phData | databricks

Hadoop to Databricks
Migration

Table of Contents

Purpose

This document is intended to serve as a plan for migrating existing on-premises Hadoop environments to Databricks. Cloudera and MapR are the two distributions of Hadoop that this document aims to cover and provide a roadmap for customer migration projects. Both distributions are made up of an ecosystem of tools and technologies that will need careful analysis and expertise to determine the appropriate mapping of technologies that will best serve the customer.

Why Databricks?

Customers familiar with on-premise Hadoop installations know all too well that one of the largest problems with the platform is the ongoing support and maintenance of the infrastructure that supports the platform. These challenges include setting up servers, networking, storage, installing software, and configuring best practices for technologies deployed. These only cover the initial setup and then there are ongoing upgrades, patches, and maintenance that require a team of operations engineers to support.

Enter Databricks and the Unified Data Analytics Platform, which aims to remove the complexities involved in running and maintaining these infrastructure decisions and instead allowing developers to focus on delivering business value. Databricks was founded by the original creators of Apache SparkTM, and is available on Microsoft Azure and Amazon Web Services as a cloud-native environment empowering data engineers and data scientists in the following ways:

- **A unified platform for data and AI:** Databricks Unified Data Analytics platform provides one cloud platform for massive scale data engineering and collaborative data science. The platform breaks down the silos between people, data and technology that typically breaks down analytics projects. In addition, Databricks brings a range of new capabilities in scalable ML and AI, enabling use cases organizations could not tackle on Hadoop.
- **Shared Notebooks:** Data teams can confidently collaborate in different languages, from Python to Scala to SQL, and share code via notebooks with revision history and GitHub integration. The collaborative nature of Databricks notebooks speeds productivity and accelerates innovation.
- **Data Availability:** Databricks created Delta Lake that brings reliability, performance, and lifecycle management to your existing data lakes on Amazon S3 or Azure Blob Storage. Delta Lake is now an open source technology managed by the Linux Foundation. Delta Lake makes all of your data more reliable and available for data science, machine learning and analytics, and with no need to make copies.

- **Data Integration**: Integrating with source systems is made easy with a wide range of connectors, and further augmented by the Data Ingestion Network to get data into Delta Lake and keep it up to date.
- **Reliable and Scalable Clusters**: Databricks provides automated cluster management, spinning up clusters, determining their optimal size for the job, and taking them down when the job is done. Combined with highly tuned performance, Databricks lowers your total cost of ownership - paying for what you use instead of investing in excess capacity to meet your peak needs.
- **Built-in Job Scheduling**: Scheduling jobs in traditional on-prem Hadoop environments has always been problematic. Oozie was the packaged service for scheduling and workflow automation but it was too complex and difficult to use that customers would choose to use their own enterprise schedulers. Databricks allows for job scheduling via Databricks Notebooks. No need to rewrite your code for production, your working Databricks Notebook can be put right into production. And Databricks Notebooks can call other Notebooks to create a workflow that enables a component architecture.

Delta Lake

One common problem in Hadoop has been the immutable nature of the datasets stored in HDFS. This often meant extensive training was needed to educate resources on how to store and process data in HDFS.

Open source Delta Lake addresses many of these problems. Delta Lake is implemented as traditional Parquet files with a transaction log that defines what data and files are the most recent so when a job queries the datasets, users are presented with accurate consistent datasets. Other key features include ACID transactions, scalable metadata handling, time travel, schema enforcement, schema evolution, audit history, and full DML support offering UPDATE, DELETE, and MERGE INTO capabilities.

Why phData?

phData is a services company dedicated to delivering solutions on Databricks. Our team has an extensive background of building and deploying solutions on Hadoop. Because of this, we have the relevant experience and background with customers to successfully plan a migration to Databricks. Our architects and engineers work closely with customers to understand their current on-premises Hadoop environments and the technologies used to pick the best cloud-native technologies paired with Databricks to ensure success. Our services include:

- **Data Engineering**: Solutions architects and data engineers pair closely with business subject matter experts to review current applications on Hadoop to plan and implement the necessary changes to deliver these data products on Databricks.

- **Managed Pipelines**: After the applications have been migrated to Databricks our team of support engineers will ensure that the applications are stable and that data is kept up-to-date and accessible to users when they need it.
- **Machine Learning**: phData provides the expertise and proven frameworks you need to get ML models built and into production. Our team of data scientists and engineers build ML based solutions for your business on top of the Databricks platform.

phData partners with Databricks to engage with customers looking to migrate to Databricks and offer consulting services to plan for the migration and to execute on that plan.

Migration Plan

The goal of the migration from on-premises Hadoop distributions to Databricks is to offer customers an easy path to becoming cloud-native and save costs on licensing and hardware. The time and duration of the migration ultimately depends on the size of the customer and the complexity of their use cases. However, the following section aims to develop a strategy for migrating these customers over three distinct phases - Discovery, Implementation, and Validation.

PHASE 1: DISCOVERY	PHASE 2: IMPLEMENTATION	PHASE 3: VALIDATION
LEARN Catalog and understand different types of workloads running in their cluster.	IMPLEMENT Cloud-native architects and engineers will construct your foundational platform.	VERIFY Data and workloads will be reviewed and verified before cutover activities.
DESIGN Size and design the new architecture that supports both data and applications.	MOVE Leveraging our experience and tooling, data and workloads will be migrated.	SUPPORT New projects, users, and any remaining Hadoop workloads.
PLAN Establish a detailed migration plan, ensuring disruptions are avoided.		EVOLVE Establish future roadmap, including ideas for the next phase of cloud.

Discovery Phase

The goal of this phase is to gather all the necessary background on the current Hadoop environment including tools and technologies, data sources, use cases, resources, integrations, and service level agreements. The output of these investigations will help develop a plan for migration.

Tools and Technologies

Creating a complete inventory of all tools and technologies used in the current Hadoop environment is imperative to creating the migration plan. This inventory should include tools native to the Hadoop environment and third-party vendor software that deliver results for the customer. From this inventory, phData will build a current state architecture detailing the Hadoop landscape. Working with the customer we will identify if the tool is needed in the go forward strategy or whether the capabilities and benefits built into Databricks can fill the void. Tools that are needed moving forward will need to go through an evaluation process to fully understand how they work in a cloud-native Databricks environment.

This inventory should be categorized as follows:

- Ingest and Data Integration
- Data processing and transformation
  - Programmatic
  - User-facing
- Data cataloging, discovery, and lineage
- SQL Interfaces
  - Programmatic
  - User-facing
- Data Visualization
- Storage
  - Storage formats
  - Structured vs Unstructured
- Streaming
  - MapR streams or Apache Kafka
  - Message format
  - Volume
  - Source and target integrations
- Third-Party vendor software
- Security and Governance

Data Sources

Equally important to the inventory of tools and technologies in the current Hadoop environment are the data sources that deliver results and provide value. These data sources may be external to the Hadoop platform such as Databases, ERP, CRM, files, streaming or event data, etc. Or alternatively, internal data sources that are used for integrations feeding data out of the platform. During this discovery, it is also a good time to identify data sets that can be deprecated and marked as unnecessary to move to the cloud. Data sources should have an owner or subject matter expert (customer identified) that can be available to answer questions and provide a detailed understanding of data sets. phData will work with these data experts to complete the following list of questions.

General Data Source Questions

- Who is the data owner/steward?
- What is the classification of data?
- What are the retention policies?
- Are there obfuscation or data masking requirements?
- Are there data encryption requirements?

Relational Database Questions

- Database type (e.g. Oracle, Microsoft SQL server, DB2, MySQL, etc)
- Are there any custom functions/UDFs that need to be migrated/replicated?
- Number of Databases
  - Number of Tables
    - Number of Columns
    - Data Types - specifically custom data types
- Required frequency of ingest
- Bulk ingest or Incremental
  - Can CDC operations be queried via audit logs
  - Do non-CDC sources have a last modified or updated column
  - How are hard deletes handled in non-CDC sources
- Do the table structures change frequently adding or removing columns, data type changes, etc
- Security requirements, roles, users, downstream consumers

MapR Streams or Kafka Questions

- What is the message type (ie AVRO, JSON, XML, delimited text, etc)?
- Is there a schema registry
- Number of topics
  - Number of partitions
    - Partition strategy, what are your message keys
  - Individual message size
  - Messages per second
  - Consuming applications
  - Data retention requirements

Delimited File Questions

- Where are the files located
- Are they accessible from the chosen cloud provider
- Does new data get delivered via a new file or append to the existing file
- File format
- Delimiter
- Contains record headers
- Can schema be inferred on read
- File encoding

API Integration Questions

- How can programmatic access to the API be provided
- What is the schema for responses
- Is there a schema registry
- Type of data
- How many requests per second
- Structure of requests - Is the response a bulk load of a dataset or incremental change
- Does the response schema change over time
- Authentication and authorization mechanisms

Use Cases (Hadoop Specific)

The most important step of the Discovery phase is to fully understand the applications running in the current Hadoop environment. This will better define the amount of effort involved in migrating the customer from the on-premises environment to the cloud. Being that Hadoop has a variety of tools and services that make up an application, there are some important distinctions to be made about how the application is deployed in the environment. phData will work with the customer and the owners of each application to understand the following:

- A complete description of the application and what value it provides to the customer
- What technologies does the application use in the Hadoop environment
  - Versions of these technologies e.g. Spark 1 vs Spark 2
- Is the application streaming or batch
- What external libraries are used
- Reference architectures
- Complete list of data sources
- Expected outputs
- Downstream consuming applications
- Security and Classification consuming and produced datasets
- Production readiness
  - Expected errors
  - Frequency of errors
  - Steps to resolution
  - Support and monitoring implementations

Resourcing

People and resources internal to the client are often concerned about change. Identifying the users of the current platform and providing required training is crucial to accelerating adoption and growth of Databricks at the customer. Understanding tools and processes in their daily workflow and making alternatives available will make sure that these people feel comfortable using Databricks.

Integrations

External applications accessing the Hadoop environment need to be inventoried as well to ensure that the cloud-native Databricks environment can serve data to these applications. Oftentimes these applications have complex access patterns for authentication and authorization which will need to be evaluated. However, external applications connecting using JDBC or ODBC can use these drivers when communicating with Databricks and should work out of the box.

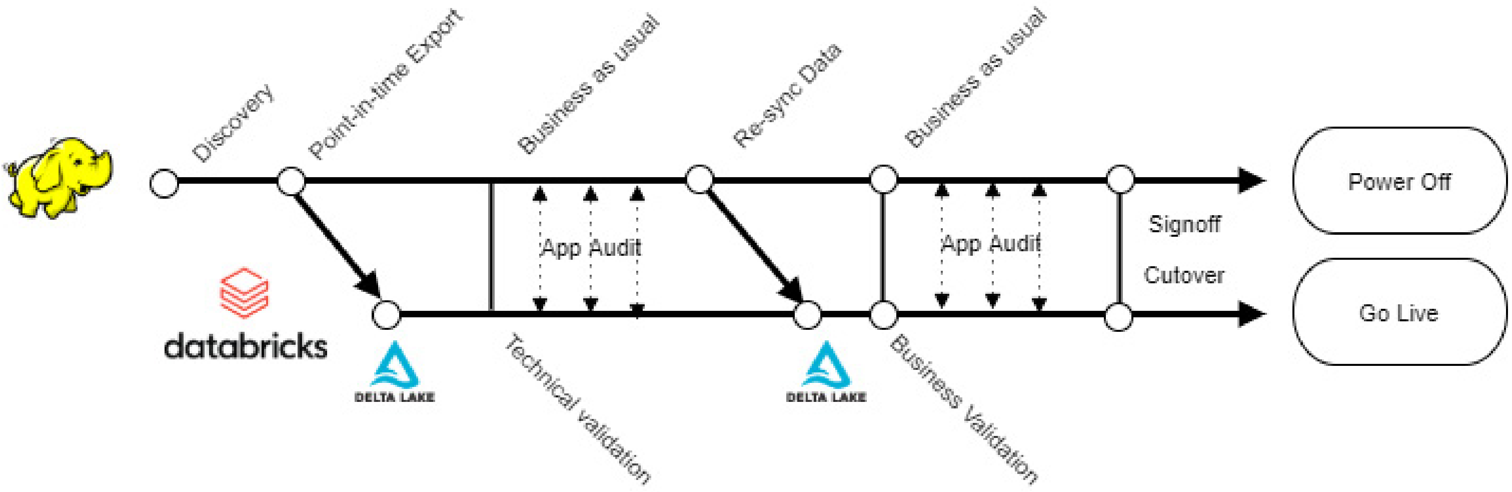
Security and Governance

A current inventory of security and governance configurations must be collected. Because of the different architectures that each Hadoop distribution puts forth, this can broaden the scope to include more tools. Tools like Sentry, Ranger, HDFS ACLs. In addition, the use of Kerberos, Active Directory, encryption-at-rest and encryption-in-transit must be taken into account. For MapR, we must look at filesystem permissions, ACLs and Access Control Expressions.

Implementation Phase

The implementation phase focuses on migrating business applications from the current Hadoop environment to Databricks. Using information gathered during the discovery phase a prioritized list of data sources, applications, and tools will be selected for migration. phData will staff small to large sized teams of architects and data engineers to work with the customer through this migration effort. This phase will be the longest and most technical to execute.

Since use cases and implementations vary greatly between customer to customer, phData does not see a generalized quick win migration tool to this implementation phase. However, when patterns develop and code or tools can be created to be used from one migration to another, our engineers will develop these tools and provide them to the broader Databricks community.

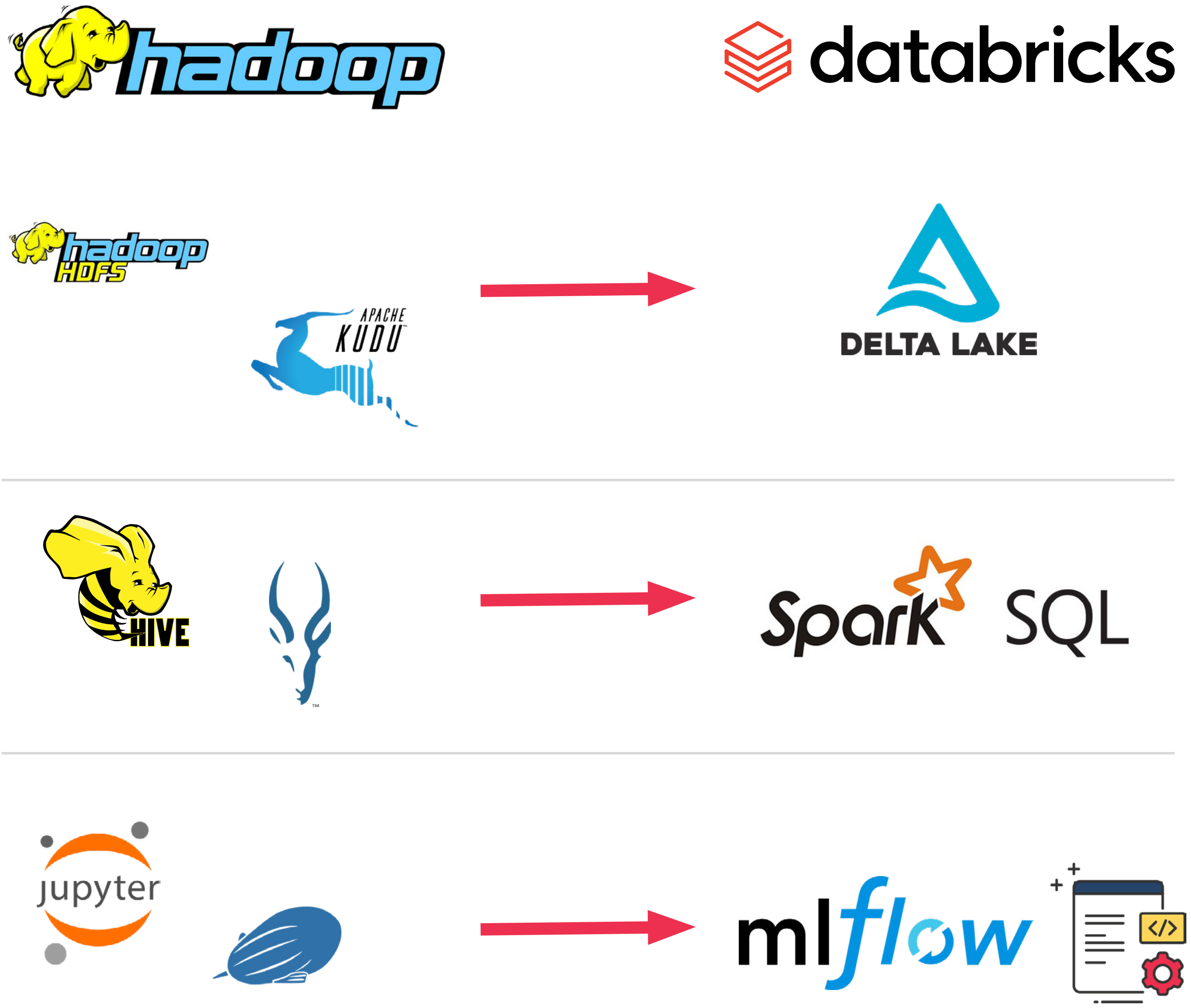


Storage Migration

Using the Data Source inventory, engineers will move data stored in HDFS to the cloud vendor's storage layer (Blob Storage or S3). The same information architecture will be applied in the new cloud storage file system and the resulting folder and file structure should be a one to one match of the HDFS file system.

Challenges

A significant challenge to the storage migration for both Cloudera and MapR implementations will be migrating role-based access controls to the new cloud storage system. Depending on the cloud vendor, the configuration of role-based access is different. Azure Blob Storage uses Active Directory with groups and users to grant access to blob containers. Whereas, Amazon AWS uses IAM policies to configure access to S3 buckets. A tool will need to be developed to migrate these policies for each cloud vendor.



Hive Metastore Migration

After the storage has been migrated, the next step is to migrate the Hive Metastore from the Hadoop environment to Databricks. The Hive Metastore contains all the location and structure of all the data assets in the Hadoop environment. Migrating the Hive Metastore will allow users to query tables in Databricks notebooks using SQL statements. During this migration process, the locations of the underlying datasets will need to be updated to reference the Databricks file system instead of the current HDFS path.

HiveQL/Impala Migration

Many times customers use Hive SQL or Impala files to execute pieces of data pipeline or workflow. These scripts are executed running `beeline -u <url> -f <path to sql file>` or `impala-shell -f <path to sql file>`, after the migration of both the storage assets and the Hive Metastore these types of workflow items can use Spark SQL within a notebook. The notebooks can then be scheduled directly in the Databricks UI or added to a workflow from a calling notebook. Overall this process will benefit from the features of Databricks notebooks including Git integration and revision history.

Hadoop Distribution Specific Technologies

Both MapR and Cloudera have tools and technologies specific to their distributions. Mainly MapR has MapR DB and MapR Streams and non-standard HDFS implementation allowing read-write capabilities. Whereas, Cloudera offers MPP query implementations like Apache Impala and Apache Kudu, a data storage layer offering DML operations.

MapR DB

MapR Database is a high-performance NoSQL database management system built for the MapR platform. The multi-model database brings together operations for analytics as well as real-time streaming applications. Depending on the applications consuming this data the migration path will vary. If the consumer is a heavy OLTP application it is better to convert this data pipeline into a Spark Streaming application on Databricks and landing data in Azure CosmoDB or DynamoDB. Traditional analytics can be implemented as Spark Streaming into Delta Lake.

MapR Streams (MapR Event Store)

MapR Streams is a service built on top of Apache Kafka. The API for publishing and subscribing to these streams is the same as Kafka. However, MapR offers tooling to enable easier integration of datasets within the platform. These types of integrations will need to be migrated to a Spark Streaming application.

Apache Kudu

Apache Kudu is a columnar storage layer developed for the Apache Hadoop platform and is distributed as part of Cloudera's distribution of Hadoop. A key benefit to Kudu is its support of DML operations (UPDATE, UPSERT, DELETE, etc). Kudu use cases will primarily be migrated to Delta Lake, as it offers comparable capabilities. Impala scripts and Spark applications utilizing Kudu will be migrated appropriately.

Apache HBase

HBase is a distributed column-oriented database built on top of the Hadoop file system and is distributed by Cloudera. HBase serves a similar purpose to that of MapR DB and as such has a very similar migration path: use cases that are primarily OLTP driven will be converted to Spark Streaming applications utilizing Azure CosmoDB or DynamoDB for storage. Analytics applications can be migrated to Spark Streaming and Delta Lake.

Apache Solr

Solr provides easy, natural language access to data stored in or ingested into Hadoop, HBase, or cloud storage. Most use cases that utilize Solr will migrate to Amazon Elasticsearch Service or Azure Cognitive Search, while data processing applications will be migrated as otherwise indicated in this document.

Apache Spark Application Migration

Apache Spark applications will need to be reviewed and fully understood before migrations can occur. Any Apache Spark version 1 application will need to be refactored to an Apache Spark version 2 application as Databricks does not support Apache Spark version 1. phData engineers are trained in converting apps like these however appropriate A B testing will need to take place.

Many Apache Spark applications may fall under the “Lift and Shift” model where as long as the source data is available to Databricks and the application can be packaged as a Jar these applications should run on Databricks.

Apache Spark to Delta Lake Migration

When going through the discovery phase we will take careful note of what Apache Spark applications should be migrated to use Delta Lake. Applications that are doing a small number of updates or deletes will be prime candidates for this refactoring. Other considerations will be around performance, ACID transactions, schema enforcement, and data consistency. Oftentimes Apache Spark apps have been written to take these features into consideration and if there is an opportunity to reduce complexity and take advantage of these features the time to implement would be during this migration.

Validation Phase

The final phase will be to validate the outcome of the migration from Hadoop to Databricks. This step should be performed using traditional A -> B testing. For some time the customer will need to be running the existing Hadoop implementation alongside the cloud-native Databricks offering. Validation scripts and processes will be developed to ensure the delivered results in Hadoop match with that in Databricks. As applications and use cases get cleared and all checkouts have been performed they can be shut down in Hadoop. The goal at the end of this phase to be able to completely remove or repurpose the Hadoop infrastructure.

Customer Success Example

phData recently worked with a large mining company to move an existing use case on their Hadoop platform to cloud-native technologies. A large driver for the use case was the complexities of their Hadoop platform, their administration efforts around the platform were getting large and they wanted to better utilize that team. With a more cloud-centric approach including Databricks, the mining company has the ability to better utilize this team in engineering efforts providing direct value to the business as opposed to administration and maintenance. The customer needed help from phData with the overall architecture of the desired implementation in addition to the engineering effort to implement their use case. The mining company has an important use case of gathering large amounts of sensor data from machines that they manufacture. This sensor data is written to Blob storage in Azure using an internally written application. Databricks is then used to normalize the data that lands in Azure storage and deliver that data back to a specified Kafka topic. The normalized data may also be stored in Delta Lake files. Depending on whether to write the normalized data to Kafka or Delta Lake is driven by a configuration that the customer can change to alter the desired output.

Getting Started

Given the proportions of Hadoop migrations, consider seeking help from phData with expertise and hands-on experience in Big Data migrations. phData is proud to be a Databricks partner. We specialize in data, ML, and long-term success with Databricks.

To start a migration plan, we recommend a Migration Workshop with phData and Databricks. This will bring our experts together with your team to outline the goals of the project, begin to understand the current systems landscape, identify critical components and discuss the appropriate methodology and processes mentioned in the paper. This is an excellent opportunity to assess the current situation and prepare for a migration project.

Migration Workshop Agenda:

- Customer use-case & goal overview
- Identify migration and customer challenges
- Overview of methodology and process
- Detailed migration case study & architecture

Duration: 90-120 minutes
Audience: Engineers, Architects, Leaders familiar with Hadoop use-cases and future organizational goals.
Contact: info@phdata.io or hadoop-migration@databricks.com