The Delta Sharing data exchange follows three efficient steps:

1. The recipient's client authenticates to the sharing server and asks to query a specific table. The client can also provide filters on the data (for example, "country=US") as a hint to read just a subset of the data.
2. The server verifies whether the client is allowed to access the data, logs the request, and then determines which data to send back. This will be a subset of the data objects in cloud storage systems that make up the table.
3. To transfer the data, the server generates short-lived presigned URLs that allow the client to read these Parquet files directly from the cloud provider, so that the transfer can happen in parallel at massive bandwidth, without streaming through the sharing server.

Learn more

Try Delta Sharing

Delta Sharing Demo

Introducing Delta Sharing: An Open Protocol for Secure Data Sharing

Introducing Data Cleanrooms for the Lakehouse

Introducing Databricks Marketplace

Delta Sharing ODSC Webinar

CHAPTER

05

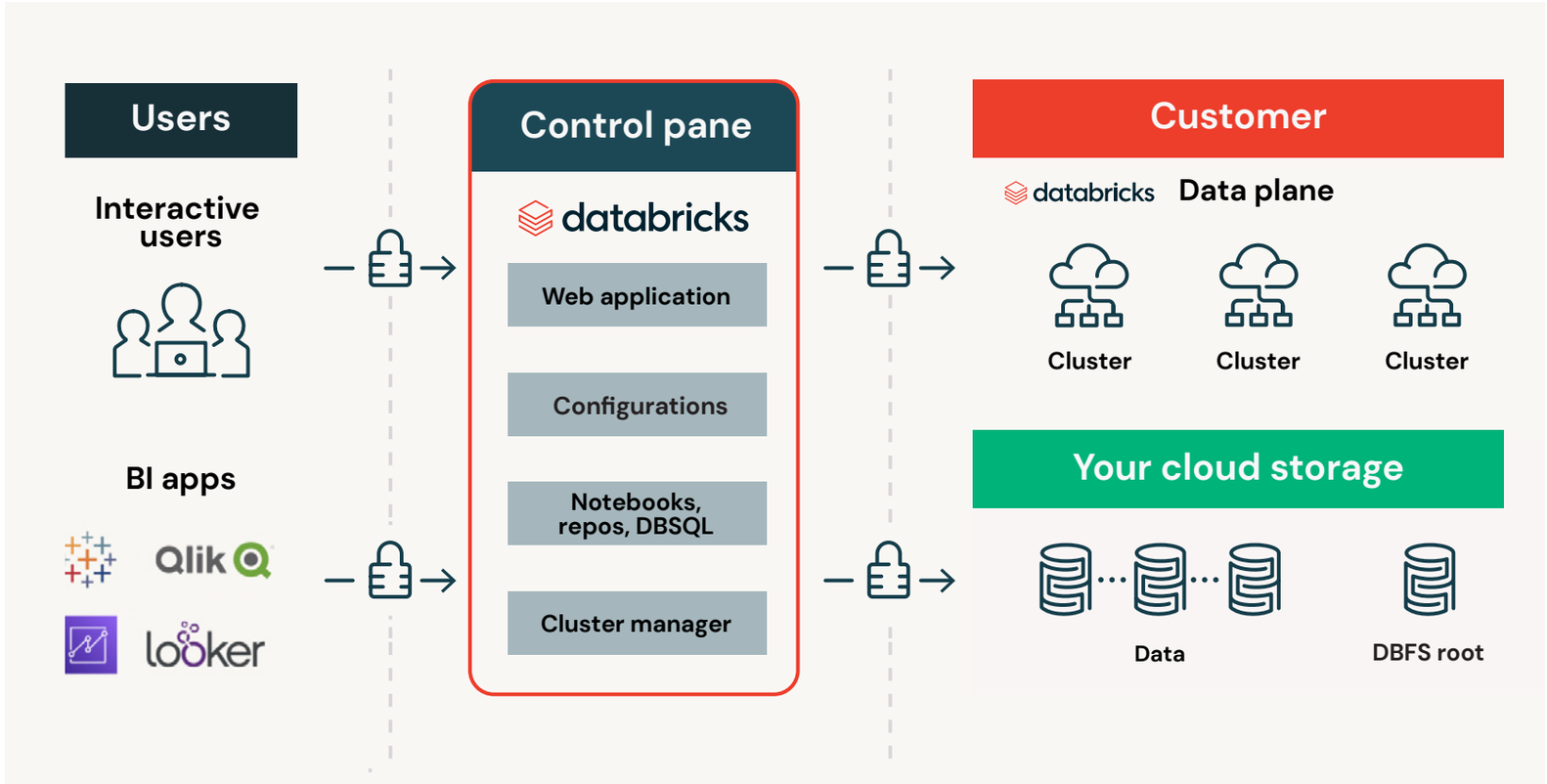
Security

Organizations that operate in multicloud environments need a unified, reliable and consistent approach to secure data. We've learned from our customers that a simple and unified approach to data security for the lakehouse is one of the most critical requirements for modern data solutions. Databricks is trusted by the world's largest organizations to provide a powerful lakehouse architecture with high security and scalability. In fact, thousands of customers trust Databricks with their most sensitive data to analyze and build data products using machine learning (ML). With significant investment in building a highly secure and scalable platform, Databricks delivers end-to-end platform security for data and users.

Platform architecture reduces risk

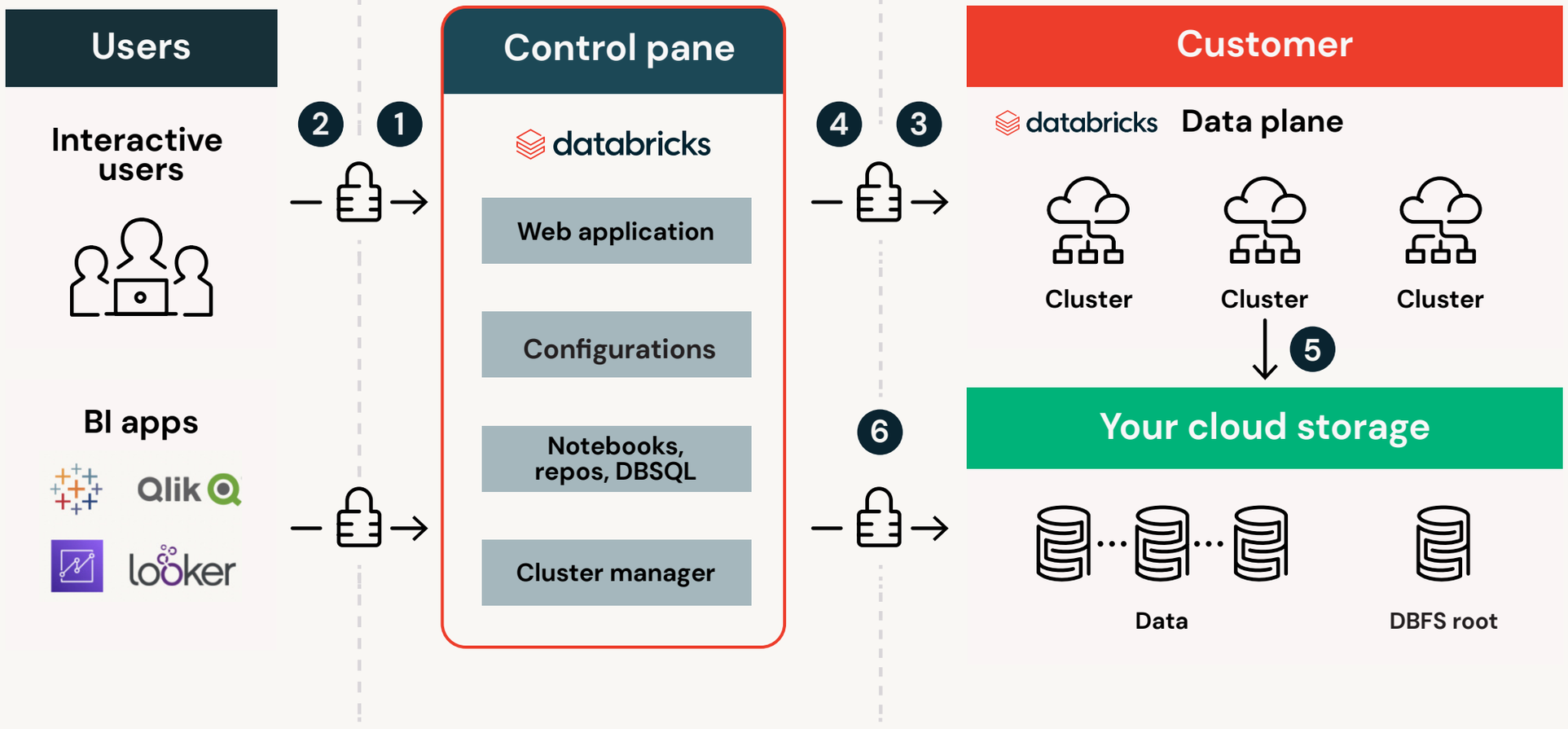
The Databricks lakehouse architecture is split into two separate planes to simplify your permissions, avoid data duplication and reduce risk. The control plane is the management plane where Databricks runs the workspace application and manages notebooks, configuration and clusters. Unless you choose to use serverless compute, the data plane runs inside your cloud service provider account, processing your data without taking it out of your account. You can embed Databricks in your data exfiltration protection architecture using features like customer-managed VPCs/VNets and admin console options that disable export.

While certain data, such as your notebooks, configurations, logs, and user information, is present within the control plane, that information is encrypted at rest, and communication to and from the control plane is encrypted in transit.



You also have choices for where certain data lives: You can host your own store of metadata about your data tables (Hive metastore), or store query results in your cloud service provider account and decide whether to use the Databricks Secrets API.

Step-by-step example



Suppose you have a data engineer that signs in to Databricks and writes a notebook that transforms raw data in Kafka to a normalized data set sent to storage such as Amazon S3 or Azure Data Lake Storage. Six steps make that happen:

1. The data engineer seamlessly authenticates, via your single sign-on if desired, to the Databricks web UI in the control plane, hosted in the Databricks account.
2. As the data engineer writes code, their web browser sends it to the control plane. JDBC/ODBC requests also follow the same path, authenticating with a token.
3. When ready, the control plane uses Cloud Service Provider APIs to create a Databricks cluster, made of new instances in the data plane, in your CSP account. Administrators can apply cluster policies to enforce security profiles.
4. Once the instances launch, the cluster manager sends the data engineer's code to the cluster.
5. The cluster pulls from Kafka in your account, transforms the data in your account and writes it to a storage in your account.
6. The cluster reports status and any outputs back to the cluster manager.

The data engineer does not need to worry about many of the details — simply write the code and Databricks runs it.

Network and server security

Here is how Databricks interacts with your cloud service provider account to manage network and server security

Networking

Regardless of where you choose to host the data plane, Databricks networking is straightforward. If you host it yourself, Databricks by default will still configure networking for you, but you can also control data plane networking with your own managed VPC or VNet.

The serverless data plane network infrastructure is managed by Databricks in a Databricks cloud service provider account and shared among customers, with additional network boundaries between workspaces and between clusters.

Databricks does not rewrite or change your data structure in your storage, nor does it change or modify any of your security and governance policies. Local firewalls complement security groups and subnet firewall policies to block unexpected inbound connections.

Customers at the enterprise tier can also use the IP access list feature on the control plane to limit which IP addresses can connect to the web UI or REST API — for example, to allow only VPN or office IPs.

Servers

In the data plane, Databricks clusters automatically run the latest hardened system image. Users cannot choose older (less secure) images or code. For AWS and Azure deployments, images are typically updated every two-to-four weeks. GCP is responsible for its system image.

Databricks runs scans for every release, including:

- System image scanning for vulnerabilities
- Container OS and library scanning
- Static and dynamic code scanning

Databricks code is peer reviewed by developers who have security training. Significant design documents go through comprehensive security reviews. Scans run fully authenticated, with all checks enabled, and issues are tracked against the timeline shown in this table.

Note that Databricks clusters are typically short-lived (often terminated after a job completes) and do not persist data after they terminate. Clusters typically share the same permission level (excluding high concurrency or Databricks SQL clusters, where more robust security controls are in place). Your code is launched in an unprivileged container to maintain system stability. This security design provides protection against persistent attackers and privilege escalation.

Severity	Remediation time
Critical	< 14 days
High	< 30 days
Medium	< 60 days
Low	When appropriate

Databricks access

Databricks access to your environment is limited to cloud service provider APIs for our automation and support access. Automated access allows the Databricks control plane to configure resources in your environment using the cloud service provider APIs. The specific APIs vary based on the cloud. For instance, an AWS cross-account IAM role, or Azure-owned automation or GKE automation do not grant access to your data sets (see the next section).

Databricks has a custom-built system that allows staff to fix issues or handle support requests — for example, when you open a support request and check the box authorizing access to your workspace. Access requires either a support ticket or engineering ticket tied expressly to your workspace and is limited to a subset of employees and for limited time periods. Additionally, if you have configured audit log delivery, the audit logs show the initial access event and the staff's actions.

Identity and access

Databricks supports robust ACLs and SCIM. AWS customers can configure SAML 2.0 and block non-SSO logins. Azure Databricks and Databricks on GCP automatically integrate with Microsoft Entra ID or GCP identity.

Databricks supports a variety of ways to enable users to access their data.

Examples include:

- The Table ACLs feature uses traditional SQL-based statements to manage access to data and enable fine-grained view-based access
- IAM instance profiles enable AWS clusters to assume an IAM role, so users of that cluster automatically access allowed resources without explicit credentials
- External storage can be mounted or accessed using a securely stored access key
- The Secrets API separates credentials from code when accessing external resources

Data security

Databricks provides encryption, isolation and auditing.

Databricks encryption capabilities are in place both at rest and in motion

For data-at-rest encryption:	For data-in-motion encryption:
• Control plane is encrypted • Data plane supports local encryption • Customers can use encrypted storage buckets • Customers at some tiers can configure customer-managed keys for managed services	• Control plane <-> data plane is encrypted • Offers optional intra-cluster encryption • Customer code can be written to avoid unencrypted services (e.g., FTP)

Customers can isolate users at multiple levels:

- **Workspace level:** Each team or department can use a separate workspace
- **Cluster level:** Cluster ACLs can restrict the users who can attach notebooks to a given cluster
- **High concurrency clusters:** Process isolation, JVM whitelisting and limited languages (SQL, Python) allow for the safe coexistence of users of different privilege levels, and is used with Table ACLs
- **Single-user cluster:** Users can create a private dedicated cluster

Activities of Databricks users are logged and can be delivered automatically to a cloud storage bucket. Customers can also monitor provisioning activities by monitoring cloud audit logs.

Compliance

Databricks supports the following compliance standards on our multi-tenant platform:

- **SOC 2 Type II**
- **ISO 27001**
- **ISO 27017**
- **ISO 27018**
- **ISO 27701**

Certain clouds support Databricks deployment options for FedRAMP High, HITRUST, HIPAA and PCI. Databricks Inc. and the Databricks Platform are also GDPR and CCPA ready.

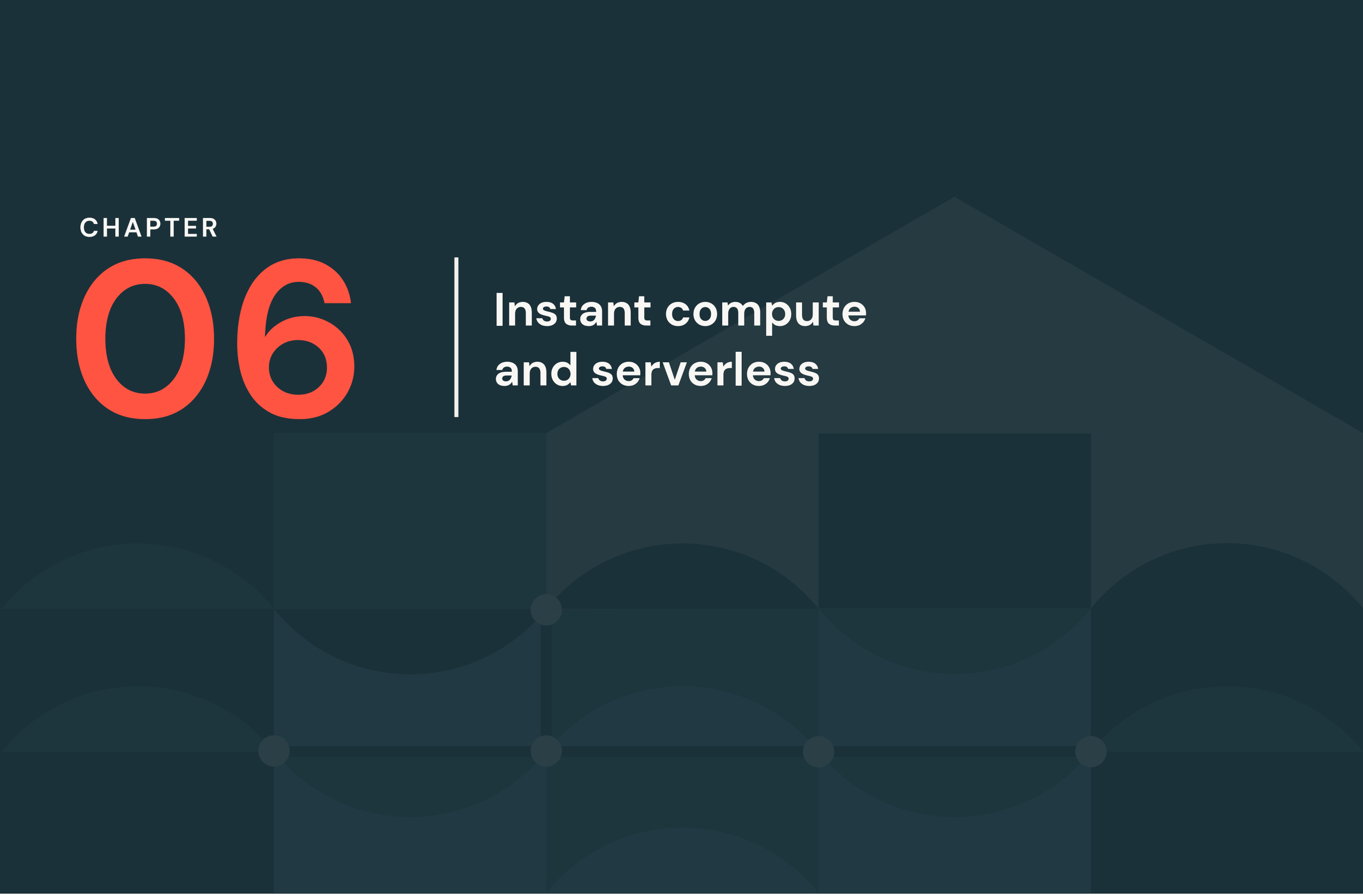
Learn more

To learn more about Databricks security, visit the Security and Trust Center

CHAPTER

06

Instant compute and serverless



What is serverless compute?

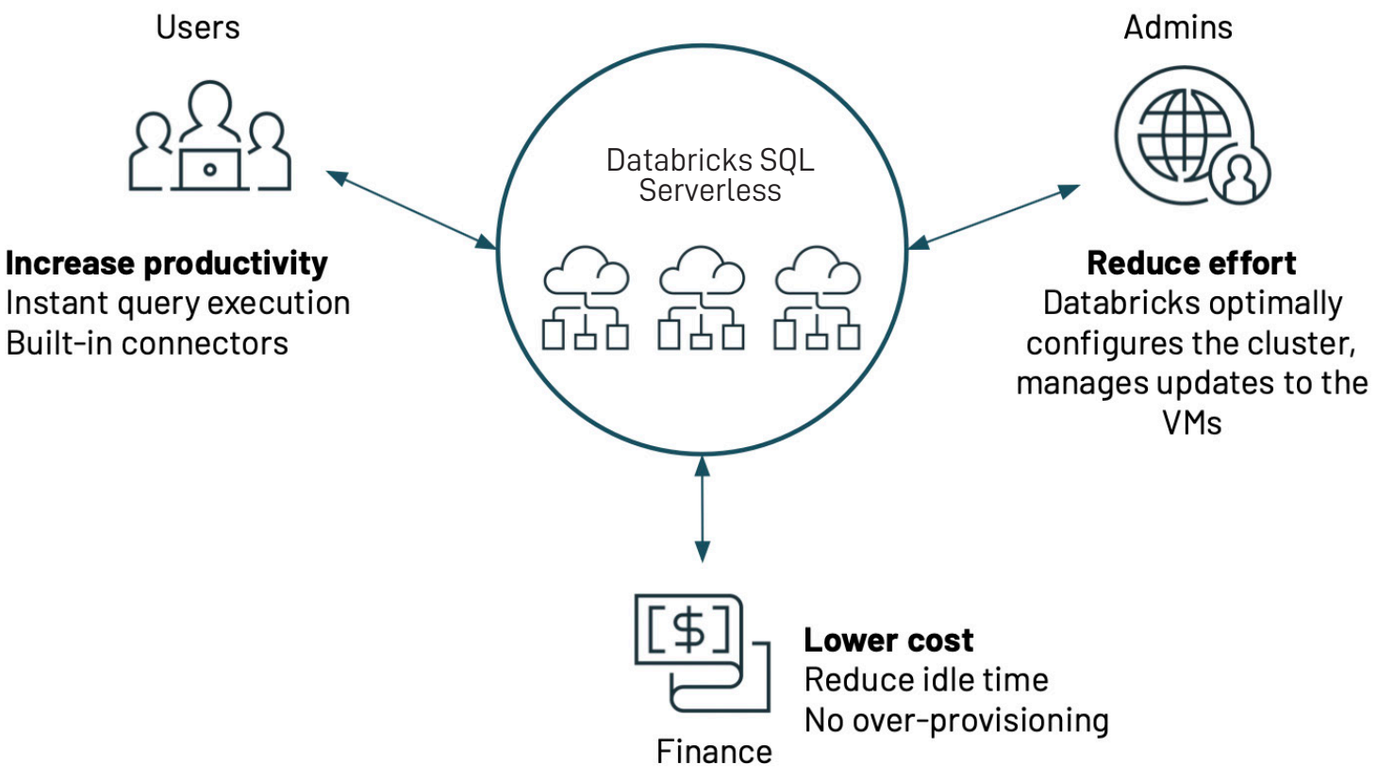
Serverless compute is a fully managed service where Databricks provisions and manages the compute layer on behalf of the customer in the Databricks cloud account instead of the customer account. As of the current release, serverless compute is supported for use with Databricks SQL, Delta Live Tables, and Notebooks.

Databricks SQL Serverless

For example, this is how serverless compute supports the capabilities for Datatabricks SQL, providing instant compute to users for their BI and SQL workloads, with minimal management required and capacity optimizations that can lower overall cost by 20%-40% on average. This makes it even easier for organizations to expand adoption of the lakehouse for business analysts who are looking to access the rich, real-time data sets of the lakehouse with simple and performant solution.

Databricks SQL Serverless is much easier to administer with Databricks taking on the responsibility of deploying, configuring and managing your cluster VMs. Databricks can transfer compute capacity to user queries typically in about 15 seconds — so you no longer need to wait for clusters to start up or scale out to run your queries.

Databricks SQL Serverless also has built-in connectors to your favorite tools such as Tableau, Power BI, Qlik, etc. These connectors use optimized JDBC/ODBC drivers for easy authentication support and high performance. And finally, you save on cost because you do not need to overprovision or pay for the idle capacity.

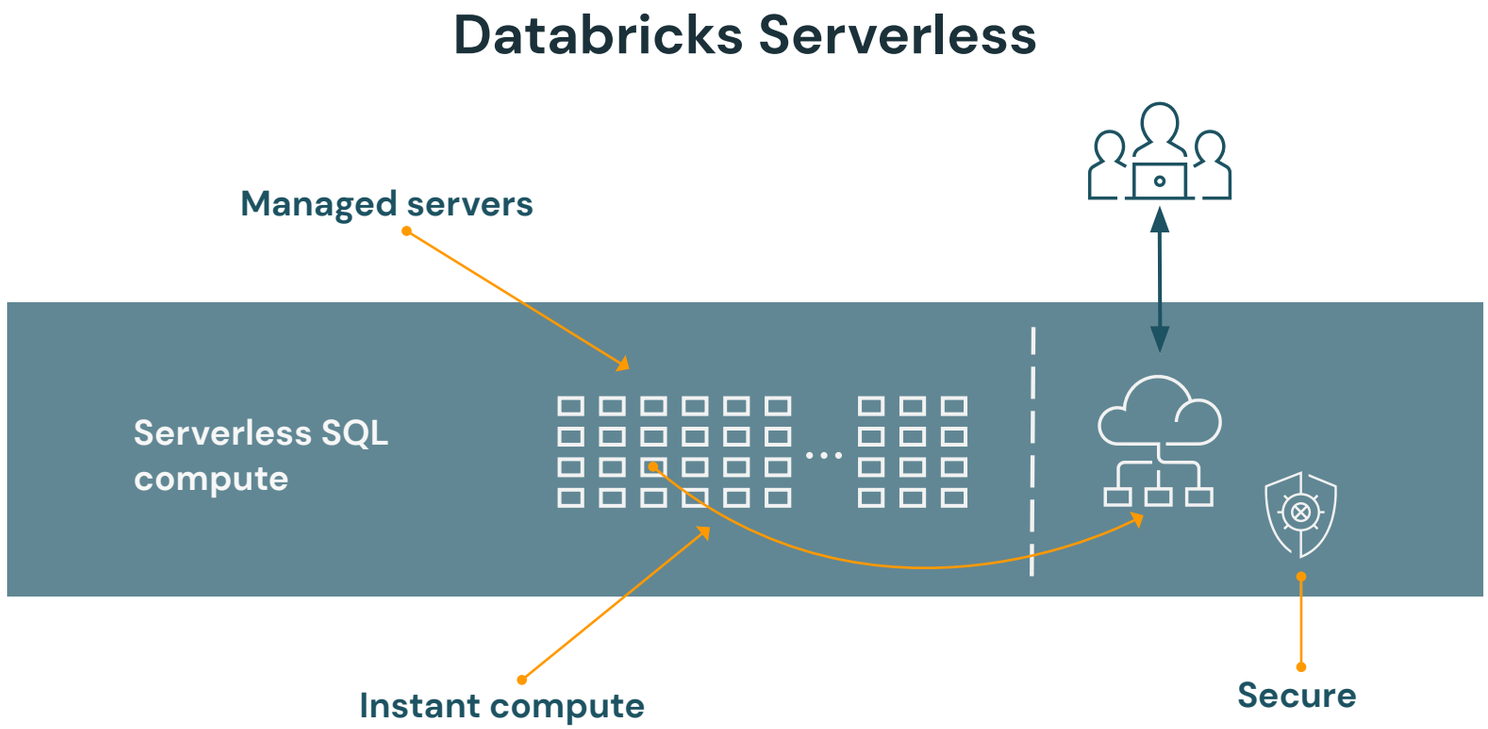


Inside Serverless

At the core of Serverless SQL is a compute platform that operates a pool of servers located in a Databricks' account, running Kubernetes containers that can be assigned to a user within seconds.

When many users are running reports or queries at the same time, the compute platform adds more servers to the cluster (again, within seconds) to handle the concurrent load. Databricks manages the entire configuration of the server and automatically performs the patching and upgrades as needed.

Each server is running a secure configuration and all processing is secured by three layers of isolation: The Kubernetes container hosting the runtime; the virtual machine (VM) hosting the container; and the virtual network for the workspace. Each layer is isolated to one workspace with no sharing or cross-network traffic allowed. The containers use hardened configurations, VMs are shut down and not reused, and network traffic is restricted to nodes in the same cluster.



Learn more

To learn more about Databricks Serverless, visit the documentation page

CHAPTER

07

Data warehousing

Data warehouses are not keeping up with today's world. The explosion of languages other than SQL and unstructured data, machine learning, IoT and streaming analytics are forcing organizations to adopt a bifurcated architecture of disjointed systems: Data warehouses for BI and data lakes for ML. While SQL is ubiquitous and known by millions of professionals, it has never been treated as a first-class citizen on data lakes, until the lakehouse.

The best data warehouse is a lakehouse

The Databricks Data Intelligence Platform provides a simplified multicloud and serverless architecture for your data warehousing workloads. Data warehousing on the lakehouse allows SQL analytics and BI at scale with a common governance model. Now you can ingest, transform and query all your data in-place — using your SQL and BI tools of choice — to deliver real-time business insights at the best price/performance. Built on open standards and APIs, the lakehouse provides the reliability, quality and performance that data lakes natively lack, and integrations with the ecosystem for maximum flexibility — no lock-in.

With data warehousing on the lakehouse, organizations can unify all analytics and simplify their architecture to enable their business with real-time business insights at the best price/performance.

Key benefits

Best price/performance

Lower costs, get the best price/performance and eliminate resource management overhead

On-premises data warehouses have reached their limits — they physically cannot scale to handle the growing volumes of data, and don't provide the elasticity customers need to respond to ever-changing business needs. Cloud data warehouses are a great alternative to on-premises data warehouses, providing greater scale and elasticity, but cloud costs for proprietary cloud data warehouses typically yield to an exponential cost increase following the growth of data volume.

The Databricks Data Intelligence Platform provides instant, elastic SQL serverless compute — decoupled from storage on cheap cloud object stores — and thousands of performance optimizations that can lower overall infrastructure costs by an average of 40%. Databricks automatically determines instance types and configuration for the best price/performance — up to 12x better than traditional cloud data warehouses — and scale for high concurrency use cases.

Built-in governance

One source of truth and one unified governance layer across all data teams

Underpinned by Delta Lake, the Databricks Data Intelligence Platform simplifies your architecture by allowing you to establish one single copy of all your data for in-place analytics and ETL/ELT on your existing data lakes — no more data movements and copies in disjointed systems. Then, seamless integration with Databricks Unity Catalog lets you easily discover, secure and manage all your data with fine-grained governance, data lineage, and standard SQL.

Rich ecosystem

Ingest, transform and query all your data in-place with your favorite tools

Very few tools exist to conduct BI on data lakes. Generally, doing so has required data analysts to submit Spark jobs or use a developer interface. While these tools are common for data scientists, they require knowledge of languages and interfaces that are not traditionally part of a data analyst's tool set. As a result, the learning curve for an analyst to make use of a data lake is too high when well-established tools and methods already exist for data warehouses.

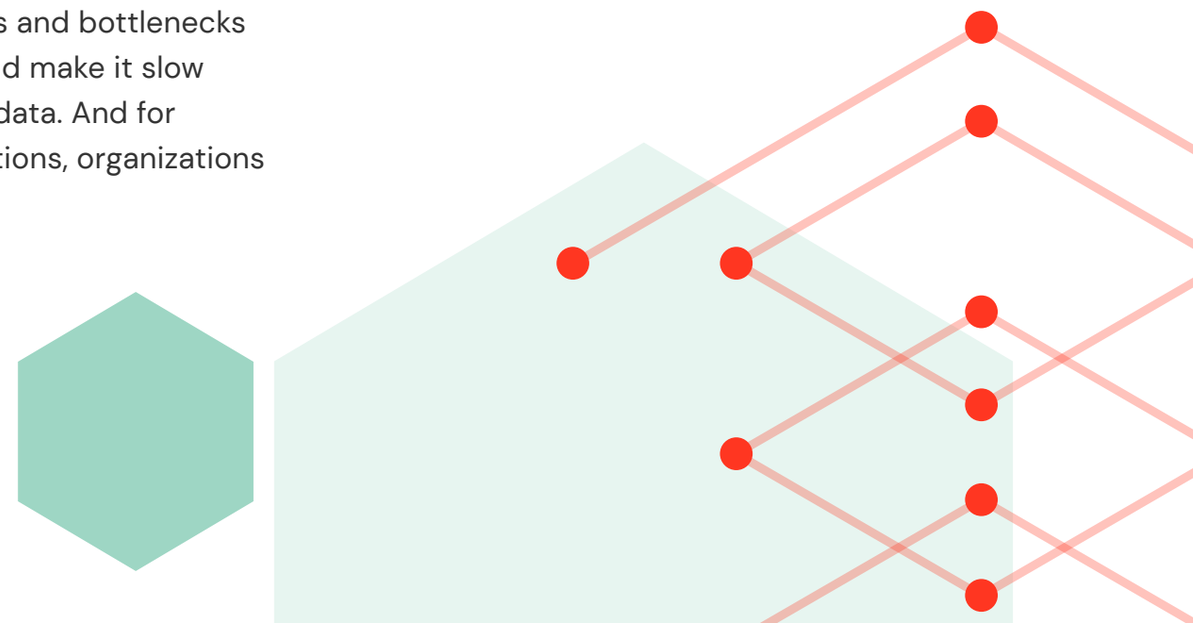
The Databricks Data Intelligence Platform works with your preferred tools like dbt, Fivetran, Power BI or Tableau, allowing analysts and analytical engineers to easily ingest, transform and query the most recent and complete data, without having to move it into a separate data warehouse. Additionally, it empowers every analyst across your organization to quickly and collaboratively find and share new insights with a built-in SQL editor, visualizations and dashboards.

Break down silos

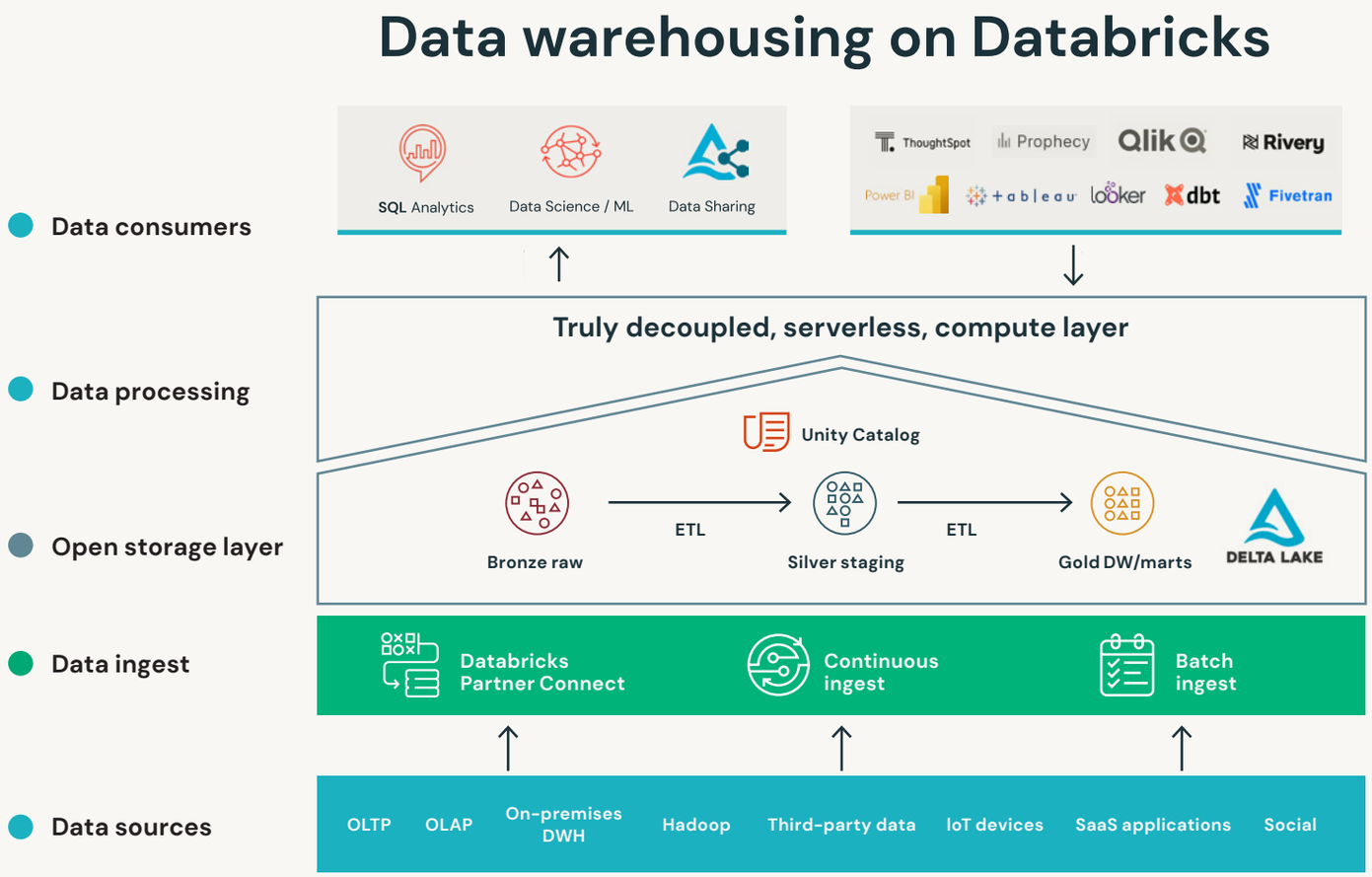
Accelerate time from raw to actionable data and go effortlessly from BI to ML

It is challenging for data engineering teams to enable analysts at the speed that the business requires. Data warehouses need data to be ingested and processed ahead of time before analysts can access and query it using BI tools. Because traditional data warehouses lack real-time processing and do not scale well for large ETL jobs, they create new data movements and bottlenecks for the data engineering team, and make it slow for analysts to access the latest data. And for advanced analytics (ML) applications, organizations will need to manage an entirely different system than their SQL-only data warehouse, slowing down collaboration and innovation.

The Databricks Data Intelligence Platform provides the most complete end-to-end data warehousing solution for all your modern analytics needs, and more. Now you can empower data teams and business users to access the latest data faster for downstream real-time analytics and go effortlessly from BI to ML. Speed up the time from raw to actionable data at any scale — in batch and streaming. And go from descriptive to advanced analytics effortlessly to uncover new insights.



databricks



Learn more

Try Databricks SQL for free

Databricks SQL Demo

Databricks SQL Data Warehousing Admin Demo

On-demand Webinar: Learn Databricks SQL From the Experts

eBook: Inner Workings of the Lakehouse for Analytics and BI

databricks

CHAPTER

08

Data engineering

Organizations realize the value data plays as a strategic asset for growing revenues, improving the customer experience, operating efficiently or improving a product or service. Data is really the driver of all these initiatives. Nowadays, data is often streamed and ingested from hundreds of different data sources, sometimes acquired from a data exchange, cleaned in various ways with different orchestrated steps, versioned and shared for analytics and AI. And increasingly, data is being monetized.

Data teams rely on getting the right data at the right time for analytics, data science and machine learning, but often are faced with challenges meeting the needs of their initiatives for data engineering.

Why data engineering is hard

One of the biggest challenges is accessing and managing the increasingly complex data that lives across the organization. Most of the complexity arises with the explosion of data volumes and data types, with organizations amassing an estimated 80% of data that is unstructured and semi-structured.

With this volume, managing data pipelines to transform and process data is slow and difficult, and increasingly expensive. And to top off the complexity, most businesses are putting an increased emphasis on multicloud environments which can be even more difficult to maintain.

Zhamak Dehghani, a principal technology consultant at Thoughtworks, wrote that data itself has become a product, and the challenging goal of the data engineer is to build and run the machinery that creates this high-fidelity data product all the way from ingestion to monetization.

Despite current technological advances data engineering remains difficult for several reasons:

Complex data ingestion methods

Data ingestion means retrieving batch and streaming data from various sources and in various formats. Ingesting data is hard and complex since you either need to use an always-running streaming platform like Apache Kafka or you need to be able to keep track of which files haven't been ingested yet. Data engineers are required to spend a lot of time hand-coding repetitive and error-prone data ingestion tasks.

Data engineering principles

These days, large operations teams are often just a memory of the past. Modern data engineering principles are based on agile software development methodologies. They apply the well-known "you build it, you run it" paradigm, use isolated development and production environments, CI/CD, and version control transformations that are pushed to production after validation. Tooling needs to support these principles.

Third-party tools

Data engineers are often required to run additional third-party tools for orchestration to automate tasks such as ELT/ETL or customer code in notebooks. Running third-party tools increases the operational overhead and decreases the reliability of the system.

Performance tuning

Finally, with all pipelines and workflows written, data engineers need to constantly focus on performance, tuning pipelines and architectures to meet SLAs. Tuning such architectures requires in-depth knowledge of the underlying architecture and constantly observing throughput parameters.

Most organizations are dealing with a complex landscape of data warehouses and data lakes these days. Each of those platforms has its own limitations, workloads, development languages and governance model.

Databricks makes modern data engineering simple

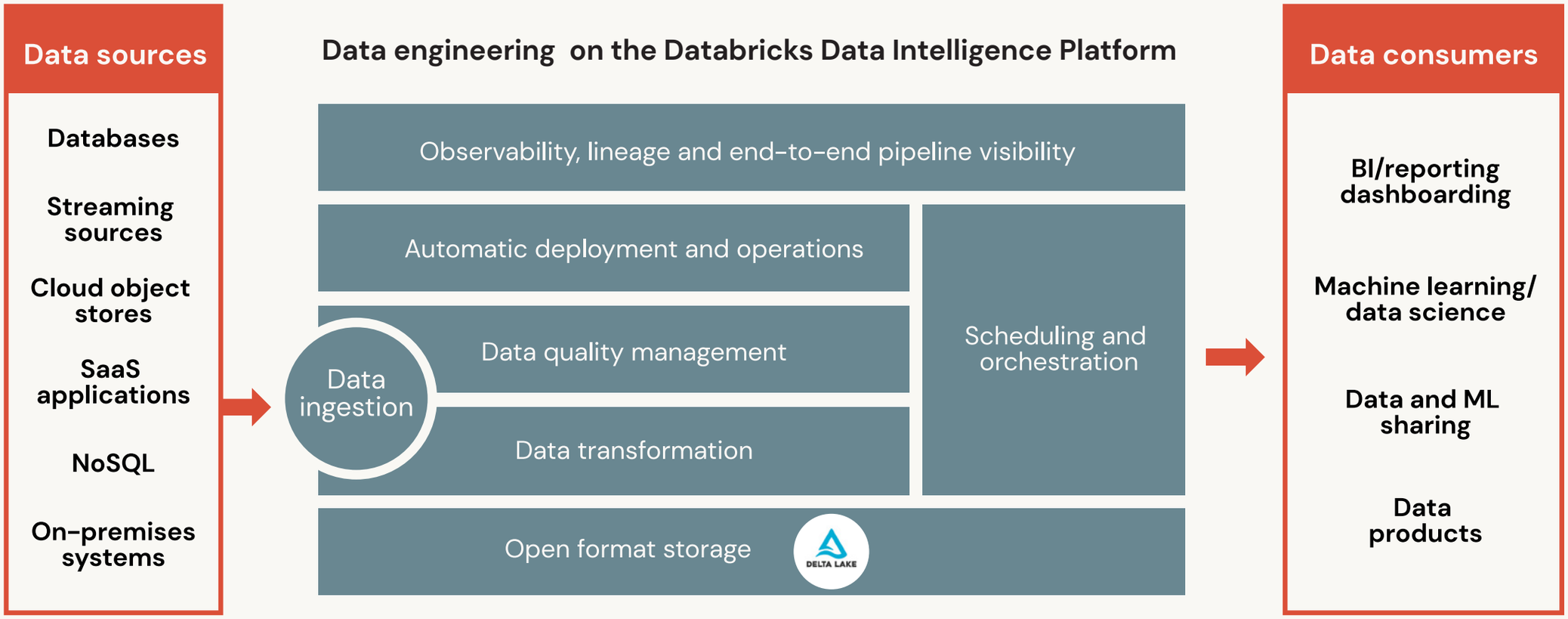
There is no industry-wide definition of modern data engineering. This should come close:

*A **unified data platform** with **managed data ingestion**, schema detection, enforcement, and evolution, paired with **declarative, auto-scaling data flow** integrated with a lakehouse **native orchestrator** that supports all kinds of workflows.*

Data engineering is an integrated part of the Data Intelligence Platform that empowers data practitioners everywhere to ingest, transform, and orchestrate data pipelines. Databricks allows both technical and non-technical users to easily deliver the right data, at the right time, at any scale.

With the Databricks Data Intelligence Platform, data engineers have access to an end-to-end data engineering solution for ingesting, transforming, processing, scheduling and delivering data. The lakehouse architecture automates the complexity of building and maintaining pipelines and running ETL workloads directly on a data lake so data engineers can focus on quality and reliability to drive valuable insights.

Data engineering on the Databricks Data Intelligence Platform serves as a robust foundation for a broad spectrum of downstream use cases, including cutting-edge applications such as Gen AI and Large Language Models (LLMs). Leveraging Databricks Assistant, AI-assistant combing natural language processing (NLP) and intelligent automation, simplifies data engineer's workflows by performing the desired actions within the platform.


Data sources
Databases
Streaming sources
Cloud object stores
SaaS applications
NoSQL
On-premises systems
Data engineering on the Databricks Data Intelligence Platform
Observability, lineage and end-to-end pipeline visibility
Automatic deployment and operations
Data quality management
Data ingestion
Data transformation
Scheduling and orchestration
Open format storage
DELTA LAKE
Data consumers
BI/reporting dashboarding
Machine learning/ data science
Data and ML sharing
Data products