UPGRADING BI AND VISUALIZATION TOOLS

The majority of BI and visualization tools interact with Databricks by connecting to SQL warehouses, so once the warehouses are Unity Catalog–enabled, they'll be able to communicate with Unity Catalog. Connection strings used to connect to the SQL warehouses by the external tools should be modified to include the catalog name if the workspace default catalog doesn't suffice.

Upgrading machine learning models

Models that are registered in the workspace model registry can be upgraded to the Databricks-hosted version of MLflow Model Registry in Unity Catalog.

To create new registered models, you need the `CREATE_MODEL` privilege on a schema, in addition to the `USE SCHEMA` and `USE CATALOG` privileges on the schema and its enclosing catalog. `CREATE_MODEL` is a new schema-level privilege that you can grant using the Catalog Explorer UI or the `SQL GRANT` command, as shown below.

```
GRANT CREATE_MODEL ON SCHEMA <schema-name> TO <principal>
```

To upgrade existing ML workflows to Unity Catalog, you can register models under a schema in Unity Catalog. By default, the MLflow Python client creates models in the Databricks workspace model registry. To use Unity Catalog, configure the MLflow client to set the model registry as **`databricks-uc`**, which points to the Unity Catalog registry, and use the three-level namespace when registering the model.

```
import mlflow
mlflow.set_registry_uri("databricks-uc")
...
mlflow.register_model(model_uri, "catalog.schema.model_name")
```

Individual models can be upgraded to Unity Catalog by using the sample Python notebook, which uses the model registry APIs to upgrade existing models in the workspace Model Registry to Unity Catalog.

UPGRADING FEATURE TABLES

Feature tables upgrade from workspace to Unity Catalog requires the underlying table to be upgraded to Unity Catalog as a requirement. Follow instructions in the "Upgrading tables" section of this chapter to make sure that the table is available prior to migrating the feature tables. Once the underlying tables are available in Unity Catalog, use `upgrade_workspace_table` to upgrade the workspace feature table metadata to Unity Catalog, as illustrated in the following code.

```
%pip install databricks-feature-engineering --upgrade

dbutils.library.restartPython()

from databricks.feature_engineering import UpgradeClient
upgrade_client = UpgradeClient()
upgrade_client.upgrade_workspace_table(
  source_workspace_table='recommender_system.customer_features',
  target_uc_table='ml.recommender_system.customer_features'
)
```

The following metadata is upgraded to Unity Catalog:

- Primary keys
- Time series columns
- Table and column comments (descriptions)
- Table and column tags
- Notebook and job lineage

Upgrade challenges and limitations

Even though Unity Catalog is a significant deviation from the way the Hive metastore is governed, the upgrade process is designed to be streamlined, with minimal impact on end users and their day-to-day operations interacting with the data. The primary challenges of upgrading to Unity Catalog include the time investment required for the upgrade process and the need to maintain both the Hive metastore and Unity Catalog during the upgrade to minimize downtime and impact on end users and downstream data consumers.

Certain limitations exist in Unity Catalog–enabled compute that didn't exist in non–Unity Catalog compute due to the strong isolation for processes running in the Unity Catalog compute enabled by Lakeguard. User and process isolation is enforced by Lakeguard on shared and single-user clusters supporting fine-grained access controls including row-level and column-level filters.

Some of these limitations are described in the following sections.

SINGLE USER ACCESS MODE LIMITATIONS ON UNITY CATALOG

Single user access mode on Unity Catalog has the following limitations.

Fine-grained access control limitations for Unity Catalog single user access mode on DBR 15.3 and earlier

- Dynamic views aren't supported
- To read from a view, you must have SELECT on all referenced tables and views
- You can't access a table that has a row filter or column mask

Compute with single user access mode on Databricks Runtime 15.4 LTS or later (Public Preview)
Databricks Runtime 15.4 LTS and later support fine-grained access control on single-user compute. To take advantage of the data filtering provided in Databricks Runtime 15.4 LTS and later, you must also verify that your workspace is enabled for serverless compute, because the data filtering functionality that supports fine-grained access controls runs on serverless compute. You might therefore be charged for serverless compute resources when you use single-user compute to run data filtering operations. See Fine-grained access control on single-user compute.

Streaming limitations for Unity Catalog single user access mode

- Asynchronous checkpointing isn't supported in Databricks Runtime 11.3 LTS and earlier
- StreamingQueryListener requires Databricks Runtime 15.1 or later to use credentials or interact with objects managed by Unity Catalog on single user compute

SHARED ACCESS MODE LIMITATIONS ON UNITY CATALOG

Shared access mode on Unity Catalog has the following limitations.

- Databricks Runtime ML and Spark Machine Learning Library (MLlib) aren't supported
- Spark-submit jobs aren't supported
- In Databricks Runtime 13.3 and later, individual rows must not exceed 128MB
- PySpark UDFs can't access Git folders, workspace files or volumes to import modules in Databricks Runtime 14.2 and earlier
- DBFS root and mounts don't support FUSE
- When you use shared access mode with credential pass-through, Unity Catalog features are disabled
- Custom containers aren't supported

Language support for Unity Catalog shared access mode

- R isn't supported
- Scala is supported in Databricks Runtime 13.3 and later
  - In Databricks Runtime 15.4 LTS and later, all Java or Scala libraries (JAR files) bundled with Databricks Runtime are available on compute in Unity Catalog access modes
  - For Databricks Runtime 15.3 or earlier on compute that uses shared access mode, set the Spark config `spark.databricks.scala.kernel.fullClasspath.enabled` to `true`

Spark API limitations for Unity Catalog shared access mode

- RDD APIs aren't supported
- DBUtils and other clients that directly read the data from cloud storage are only supported when you use an external location to access the storage location. See Create an external location to connect cloud storage to Databricks.
- Spark Context (sc), spark.sparkContext and sqlContext aren't supported for Scala in any Databricks Runtime and aren't supported for Python in Databricks Runtime 14.0 and later
  - Databricks recommends using the spark variable to interact with the SparkSession instance
  - The following sc functions are also not supported: emptyRDD, range, init_batched_serializer, parallelize, pickleFile, textFile, wholeTextFiles, binaryFiles, binaryRecords, sequenceFile, newAPIHadoopFile, newAPIHadoopRDD, hadoopFile, hadoopRDD, union, runJob, setSystemProperty, uiWebUrl, stop, setJobGroup, setLocalProperty, getConf
  - The following Scala Dataset API operations require Databricks Runtime 15.4 LTS or later: map, mapPartitions, foreachPartition, flatMap, reduce and filter

UDF limitations for Unity Catalog shared access mode

User-defined functions (UDFs) have the following limitations with shared access mode:

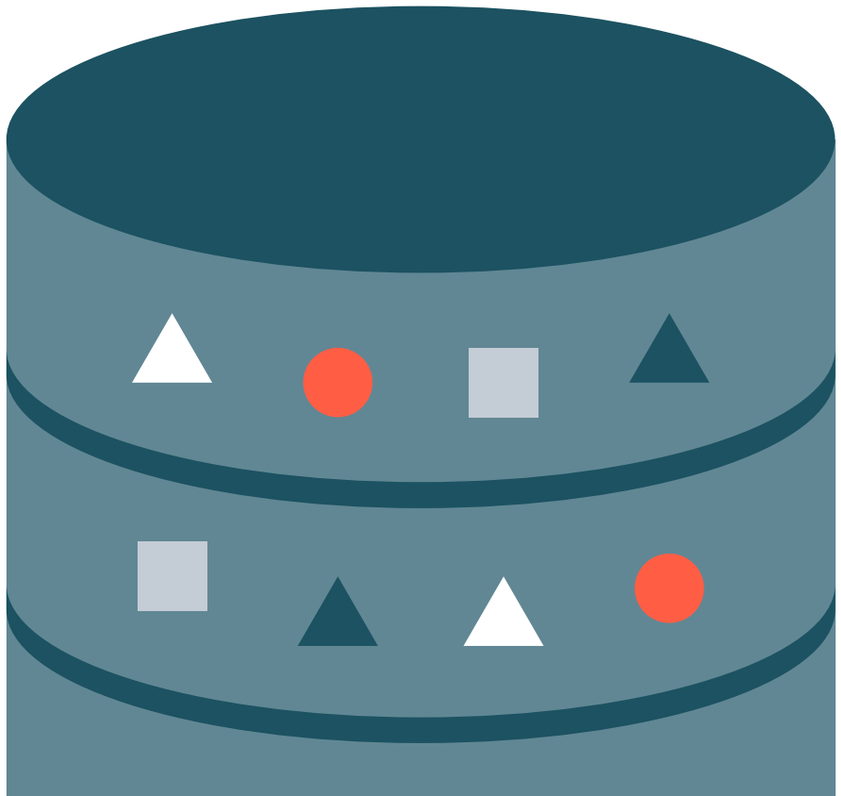
- Hive UDFs aren't supported
- applyInPandas and mapInPandas require Databricks Runtime 14.3 or later
- Scala scalar UDFs require Databricks Runtime 14.2 or later. Other Scala UDFs and UDAFs aren't supported.
- In Databricks Runtime 14.2 and earlier, using a custom version of grpc, pyarrow or protobuf in a PySpark UDF through notebook-scoped or cluster-scoped libraries isn't supported because the installed version is always preferred. To find the version of installed libraries, see the System Environment section of the specific Databricks Runtime version release notes.
- Python scalar UDFs and Pandas UDFs require Databricks Runtime 13.3 LTS or later. Other Python UDFs, including UDAFs, UDTFs and Pandas on Spark aren't supported.

For a complete list of limitations, refer to the documentation.

Rollback strategy

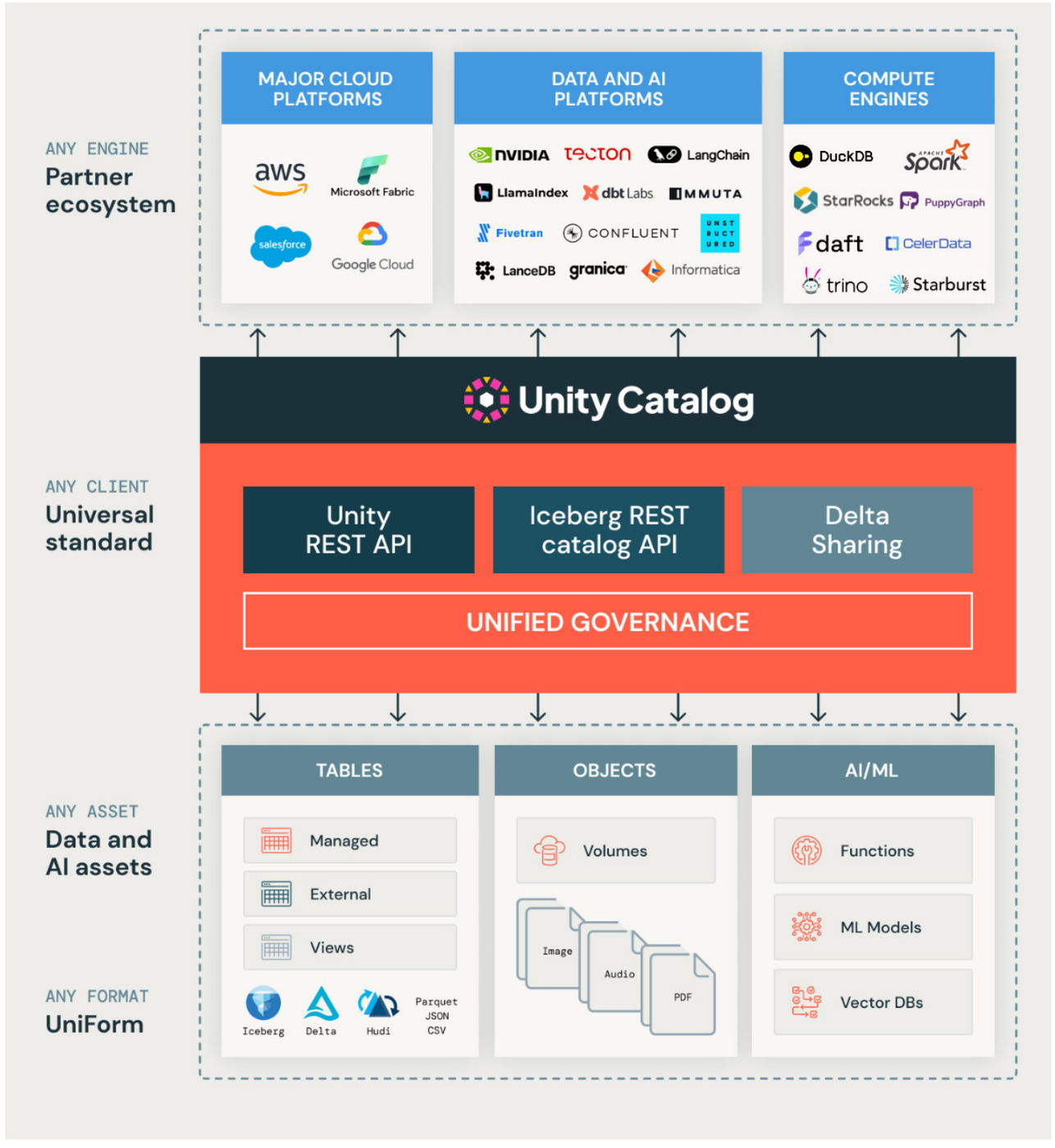
The Unity Catalog upgrade is a significant change to the Databricks Platform, requiring modifications to code, operations and infrastructure. It's important to have a rollback strategy in case an issue arises that prevents the upgrade from proceeding.

- We recommend that you continue hydrating the Hive metastore during the Unity Catalog upgrade to ensure a fallback option is available in case of any issues
- A phased approach is recommended for the Unity Catalog upgrade instead of a big bang approach. This gradual implementation can help minimize risks and disruptions.
- All the code changes should only come via a code repository and CI/CD process
- Make use of Databricks Terraform Provider to automate the Unity Catalog deployment process



Open Data Accessibility

Here at Databricks, we're excited to announce the open sourcing of Unity Catalog, the industry's first open source catalog for data and AI governance across clouds, data formats and data platforms. The project is now available on GitHub, marking the first step in our journey to bring the Unity Catalog vision into open source. Unity Catalog is hosted by LF AI & Data, a part of the Linux Foundation that fosters open source innovation in AI and data.



Why open source?

In the past two years since Unity Catalog's General Availability we've seen widespread adoption. Organizations have consistently expressed the need for an open foundation for their data and AI applications, not just for today but for the innovations of the coming decades.

Unfortunately, most data platforms today function as walled gardens. Many cloud data warehouses use "native tables" that aren't in open formats, while other platforms require customers to pay for always-on compute even when accessing data from external engines, de facto doubling the computation costs. Additionally, many platforms limit the data formats and clients they support.

This leads to siloed data and fragmented governance across assets. Without a multimodal interface for tabular data and AI assets, organizations are forced to piece together multiple disjointed solutions. Databricks has already taken a strong stance in the industry by being the only major platform where all tables are in open formats by default and by opening up Delta tables to Apache Iceberg™ clients with Delta Lake UniForm last year. By open sourcing Unity Catalog, we're providing organizations with an open foundation for their current and future workloads.

INTEROPERABLE, OPEN, UNIFIED

Among all the characteristics of Unity Catalog, the three most important ones you need to remember are: *interoperability*, *openness* and *unified governance*.

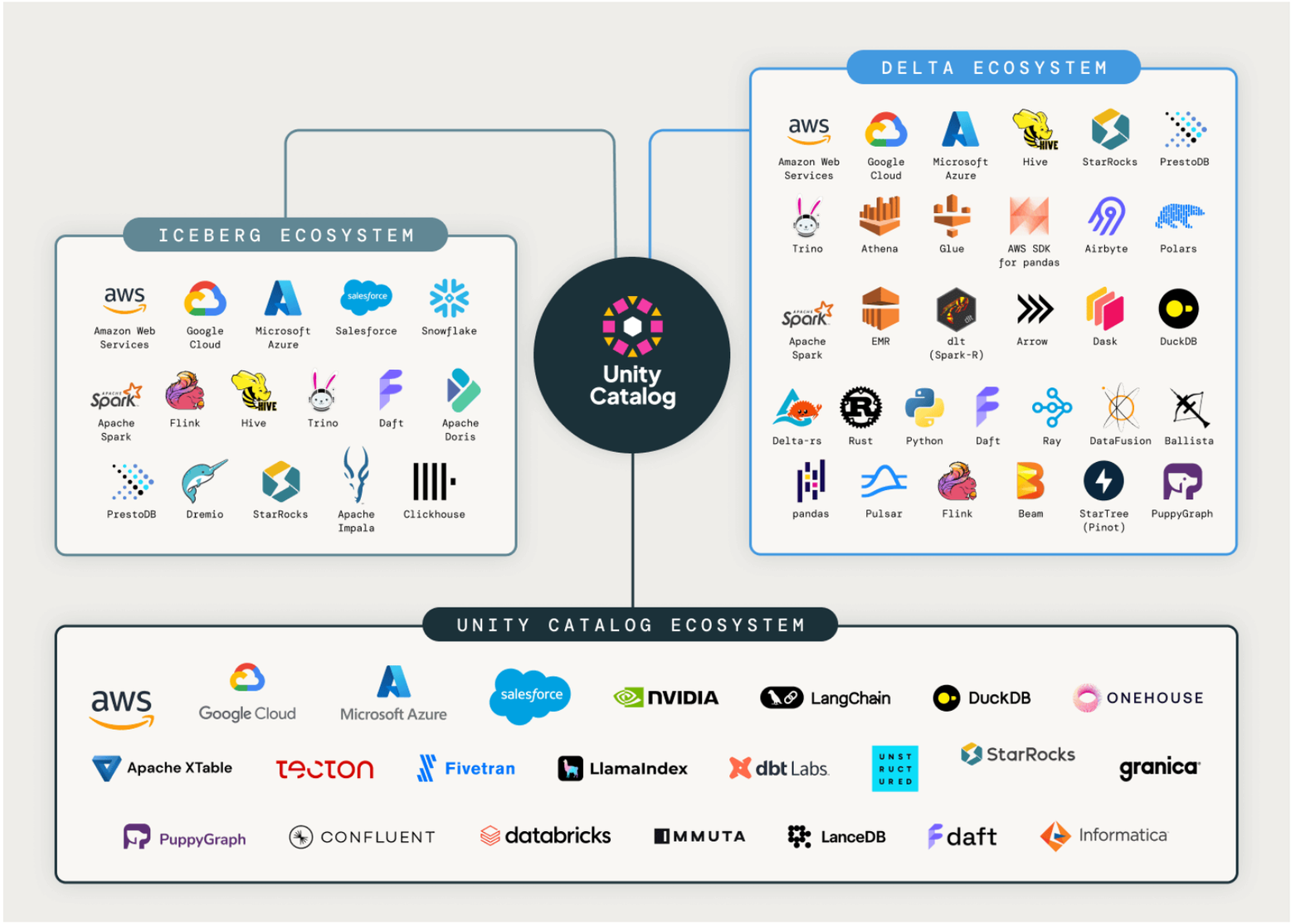
Unity Catalog is interoperable across any format and any engine. It supports Delta Lake, Apache Iceberg via UniForm, Parquet, CSV, JSON and many other formats. It also implements the Iceberg REST Catalog API to interoperate with a broad ecosystem of engines.

Unity Catalog is now also open source under the Apache 2.0 license, including an OpenAPI specification, with both server and clients. Customers need to be flexible and must be able to choose which engine to use, and open standards maximize this by ensuring extensive interoperability across engines, tools and platforms.

Unity Catalog is a universal catalog that's able to govern and secure any type of data, be it structured Delta tables, unstructured folders of images and documents or GenAI models and tools with a unified solution. It's able to secure access with strong authentication, secure credential vending and asset-level access control.

ECOSYSTEM

Unity Catalog open source is supported by a vibrant community comprising all three main cloud providers, data and AI platforms and many software vendors already. Choosing Unity Catalog today makes your company future-ready.



JOIN THE COMMUNITY TODAY

Join the Unity Catalog open source community on LinkedIn, GitHub or Slack.

Summary

The Databricks Data Intelligence Platform is a versatile, cloud-native solution that operates seamlessly across AWS, Azure and Google Cloud Platform (GCP). This platform empowers organizations with the flexibility to choose their preferred cloud provider or adopt a multicloud strategy, enabling efficient execution and integration of workloads across different environments. However, as organizations scale their use of data, effective governance becomes increasingly critical, particularly in the face of challenges like decentralized responsibilities and evolving roles such as the chief data officer (CDO).

To address these governance challenges, Databricks introduces **Unity Catalog**, a unified governance framework that centralizes the management of structured and unstructured data, AI models and data sharing across all cloud environments. Unity Catalog simplifies governance by providing a consistent experience across clouds, making it easier for organizations to implement and enforce data governance policies at scale.

Unity Catalog's cross-cloud data sharing capabilities are further enhanced by **Delta Sharing**, an open protocol for secure data sharing across platforms, clouds and regions. Delta Sharing allows organizations to share live data with external partners, customers and vendors, regardless of their infrastructure. This open source protocol supports various data formats and is designed to integrate seamlessly with Unity Catalog, enabling secure, scalable and governed data sharing.

Delta Sharing, a key feature of Unity Catalog, revolutionizes how data is shared across organizations and platforms. It provides a secure, flexible and open method for sharing live data with any recipient, regardless of their infrastructure or cloud provider. Delta Sharing is the first open source data sharing protocol designed to facilitate collaboration across different platforms and ecosystems.

In addition to its robust governance and data sharing capabilities, Unity Catalog is now available as an open source project, hosted by the Linux Foundation. By open sourcing Unity Catalog, Databricks provides organizations with an open foundation for data and AI governance that's interoperable across clouds, data formats and platforms.

The Databricks Data Intelligence Platform, with the integration of Unity Catalog, provides a powerful and unified solution for managing data governance, security and sharing across multicloud environments. Unity Catalog's ability to centralize governance, abstract cloud-specific complexities and facilitate secure data sharing through Delta Sharing makes it an indispensable tool for modern organizations. By open sourcing Unity Catalog, Databricks extends its commitment to openness and interoperability, empowering organizations to build scalable, secure and innovative data and AI solutions.

As businesses continue to navigate the complexities of data governance and multicloud strategies, Unity Catalog stands out as a future-proof solution that not only simplifies data management but also enhances security, compliance and collaboration across the enterprise.

Resources

Demos

- Watch the Unity Catalog demos

Documentation

- AWS documentation
- Azure documentation
- GCP documentation

Featured talks

- [Keynote] Evolving Data Governance With Unity Catalog Presented by Matei Zaharia at Data + AI Summit 2024
- What's New in Unity Catalog — With Live Demos
- Technical Deep Dive for Practitioners: Databricks Unity Catalog from A-Z

About the Authors

Pearl Ubaru is a Senior Technical Product Marketing Engineer at Databricks with skills in data sharing, data warehousing, data analytics and BI, and data governance. You can reach out to Pearl on LinkedIn.

Fabian Lanz is a Senior Solutions Architect at Databricks with a primary focus on AI and governance. He works with customers to develop secure and scalable end-to-end data and AI solutions. You can reach out to Fabian on LinkedIn.

Karthik Subbarao is a Specialist Solutions Architect at Databricks, focusing on platform, security and governance topics. He helps organizations design and implement secure, scalable and well-governed data platforms. You can reach Karthik on LinkedIn.

Kiran Sreekumar is a Specialist Solutions Architect at Databricks, focusing on data governance. He works with customers to implement their enterprise data governance strategies. You can reach Kiran on LinkedIn.

Mattia Zeni is a Senior Solutions Architect at Databricks, where he collaborates with strategic customers to maximize the value of their data using open data platforms. His primary areas of focus are data governance and managed serverless compute. Connect with Mattia on LinkedIn.

Jince James is a Senior Resident Solutions Architect at Databricks, focusing on data engineering, governance and data strategy. He helps organizations navigate and solve complex data and AI challenges. You can connect with Jince on LinkedIn.

Ivan Trusov is a Senior Specialist Solutions Architect on the Databricks EMEA team. His focus areas are governance, platform automation and security. He enables data teams to build efficient and secure data platforms. You can reach out to Ivan on LinkedIn.

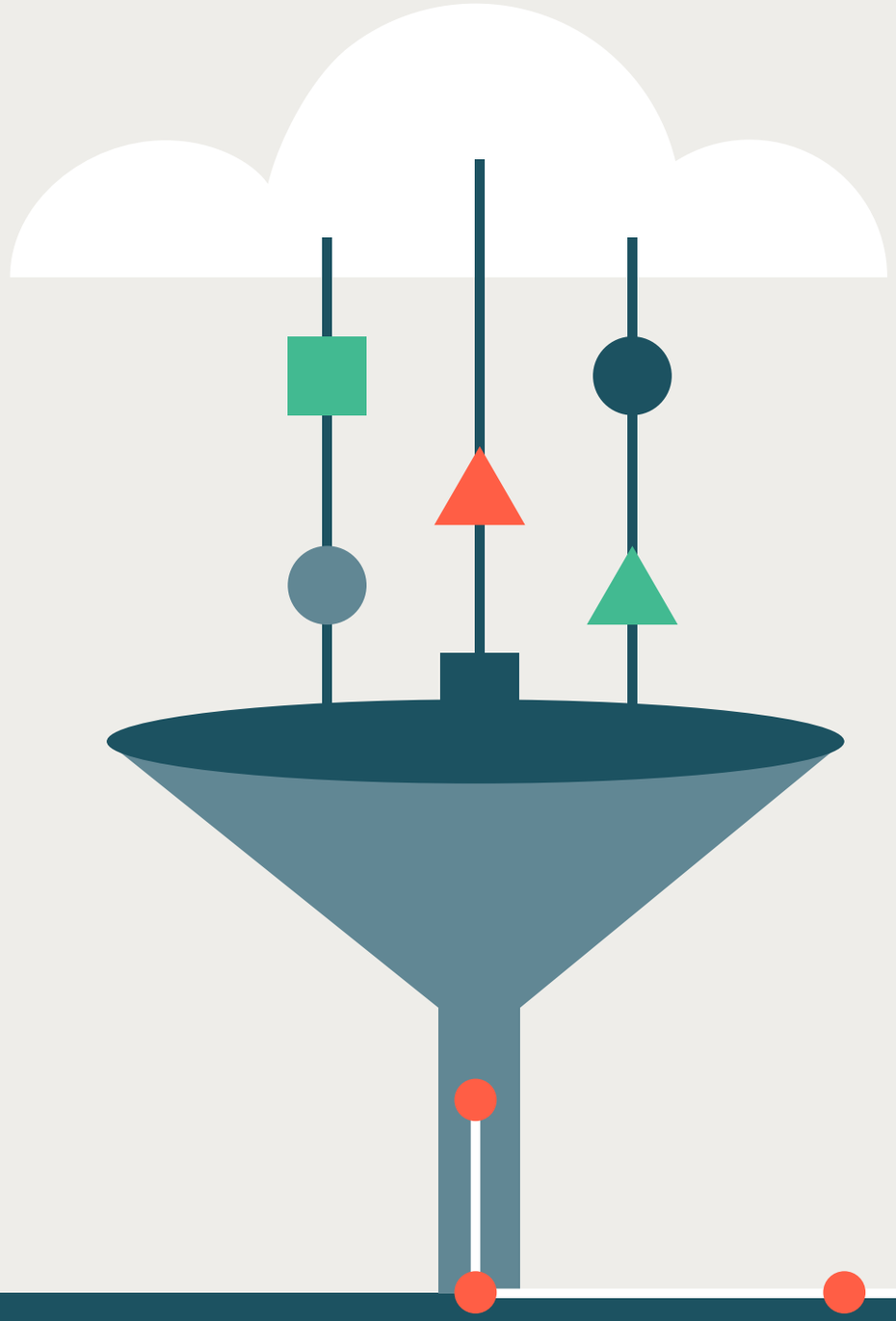
Saurabh Shukla is a Senior Specialist Solutions Architect at Databricks with deep expertise in big data engineering, real-time analytics, cloud computing, data architecture and advanced software development. Saurabh crafts scalable data solutions and leverages advanced technology to solve complex challenges. You can reach out to Saurabh on LinkedIn.

Amit Kara, Director of Technical Marketing at Databricks, is a seasoned expert in data management, combining technical expertise with strategic vision to help organizations realize the full potential of their data. Connect with Amit on LinkedIn.

About Databricks

Databricks is the data and AI company. More than 10,000 organizations worldwide — including Block, Comcast, Condé Nast, Rivian, Shell and over 60% of the Fortune 500 — rely on the Databricks Data Intelligence Platform to take control of their data and put it to work with AI. Databricks is headquartered in San Francisco, with offices around the globe, and was founded by the original creators of Lakehouse, Apache Spark™, Delta Lake and MLflow. To learn more, follow Databricks on LinkedIn, X and Facebook.

TRY DATABRICKS FREE



databricks