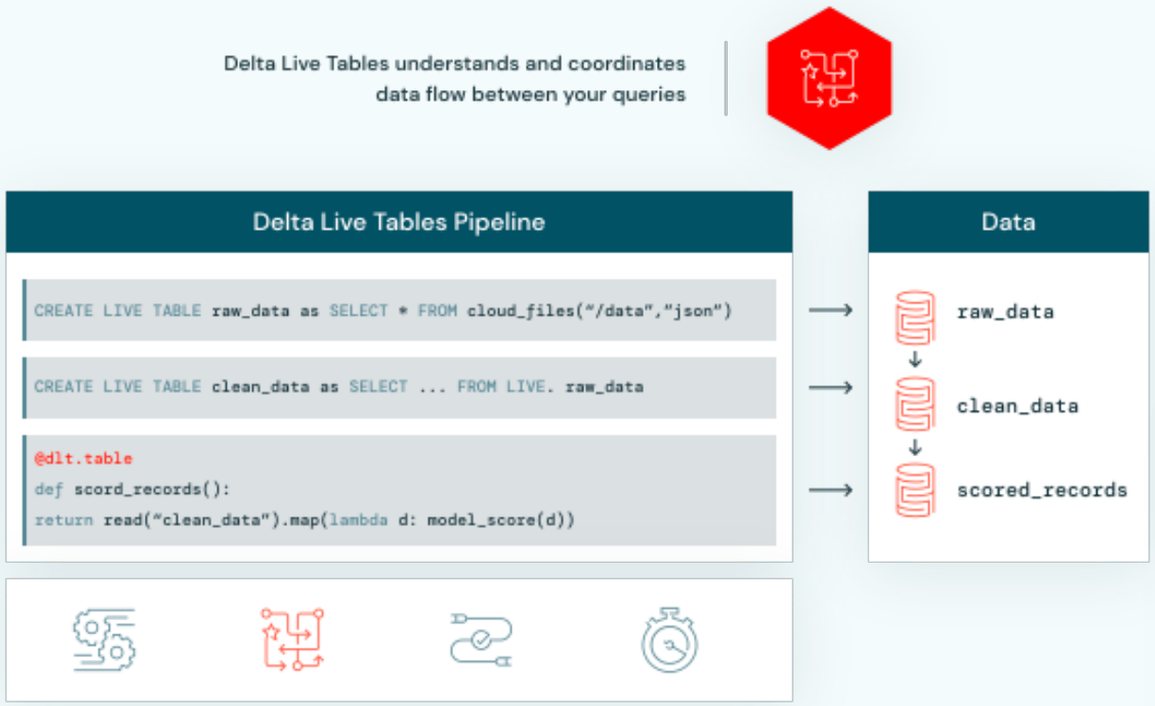
Abnormal also required a data analytics infrastructure robust enough to meet the scale requirements for its data pipelines and constantly refined ML models.

"We were spending too much time managing our Spark infrastructure," said Carlos Gasperi, Software Engineer at Abnormal Security. "What we needed to be doing with that time was building the pipelines that would make the product better."

The company implemented the Databricks Lakehouse Platform, which simplified its data architecture and maximized the performance of data pipelines and analytics. Data practitioners are now able to ingest data directly from S3 and query it in near real-time with the help of Delta Lake, an open-format storage layer that delivers reliability, security and performance on the data lake for both streaming and batch operations. With Databricks SQL, data scientists are then able to create visualizations using rich dashboards to drive product decisions and improve detection efficacy.

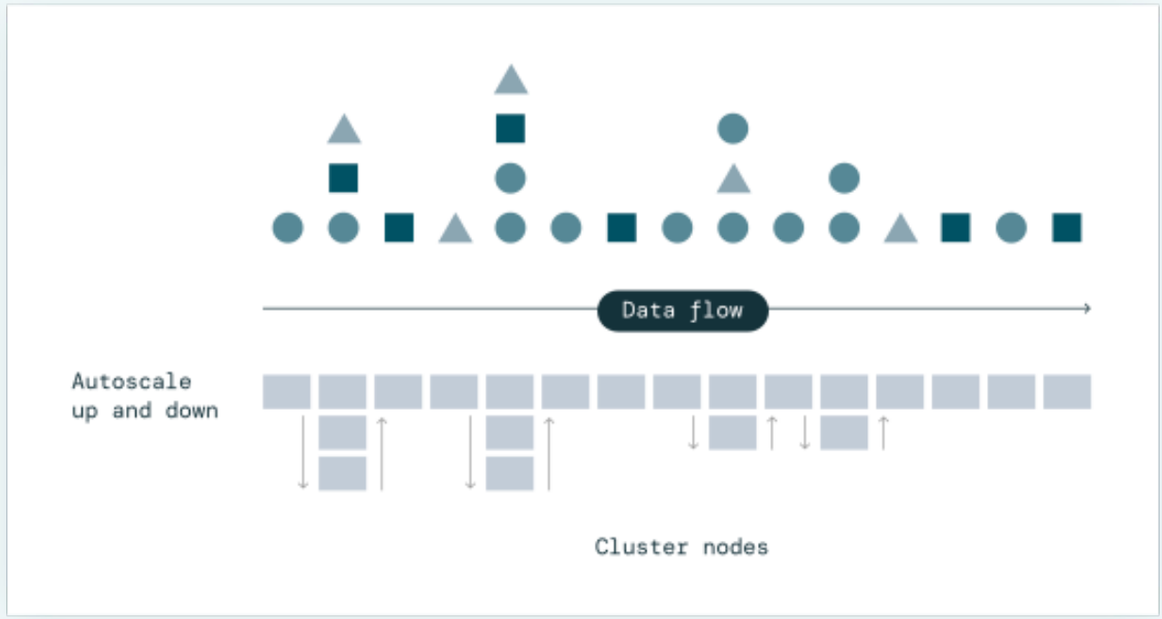
Databricks also provided the collaborative environment that Abnormal's data teams needed to increase their productivity and work in the same space without constantly competing for compute resources.

With Databricks, Abnormal has seen a 20% reduction in successful email attacks, a 40% reduction in infrastructure costs and a 30% increase in productivity. Read the full story here.



Delta Live Tables helps prevent bad data from flowing into tables through validation, integrity checks and predefined error policies. In addition, you can monitor data quality trends over time to get insight into how your data is evolving and where changes may be necessary.

Delta Live Tables Enhanced Autoscaling is designed to handle streaming workloads that trigger intermittently and are unpredictable. It optimizes cluster utilization by only scaling up to the necessary number of nodes while maintaining end-to-end SLAs, and gracefully shuts down nodes when utilization is low to avoid unnecessary idle node capacity.



Step 5

Use Databricks SQL for serverless compute

Databricks SQL (DB SQL) is a serverless data warehouse on the Lakehouse Platform for running your SQL and BI applications at scale with up to 12x better price/performance. It's imperative for younger, growing companies to reduce resource contention, and one way to accomplish that is with serverless compute. Running serverless removes the need to manage, configure or scale cloud infrastructure on the lakehouse, freeing up your data team for what they do best.

Databricks SQL warehouses provide instant, elastic SQL compute — decoupled from storage — and will automatically scale to provide unlimited concurrency without disruption, for high concurrency use cases. DB SQL has data governance and security built in. Handle high concurrency with fully managed load balancing and scaling of compute resources.

See for yourself in this tutorial on how to run and visualize a query in Databrick SQL and create dashboards on data stored in your data lake.

The Databricks SQL REST API supports services to manage queries and dashboards, query history and SQL warehouses.

Faster queries with Photon

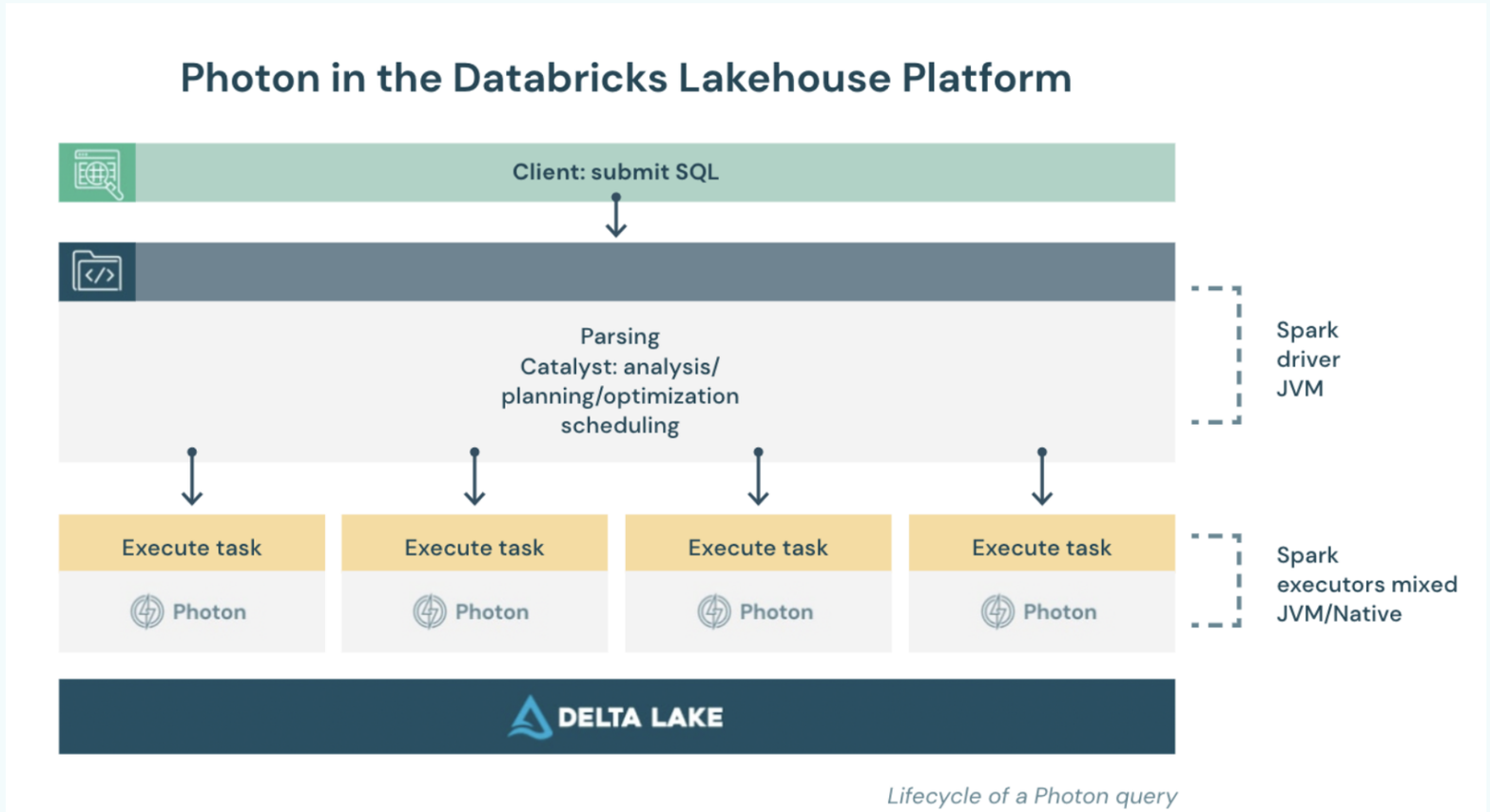
Photon is a new vectorized query engine designed to deliver dramatic infrastructure cost savings and accelerate all data and analytics workloads: data ingestion, ETL, streaming, interactive queries, data science and machine learning.

Photon is used by default in Databricks SQL. To enable Photon acceleration, select the **Use Photon Acceleration** checkbox when you create the cluster. If you create the cluster using the clusters API, set runtime_engine to PHOTON.

Photon supports a number of instance types on the driver and worker nodes. Photon instance types consume DBUs at a different rate than the same instance type running the non-Photon runtime. For more information about Photon instances and DBU consumption, see the Databricks pricing page.

Photon will seamlessly coordinate work and resources and transparently accelerate portions of your SQL and Spark queries. No tuning or user intervention required. Photon is compatible with Apache Spark APIs, so getting started is as easy as turning it on — no code change and no lock- in. Written entirely in C++, Photon provides an additional 2x speedup over Apache Spark per the TPC-DS 1TB benchmark, and customers have observed 3x–8x speedups on average.

With Photon, typical customers are seeing up to 80% TCO savings over traditional Databricks Runtime (Apache Spark) and up to 85% reduction in VM compute hours.



Lifecycle of a Photon query

Learn how to connect BI tools to Databricks SQL compute resources with the following user guides:

- Queries
- Visualizations
- Dashboards
- Alerts
- Favorites and tags
- Workspace browser

Step 6

Orchestrate workflows

Databricks provides a comprehensive suite of tools and integrations to support your data processing workflows.

Databricks Workflows removes operational overhead by offering fully managed orchestration service for all your teams, so you can focus on your workflows, not on managing your infrastructure. Orchestrate diverse workloads for the full lifecycle including Delta Live Tables, Jobs for SQL, Spark, notebooks, dbt, ML models and more.

Here's a tutorial on how to create your first workflow with a Databricks job. You will learn how to create notebooks, create and run a job, view the run details, and run jobs with different parameters.

“Databricks Workflows allows our analysts to easily create, run, monitor and repair data pipelines without managing any infrastructure. This enables them to have full autonomy in designing and improving ETL processes that produce must-have insights for our clients. We are excited to move our Airflow pipelines over to Databricks Workflows.”

—Anup Segu, Senior Software Engineer, YipitData

Learn more.

Step 7

Run an end-to-end analytics pipeline

This where you can see how everything works together to run efficiently at scale. First take the quickstart: Running end-to-end lakehouse analytics pipelines, where you will write to and read data from an external location managed by Unity Catalog and configure Auto Loader to ingest data to Unity Catalog.

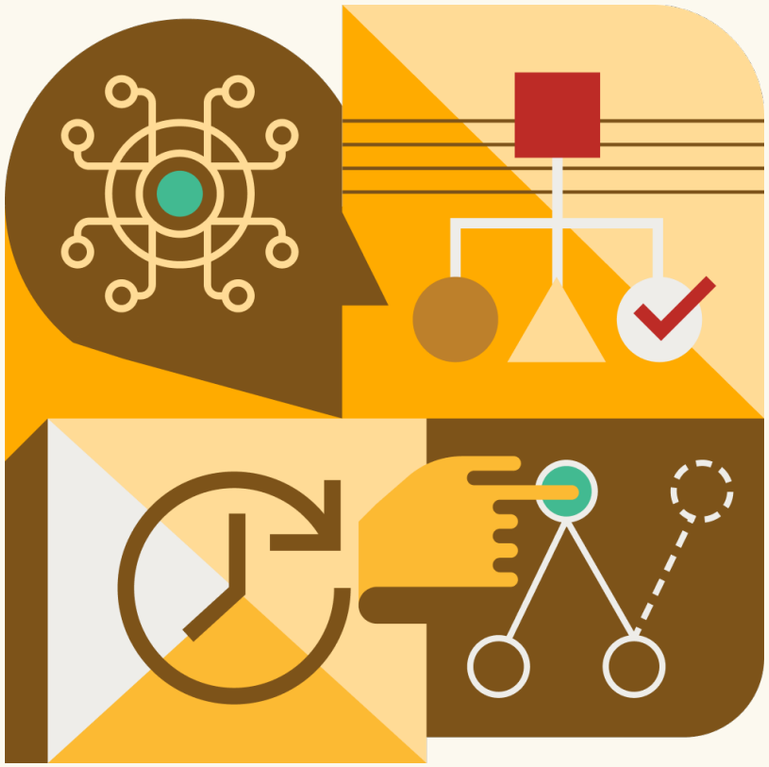
Resources:

- Databricks Lakehouse free trial
- The Lakehouse for companies born in the cloud
- How DuPont achieved 11x latency reduction and 4x cost reduction with Photon
- Apache Spark on Databricks
- Discover Lakehouse solutions
- Databricks documentation

03

CHALLENGE:

Building effective machine-learning operations



CHALLENGE 03

Building effective machine-learning operations

Growing startups and digital native companies face several challenges when they start building, maintaining and scaling machine learning operations (MLOps) for their data science teams.

1. Data teams often perform development in disjointed, siloed stacks spanning DataOps, ModelOps and DevOps

2. Development and training environment disconnect. Moving code and data between personal development environments and machine learning platforms for model training at scale is error prone and cumbersome. The “it worked on my machine” problem.

3. Gathering high-quality data. Data that is siloed across the organization is hard to discover, collect, clean and use. This leads to stale data and delays in development of models.

See **Create a unified data architecture.**

4. MLOps is different from DevOps. DevOps practices and tooling alone are insufficient because ML applications rely on an assortment of artifacts (e.g., models, data, code) that can each require different methods of experiment tracking, model training, feature development, governance, feature and model serving.

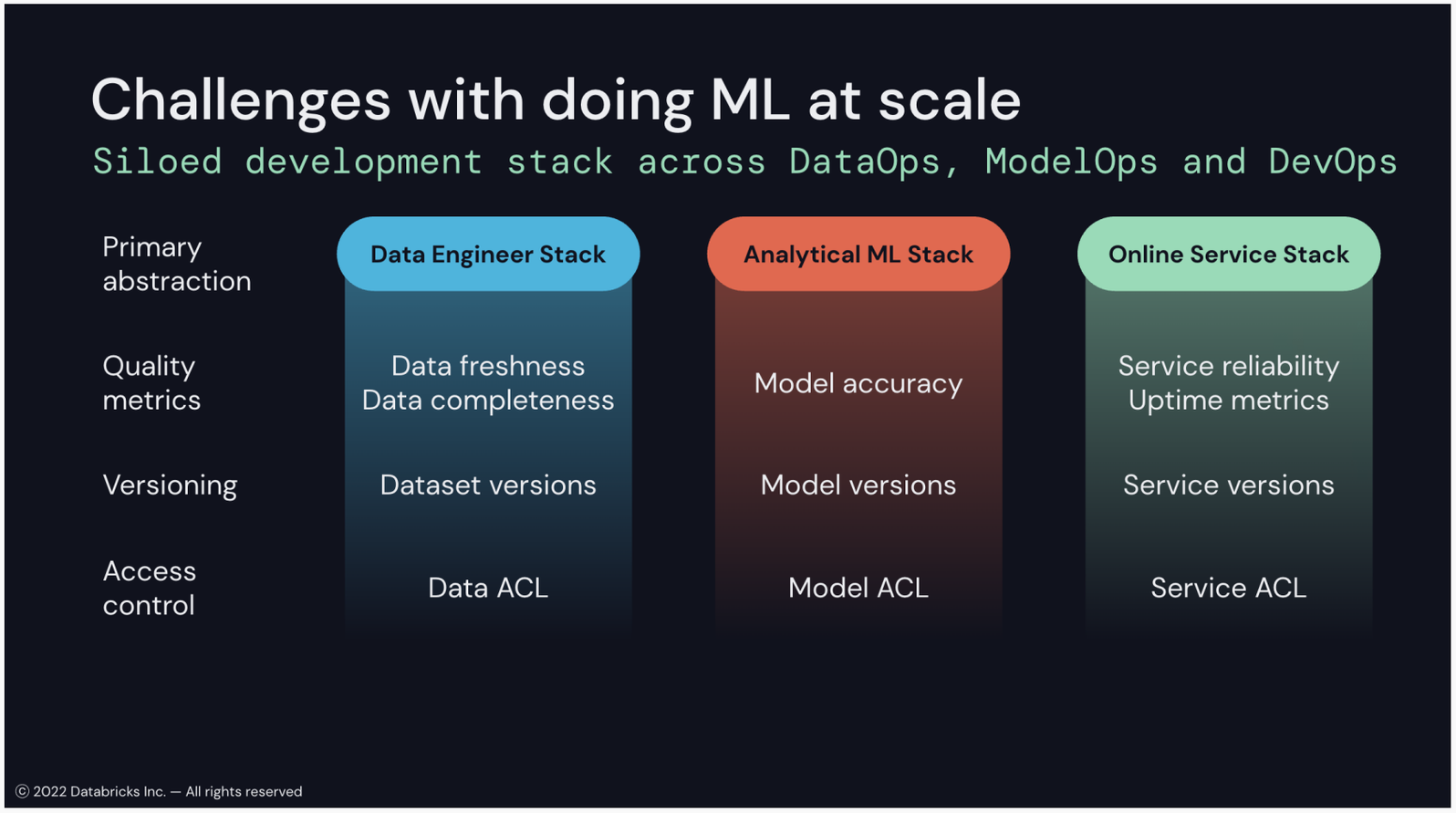
5. For data teams beginning their machine learning journeys, the challenge of training data models can be labor-intensive and not cost-effective because the data has to be converted into features and trained on a separate machine learning platform

Siloed stacks spanning DataOps, ModelOps and DevOps

When data engineers help ingest, refine and prep data, they do so on their own stack. This data has to be converted into features and then trained on a separate machine learning platform. This cross-platform handoff often results in data staleness, difficulty in maintaining versions, and eventually, poorly performing models. Even after you have trained your model, you have to deal with yet another tech stack for model deployment. It's challenging to serve features in real time and difficult to trace problems in production back to the data.

The downstream business impact is massive — longer and more expensive projects, and lower model accuracy in production leading to declining business metrics.

If you are looking at launching or scaling your MLOps, you should probably focus on an incremental strategy. At Databricks, we see firsthand how customers develop their MLOps approaches across a huge variety of teams and businesses. Check out this Data +AI Summit session to learn more about building robust MLOps practices.



Databricks solution:

Databricks Machine Learning is an integrated end-to-end machine learning environment incorporating managed services for experiment tracking, model training, feature development and management, and model serving. The capabilities of Databricks map directly to the steps of model development and deployment. With Databricks Machine Learning, you can:

- Train models either manually or with AutoML
- Track training parameters and models using experiments with MLflow tracking
- Create feature tables and access them for model training and inference
- Share, manage and serve models using MLflow Model Registry
- Deploy models for Serverless Real-time Inference

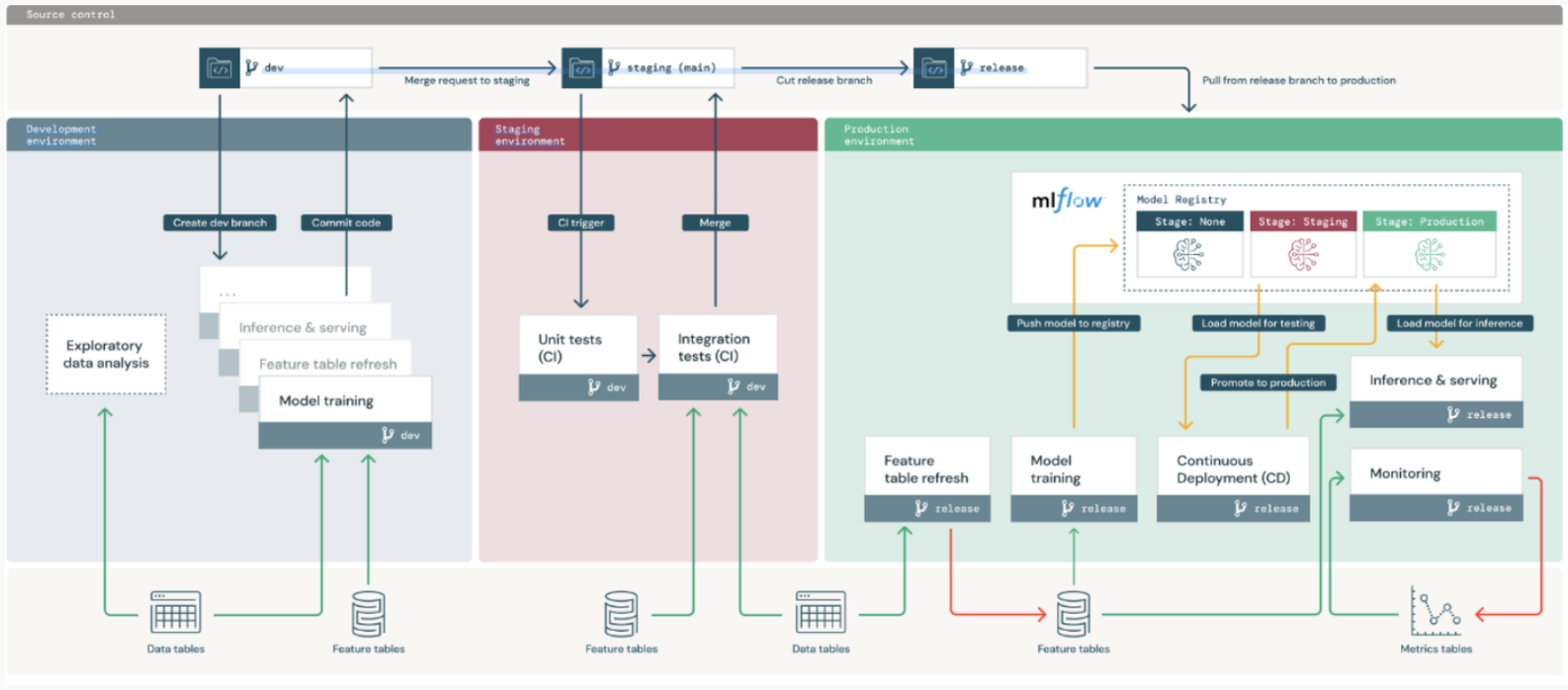
Use MLOps on the Databricks Lakehouse Platform

To gain efficiencies and reduce costs, many smaller digital companies are employing machine learning operations. MLOps is a set of processes and automation for managing models, data and code, and unique library dependencies to improve performance stability and long-term efficiency in ML systems.

To describe it simply, MLOps = ModelOps + DataOps + DevOps. The aim of MLOps is to improve the long-term performance, stability and success rate of ML systems while maximizing the efficiency of the teams who build them.

Not only does MLOps improve organizational efficiency, it also allows the models to iterate faster and react to real-life changes in the data. This ability separates companies that can grow to meet their customer's challenges in a reactive manner versus those that will spend significant time on data updates/processes and miss the opportunity to do something with their models.

The absence of MLOps is typically marked by an overabundance of manual processes which are slower



and more prone to error, affecting the quality of models, data and code. Eventually they form a bottleneck, capping the ability for a data team to take on new projects. The process is complex. In larger organizations, several specialists and stakeholders can be involved in one ML project. But data practitioners at smaller digital natives and high-growth startups may be forced to wear several hats.

And once an ML project goes into production, the MLOps continues, since the models, data and code change over time due to regulatory and business requirements. But the ML system must be resilient and flexible. Addressing these challenges with a defined MLOps strategy can dramatically reduce the iteration cycle of delivering models to production.

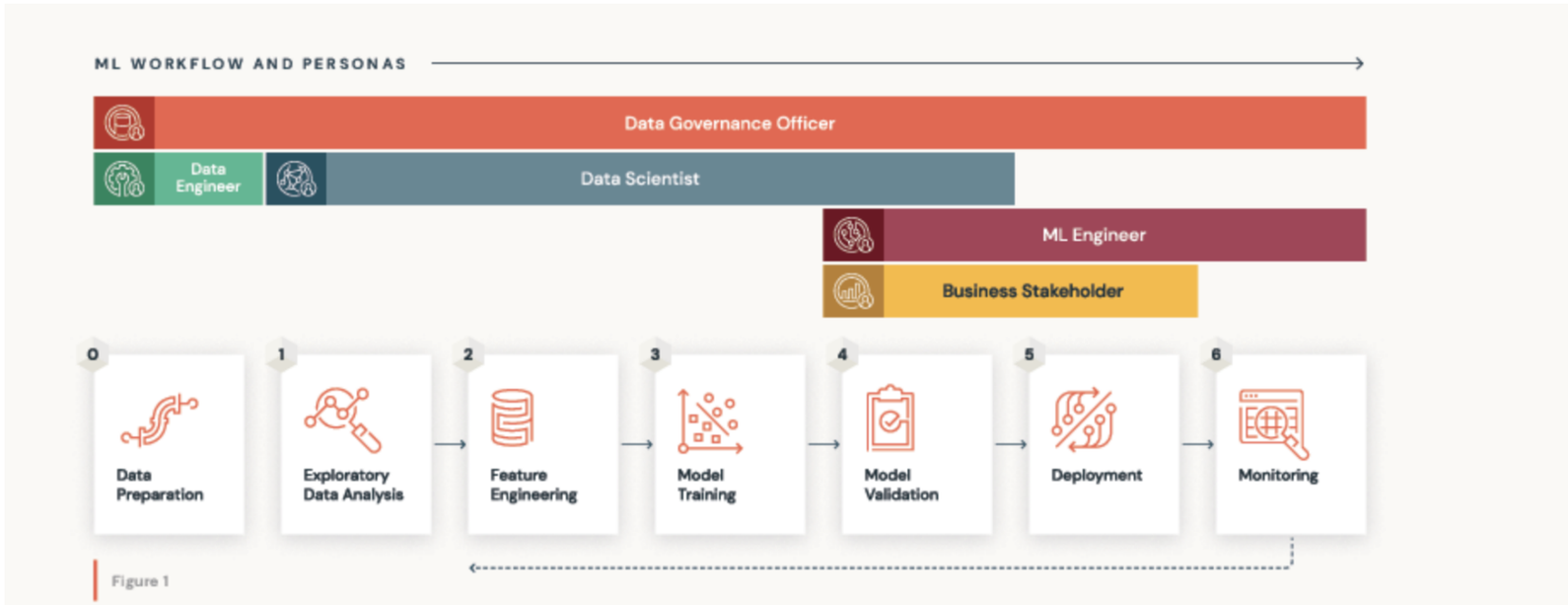


Figure 1

databricks

Steps in machine learning model development and deployment:

Step 1

Data preparation

Manually preparing and labeling data is a thankless, time-consuming job. With Databricks, teams can label data with human effort, machine learning models in Databricks, or a combination of both. Teams can also employ a model-assisted labeling workflow that allows humans to easily inspect and correct a model's predicted labels. This process can drastically reduce the amount of unstructured data you need to achieve strong model performance.

The Databricks Runtime for Machine Learning is a ready-to-go environment with many external libraries, including TensorFlow, PyTorch, Horovod, scikit-learn and XGBoost. It provides extensions to improve performance, including GPU acceleration in XGBoost, distributed deep learning using HorovodRunner, and model checkpointing.

To use Databricks Runtime ML, select the ML version of the runtime when you create your cluster. To access data in Unity Catalog for machine learning workflows, you must use a single user cluster. User isolation clusters are not compatible with Databricks Runtime for Machine Learning.

Machine learning applications often need to use shared storage for data loading and model checkpointing. You can load tabular data from tables or files. A table is a collection of structured data stored as a directory on cloud object storage.

For data preprocessing, you can use Databricks Feature Store to create new features, explore and reuse existing features, track lineage and feature creation code, and publish features to low-latency online stores for real-time inference. The Feature Store is a centralized repository that enables data scientists to find and share features. It ensures that the same code used to compute the feature values is used for model training and inference. The Feature Store library is available only on Databricks Runtime for Machine Learning and is accessible through Databricks notebooks and workflows.

Resources:

- The Comprehensive Guide to Feature Stores
- Load data for machine learning and deep learning
- Preprocess data for machine learning and deep learning

CUSTOMER STORY: ZIPLINE

Data-driven drones deliver lifesaving medical aid around the world

Automated logistics and delivery system provider Zipline is redefining logistics by using cutting-edge drone technology and a global autonomous logistics network to save lives by giving remote communities access to emergency and preparatory medical aid and resources, regardless of where they are in the world.

Doing so requires the ability to ingest and analyze huge chunks of time series data in real time. This data is produced every time a drone takes flight and includes performance data, in-flight battery management, regional weather patterns, geographic obstacles, landing errors and a litany of other information that must be processed.

Every Zipline flight generates a gigabyte of data with potential life-or-death consequences, but accessing and federating the data for both internal and external decision-making was challenging. With Databricks as the common platform, Zipline's data team can access all the information they need to accurately measure success, find the metrics that relate to customer experiences or logistics, and improve on them exponentially as more data is ingested and machine learning models are refined.

"About 30% of the deliveries we do are lifesaving emergency deliveries, where the product being delivered does not exist at the hospital. We have to be fast, and we have to be able to rely on all the different kinds of data to predict failures before they occur so that we can guarantee a really, really high service level to the people who are literally depending on us with their lives," said Zipline CEO Keller Rinaudo. "Databricks gives us confidence in our operations, and enables us to continuously improve our technology, expand our impact, and provide lifesaving aid where and when it's needed, every single day."

Read full story here.

Step 2

Model training

For training machine learning and deep learning models, you can use AutoML, which automatically prepares a data set for model training, performs a set of trials using open-source libraries such as scikit-learn and XGBoost, and creates a Python notebook with the source code for each trial run so you can review, reproduce and modify the code.

In Databricks, notebooks are the primary tool for creating data science and machine learning workflows and collaborating with colleagues. Databricks notebooks provide real-time coauthoring in multiple languages, automatic versioning and built-in data visualizations.

Resources:

- Model training examples
- Training models with Feature Store
- Best practices for deep learning on Databricks
- Machine learning quickstart notebook

Step 3

Track model development

The model development process is iterative, and can be challenging. You can use MLflow tracking to help you keep track of the model development process, including parameter settings or combinations you have tried and how they affected the model's performance.

MLflow tracking uses experiments and runs to log and track your model development. A run is a single execution of model code. An experiment is a collection of related runs. Within an experiment, you can compare and filter runs to understand how your model performs and how its performance depends on the parameter settings, input data, etc.

MLflow can automatically log training code written in many ML frameworks. This is the easiest way to get started using MLflow tracking. With MLflow's autologging capabilities, a single line of code automatically logs the resulting model.

A hosted version of MLflow Model Registry can help manage the full lifecycle of MLflow models. You can apply webhooks to automatically trigger actions based on registry events. For example, you can trigger CI builds when a new model version is created or notify your team members through Slack each time a model transition to production is requested. This promotes a traceable version control work process. You can leverage this feature for web traffic A/B testing and funneled to versions of deployed models for more precise population studies.

Step 4

Deploy machine learning models

You can use MLflow to deploy models for batch or streaming inference or to set up a REST endpoint to serve the model. Simplify your model deployment by registering models to the MLflow Model Registry. After you have registered your model, you can automatically generate a notebook for batch inference or configure the model for online serving with Serverless Real-Time Inference or Classic MLflow Model Serving on Databricks. For model inference for deep learning applications, Databricks recommends the following workflow.

To debug and tune model inference on Databricks, using GPUs (graphics processing units) can efficiently optimize the running speed for model inference. As GPUs and other accelerators become faster, it is important that the data input pipeline keep up with demand. The data input pipeline reads the data into Spark DataFrames, transforms it and loads it as the input for model inference.

Resources:

- MLflow quickstart (Python)
- Track machine learning training runs
- Automatically log training runs to MLflow
- Track ML Model training data with Delta Lake
- Log, load, register, and deploy MLflow models

CUSTOMER STORY: ITERABLE

Optimizing touch points across the entire customer journey

“With Databricks Lakehouse, we can efficiently deploy powerful ML and AI solutions to help our customers meet rising consumer demands for more personalized experiences that drive revenue and results.” —Sinéad Cheung, Principal Product Manager, Iterable

Captivating an audience and understanding customer journeys are essential to creating deeper brand- customer connections that drive growth, loyalty and revenue. From helping medical practitioners build trust with new patients to ensuring that food delivery users feel connected to their culinary community, Iterable helps more than 1,000 brands optimize and humanize their marketing in today’s competitive landscape.

This need to build personalized and automated customer experiences for its clients drove the company to find a fully managed platform that would simplify infrastructure management, make collaboration possible, and give it the ability to scale for analytics and AI.

With Databricks Lakehouse, Iterable can harness diverse, complex data sets — including conversion events, unique user labels, engagement patterns and business insights — and facilitate rapid prototyping of machine learning models that deliver top-notch and personalized user experiences for higher-converting marketing campaigns. Read the full story here.

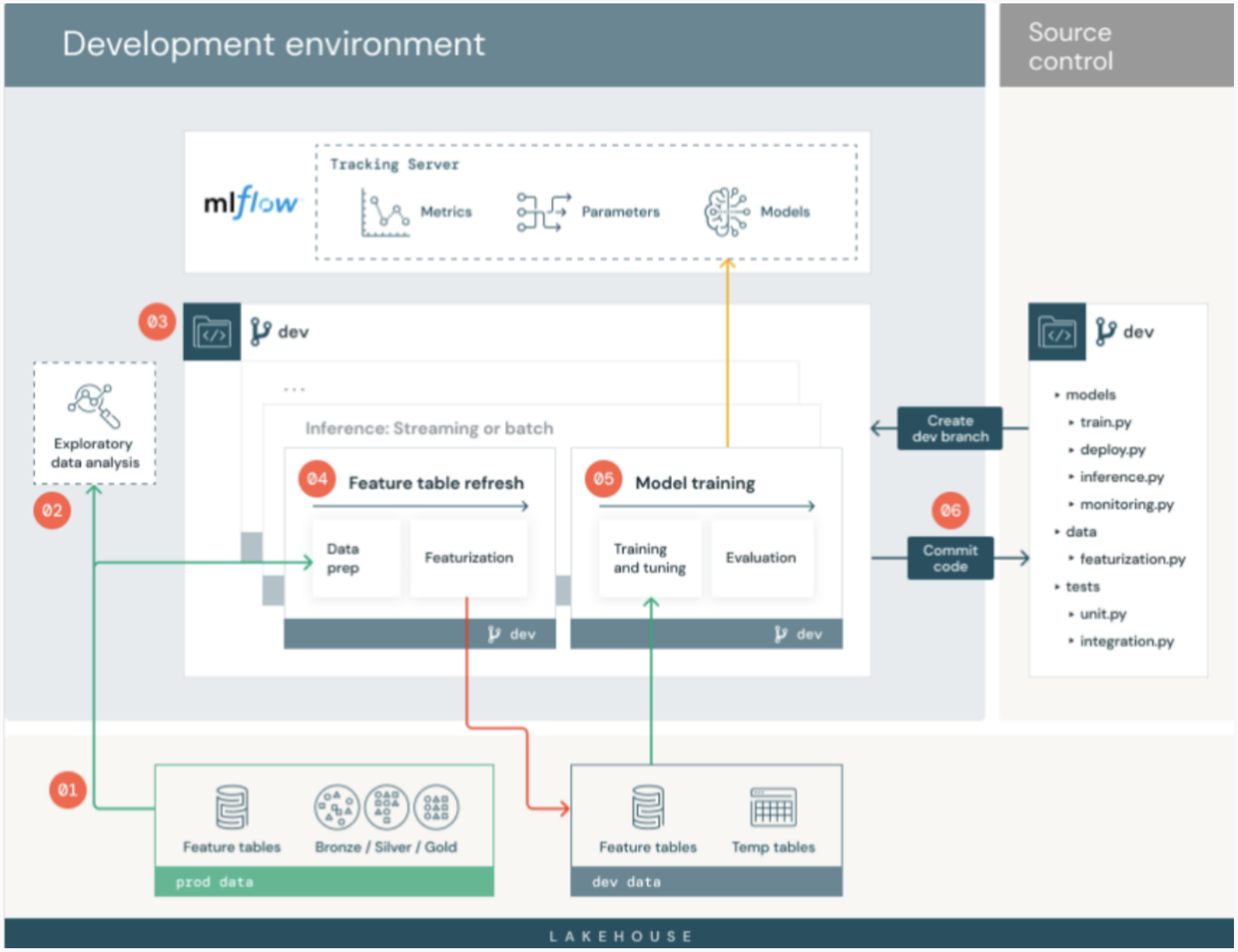
ML Stages

ML workflows include the following key assets: code, models and data. These assets need to be developed (dev), tested (staging) and deployed (production). Each stage needs to operate within an execution environment. So the execution environments, code, models and data are divided into dev, staging and production.

ML project code is often stored in a version control repository (such as Git), with most organizations using branches corresponding to the lifecycle phases of development, staging or production.

Since model lifecycles do not correspond one-to-one with code lifecycles, it makes sense for model management to have its own service. MLflow and its Model Registry support managing model artifacts directly via UI and APIs. The loose coupling of model artifacts and code provides flexibility to update production models without code changes, streamlining the deployment process in many cases.

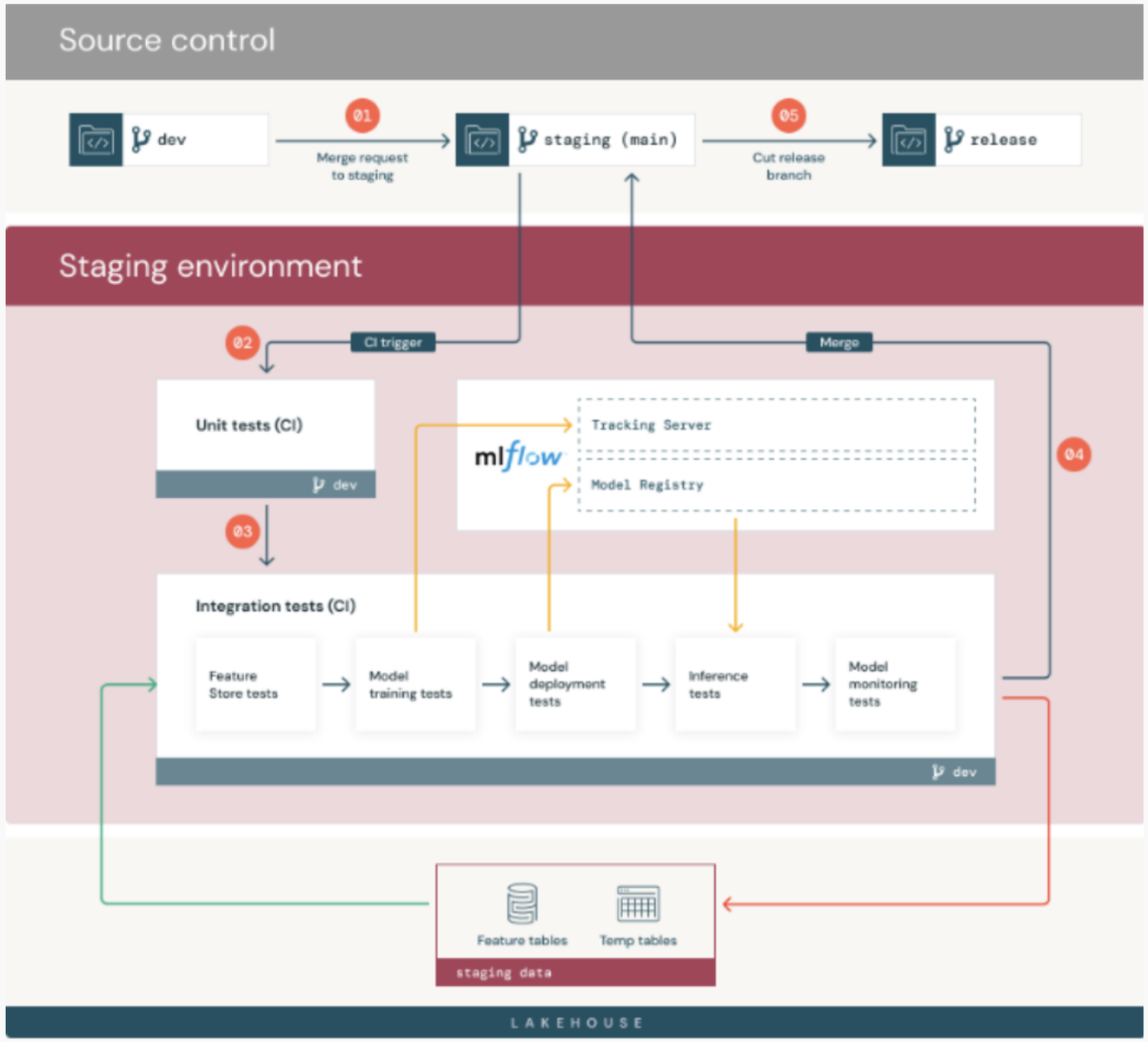
Databricks recommends creating separate environments for the different stages of ML code and model development with clearly defined transitions between stages. The recommended MLOps workflow is broken into these three stages:



Development — The focus of the development stage is experimentation. Data scientists develop features and models and run experiments to optimize model performance. The output of the development process is ML pipeline code that can include feature computation, model training, inference and monitoring.

Staging

This stage focuses on testing the ML pipeline code for production readiness, including code for model training as well as feature engineering pipelines and inference code. The output of the staging process is a release branch that triggers the CI/CD system to start the production stage.

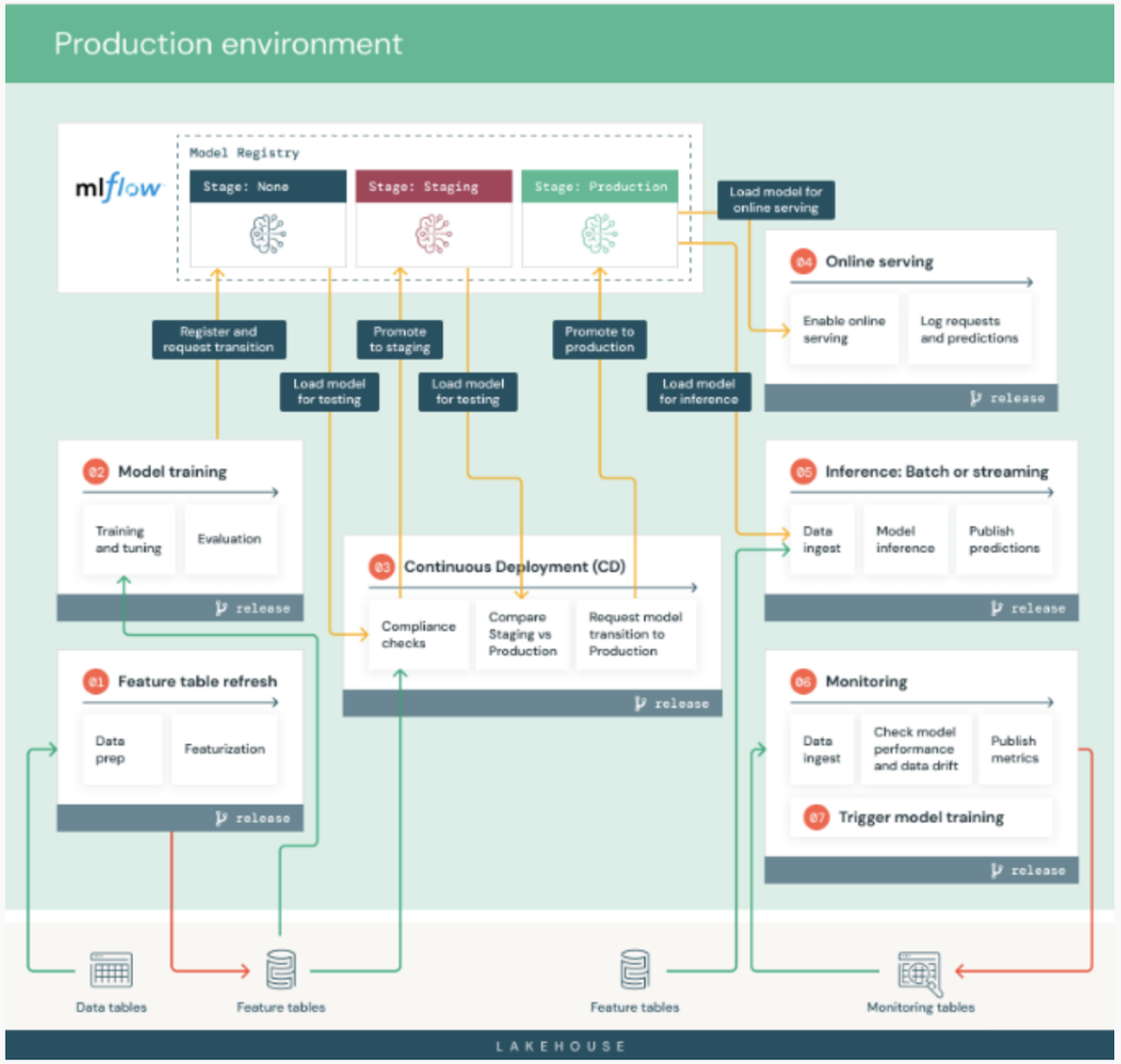


Production

ML engineers own the production environment where ML pipelines are deployed. These pipelines compute fresh feature values, train and test new model versions, publish predictions to downstream tables or applications, and monitor the entire process to avoid performance degradation and instability. Data scientists have visibility to test results, logs, model artifacts and production pipeline status to allow them to identify and diagnose problems in production.

The Databricks Machine Learning home page provides quick access to all the machine learning resources. To access this page, move your mouse or pointer over the left sidebar in the Databricks workspace. From the persona switcher at the top of the sidebar, select Machine Learning.

From the shortcuts menu, you can create a notebook, start AutoML or open a tutorial notebook. The center of the screen includes any recently viewed items, and the sidebar provides quick access to the Experiments page, Databricks Feature Store and Model Registry.
New users can get started with a series of tutorials that illustrate how to use Databricks throughout the



ML lifecycle or access the in-product quickstart for a model-training tutorial notebook that steps through loading data, training and tuning a model, comparing and analyzing model performance and using the model for inference.

Also be sure to download the Big Book of MLOps to learn how your organization can build a robust MLOPs practice incrementally.

Resources:

- MLOps Virtual Event: Standardizing MLOps at Scale
- Virtual Event — Automating the ML Lifecycle With Databricks Machine Learning
- MLOps Virtual Event "Operationalizing Machine Learning at Scale"
- The Big Book of MLOps
- Machine learning on Databricks
- Watch the demos

04

SUMMARY:

The Databricks Lakehouse Platform addresses these challenges

Summary

We've organized the common data challenges for startups and growing digital native businesses into three main buckets: Building a **unified data architecture** — one that supports **scalability and performance**; and building effective **machine learning operations**, all with an eye on cost efficiency and increased productivity.

The Lakehouse Platform provides an efficient and scalable architecture that solves these challenges and will support your data, analytics and AI workloads now and as you scale.

With Databricks you can unify all your data with cost-efficient architecture for highly performant digital native applications and analytic workloads — designed to scale as you grow. Use your data however and wherever you want with open-source flexibility, leverage open formats, APIs and your tools of choice. Ensure reliable, high-performing data workloads while Databricks automatically manages your infrastructure as you scale. Leverage serverless Databricks SQL to increase productivity and scale on demand with up to 12x better price/performance.

Easily access data for ML models and accelerate the full ML lifecycle from experimentation to production.

Discover more about the lakehouse for companies born in the cloud.

databricks

Get started with Databricks Trial

Get a collaborative environment for data teams to build solutions together with interactive notebooks to use Apache Spark™, SQL, Python, Scala, Delta Lake, MLflow, TensorFlow, Keras, scikit-learn and more.

- Available as a 14-day full trial in your own cloud or as a lightweight trial hosted by Databricks.

TRY DATABRICKS FOR FREE

About Databricks

Databricks is the lakehouse company. More than 7,000 organizations worldwide — including Comcast, Condé Nast and over 50% of the Fortune 500 — rely on the Databricks Lakehouse Platform to unify their data, analytics and AI. Databricks is headquartered in San Francisco, with offices around the globe. Founded by the original creators of Apache Spark™, Delta Lake and MLflow, Databricks is on a mission to help data teams solve the world's toughest problems. To learn more, follow Databricks on Twitter, LinkedIn and Facebook.

databricks