



PLURALSIGHT

Data Engineering



Tech Catalyst Bootcamp



Tarek Atwan
Instructor, Pluralsight

Proprietary and confidential

 PLURALSIGHT

STANDUP

What are some of the challenges you faced yesterday in the lab.

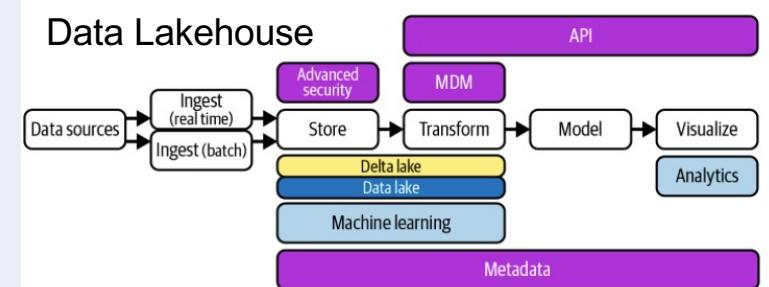
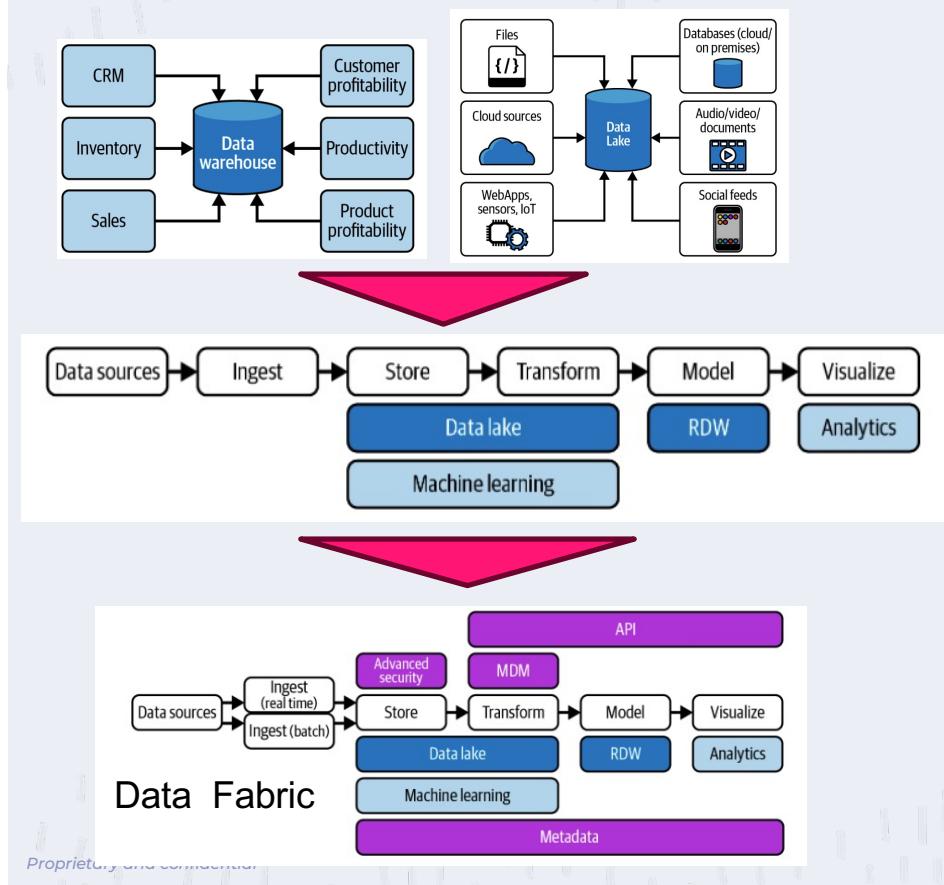
How did you resolve it?

What did you learn from it?

What was something “new” or interesting that you learned from the lab yesterday?

Quick Review

Evolution Summary



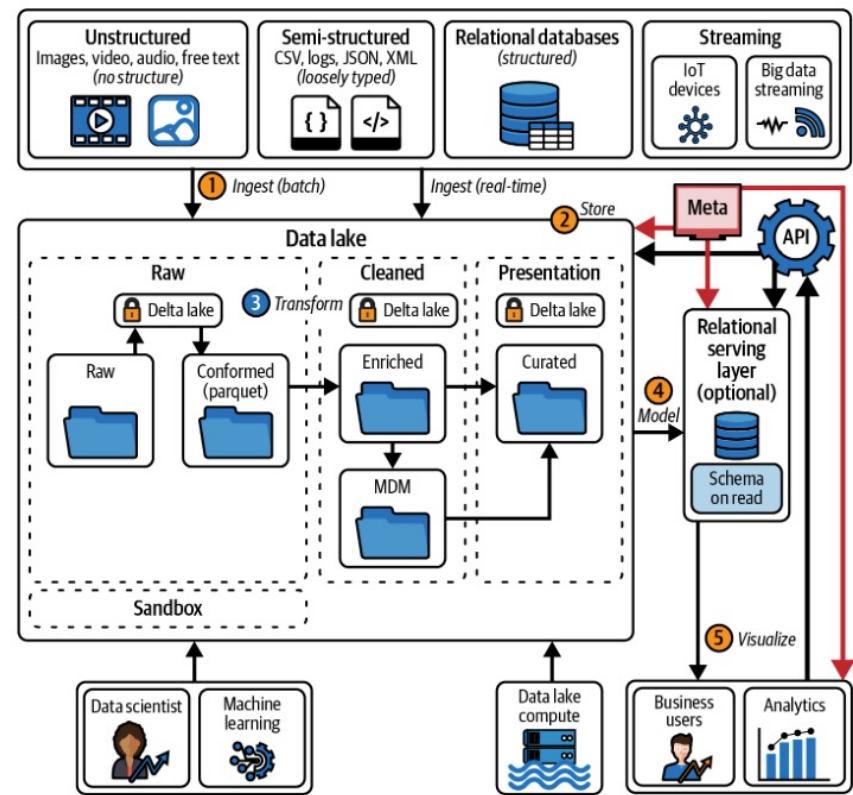
Data Lakehouse Architecture

It adds

- Transaction Support (ACID)
- Time Travel
- Caching
- Query optimization

Drawback

- Can be resource intensive to build this



Databases



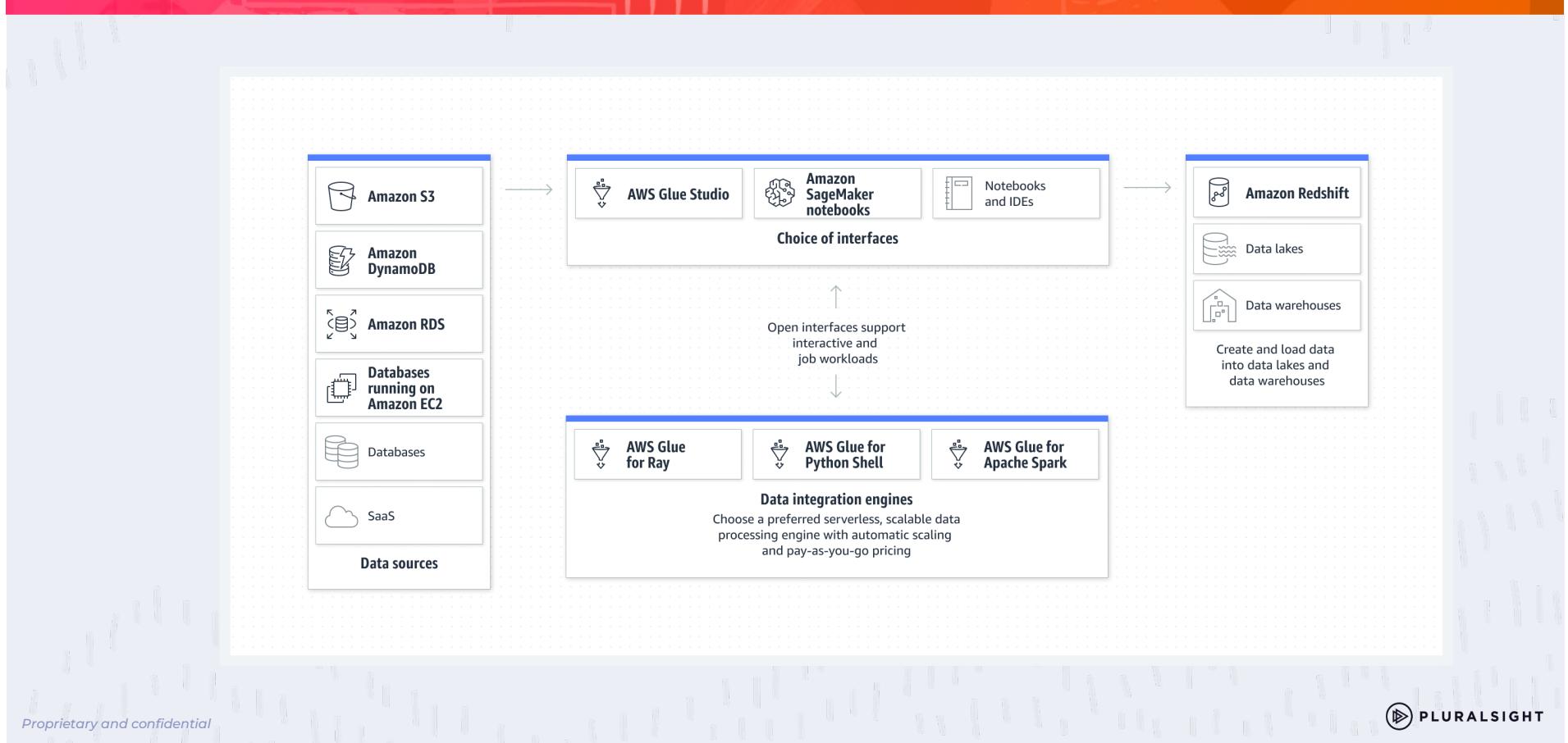
<https://docs.aws.amazon.com/whitepapers/latest/aws-overview/database.html>

AWS Analytics Services

Proprietary and confidential



AWS Glue



AWS Glue

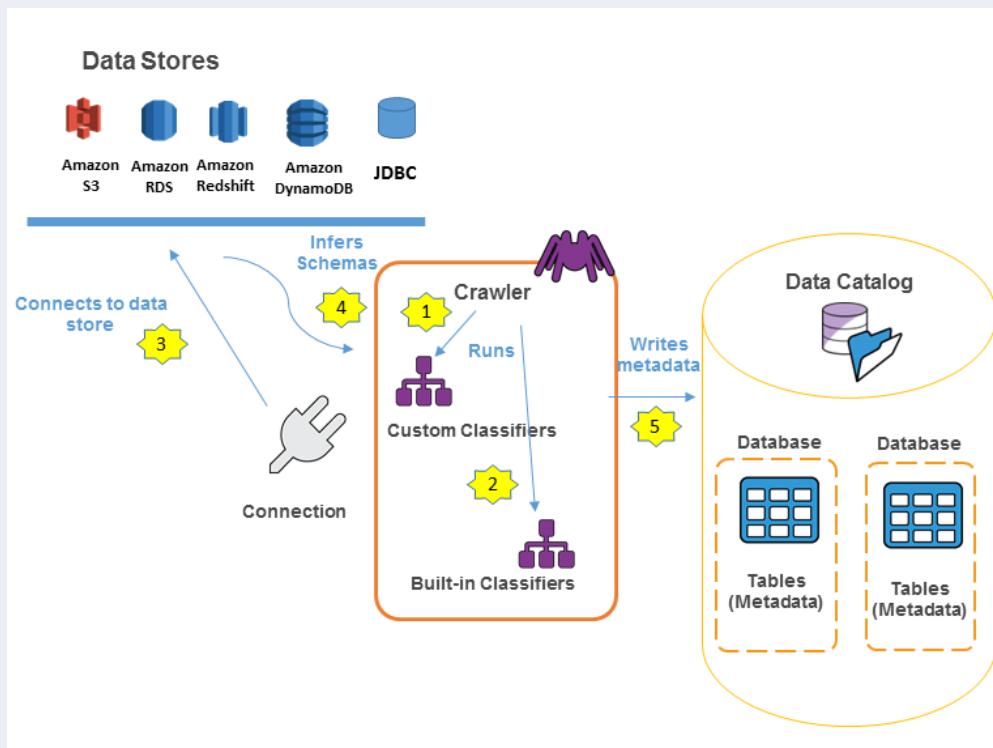
Serverless discovery and definition of table definitions and schema

- S3 “data lakes”
- RDS
- Redshift
- DynamoDB
- Most other SQL databases

Custom ETL jobs

- Trigger-driven, on a schedule, or on demand
- Fully managed

Glue Crawler / Data Catalog



Glue Crawler / Data Catalog

- Glue crawler scans data in S3, creates schema
- Can run periodically
- Populates the Glue Data Catalog
 - Stores only table definition
 - Original data stays in S3
- Once cataloged, you can treat your unstructured data like it's structured
 - Redshift Spectrum
 - Athena
 - EMR
 - Quicksight



Glue ETL

- Automatic code generation (Scala or Python)
- Encryption
 - Server-side (at rest)
 - SSL (in transit)
- Can be event-driven
- Can provision additional “DPU’s” (data processing units) to increase performance of underlying Spark jobs
 - Enabling job metrics can help you understand the maximum capacity in DPU’s you need
- Errors reported to CloudWatch
 - Could tie into SNS for notification

Glue ETL

- Transform data, Clean Data, Enrich Data (before doing analysis)
 - Generate ETL code in Python or Scala, you can modify the code
 - Can provide your own Spark or PySpark scripts
 - Target can be S3, JDBC (RDS, Redshift), or in Glue Data Catalog
- Fully managed, cost effective, pay only for the resources consumed
- Jobs are run on a serverless Spark platform
- Glue Scheduler to schedule the jobs
- Glue Triggers to automate job runs based on “events”

Glue ETL - Transformations

- Bundled Transformations:

- DropFields, DropNullFields – remove (null) fields

- Filter – specify a function to filter records

- Join – to enrich data

- Map - add fields, delete fields, perform external lookups

- Machine Learning Transformations:

- FindMatches ML:** identify duplicate or matching records in your dataset, even when the records do not have a common unique identifier and no fields match exactly.

- Format conversions: CSV, JSON, Avro, Parquet, ORC, XML

- Apache Spark transformations (example: K-Means)

- Can convert between Spark DataFrame and Glue DynamicFrame

Amazon Athena – Typical Pattern



AWS Lake Formation

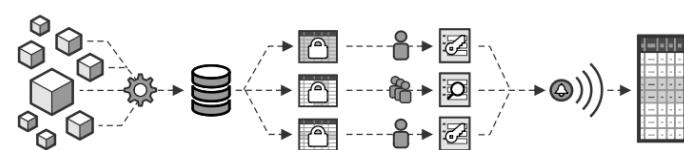
DATA LAKES AND ANALYTICS

AWS Lake Formation

Create and manage data lakes with centralized access control across Amazon Web Services

The easiest way to create and manage your data lake on Amazon S3

How it works



Ingest & Organize

Automatically ingest, clean, encrypt, and register existing Amazon S3 bucket content, including log data from CloudTrail, CloudFront, and Amazon services.

Secure & Control

Define access control that provides the right data to the right users, groups, and access is checked against policy, so your data is protected even if tools change or new data arrives.

Collaborate & Use

Search and discover using catalog metadata. All access is checked against policy, so your data is protected even if tools change or new data arrives.

Monitor & Audit

Be alerted of access requests and policy exceptions. Review activity history with detailed change logs and data lineage.

Get started

With AWS Lake Formation, it is easy to create and manage your data lake without having to configure and integrate each underlying AWS service.

[Get started](#)

Pricing (US)

With AWS Lake Formation, you pay minimal fees for metadata and API costs on top of your current storage, analytic, and machine learning service usage charges.

[Learn more](#)

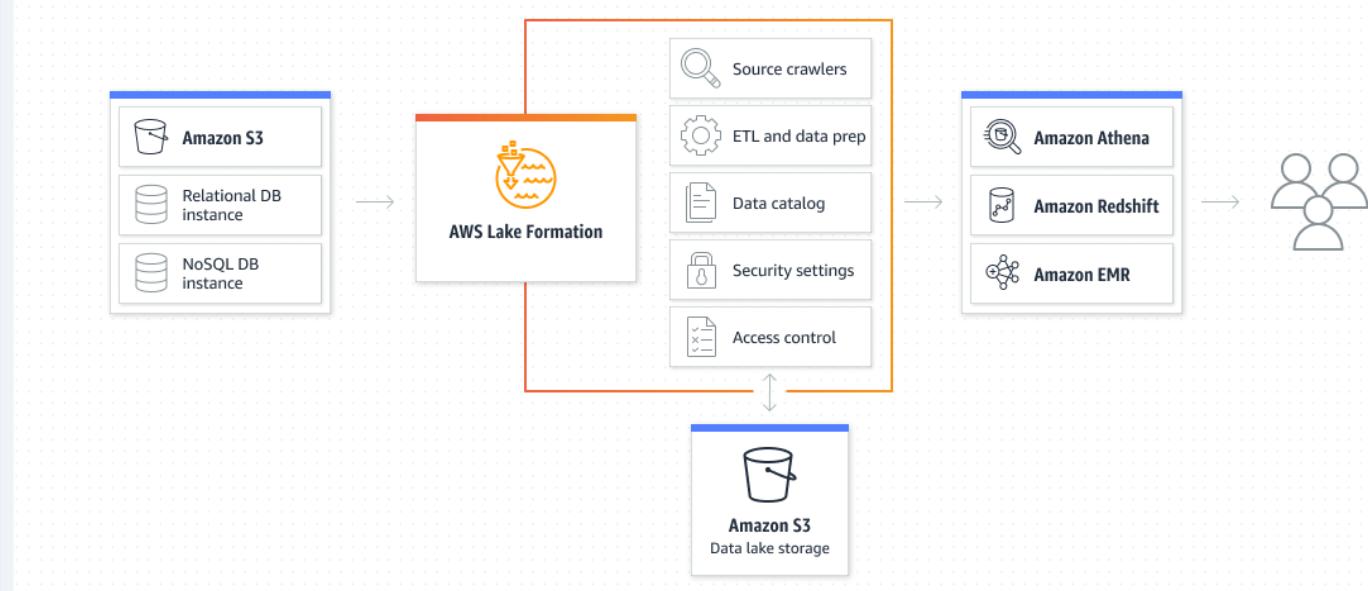
Related services

[AWS S3](#)

[AWS Glue](#)

[AWS Athena](#)

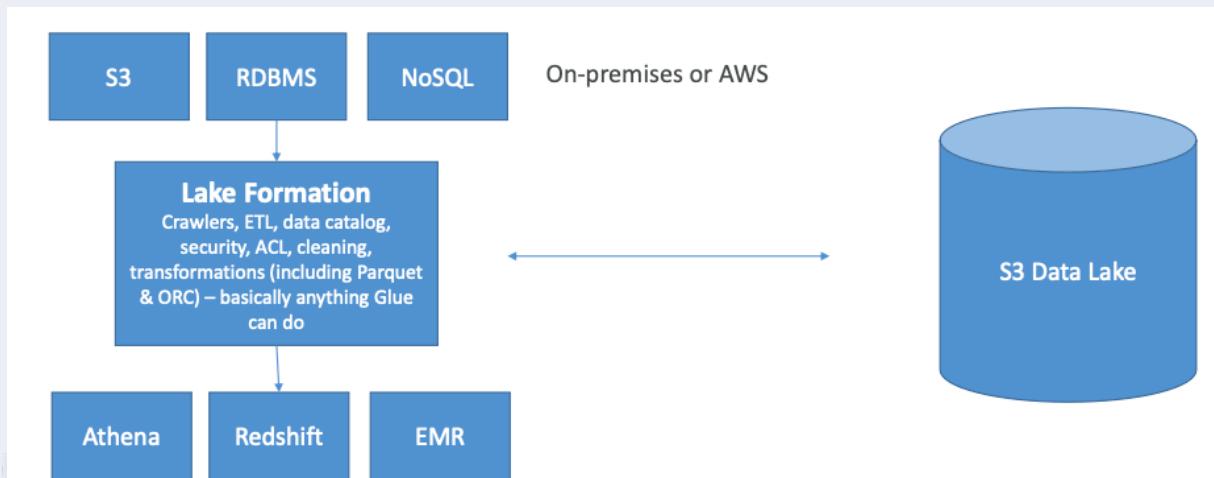
AWS Lake Formation



Proprietary and confidential

AWS Lake Formation

AWS Lake Formation helps you centrally **govern**, **secure**, and globally share data for analytics and machine learning. With Lake Formation, you can manage **fine-grained access control** for your data lake data on Amazon Simple Storage Service (Amazon S3) and its metadata in AWS Glue Data Catalog.



AWS Lake Formation

- ⓘ You can now create Apache Iceberg tables in the Data Catalog. To learn more, visit the [documentation](#).
- ⓘ You can now enable auto-compaction for managed tables in Data Catalog. To learn more, visit the [documentation](#).
- ⓘ You can now manage views in Data Catalog. To create a new view, visit the [documentation](#).

AWS Lake Formation – Databases

AWS Lake Formation > Databases

Databases (12)

Name	Owner account	Lake Formation	Default permission	Shared resource	Shared resource	Shared resource	Amazon S3
default	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
ginamastrorilli_nyctaxi_db	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
nyctaxi_db	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
nyctaxi_db_aryan	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
nyctaxi_db_ir	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
nyctaxi_db_jason	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
nyctaxi_db_nabil	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
nyctaxi_db_nithila	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
nyctaxi_db_pa	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
nyctaxi_db-bc	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
nyctaxi-db-samee	535146832369	No users	IAM/AWS Glue p...	-	-	-	-
zayd_nyctaxi_db	535146832369	No users	IAM/AWS Glue p...	-	-	-	-

Find databases

C Actions View Create database

< 1 > ⌂

Proprietary and confidential

PLURALSIGHT

AWS Lake Formation - Tables

AWS Lake Formation > Tables

Tables (35 loaded, more available)

Actions

<input type="checkbox"/>	Name	Database	Data ac...	Lake Fo...	Governance	Owner ...	Shared ...	Shared ...	Shared ...	Location	Classification
<input type="checkbox"/>	taxi_zone_lookup	nyctaxi_db_nabila	Other	No users	-	53514683...	-	-	-	s3://nabil...	CSV
<input type="checkbox"/>	raw_yellow_tripdata	nyctaxi_db_nabila	Other	No users	-	53514683...	-	-	-	s3://nabil...	CSV
<input type="checkbox"/>	zaydghaffar_s3_tr...	zayd_nyctaxi_db	Other	No users	-	53514683...	-	-	-	s3://zayd...	Parquet
<input type="checkbox"/>	yellow_tripdata	ginamastrorilli_n...	Other	No users	-	53514683...	-	-	-	s3://gina...	Parquet
<input type="checkbox"/>	samee_hartford....	nyctaxi-db-samee	Other	No users	-	53514683...	-	-	-	s3://samee...	Parquet
<input type="checkbox"/>	aryan_aws_glue_a...	nyctaxi_db_aryan	Other	No users	-	53514683...	-	-	-	s3://arya...	Parquet
<input type="checkbox"/>	yellow_tripdata	nyctaxi_db_pa	Other	No users	-	53514683...	-	-	-	s3://peter...	Parquet
<input type="checkbox"/>	yellow_tripdata	nyctaxi_db_ir	Other	No users	-	53514683...	-	-	-	s3://ir-my...	Parquet
<input type="checkbox"/>	ben_bucket_trans...	nyctaxi_db-bc	Other	No users	-	53514683...	-	-	-	s3://ben-...	Parquet
<input type="checkbox"/>	day_3_lab_jason_...	nyctaxi_db_jason	Other	No users	-	53514683...	-	-	-	s3://day-...	Parquet
<input type="checkbox"/>	nithila_demo_tra...	nyctaxi_db_nithila	Other	No users	-	53514683...	-	-	-	s3://nithil...	Parquet
<input type="checkbox"/>	taxi_zone_lookup	nyctaxi_db_pa	Other	No users	-	53514683...	-	-	-	s3://peter...	CSV

AWS Lake Formation - Views

AWS Lake Formation > Views

Views (8+)

A view is a logical table in Data Catalog. View and manage all available views.

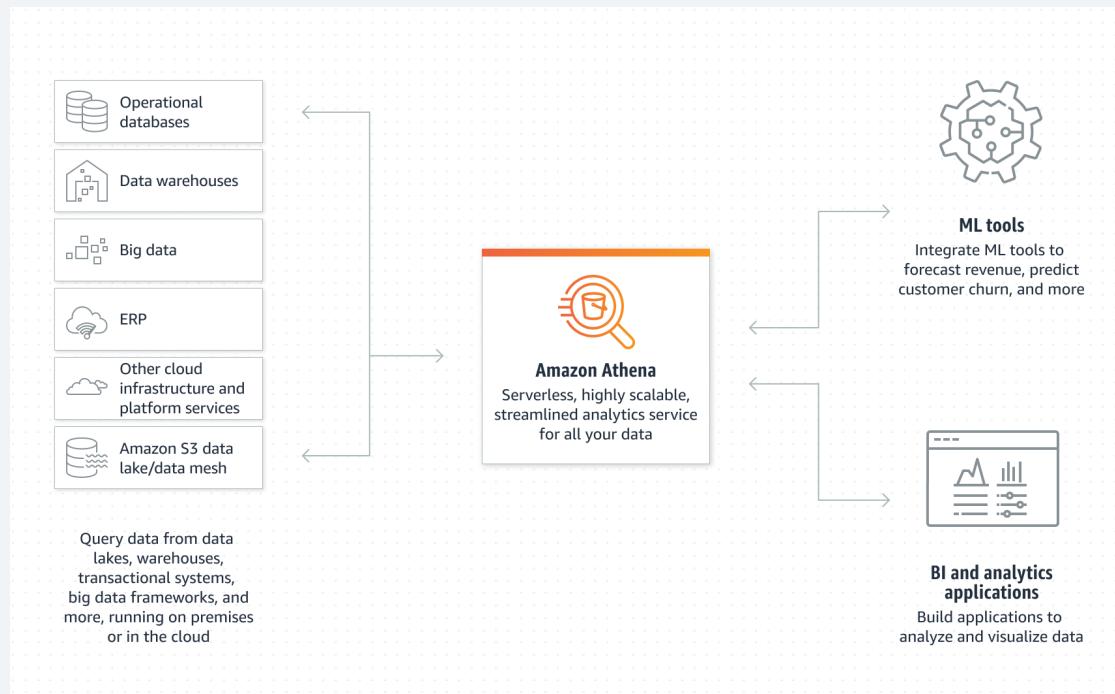
Name	Database	Status	Shared resource	Shared resource owner	Shared resource owner	Last updated
v_yellow_tripdata	zayd_nyctaxi_db	-	-	-	-	June 20, 2024 at 9:04 PM UTC
v_yellow_tripdata	ginamastrorilli_nyctaxi...	-	-	-	-	June 20, 2024 at 8:45 PM UTC
v_yellow_tripdata	nyctaxi_db_aryan	-	-	-	-	June 20, 2024 at 8:31 PM UTC
v_yellow_tripdata	nyctaxi_db_ir	-	-	-	-	June 20, 2024 at 8:24 PM UTC
v_yellow_tripdata	nyctaxi_db_pa	-	-	-	-	June 20, 2024 at 8:24 PM UTC
v_yellow_tripdata	nyctaxi_db_jason	-	-	-	-	June 20, 2024 at 8:19 PM UTC
v_yellow_tripdata	nyctaxi_db_nithila	-	-	-	-	June 20, 2024 at 8:16 PM UTC
v_yellow_tripdata	nyctaxi_db	-	-	-	-	June 19, 2024 at 4:38 PM UTC

Proprietary and confidential

PLURALSIGHT

Amazon Athena

Amazon Athena is an **interactive query service** that makes it easy to analyze data directly in Amazon Simple Storage Service (Amazon S3) using standard **SQL**. With a few actions in the AWS Management Console, you can point Athena at your data stored in **Amazon S3** and begin using standard SQL to run ad-hoc queries and get results in seconds.



Amazon Athena – Presto under the hood

The screenshot shows the official Presto website. At the top, there's a red header with the text "Amazon Athena – Presto under the hood". Below this is a blue navigation bar with the "presto" logo, "Getting Started", "Learn", "Community", "Blog", and "Docs" links, and social media icons for GitHub, Docker, Slack, Twitter, and LinkedIn. The main content area features the Presto logo and the tagline "Fast and Reliable SQL Engine for Data Analytics and the Open Lakehouse". A blue button invites users to "Learn more about our active development of Presto 2.0, the C++ native engine and next-generation version of Presto. →". Below this are two buttons: "Get Started" and "Join the Presto Foundation". A GitHub icon with "15.7K" indicates the repository size. The bottom of the page includes a "Proprietary and confidential" notice and the Pluralsight logo.

presto

Getting Started Learn Community Blog Docs

GitHub Docker Slack Twitter LinkedIn

presto

Fast and Reliable SQL Engine for Data Analytics and the Open Lakehouse

Learn more about our active development of **Presto 2.0**, the C++ native engine and next-generation version of Presto. →

Get Started

Join the Presto Foundation

GitHub 15.7K

Proprietary and confidential

PLURALSIGHT

Amazon Athena – When to use

Data Analysis:

- Analyzes unstructured, semi-structured, and structured data stored in Amazon S3.
- Supports various data formats including CSV, JSON, Apache Parquet, and Apache ORC.

Ad-Hoc Queries:

- Run ad-hoc queries using ANSI SQL.
- No need to aggregate or load data into Athena.

Integration with Visualization and BI Tools:

- Integrates with Amazon QuickSight for easy data visualization.
- Supports generating reports and data exploration with BI tools or SQL clients via JDBC or ODBC drivers.

Amazon Athena – When to use

AWS Glue Integration:

- Works with AWS Glue Data Catalog for a persistent metadata store.
- Enables table creation and querying based on a central metadata store.
- Integrated with AWS Glue's ETL and data discovery features.

Ease of Use:

- Allows running interactive queries directly against data in Amazon S3.
- Requires no data formatting or infrastructure management.
- Quick setup: define a table and start querying using standard SQL.

Ideal Use Cases:

- Suitable for running interactive ad-hoc SQL queries against Amazon S3 data.
- Eliminates the need to manage infrastructure or clusters.
- Perfect for quick queries, such as troubleshooting web log performance issues.

Amazon Athena – When to use

Athena uses the following terms to refer to hierarchies of data objects:

- **Data source** – a group of databases
- **Database** – a group of tables
- **Table** – data organized as a group of rows or columns

Sometimes these objects are also referred to with alternate but equivalent names such as the following:

- A **data source** is sometimes referred to as a **catalog**.
- A **database** is sometimes referred to as a **schema**.

The screenshot shows the Amazon Athena console interface. On the left, there's a sidebar titled "Data" with dropdown menus for "Data source" (set to "AwsDataCatalog") and "Database" (set to "nyctaxi_db"). Below these are buttons for "Tables and views" and "Create". On the right, there's a query editor with four tabs: "Query 1", "Query 2", "Query 3", and "Query 4". The fourth tab, "Query 4", is active and contains the following SQL code:

```
1 SELECT * FROM "AwsDataCatalog"."nyctaxi_db"."raw_yellow_tripdata" limit 10;
```

The "Data source" and "Database" dropdowns in the sidebar are highlighted with red boxes. The SQL code in the query editor is also highlighted with a red box.

Amazon Athena – When to use

Athena uses the following terms to refer to hierarchies of data objects:

- **Data source** – a group of databases
- **Database** – a group of tables
- **Table** – data organized as a group of rows or columns

Sometimes these objects are also referred to with alternate but equivalent names such as the following:

- A **data source** is sometimes referred to as a **catalog**.
- A **database** is sometimes referred to as a **schema**.

The screenshot shows the Amazon Athena console interface. On the left, there's a sidebar titled "Data" with dropdown menus for "Data source" (set to "AwsDataCatalog") and "Database" (set to "nyctaxi_db"). Below these are buttons for "Tables and views" and "Create". On the right, there's a query editor with four tabs: "Query 1", "Query 2", "Query 3", and "Query 4". The fourth tab, "Query 4", is active and contains the following SQL code:

```
1 SELECT * FROM "AwsDataCatalog"."nyctaxi_db"."raw_yellow_tripdata" limit 10;
```

The "Data source" and "Database" dropdowns in the sidebar are highlighted with red boxes. The SQL code in the query editor is also highlighted with a red box.

Amazon Athena – CTAS

- A **CREATE TABLE AS SELECT** (CTAS) query creates a new table in Athena from the results of a SELECT statement from another query. Athena stores data files created by the CTAS statement in a specified location in Amazon S3.
- **CREATE TABLE AS** combines a **CREATE TABLE** DDL statement with a **SELECT** DML statement and therefore technically contains both DDL and DML. However, note that for Service Quotas purposes, CTAS queries in Athena are treated as DML

Amazon Athena – ACID Transactions

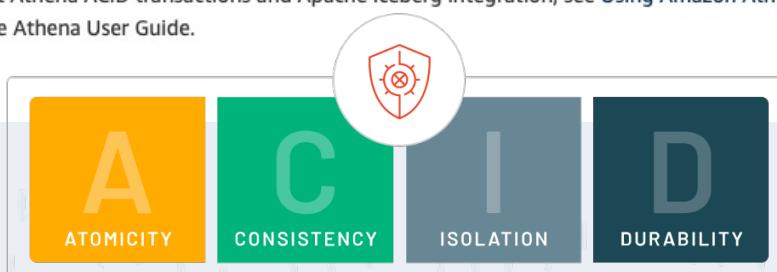
Announcing general availability of Amazon Athena ACID transactions, powered by Apache Iceberg

Posted On: Apr 5, 2022

We are excited to announce the general availability of Amazon Athena ACID transactions, a new capability that adds insert, update, delete, and time travel operations to Athena's SQL data manipulation language (DML). Athena ACID transactions enable multiple concurrent users to make reliable, row-level modifications to their [Amazon S3](#) data from Athena's console, API, and ODBC and JDBC drivers. Built on the Apache Iceberg table format, Athena ACID transactions are optimized for Amazon S3 storage, support seamless schema evolution, and ensure atomic operations across other services and engines that support the Iceberg table format such as [Amazon EMR](#), [Apache Spark](#), and [Apache Flink](#).

Athena ACID transactions can help you make business- and regulatory-driven updates to your data using familiar SQL syntax and without requiring a custom record locking solution. Responding to a data erasure request is as simple as issuing a SQL DELETE operation. Making manual record corrections can be accomplished via a single UPDATE statement. And with time travel capability, you can recover data that was recently deleted using just a SELECT statement.

To learn more about Athena ACID transactions and Apache Iceberg integration, see [Using Amazon Athena Transactions](#) and [Using Iceberg Tables](#) in the Athena User Guide.



Proprietary and confidential

 PLURALSIGHT

What are ACID Transactions

If a database operation has these ACID properties, it can be called an ACID transaction, and data storage systems that apply these operations are called transactional systems. ACID transactions guarantee that each read, write, or modification of a table has the following properties:

Atomicity

- This means that all parts of a transaction must be completed successfully. If any part fails, the entire transaction is rolled back, leaving the system as if the transaction never happened. Think of it as an all-or-nothing rule.

Consistency

- A transaction must bring the database from one valid state to another, maintaining all predefined rules, such as data integrity constraints. This ensures that the database remains accurate and reliable throughout the transaction process.

Isolation

- Transactions are isolated from each other, meaning the operations of one transaction cannot interfere with those of another. This is like making sure that if multiple people are making changes at the same time, each change is processed separately without causing conflicts.

Durability

- Once a transaction is successfully completed, the changes it made are permanent, even in the event of a system failure. This ensures that once you have made a change and the system confirms it, that change will not be lost.

Amazon Athena – ACID Transactions

- The term "ACID transactions" refers to a set of properties (**atomicity, consistency, isolation, and durability**) that ensure data integrity in database transactions.
- ACID transactions enable multiple users to **concurrently** and reliably **add** and **delete** Amazon S3 objects in an atomic manner, while isolating any existing queries by maintaining read consistency for queries against the data lake.
- Athena ACID transactions add single-table support for insert, delete, update, and time travel operations to the Athena SQL data manipulation language (DML).
- You and multiple concurrent users can use Athena ACID transactions to make reliable, row-level modifications to Amazon S3 data. Athena transactions automatically manage locking semantics and coordination and do not require a custom record locking solution.

What is Apache Iceberg

The screenshot shows the Apache Iceberg website homepage. The header features the "ICEBERG" logo with a blue globe icon, the text "Apache Iceberg", a search bar with a magnifying glass icon, and social media links for GitHub, YouTube, and Slack. Below the header is a navigation menu with links to Home, Quickstart, Docs, Releases, Blogs, Talks, Vendors, Project, and Concepts. The main content area has a teal background with a stylized illustration of two penguins on icebergs. The text "Apache Iceberg" is prominently displayed in large white letters, followed by the subtitle "The open table format for analytic datasets." Below this are four buttons for "COMMUNITY", "GITHUB", "YOUTUBE", and "SLACK". A small watermark "Proprietary and Confidential" is visible in the bottom left corner, and the word "URALSIGHT" is in the bottom right corner.

ICEBERG

Apache Iceberg

Search

Home Quickstart Docs Releases Blogs Talks Vendors Project Concepts

Apache Iceberg

The open table format for analytic datasets.

COMMUNITY GITHUB YOUTUBE SLACK

Proprietary and Confidential

URALSIGHT

What is Apache Iceberg

Apache Iceberg is an open-source high-performance table format designed for massive analytic tables. It enables the use of **SQL tables** for big data and allows multiple engines like Apache Spark, Trino, Flink, Presto, Hive, Impala, and others to safely work with the same tables **simultaneously**. Iceberg addresses the performance and usability challenges of using Apache Hive tables in large and demanding data lake environments by providing features such as:

- **ACID Transactions:** Ensures data accuracy by guaranteeing atomicity, consistency, isolation, and durability.
- **Schema Evolution:** Supports add, drop, update, or rename operations without side effects.
- **Partition Evolution:** Allows updates to partition schemes as queries and data volumes change.
- **Time Travel:** Enables reproducible queries using exact table snapshots or examining changes.
- **Version Rollback:** Quickly corrects problems by resetting tables to a known good state.

Amazon EMR (Elastic MapReduce)

Analytics

Amazon EMR

Easily run and scale Apache Spark, Apache Hive, Presto, and other big data workloads.

Amazon EMR is a cloud big data platform for running large-scale distributed data processing jobs, interactive SQL queries, and machine learning (ML) applications using open-source analytics framework such as Apache Spark, Apache Hive, Presto, and more.

Explore Amazon EMR now

- Amazon EMR running on Amazon EC2**
Process and analyze data for Machine Learning, scientific simulation, data mining, web indexing, log file analysis, and data warehousing.
- Amazon EMR Studio**
Manage Jupyter notebooks that run on Amazon EMR clusters and debug applications such as Apache Spark.
- Amazon EMR Serverless**
Run big data applications using open-source frameworks without managing clusters and servers.
- Amazon EMR on EKS**
Run open-source big data frameworks on Amazon Elastic Kubernetes Service (Amazon EKS).

[Create cluster](#)

Pricing

[Amazon EMR on Amazon EC2](#)

[Amazon EMR on Amazon EKS](#)

[Amazon EMR on AWS Outposts](#)

[Amazon EMR Serverless](#)

[Cost calculator](#)

Get started

How it works



Proprietary and confidential

 PLURALSIGHT

Amazon EMR Serverless

Try EMR Serverless for batch jobs, and now with interactive notebooks from EMR Studio.

Analytics

Amazon EMR Serverless

Run big data applications without managing clusters and servers

Amazon EMR Serverless is a serverless option in Amazon EMR that makes it easy for data analysts and engineers to run batch jobs and interactive workloads using open-source big data analytics frameworks without configuring, managing, and scaling clusters or servers. You get all the features and benefits of Amazon EMR without the need for experts to plan and manage clusters.

Introduction



Get started

EMR Serverless provides a runtime environment that simplifies running analytics applications using the latest open-source frameworks. Get started with your first application in seconds.

[Get started](#)

What's new

[Monitor Amazon EMR Serverless applications in near real-time with CloudWatch metrics](#)

[Monitor Amazon EMR Serverless jobs in real-time with native Spark and Hive Tez UI](#)

Documentation

[What is Amazon EMR Serverless?](#)

[Getting started with EMR Serverless](#)

[Amazon CodeWhisperer integration](#)

Pricing (US)

Proprietary and

JRALSIGHT

Hadoop

Hadoop consists of a cluster of tools designed to meet the challenges of processing large amounts of data, including:



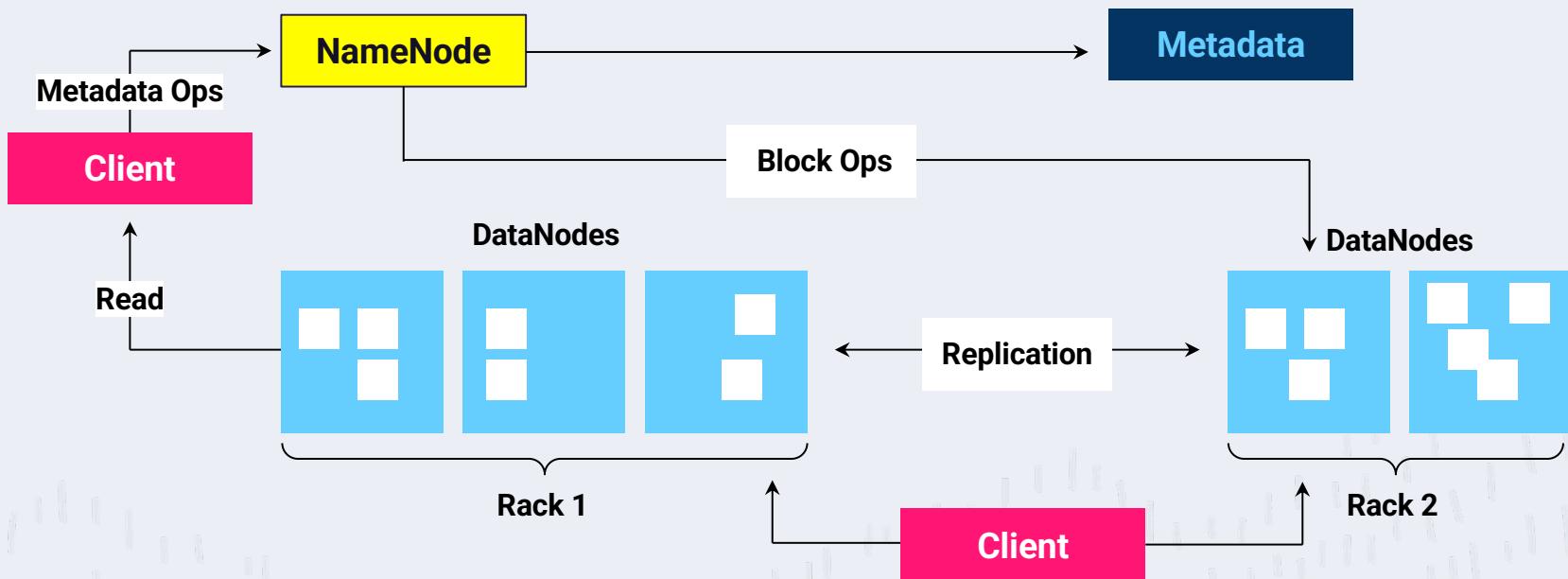
The Hadoop Distributed
File System, or HDFS

Apache Hive

MapReduce

The Hadoop Distributed File System

The **Hadoop Distributed File System**, or **HDFS**, allows Hadoop to store large quantities of data across multiple servers efficiently (and inexpensively) while minimizing the risk of data loss.



What is HIVE?

The screenshot shows the Apache Hive website homepage. The header features a navigation bar with links for Apache Hive, Release, Documentation, General, Development, Community, Blogs, and ASF. The main background is a blue network graph with red nodes and white edges. In the center, there is a large yellow cartoon bee logo with the word "HIVE" written below it in a stylized font. Below the logo, the text "Apache Hive" is displayed. A descriptive paragraph at the bottom left states: "The Apache Hive™ is a distributed, fault-tolerant data warehouse system that enables analytics at a massive scale and facilitates reading, writing, and managing petabytes of data residing in distributed storage using SQL." At the bottom, there are four buttons: "Github" with a GitHub icon, "Mail" with an envelope icon, "Docker" with a Docker icon, and "Community" with a speech bubble icon. The footer contains the text "Proprietary and Confidential" and "URALSIGHT".

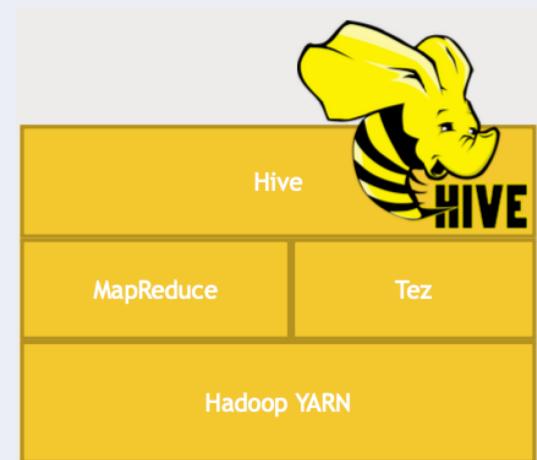
What is HIVE?

Apache Hive is an SQL-like query tool for big data.

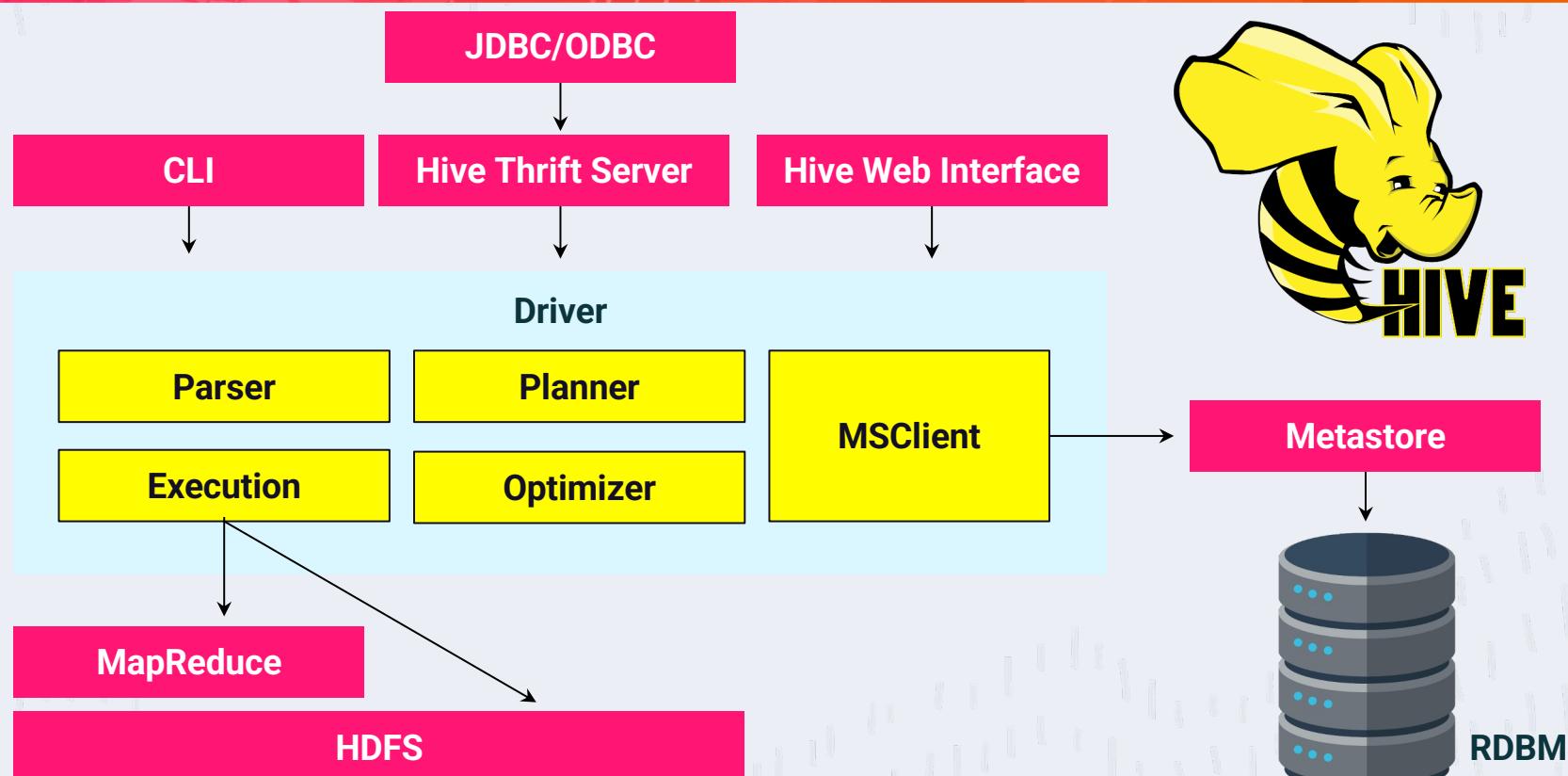


What is HIVE?

- Apache Hive is a **distributed**, fault-tolerant **data warehouse** system that enables analytics at a massive scale. A data warehouse provides a central store of information that can easily be analyzed to make informed, data driven decisions.
- Hive allows users to read, write, and manage petabytes of data using SQL.
- Hive is built on top of Apache Hadoop, which is an open-source framework used to efficiently store and process large datasets.
- As a result, Hive is closely integrated with **Hadoop**, and is designed to work quickly on petabytes of data. What makes Hive unique is the ability to query large datasets, leveraging Apache Tez or MapReduce, with a SQL-like interface.



Apache Hive Architecture

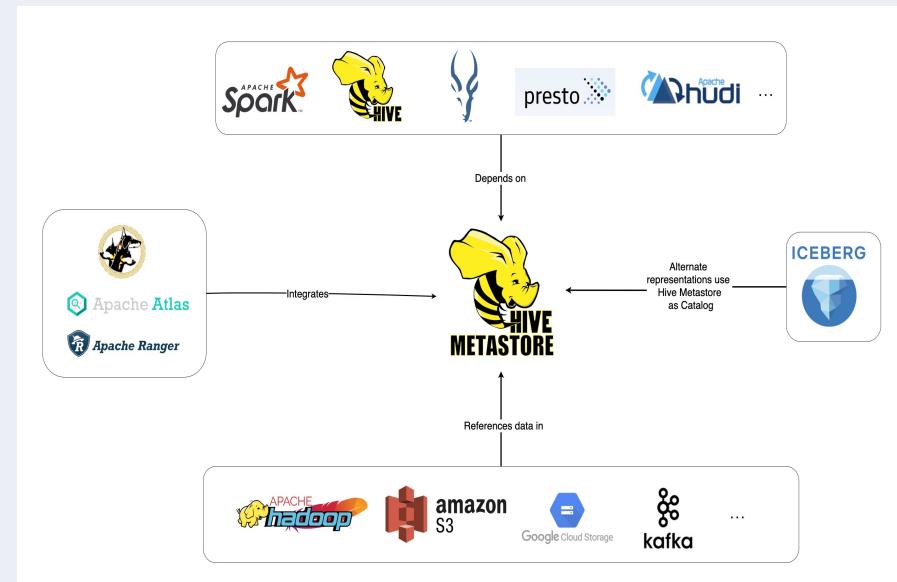


Proprietary and confidential

PLURALSIGHT

What is Hive Metastore?

- The **Hive Metastore (HMS)** is a central repository of metadata for Hive tables and partitions in a relational database, and provides clients (including Hive, Impala and Spark) access to this information using the metastore service API.
- It has become a building block for data lakes that utilize the diverse world of open-source software, such as Apache Spark and Presto. I
- In fact, a whole ecosystem of tools, open-source and otherwise, are built around the Hive Metastore, some of which this diagram illustrates.



What is HiveQL?

HiveQL is a query language for Hive to analyze and process structured data in a Meta-store. It is a mixture of SQL-92, MySQL, and Oracle's SQL.

It is very much **similar to SQL** and highly scalable. It reuses familiar concepts from the relational database world, such as tables, rows, columns and schema, to ease learning. Hive supports four file formats those are TEXT FILE, SEQUENCE FILE, ORC and RC FILE (Record Columnar File).

MapReduce

MapReduce distributes large data tasks across multiple servers and then assembles the results.



MapReduce

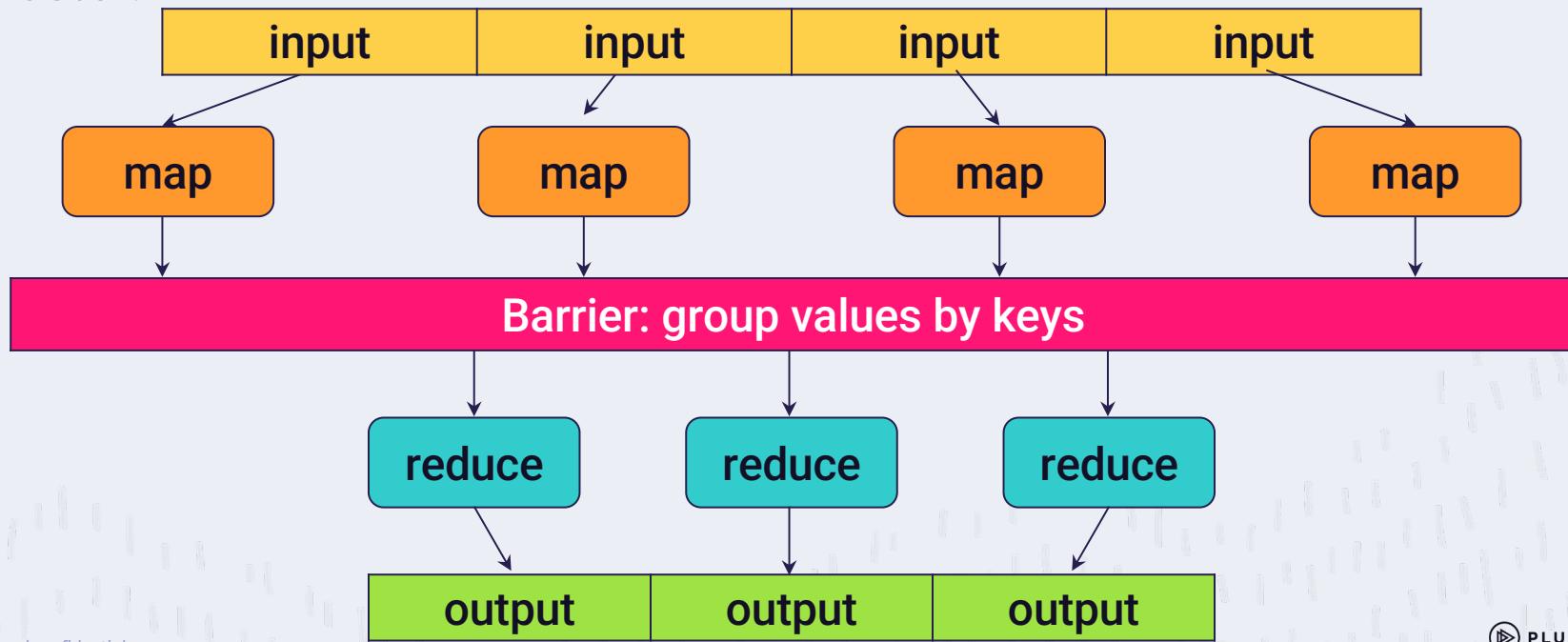
An analogy for MapReduce

If we want to count all the books in a library, we can count each section (**Map**), and then add all the results (**Reduce**).



MapReduce

Google created MapReduce to index all information on the internet. You can use MapReduce to distribute and process the data on your cluster.



MapReduce

Mapping is the process of splitting up data, preprocessing it, and then converting the data into key-value pairs.

Reducing is the process of aggregating the results.

Apache Spark



Download Libraries Documentation Examples Community Developers

Apache Software Foundation

Unified engine for large-scale
data analytics

GET STARTED

What is Apache Spark™?

Apache Spark™ is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters.



Simple. Fast.

Key features



Batch/streaming data

Unify the processing of your data in batches and real-time streaming, using your preferred language: Python, SQL, Scala, Java or R.



SQL analytics

Execute fast, distributed ANSI SQL queries for dashboarding and ad-hoc reporting. Runs faster than most data warehouses.

Proprietary and

JURALSIGHT

Apache Spark

Ecosystem

Apache Spark™ integrates with your favorite frameworks, helping to scale them to thousands of machines.

Data science and Machine learning



SQL analytics and BI



Storage and Infrastructure



Proprietary and confidential

PLURALSIGHT

Apache Spark

Spark SQL engine: under the hood

Apache Spark™ is built on an advanced distributed SQL engine for large-scale data

Adaptive Query Execution

Spark SQL adapts the execution plan at runtime, such as automatically setting the number of reducers and join algorithms.

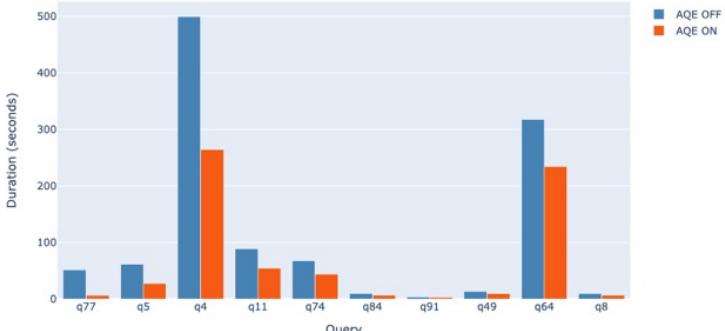
Support for ANSI SQL

Use the same SQL you're already comfortable with.

Structured and unstructured data

Spark SQL works on structured tables and unstructured data such as JSON or images.

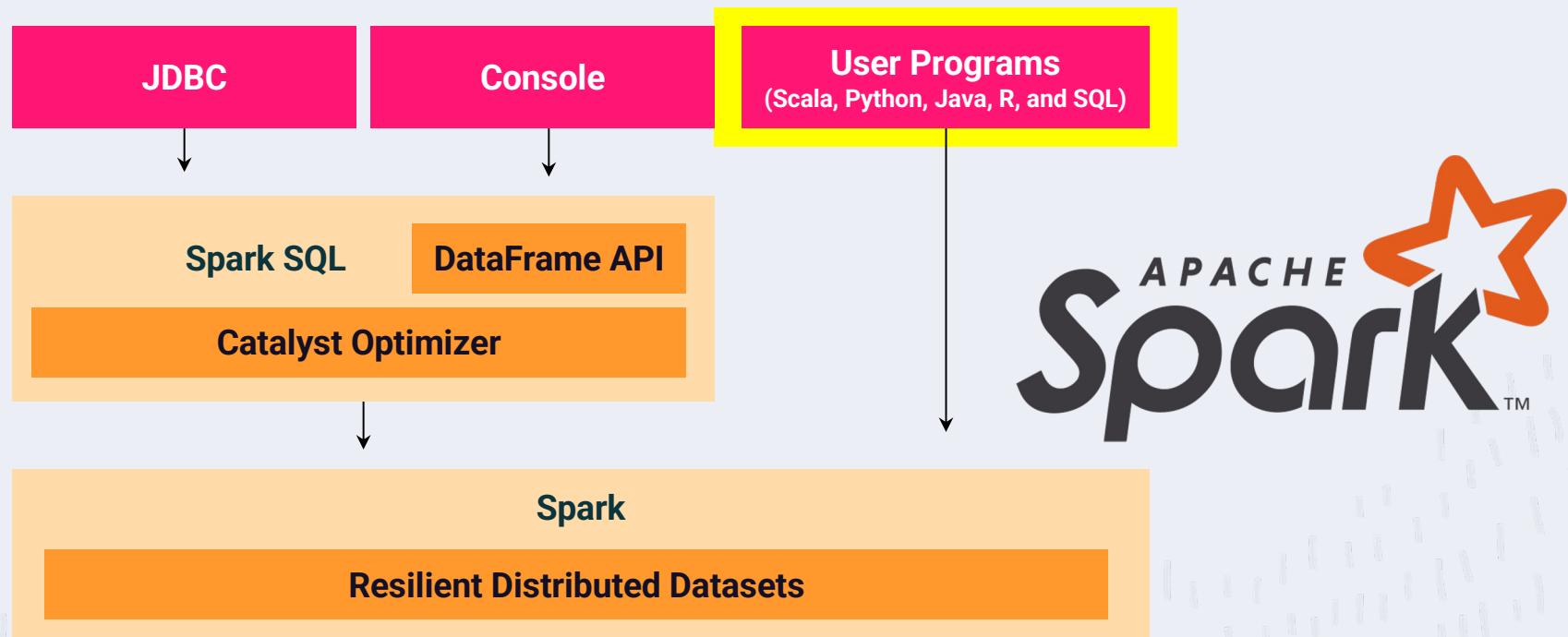
TPC-DS 1TB No-Stats With vs. Without Adaptive Query Execution



Accelerates TPC-DS queries up to 8x

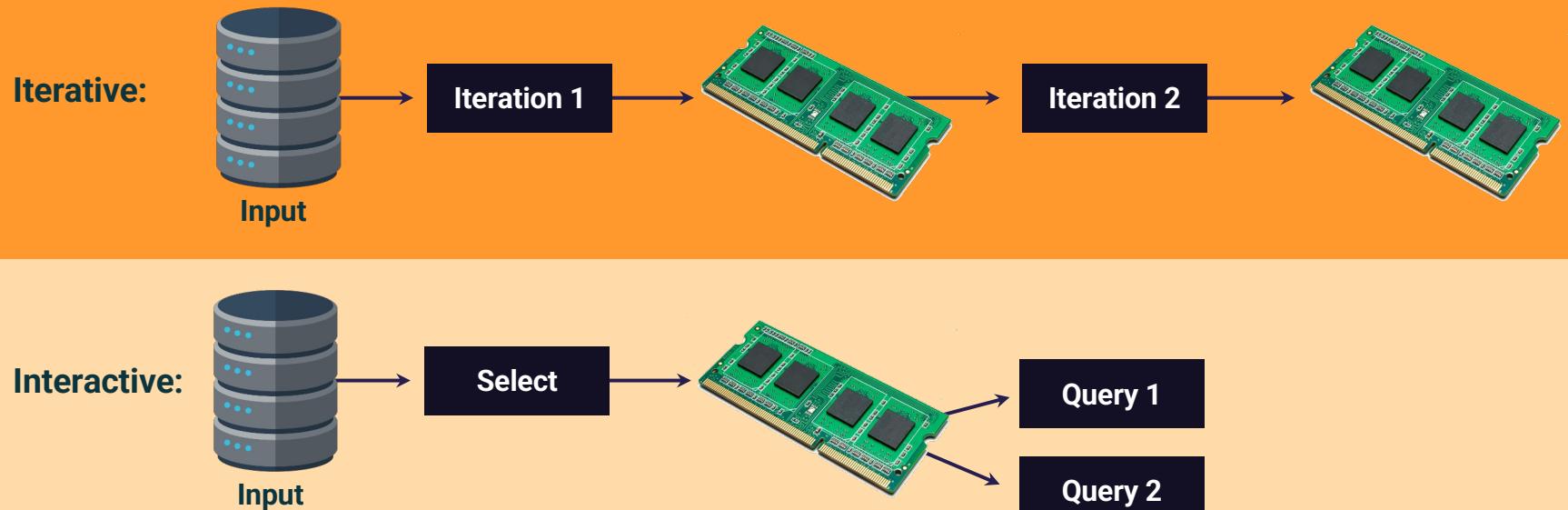
Spark Overview

Spark uses **scripts** from programming languages like Python, has a rich ecosystem, and is very scalable.



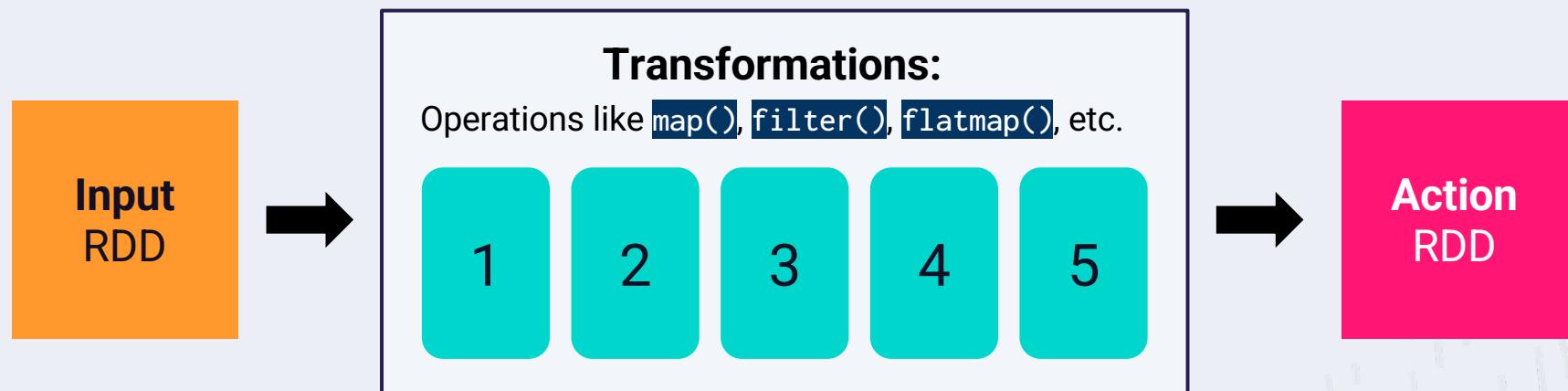
Spark Overview

Spark uses **in-memory computation** instead of a disk-based solution. It doesn't need to talk to the Hadoop Distributed File System (HDFS) for every query, and it retains as much as it can in memory.



Spark Overview

Spark uses **lazy evaluation**, which delays the evaluation of an expression until its value is needed.



Spark **maintains the lineage of transformations** but does not evaluate them until an action is called.

Amazon SageMaker

Amazon SageMaker provides machine learning (ML) capabilities for data scientists and developers to prepare, build, train, and deploy high-quality ML models efficiently.

The screenshot shows the Amazon SageMaker homepage with a dark blue header and a white sidebar. The header features the "Amazon SageMaker" logo and a search bar. The sidebar contains links for "Getting started", "Studio", "Studio Lab", "Canvas", "RStudio", "TensorBoard", "Admin configurations", "JumpStart", "Governance", "HyperPod Clusters", and "Ground Truth". The main content area has a "MACHINE LEARNING" section with the heading "Amazon SageMaker: Build, train, and deploy machine learning models at scale". It includes a sub-section "How it works" with a "What is Amazon SageMaker?" paragraph and a "New user onboarding guide" button. A sidebar on the right titled "New to SageMaker?" offers "Quick setup for a single user" and "Advanced setup for organizations", each with a "Set up for [user type]" button. Another sidebar titled "Documentation" lists links for "Getting started", "Tutorials", "Documentation", "Developer Resources", "AWS Developer Forum", and "Contact us". The footer contains the "ALSIGHT" logo.

Amazon SageMaker

Typical SageMaker workflow



1. Label data

Set up and manage labeling jobs for highly accurate training datasets within Amazon SageMaker, using active learning and human labeling.



2. Build

Connect to other AWS services and transform data in Amazon SageMaker notebooks.



3. Train

Use Amazon SageMaker's algorithms and frameworks, or bring your own, for distributed training.



4. Tune

Amazon SageMaker automatically tunes your model by adjusting multiple combinations of algorithm parameters.



5. Deploy

After training is completed, models can be deployed to Amazon SageMaker endpoints, for real-time predictions.



6. Discover

Find, buy, and deploy ready-to-use model packages, algorithms, and data products in AWS Marketplace.

Thank you!

**If you have any additional questions, please
reach out to me at: (email address).**



PLURALSIGHT