**Databricks Documents**

**Introduction to Databricks**

Databricks is a unified analytics platform designed to help organizations harness the power of big data and AI. It was founded by the creators of Apache Spark, an open-source distributed computing system, and it provides a managed environment for running Spark-based applications.

**What is Databricks SQL?**

Databricks SQL is the collection of services that bring data warehousing capabilities and performance to your existing data lakes. Databricks SQL supports open formats and standard ANSI SQL. An in-platform SQL editor and dashboarding tools allow team members to collaborate with other Databricks users directly in the workspace. Databricks SQL also integrates with a variety of tools so that analysts can author queries and dashboards in their favorite environments without adjusting to a new platform.

Databricks is a unified analytics platform that empowers organizations to harness the power of big data and artificial intelligence (AI) through a collaborative and managed environment.

Founding: Founded by the creators of Apache Spark, Databricks leverages Spark's capabilities for distributed computing and data processing.

A long, long time ago database systems ran on a single computer with some associated storage.

• The computers and storage got bigger and faster every year, but the basic architecture remained the same

• If you wanted answers faster, you bought a better computer. If you wanted to store more data, you bought a more expensive storage system

• If the speed you desired or the amount of data you had exceeded the capacity of the best available hardware, you were out of luck; you simply could not create such a database system

• Databricks is an organization and big data processing platform founded by the creators of Apache Spark.

• Azure Databricks is an Apache Spark-based analytics platform optimized for the Microsoft Azure cloud services platform.

• Databricks is integrated with Azure to provide one-click setup, streamlined workflows, and an interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.

**Key features of Databricks include:**

**Unified Analytics Platform:** Databricks provides a collaborative workspace where data scientists, data engineers, and analysts can work together on big data and machine learning projects.

**Managed Spark Cluster:** Databricks manages the underlying Spark cluster, allowing users to focus on their data and analysis without worrying about infrastructure management.

**Data Engineering:** Databricks supports data engineering tasks such as data ingestion, ETL (Extract, Transform, Load) processes, and data integration.

**Data Science and Machine Learning:** The platform includes tools and libraries for building and deploying machine learning models at scale. It integrates with popular machine learning frameworks like TensorFlow, PyTorch, and scikit-learn.

**Collaboration and Sharing:** Databricks provides features for collaboration, version control, and sharing of code and analysis results within teams.

**Scalability:** Databricks are built to scale horizontally, allowing organizations to handle massive amounts of data and computer resources. It can automatically scale up or down based on workload demands, ensuring optimal performance and resource utilization.

**Cost Efficiency:** By providing a managed environment, Databricks helps reduce the operational overhead associated with setting up and maintaining Spark clusters. It also offers cost optimization features such as auto-scaling and instance types of selection, which can help minimize infrastructure costs.

**Security and Compliance:** Databricks includes built-in security features to protect data and infrastructure. This includes encryption at rest and in transit, role-based access control (RBAC), audit logging, and compliance certifications such as SOC 2 Type II and HIPAA.

**Streaming and Real-time Analytics:** Databricks supports real-time data processing and analytics through integration with Apache Spark Streaming and Structured Streaming. This enables organizations to analyze streaming data sources such as IoT devices, social media feeds, and financial transactions in real-time.

**Multi-language Support:** While Databricks is primarily associated with Apache Spark and Python, it also supports other programming languages such as Scala, R, SQL, and Java. This allows users to leverage their existing skills and libraries when working with data on the platform.

**Machine Learning Lifecycle Management:** Databricks provides capabilities for managing the entire machine learning lifecycle, from data preparation and feature engineering to model training, deployment, and monitoring. It includes tools like MLflow for experiment tracking, model versioning, and model serving.

**Community and Ecosystem:** Databricks has a thriving community of users, developers, and contributors who share knowledge, best practices, and code libraries. It also integrates with popular data science and analytics tools, cloud services, and third-party platforms, enhancing its ecosystem and extensibility.

**Industry Applications:** Databricks is used across various industries such as finance, healthcare, retail, telecommunications, and manufacturing for a wide range of applications including fraud detection, customer analytics, recommendation systems, risk management, and supply chain optimization.

It integrates with various data sources, storage systems, and analytics tools, making it easier to work with existing data infrastructure.

Databricks is widely used across industries for various use cases, including data exploration and visualization, predictive analytics, real-time data processing, and large-scale machine learning. It's particularly popular among organizations dealing with large volumes of data and complex analytics requirements.

Overall, Databricks provides a comprehensive platform for organizations to unlock the value of their data, accelerate innovation, and drive business outcomes through advanced analytics and AI.

**Prerequisites:**

• The pre-requisites for Databricks course include a basic understanding of Computer Programming Languages E.g. Python, Scala.

• Fundamentals of Data Analysis practiced over any of the data analysis tools like R will be a plus.

• Spark Concepts (RDD, Transformations, Actions, Join Operation, File Formats, Read/Write to AWS S3, partition)

• Azure/AWS (S3, Redshift, Azure Blob Storage, Azure Data Lake Storage, Azure SQL Data warehouse).

• Hadoop/Hive Optional: (Airflow, Docker, Datawarehouse concepts)

• However, you will be provided with complimentary study material for “Basics of Bigdata” as a self-paced course once you enroll for the course.

**Sign up for Azure free Trial**

1. Go to Create your Azure free account today.

2. Click Start free.

3. Choose a credential associated with a Microsoft account or click Use another account.

4. Provide a credential and click Next.

5. Enter a password and click Sign in.

6. Fill in the Email address and Phone fields and click Next.

7. Fill in the Phone number field and choose an identity verification method.

8. Complete the identify verification process.

9. Provide credit card details and click Next.

10. Check the Agreement checkbox and click Sign up.

**Create an Azure Databricks workspace**

1. Log in to the Azure portal.

2. Select + Create a resource > Analytics > Azure Databricks.

3. In the Azure Databricks Service dialog, provide the workspace configuration.

• **Workspace name:** Enter a name for your Azure Databricks workspace. • Subscription: Select your Azure subscription.

• **Resource group:** Create a new resource group or use an existing one. • Location: Select a geographical region.

• **Pricing Tier:** Select a pricing tier. If you select Trial (Premium - 14-Days Free DBUs), the workspace has access to free Premium Azure Databricks DBUs for 14 days (about 2 weeks).

4. Select Pin to dashboard and then click Create. The portal displays “Deployment in progress”. After a few minutes the Azure Databricks Service page displays.

Launch the workspace 1. In the Azure Databricks Service page, click Launch Workspace. You’ll see the Azure Databricks workspace home page:

Introduction to Databricks UI From the sidebar at the left and the Common Tasks list on the home page, you access fundamental Azure Databricks entities: Workspace, clusters, tables, notebooks, jobs, and libraries. The Workspace is the special root folder that stores your Azure Databricks assets, such as notebooks and libraries, and the data that you import.

Purpose of this Exam Guide The purpose of this exam guide is to give you an overview of the exam and what is covered on the exam to help you determine your exam readiness.

This document will be updated anytime there are any changes to an exam (and when those changes take effect on an exam) so that you can be prepared.

This version covers the currently live exam as of April 1, 2024. Please check back two weeks before you take your exam to make sure you have the most current version.

Audience Description the Databricks Certified Data Engineer Associate certification exam assesses an individual’s ability to use the Databricks Lakehouse Platform to complete introductory data engineering tasks.

This includes an understanding of the Lakehouse Platform and its workspace, its architecture, and its capabilities. It also assesses the ability to perform multi-hop architecture ETL tasks using Apache Spark SQL and Python in both batch and incrementally processed paradigms. Finally, the exam assesses the tester’s ability to put basic ETL pipelines and Databricks SQL queries and dashboards into production while maintaining entity permissions. Individuals who pass this certification exam can be expected to complete basic data engineering tasks using Databricks and its associated tool.

**Hadoop**

• In 2006, Doug Cutting et al at Yahoo created Hadoop

• Unlimited horizontal scaling on cheap computers!

• Key ideas…

• HDFS, all disks became one file system

• MapReduce, a way to run parallel code on all the CPUs

• Soon there were Hadoop clusters with 100s of nodes, then 1000s

• You could do database things that were simply impossible before!

**Hadoop Drawbacks**

• MapReduce is hard to program

• A new way of thinking about coding

• Does not magically parallelize algorithms for you

• Requires trial/error, tuning, multiple stages of MR

• Hadoop wrote MR intermediate results to disk

• Often many times for one job

• Much slower than a memory write/read

• Hadoop was a “batch” system, not interactive queries

**Hive**

• Introduced in 2009, it solved the "hard to program" problem

• SQL abstraction on top of HDFS and MapReduce

• Data appears as normal relational-like tables

• Database jobs can be written in SQL

• Essentially a compiler that translates SQL into Java MapReduce code

• Generated code usually better than human would create

• But still… lots of disk I/O

• Simple Hive query on small table = ~15 secs

Spark

• In 2011, Spark project to solve Hadoop disk I/O problem

• Goal: do as many operations as possible completely within memory

• Spark delivered 10 – 100x speedup on fewer machines.

**Spark, Issues**

• Complexity

• Software installs

• Hardware clusters

• File system setup

• Performance tuning

• Security

• Clusters

• Code / Jobs

• Data

• Enter Databricks (2015)

**Databricks**

• Databricks is a way to use Spark more conveniently

• Databricks is Spark, but with a GUI and many automated features

• Creation and configuration of server clusters

• Auto-scaling and shutdown of clusters

• Connections to various file systems and formats

• Programming interfaces for Python, Scala, SQL, R

• Integration with other Azure services

• Available only as a cloud service, both Amazon and Azure

**Databricks Connections**

• Getting data in

• CSV, JSON, Parquet, LZO, Zip, Avro

• Hive tables

• Azure Blob or Data Lake as DBFS directory

• Any RDBMS with JDBC

• Azure Data Hub, which has many sources connectors

• Getting data out • Write to many files formats

• JBDC and ODBC for programmatic inbound reads

• REST API

• Clusters, DBFS, jobs, libraries, workspaces

**Databricks Goodies**

• Databricks Delta, ACID compliant transactions

• Security integration with Azure Active Directory

• See my article on LinkedIn for details

**Graph Frames**

• A library of routines for creating and calculating node/edge data structures

• Ex: shortest path, PageRank

• Machine Learning

• A library and workflow for many common ML techniques

• Support for many third-party ML libs – H2O, scikit-learn, DataRobot, XGBoost

• R language.

**Unified Analytics Platform**

Collaboration: Databricks provides a unified workspace where data scientists, data engineers, and analysts can collaborate on big data and machine learning projects seamlessly.

Workspaces: Users can access shared notebooks, dashboards, and data sets, facilitating collaboration and knowledge sharing within teams.

**Managed Spark Cluster**

Infrastructure Management: Databricks abstracts away the complexities of managing Spark clusters, allowing users to focus on data analysis and insights without worrying about infrastructure provisioning or maintenance.

Autoscaling: The platform automatically scales the underlying compute resources up or down based on workload demands, optimizing resource utilization and performance.

**Data Engineering Capabilities**

Data Ingestion: Databricks supports various data ingestion methods, including batch and streaming data sources, enabling organizations to ingest data from multiple sources seamlessly.

**ETL Processes:** Users can perform Extract, Transform, Load (ETL) processes within Databricks, leveraging Spark's capabilities for data manipulation and transformation.

**Data Integration:** The platform integrates with popular data storage systems and services such as Amazon S3, Azure Data Lake Storage, and Google Cloud Storage, facilitating data integration and interoperability.

**Data Science and Machine Learning**

Machine Learning Libraries: Databricks provides access to a wide range of machine learning libraries and frameworks, including TensorFlow, PyTorch, scikit-learn, and MLlib, enabling users to build and deploy machine learning models at scale.

**Model Deployment:** Users can deploy machine learning models trained within Databricks to production environments seamlessly, leveraging built-in deployment tools and integrations with containerization platforms.

Collaboration and Sharing

**Version Control:** Databricks offers version control capabilities, allowing users to track changes to notebooks, code, and models over time.

Sharing Insights: Users can share insights, visualizations, and analysis results with stakeholders through built-in sharing features, enhancing collaboration and communication within organizations.

**Security and Compliance**

Data Encryption: Databricks encrypts data at rest and in transit, ensuring data security and compliance with industry regulations and standards.

Role-Based Access Control (RBAC): The platform offers RBAC capabilities, allowing organizations to enforce fine-grained access controls and permissions based on user roles and responsibilities.

**Scalability and Cost Efficiency**

**Horizontal Scaling:** Databricks is designed to scale horizontally, enabling organizations to handle massive amounts of data and compute resources efficiently.

**Cost Optimization:** The platform offers cost optimization features such as autoscaling and instance type selection, helping organizations minimize infrastructure costs and optimize resource utilization.

**Real-time Analytics and Streaming**

Real-time Processing: Databricks supports real-time data processing and analytics through integration with Apache Spark Streaming and Structured Streaming, enabling organizations to analyze streaming data sources in real-time.

Use Cases: Real-time analytics use cases include IoT data processing, fraud detection, social media analytics, and monitoring systems.

**Multi-language Support**

Programming Languages: Databricks supports multiple programming languages, including Python, Scala, R, SQL, and Java, allowing users to leverage their existing skills and libraries when working with data on the platform.

Language Interoperability: Users can seamlessly integrate code written in different languages within Databricks notebooks and workflows, enhancing flexibility and productivity.

**Machine Learning Lifecycle Management**

MLflow Integration: Databricks integrates with MLflow, an open-source platform for managing the machine learning lifecycle, enabling organizations to track experiments, collaborate on model development, and deploy models to production environments seamlessly.

Model Versioning: MLflow provides capabilities for versioning machine learning models, enabling organizations to track model changes and reproduce experiments reliably.

**Community and Ecosystem**

Thriving Community: Databricks has a vibrant community of users, developers, and contributors who share knowledge, best practices, and code libraries through forums, meetups, and online communities.

Ecosystem Integrations: Databricks integrates with a wide range of data science and analytics tools, cloud services, and third-party platforms, enhancing its ecosystem and extensibility.

**Industry Applications**

**Finance:** Use cases in finance include risk management, fraud detection, algorithmic trading, and customer segmentation.

**Healthcare:** Healthcare applications include predictive analytics, patient monitoring, disease outbreak detection, and personalized medicine.

**Retail:** Retail use cases include demand forecasting, customer segmentation, recommendation systems, and supply chain optimization.

**Telecommunications:** Telecommunications applications include network optimization, customer churn prediction, fraud detection, and service quality monitoring.

**Manufacturing:** Manufacturing use cases include predictive maintenance, quality control, supply chain optimization, and production optimization.

# **Discover data**

Databricks provides a suite of tools and products that simplify the discovery of data assets that are accessible through the Databricks Data Intelligence Platform. This article provides an opinionated overview of how you can discover and preview data that has already been configured for access in your workspace.

* To connect to data sources, see [Connect to data sources](https://docs.databricks.com/en/connect/index.html).
* For information about gaining access to data in the Databricks Marketplace, see What is Databricks Marketplace.

Topics in this section focus on exploring data objects and data files. If you’re looking for information about working with assets such as notebooks, SQL queries, libraries, and models, see [Navigate the workspace](https://docs.databricks.com/en/workspace/index.html).

If you’re seeking guidance around generating summary statistics for datasets or other tasks associated with exploratory data analysis (EDA), see [Exploratory data analysis on Databricks: Tools and techniques](https://docs.databricks.com/en/exploratory-data-analysis/index.html).

**How can you discover data assets?**

Data discovery tools on Databricks fall into the following general categories:

* AI-assisted insights, summary, and search.
* Keyword search.
* Catalog exploration using the UI.
* Programmatic listing and metadata exploration.

Data discovery tools are optimized for data governed by Unity Catalog. Data assets that have not been registered as Unity Catalog objects might not be discoverable using some of these approaches.

**Find data using the UI**

Catalog Explorer provides tools for exploring and governing data assets. You access Catalog Explorer using the **Catalog** in the workspace sidebar. See [What is Catalog Explorer?](https://docs.databricks.com/en/catalog-explorer/index.html).

Notebooks and the SQL query editor also provide a catalog navigator for exploring database objects. Click the **Catalog** icon in these interfaces to expand or collapse the catalog navigator without leaving from your code editor.

Once you’ve discovered a dataset of interest, you can use the **Insights** tab to learn how the data is being used in your workspace. See [View frequent queries and users of a table](https://docs.databricks.com/en/discover/table-insights.html).

**Explore data programmatically**

You can use the SHOW command on all database objects to discover assets registered to Unity Catalog. Use the LIST command, the %fs magic command, or Databricks Utilities to list files.

See [Explore storage and find data files](https://docs.databricks.com/en/discover/files.html) and [Explore database objects](https://docs.databricks.com/en/discover/database-objects.html).

## **Review data comments**

You can review comments to learn about the contents of datasets available in your lakehouse. Comments can be set on data objects including catalogs, schemas, tables, and columns. You can view comments in Catalog Explorer or using the DESCRIBE command for an object.

Catalog Explorer can provide AI-generated comments for tables, which makes it easy for data asset owners to provide a rich overview of datasets. See [Add AI-generated comments to a table](https://docs.databricks.com/en/catalog-explorer/ai-comments.html).

Users can also optionally provide comments on tables and other database objects using markdown, which is rendered in Catalog Explorer. See [Document data in Catalog Explorer using markdown comments](https://docs.databricks.com/en/catalog-explorer/markdown-data-comments.html).

**Search for tables in your lakehouse**

You can use the search bar in Databricks to find tables registered to Unity Catalog. You can either perform a keyword search or use semantic search to find datasets or columns that relate to your search query. Search only returns results for tables that you have permission to see. Search reviews table names, column names, table comments, and column comments. See [Search for workspace objects](https://docs.databricks.com/en/search/index.html).

Was this article helpful?

**Databricks data engineering**

Databricks data engineering features are a robust environment for collaboration among data scientists, data engineers, and data analysts. Data engineering tasks are also the backbone of Databricks machine learning solutions.

Note:If you are a data analyst who works primarily with SQL queries and BI tools, you might prefer Databricks SQL.

The data engineering documentation provides how-to guidance to help you get the most out of the Databricks collaborative analytics platform. For getting started tutorials and introductory information, see Get started: Account and workspace setup and What is Databricks?.

**Delta Live Tables**

Learn how to build data pipelines for ingestion and transformation with Databricks Delta Live Tables.

**Structured Streaming**

Learn about streaming, incremental, and real-time workloads powered by Structured Streaming on Databricks.

**Apache Spark**

Learn how Apache Spark works on Databricks and the Databricks platform.

**Compute**

Learn about the types of Databricks compute available in your workspace.

**Notebooks**

Learn what a Databricks notebook is, and how to use and manage notebooks to process, analyze, and visualize your data.

**Workflows**

Learn how to orchestrate data processing, machine learning, and data analysis workflows on the Databricks Data Intelligence Platform.

**Libraries**

Learn how to make third-party or custom code available in Databricks using libraries. Learn about the different modes for installing libraries on Databricks.

Init scripts

Learn how to use initialization (init) scripts to install packages and libraries, set system properties and environment variables, modify Apache Spark config parameters, and set other configurations on Databricks clusters.

**Git folders**

Learn how to use Git to version control your notebooks and other files for development in Databricks.

**DBFS**

Learn about Databricks File System (DBFS), a distributed file system mounted into a Databricks workspace and available on Databricks clusters

**Files**

Learn about options for working with files on Databricks.

Migration

Learn how to migrate data applications such as ETL jobs, enterprise data warehouses, ML, data science, and analytics to Databricks.

Optimization & performance

Learn about optimizations and performance recommendations on Databricks.

Generative AI and large language models (LLMs) on Databricks

This article provides an overview of generative AI on Databricks and includes links to example notebooks and demos.

**What is generative AI?**

Generative AI is a type of artificial intelligence focused on the ability of computers to use models to create content like images, text, code, and synthetic data.

Generative AI applications are built on top of large language models (LLMs) and foundation models.

LLMs are deep learning models that consume and train on massive datasets to excel in language processing tasks. They create new combinations of text that mimic natural language based on its training data.

Foundation models are large ML models pre-trained to be fine-tuned for more specific language understanding and generation tasks. These models are utilized to discern patterns within the input data.

After these models have completed their learning processes, together they generate statistically probable outputs when prompted and they can be employed to accomplish various tasks, including:

Image generation based on existing ones or utilizing the style of one image to modify or create a new one.

Speech tasks such as transcription, translation, question/answer generation, and interpretation of the intent or meaning of text.

Important While many LLMs or other generative AI models have safeguards, they can still generate harmful or inaccurate information.

Generative AI has the following design patterns:

Prompt Engineering: Crafting specialized prompts to guide LLM behavior

Retrieval Augmented Generation (RAG): Combining an LLM with external knowledge retrieval

Fine-tuning: Adapting a pre-trained LLM to specific data sets of domains

Pre-training: Training an LLM from scratch

Develop generative AI and LLMs on Databricks

Databricks unifies the AI lifecycle from data collection and preparation, to model development and LLMOps, to serving and monitoring. The following features are specifically optimized to facilitate the development of generative AI applications:

Unity Catalog for governance, discovery, versioning, and access control for data, features, models, and functions.

MLflow for model development tracking and LLM evaluation.

Feature engineering and serving.

Databricks Model Serving for deploying LLMs. You can configure a model serving endpoint specifically for accessing foundation models:

State-of-the-art open LLMs using Foundation Model APIs

Third-party models hosted outside of Databricks. See External models in Databricks Model Serving.

Databricks Vector Search provides a queryable vector database that stores embedding vectors and can be configured to automatically sync to your knowledge base.

Lakehouse Monitoring for data monitoring and tracking model prediction quality and drift using automatic payload logging with inference tables.

AI Playground for testing foundation models from your Databricks workspace. You can prompt, compare and adjust settings such as system prompt and inference parameters.

Additional resources

See Retrieval Augmented Generation (RAG) on Databricks.

See Build a Q&A chatbot with LLama2 and Databricks.

For information about using Hugging Face models on Databricks, see Hugging Face Transformers.

The databricks-ml-examples repo in Github contains example implementations of state-of-the-art (SOTA) LLMs.

AI and Machine Learning on Databricks

March 18, 2024

This article describes the tools that Databricks provides to help you build and monitor AI and ML workflows. The diagram shows how these components work together to help you implement your model development and deployment process.

**Why use Databricks for machine learning and deep learning**?

With Databricks, you can implement the full ML lifecycle on a single platform with end-to-end governance throughout the ML pipeline. Databricks includes the following built-in tools to support ML workflows:

Unity Catalog for governance, discovery, versioning, and access control for data, features, models, and functions.

Lakehouse Monitoring for data monitoring.

Feature engineering and serving.

Support for the model lifecycle:

Databricks AutoML for automated model training.

MLflow for model development tracking.

Unity Catalog for model management.

Databricks Model Serving for high-availability, low-latency model serving. This includes deploying LLMs using:

Foundation Model APIs which allow you to access and query state-of-the-art open models from a serving endpoint.

External models which allow you to access models hosted outside of Databricks.

Lakehouse Monitoring to track model prediction quality and drift.

Databricks Workflows for automated workflows and production-ready ETL pipelines.

Databricks Git folders for code management and Git integration.

**Deep learning on Databricks**

Configuring infrastructure for deep learning applications can be difficult.

Databricks Runtime for Machine Learning takes care of that for you, with clusters that have built-in compatible versions of the most common deep learning libraries like TensorFlow, PyTorch, and Keras, and supporting libraries such as Petastorm, Hyperopt, and Horovod. Databricks Runtime ML clusters also include pre-configured GPU support with drivers and supporting libraries. It also supports libraries like Ray to parallelize compute processing for scaling ML workflows and AI applications.

Databricks Runtime ML clusters also include pre-configured GPU support with drivers and supporting libraries. Databricks Model Serving enables creation of scalable GPU endpoints for deep learning models with no extra configuration.

For machine learning applications, Databricks recommends using a cluster running Databricks Runtime for Machine Learning. See Create a cluster using Databricks Runtime ML.

To get started with deep learning on Databricks, see:

Best practices for deep learning on Databricks

Deep learning on Databricks

Reference solutions for deep learning

**Large language models (LLMs) and generative AI on Databricks**

Databricks Runtime for Machine Learning includes libraries like Hugging Face Transformers and LangChain that allow you to integrate existing pre-trained models or other open-source libraries into your workflow. The Databricks MLflow integration makes it easy to use the MLflow tracking service with transformer pipelines, models, and processing components. In addition, you can integrate OpenAI models or solutions from partners like John Snow Labs in your Databricks workflows.

With Databricks, you can customize a LLM on your data for your specific task. With the support of open source tooling, such as Hugging Face and DeepSpeed, you can efficiently take a foundation LLM and train it with your own data to improve its accuracy for your specific domain and workload. You can then leverage the custom LLM in your generative AI applications.

In addition, Databricks provides Foundation Model APIs and external models which allows you to access and query state-of-the-art open models from a serving endpoint. Using Foundation Model APIs, developers can quickly and easily build applications that leverage a high-quality generative AI model without maintaining their own model deployment.

For SQL users, Databricks provides AI functions that SQL data analysts can use to access LLM models, including from OpenAI, directly within their data pipelines and workflows. See AI Functions on Databricks.

**Databricks Runtime for Machine Learning**

Databricks Runtime for Machine Learning (Databricks Runtime ML) automates the creation of a cluster with pre-built machine learning and deep learning infrastructure including the most common ML and DL libraries. For the full list of libraries in each version of Databricks Runtime ML, see the release notes.

To access data in Unity Catalog for machine learning workflows, the access mode for the cluster must be single user (assigned). Shared clusters are not compatible with Databricks Runtime for Machine Learning. In addition, Databricks Runtime ML is not supported on TableACLs clusters or clusters with spark.databricks.pyspark.enableProcessIsolation config set to true.

Create a cluster using Databricks Runtime ML

When you create a cluster, select a Databricks Runtime ML version from the Databricks runtime version drop-down menu. Both CPU and GPU-enabled ML runtimes are available.

If you select a cluster from the drop-down menu in the notebook, the Databricks Runtime version appears at the right of the cluster name:

If you select a GPU-enabled ML runtime, you are prompted to select a compatible Driver type and Worker type. Incompatible instance types are grayed out in the drop-down menu. GPU-enabled instance types are listed under the GPU accelerated label.

Note

To access data in Unity Catalog for machine learning workflows, the access mode for the cluster must be single user (assigned). Shared clusters are not compatible with Databricks Runtime for Machine Learning.

Libraries included in Databricks Runtime ML

Databricks Runtime ML includes a variety of popular ML libraries. The libraries are updated with each release to include new features and fixes.

Databricks has designated a subset of the supported libraries as top-tier libraries. For these libraries, Databricks provides a faster update cadence, updating to the latest package releases with each runtime release (barring dependency conflicts). Databricks also provides advanced support, testing, and embedded optimizations for top-tier libraries.

For a full list of top-tier and other provided libraries, see the release notes for Databricks Runtime ML.

Next steps

To get started, see:

Tutorials: Get started with ML

For a recommended MLOps workflow on Databricks Machine Learning, see:

MLOps workflows on Databricks

To learn about key Databricks Machine Learning features, see:

What is AutoML?

What is a feature store?

Model serving with Databricks

Lakehouse Monitoring

Manage model lifecycle

MLflow experiment tracking

Model serving with Databricks

April 03, 2024

This article describes Databricks Model Serving, including its advantages and limitations.

**What is Model Serving?**

Databricks Model Serving provides a unified interface to deploy, govern, and query AI models. Each model you serve is available as a REST API that you can integrate into your web or client application.

Model Serving provides a highly available and low-latency service for deploying models. The service automatically scales up or down to meet demand changes, saving infrastructure costs while optimizing latency performance. This functionality uses serverless compute. See the Model Serving pricing page for more details.

Model serving supports serving:

Custom models. These are Python models packaged in the MLflow format. They can be registered either in Unity Catalog or in the workspace model registry. Examples include scikit-learn, XGBoost, PyTorch, and Hugging Face transformer models.

State-of-the-art open models made available by Foundation Model APIs. These models are curated foundation model architectures that support optimized inference. Base models, like Llama-2-70B-chat, BGE-Large, and Mistral-7B are available for immediate use with pay-per-token pricing, and workloads that require performance guarantees and fine-tuned model variants can be deployed with provisioned throughput.

External models. These are models that are hosted outside of Databricks. Endpoints that serve external models can be centrally governed and customers can establish rate limits and access control for them. Examples include foundation models like, OpenAI’s GPT-4, Anthropic’s Claude, and others.

Note

You can interact with supported large language models using the AI Playground. The AI Playground is a chat-like environment where you can test, prompt, and compare LLMs. This functionality is available in your Databricks workspace.

Model serving offers a unified REST API and MLflow Deployment API for CRUD and querying tasks. In addition, it provides a single UI to manage all your models and their respective serving endpoints. You can also access models directly from SQL using AI functions for easy integration into analytics workflows.

For an introductory tutorial on how to serve custom models on Databricks, see Tutorial: Deploy and query a custom model.

For a getting started tutorial on how to query a foundation model on Databricks, see Get started querying LLMs on Databricks.

**Why use Model Serving?**

Deploy and query any models: Model Serving provides a unified interface that so you can manage all models in one location and query them with a single API, regardless of whether they are hosted on Databricks or externally. This approach simplifies the process of experimenting with, customizing, and deploying models in production across various clouds and providers.

Securely customize models with your private data: Built on a Data Intelligence Platform, Model Serving simplifies the integration of features and embeddings into models through native integration with the Databricks Feature Store and Databricks Vector Search. For even more improved accuracy and contextual understanding, models can be fine-tuned with proprietary data and deployed effortlessly on Model Serving.

Govern and monitor models: The Serving UI allows you to centrally manage all model endpoints in one place, including those that are externally hosted. You can manage permissions, track and set usage limits, and monitor the quality of all types of models. This enables you to democratize access to SaaS and open LLMs within your organization while ensuring appropriate guardrails are in place.

Reduce cost with optimized inference and fast scaling: Databricks has implemented a range of optimizations to ensure you get the best throughput and latency for large models. The endpoints automatically scale up or down to meet demand changes, saving infrastructure costs while optimizing latency performance.

Bring reliability and security to Model Serving: Model Serving is designed for high-availability, low-latency production use and can support over 25K queries per second with an overhead latency of less than 50 ms. The serving workloads are protected by multiple layers of security, ensuring a secure and reliable environment for even the most sensitive tasks.

Requirements

Registered model in Unity Catalog or the Workspace Model Registry.

Permissions on the registered models as described in Serving endpoint ACLs.

MLflow 1.29 or higher

Enable Model Serving for your workspace

To use Model Serving, your account admin must read and accept the terms and conditions for enabling serverless compute in the account console.

Note

If your account was created after March 28, 2022, serverless compute is enabled by default for your workspaces.

If you are not an account admin, you cannot perform these steps. Contact an account admin if your workspace needs access to serverless compute.

As an account admin, go to the feature enablement tab of the account console settings page.

A banner at the top of the page prompts you to accept the additional terms. Once you read the terms, click Accept. If you do not see the banner asking you to accept the terms, this step has been completed already.

After you’ve accepted the terms, your account is enabled for serverless.

No additional steps are required to enable Model Serving in your workspace.

Limitations and region availability

Databricks Model Serving imposes default limits to ensure reliable performance. See Model Serving limits and regions. If you have feedback on these limits or an endpoint in an unsupported region, reach out to your Databricks account team.

Data protection in Model Serving

Databricks takes data security seriously. Databricks understands the importance of the data you analyze using Databricks Model Serving, and implements the following security controls to protect your data.

Every customer request to Model Serving is logically isolated, authenticated, and authorized.

Databricks Model Serving encrypts all data at rest (AES-256) and in transit (TLS 1.2+).

For all paid accounts, Databricks Model Serving does not use user inputs submitted to the service or outputs from the service to train any models or improve any Databricks services.

For Databricks Foundation Model APIs, as part of providing the service, Databricks may temporarily process and store inputs and outputs to prevent, detect, and mitigate abuse or harmful uses. Your inputs and outputs are isolated from those of other customers, stored in the same region as your workspace for up to thirty (30) days, and only accessible for detecting and responding to security or abuse concerns.

Additional resources

Get started querying LLMs on Databricks.

Tutorial: Deploy and query a custom model

Deploy generative AI foundation models

Deploy custom models.

Migrate to Model Serving

Migrate optimized LLM serving endpoints to provisioned throughput

**What is data warehousing on Databricks?**

Data warehousing refers to collecting and storing data from multiple sources so it can be quickly accessed for business insights and reporting. This article contains key concepts for building a data warehouse in your data lakehouse.

The lakehouse architecture and Databricks SQL bring cloud data warehousing capabilities to your data lakes. Using familiar data structures, relations, and management tools, you can model a highly-performant, cost-effective data warehouse that runs directly on your data lake. For more information, see

**What is a data lakehouse?**

As with a traditional data warehouse, you model data according to business requirements and then serve it to your end users for analytics and reports. Unlike a traditional data warehouse, you can avoid siloing your business analytics data or creating redundant copies that quickly become stale.

Building a data warehouse inside your lakehouse lets you bring all your data into a single system and lets you take advantage of features such as Unity Catalog and Delta Lake.

Unity Catalog adds a unified governance model so that you can secure and audit data access and provide lineage information on downstream tables. Delta Lake adds ACID transactions and schema evolution, among other powerful tools for keeping your data reliable, scalable, and high-quality.

Databricks SQL provides general compute resources that are executed against the tables in the lakehouse. Databricks SQL is powered by SQL warehouses, offering scalable SQL compute resources decoupled from storage.

What is a SQL warehouse? for more information on SQL Warehouse defaults and options.

Databricks SQL integrates with Unity Catalog so that you can discover, audit, and govern data assets from one place. To learn more, see What is Unity Catalog

**Data modeling on Databricks**

A lakehouse supports a variety of modeling styles. The following image shows how data is curated and modeled as it moves through different layers of a lakehouse.

**Medallion architecture**

The medallion architecture is a data design pattern that describes a series of incrementally refined data layers that provide a basic structure in the lakehouse. The bronze, silver, and gold layers signify increasing data quality at each level, with gold representing the highest quality. For more information, see What is the medallion lakehouse architecture?.

Inside a lakehouse, each layer can contain one or more tables. The data warehouse is modeled at the silver layer and feeds specialized data marts in the gold layer.

**Bronze layer**

Data can enter your lakehouse in any format and through any combination of batch or steaming transactions. The bronze layer provides the landing space for all of your raw data in its original format. That data is converted to Delta tables.

**Silver layer**

The silver layer brings the data from different sources together. For the part of the business that focuses on data science and machine learning applications, this is where you start to curate meaningful data assets. This process is often marked by a focus on speed and agility.

The silver layer is also where you can carefully integrate data from disparate sources to build a data warehouse in alignment with your existing business processes. Often, this data follows a Third Normal Form (3NF) or Data Vault model. Specifying primary and foreign key constraints allows end users to understand table relationships when using Unity Catalog. Your data warehouse should serve as the single source of truth for your data marts.

The data warehouse itself is schema-on-write and atomic. It is optimized for change, so you can quickly modify the data warehouse to match your current needs when your business processes change or evolve.

**Gold layer**

The gold layer is the presentation layer, which can contain one or more data marts. Frequently, data marts are dimensional models in the form of a set of related tables that capture a specific business perspective.

The gold layer also houses departmental and data science sandboxes to enable self-service analytics and data science across the enterprise. Providing these sandboxes and their own separate compute clusters prevents the Business teams from creating copies of data outside the lakehouse.