Migrating to Enterprise Data Lake 3.0

Journey towards Serverless Data Lake

Last Updated Jul 2019

Authors

Sudheesh Narayanan
Founder & CEO, Knowledge Lens

Ganesh Iyer
Chief Technology Officer, Knowledge Lens

Gopala M
Chief Architect, Knowledge Lens

Kumar Chinnakali
Principal Architect, Knowledge Lens

Knowledge Lens
Accelerating Actionable Insights

Notice

This paper consolidates the learnings from our various Enterprise Data Lake initiatives, and how the technology has evolved over the last decade to the new Serverless Data Lake. Today, Enterprises are migrating from Legacy Data Lake built on static clusters to Serverless Data Lake or Enterprise Data Lake 3.0

It is important to note a significant difference in terminology as Enterprise Data Lake 3.0 was mostly used in the context of Hortonworks Data Lake 3.0, however here we are referring to the 3rd generation of the Enterprise Data Lake and not specifically to Hortonworks Data Lake 3.0. The third generation of Enterprise Data Lake is based on managed infrastructure using serverless architecture.

This paper highlights the workload migration challenges, technology issues and solutions for migrating to the Serverless Data Lake or Enterprise Data Lake 3.0. The discussion pointers in this paper are presented to create further discussion in the data community and is not to be considered as an assumed prototype or standard of any kind. It represents Knowledge Lens' current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of Knowledge Lens' products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from Knowledge Lens, its affiliates, suppliers or licensors. The responsibilities and liabilities of Knowledge Lens to its customers are controlled by Knowledge Lens' agreements, and this document is not part of, nor does it modify, any agreement between Knowledge Lens and its customers.

Executive Summary

The Big Data world has undergone a significant transformation during the last decade. From the traditional Big Data Hub in the early 2010s to the Serverless Data Lake in recent times, there is a fundamental change in how enterprises have navigated the Big Data Analytics Journey. During the initial stage of the Big Data journey, the focus was more on moving the compute closer to the storage (Data Local Processing) to avoid network latency for large scale data processing. However, the paradigm has undergone a dramatic change with In-Memory Compute and Elastic Serverless execution model. There is a complete change in the fundamentals of how we did Big Data Processing in the early 2010s Vs. how we do Big Data Processing today. It is critical that we understand this paradigm shift and reasons behind it to ensure we have a successful Serverless Data Lake implementation.

This white paper looks at this fundamental change and details how traditional enterprises can upgrade themselves to the new Serverless Data Lake with ease. We explore the various Serverless Data Lake options that are available and how there is a dramatic change in the adoption of some of these technologies as we move to this new paradigm. We discuss in detail what tried and tested approach you can adopt to move quickly to this new paradigm, rather than embark on another long-winding journey of Technology Migration.

This document describes the imminent need, the rationale and complexity involved to automate the migration of data and computational workloads from Traditional Big Data Lake to Serverless Data Lake.

Evolution of the Data Lake towards Serverless Data Lake

The Enterprise Data Lake has been through an evolution during the last decade. Let's look at some of the key drivers of this evolution from Data Lake 1.0 to today's Data Lake 3.0.

Enterprise Data Lake 1.0

The traditional data lake (First Generation of Data Lake) was built on HDFS (Hadoop Distributed File System). The key driver for this data lake was due to the following:

- Traditional RDBMS and Parallel Processing Technology becoming cost prohibitive to scale beyond the few-Terabyte size.
- Pricing monopoly of the traditional RDBMS and Parallel Data Warehousing vendors
- Performance and maintenance overheads of moving data across various machines to generate any insights using traditional technology
- Inability to handle the unstructured and streaming data wave that was generated due to the Internet revolution
- Open source technology availability like Hadoop and focused vendors like Cloudera who eliminated enterprise apprehensions of working with Open Source Technologies by providing dedicated support and managed distributions for open source technology.

Hadoop was the natural choice during this time as enterprises were not very cloud-savvy and had data security apprehensions. Enterprises preferred to keep the data on-premise under tight security. So, Data Lake leveraging HDFS as the de-facto storage emerged. And this was boosted by companies like Cloudera and Hortonworks who gave the required confidence to enterprises on the usage of Open Source Technology, and the ability to manage this on their own.

Enterprise Data Lake 2.0

While enterprises were still adopting Big Data technologies and trying to derive value on their accumulated data in the Data Lake, speed of analytics became a major constraint. Traditional technologies like Hive with MapReduce didn't help for Interactive Analysis. Thus, the need for in-memory technology evolved and the first-generation Impala solved some of the challenges enterprises faced with their Enterprise Data Lake 1.0.

However, this was short-lived as various alternatives to MapReduce started surfacing and Apache Spark emerged as the winner in this race. The second generation Enterprise Data Lake (Enterprise Data Lake 2.0) was focused on In-Memory Compute based Data Lake. The key drivers for this change were:

- Advent of Apache Spark as the mainstream processing system. Impala, Drill, Presto and Tez contributed to the new normal that In-Memory Computing is the way to go.
- Hive on Spark helped enterprises realize the value of Spark leveraging the traditional Hive query models. Spark gradually emerged as one of the de-facto distributed in-memory processing engines.
- Amazon showed the path of Elasticity with its Elastic Map Reduce and drove home the point that ease of infrastructure provisioning and cost saving is key for Big Data adoption.
- The adoption of Cloud for non-critical data became the norm and the cloud adoption journey became the cornerstone for enterprises.
- Significant advancements in Object store technologies like Amazon S3 and Azure Data Storage started showing the path towards Cloud.
- The infrastructure maintenance headaches of on-prem clusters started to show up and the cluster administration efforts became cost drivers for moving to the Cloud.

Gradually, enterprises started realizing that the Cloud is not necessarily bad if the right security model is enforced, as the enterprise extends their on-prem data centers to the Cloud. Cloud Storage costs were comparable or lesser than the on-prem storage cost and thus, storing large volumes of data on Cloud Storage suddenly looked more attractive than ever before. The Cloud storage API performance improvement proved to be the icing on the cake and thus, enterprises began seriously considering Cloud as one of the strategic options for Enterprise Data Lake. Thus, the Enterprise Data Lake 2.0 was born with some of the enterprises moving to the hybrid or pure cloud model.

Enterprise Data Lake 3.0

As enterprises started to move data to the Cloud, the driving focus for Cloud Managed Processing became mainstream. Companies like Databricks and Qubole emerged along with Amazon EMR, Azure to shift the data processing paradigm completely to the cloud. Suddenly, we see a big shift in enterprises switching to managed cloud-based processing for the following reasons:

- Frustrations over managing Hadoop Infrastructure on-premise and experiencing a seamless managed service on the cloud. Inability to handle the unstructured and streaming data wave that was generated due to the Internet revolution
- Databricks showed how cloud-based processing was faster than on-premise Spark execution on static clusters, using their secret optimized Spark runtime.
- Significant cost savings in the overall infrastructure due to on-demand compute provisioning (Serverless execution model)

But above all, four significant events changed the game completely in favor of Enterprise Data Lake 3.0 on the Cloud.

1. **Big Data Pioneers losing market share**
   The #1 Big Data Vendor Cloudera acquired the #2 Hortonworks. The merger announcement along with a vague 'way forward' strategy collapsed the initial expectations of a converged data platform from the merger; Cloudera reported missed revenues and a bleak outlook for the future. To add to the fiasco, the #3 vendor MapR is reportedly close to shutting down and desperately seeking investments, further lending itself to the idea that the era of Static Data Lake is coming to an end.

2. **Increased Cloud Adoption by Enterprise**
   AWS reported 300% rise in adoption of its managed Big Data solutions, Azure & other public cloud reported 200% rise in adoption of its managed Big Data solutions. The investments in cloud native solutions boosted the confidence that Cloud adoption is not a theory anymore and enterprises that haven't yet looked into Cloud, have to begin considering these options.

3. **Serverless Big Data Processing goes mainstream**
   Both traditional vendors and cloud native solutions adopted Apache Spark as the de-facto standard. Cloud providers like AWS and Azure, released reference architectures supporting 'Serverless Compute' with Spark as the default compute engine for the Data Lake. This has reinforced the confidence on Apache Spark.

4. **Rise of Containers**
   In addition to the serverless paradigm, Microservice and Container Technology using Docker has been the new wave. Enterprises are increasingly deploying 'microservices' and 'containers' and want to see how it can be leveraged into

their Big Data journey. The cloud-managed services like Databricks uses this technology as the fundamental way to scale and manage the infrastructure, thus promoting the way for elastic compute processing using serverless infrastructure and containers.

Thus, there has been a gradual shift in favor of Enterprise Data Lake 3.0 leveraging managed cloud service for Data Lake.

Databricks, a company that provides value-added products on top of Spark, gained increasing traction and saw an increase in adoption on both AWS and Azure. The research project Apache Spark that started at UC Berkeley university and later launched with Databricks, provided a Unified Analytics Platform powered by Apache Spark for business, data engineering and data science to build data products; with this users can achieve very high speed of time-to-value with Databricks by creating an analytic data pipeline that goes from ETL and interactive exploration to production.

Enterprise Data Lake 3.0

Understanding the technology choices in the new world of Data Lake 3.0 is one of the questions we typically hear from most of our customers. While each enterprise has their own choice of technology stack in the Serverless Data Lake Model, some of the choices are straightforward.

To represent this technology map, we present the high-level building blocks which are essential for an Enterprise Data Lake. Then we map these high level building blocks to technology choices available as we migrate from Cloudera / Hortonworks/MapR to Serverless Data Lake 3.0.

The logical view of the Enterprise Data Lake is presented below.

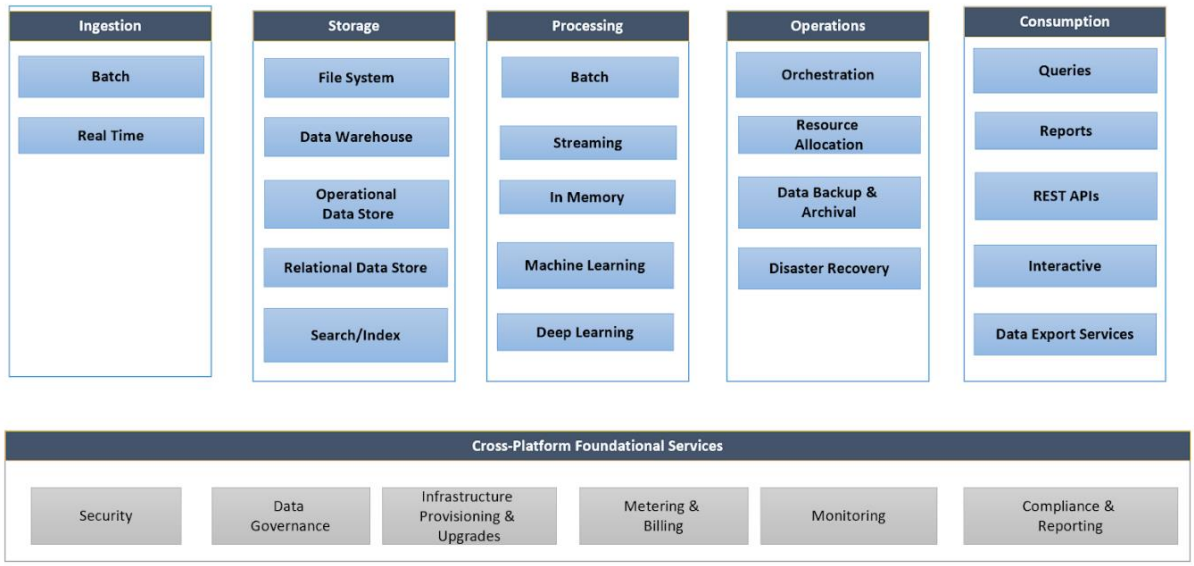

Fig. 1 – Logical view of the Enterprise Data Lake

Enterprise Data Lake 3.0 on Azure

The diagram below shows the various components that are currently deployed in building Enterprise Data Lake 1st generation or 2nd generation. As we move towards the 3rd generation of Enterprise Data Lake towards a Serverless Data Lake on Azure, the technology landscape changes significantly and the right choice of these technology is critical for the success of the Data Lake. The focus is more towards managed infrastructure where the focus is only towards building applications that serve on the Data Lake rather than the infrastructure itself. So, the Platform as a Service (PaaS) offerings are quite preferred over self-managed infrastructure.

Migrating from Cloudera Data Lake to Azure Data Lake 3.0

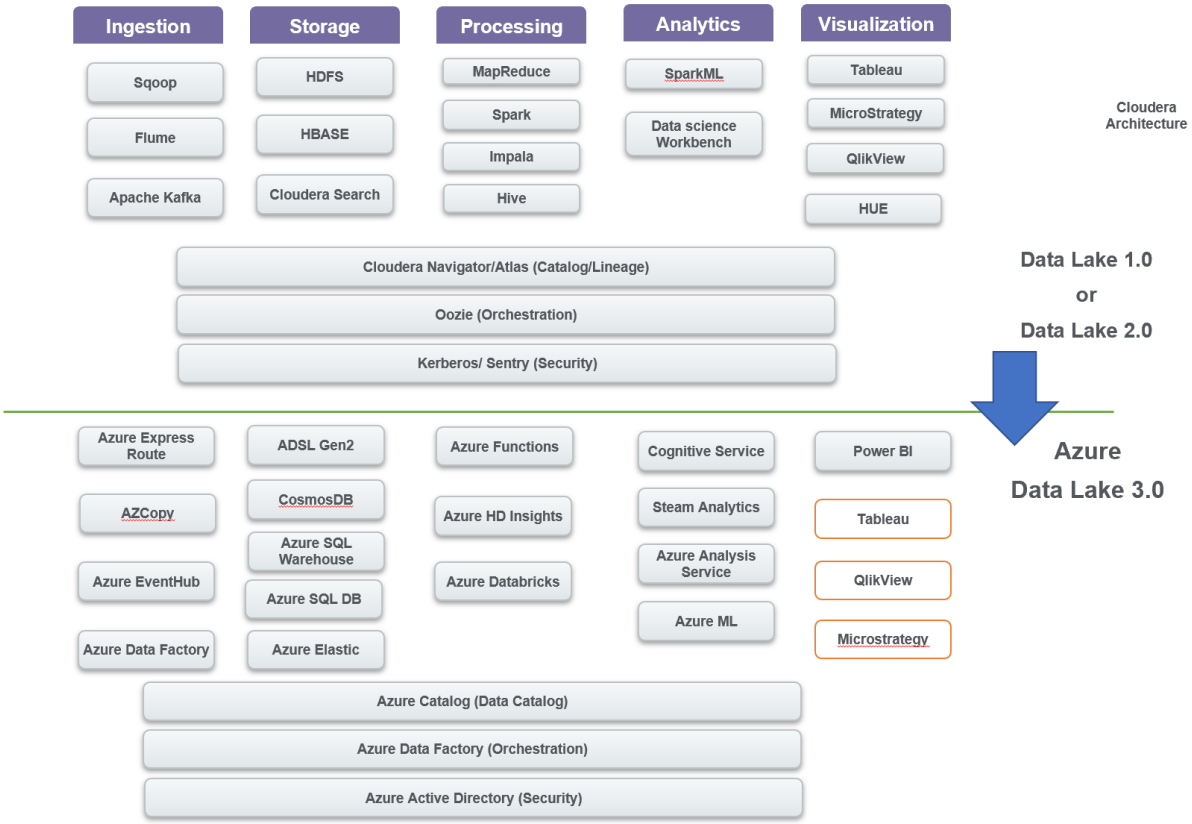

Fig. 2 – Migrating from Cloudera Data Lake to Azure Data Lake 3.0

Enterprise Data Lake 3.0 on AWS

AWS has similar offering as Microsoft Azure. The right choice of components especially the Data Storage and Analytics layer are critical for the successful implementation. Security on the cloud is another important consideration and having a well-defined cloud security architecture is a must irrespective of what approach we take for migration. The security key management, ACL migration and enforcing data security both at rest and motion are critical.

Migrating from Cloudera Data Lake to AWS Cloud Data Lake 3.0

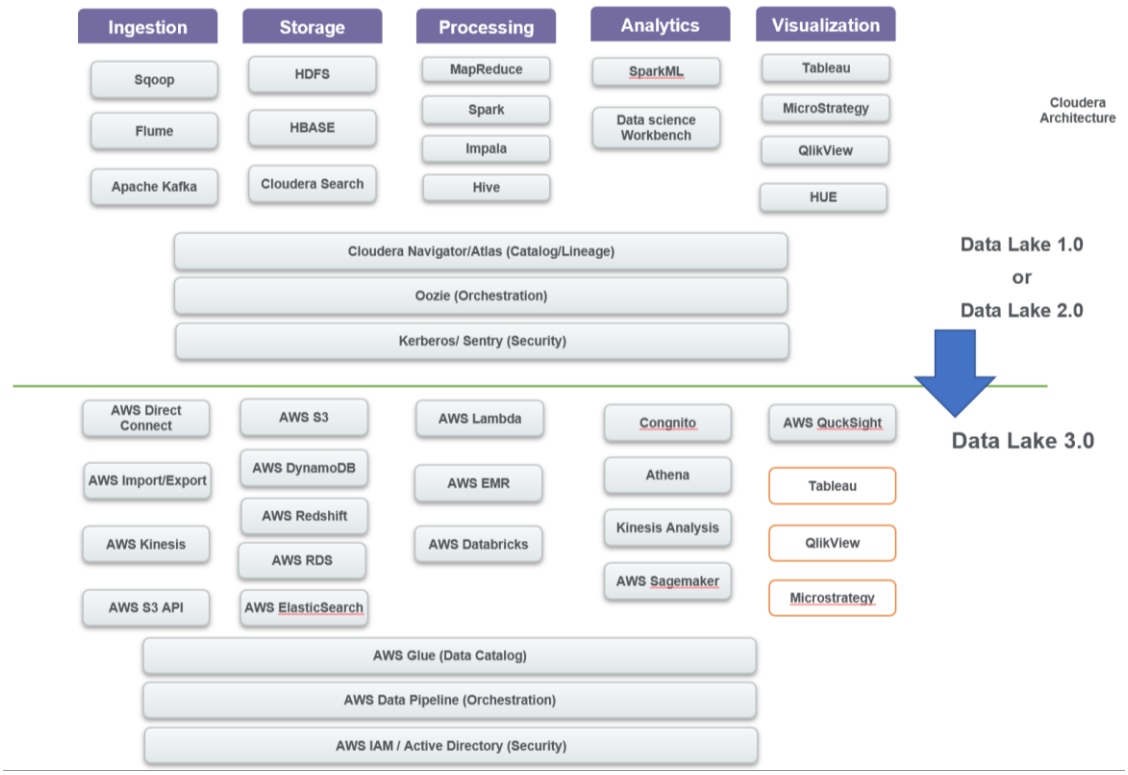

Fig. 3 – Migrating from Cloudera Data Lake to AWS Cloud Data Lake 3.0

Enterprise Data Lake 3.0 and its benefits

One of the key questions every enterprise running on-premise Hadoop clusters asks is –

1. What are the benefits of Enterprise Data Lake 3.0?
2. Should we continue on-prem using Data Lake 3.0 based on the containerized infrastructure or should we move to managed services on the cloud?
3. When migrating to Enterprise Data Lake 3.0, what would this journey look like? What are the pitfalls in this journey?
4. What tools are available that can be leveraged for this migration?
5. Is Enterprise Data Lake 3.0 beneficial and if yes, how?

Let's look at the answers to these questions:

In the new world of Big Data Analytics, there are only two options that enterprises have today:

a. Move to Elastic Data Lake based on containerized Infrastructure on a converged strategy laid out by Cloudera and Hortonworks.
b. Move to cloud managed Services on AWS or Azure, or look at GCP for analytics.

The first option looks very risky, given the vague strategy of the Cloudera-Hortonworks merger and their financial worries, and the significant amount of changes and effort that enterprises have to deploy as they move to new versions of Cloudera-Hortonworks. This would accelerate the migration out of Cloudera-Hortworks or MapR clusters to a Cloud Native or Hybrid Data Lake on Cloud. Enterprises are already burdened with Infrastructure management, regular upgrade challenges, performance tuning, data growth worry and elasticity challenges. There is already a strong business case to move to Managed Big Data Lake on Cloud.

So, the natural choice for all enterprise is to move to Managed Services on the Cloud. While there are currently multiple options for managed services on the cloud, some of the options are a clear advantage, compared to others. Let's look at some of the available options in detail.

1. AWS EMR based Data Lake with S3 as the storage layer and Hive on Spark as the processing engine
2. Azure Data Lake with Azure Data Lake Storage Gen 2 as the storage layer and Azure HD Insights as the processing layer.

3. AWS S3 based storage but leverage Databricks or Qubole as the processing layer
4. Azure Data Lake Storage Gen 2 as the Storage layer but leverage Databricks or Qubole as the processing layer.

There are pros and cons for each of these approaches and we will leave that discussion for another publishing. But one of the key differences in all the approaches is how Databricks Runtime stands out clearly with the rest of the processing layer with its enhanced performance and flexibility.

For example, an AWS EMR based strategy doesn't seem to perform well, given that the performance of Spark on EMR is significantly lower than the performance of Databricks Runtime. This Databricks strategy of keeping the performance improvement of Apache Spark delayed, compared to the Databricks Runtime, has steered enterprises to adopt Databricks Runtime as the de-facto Distributed In-Memory Processing Engine.

Thus, the major advantage for enterprises is that there is no vendor lock-in; today we could run Databricks on Azure or AWS with the same efficiency, while getting improved performance over EMR Spark or Azure HD Insights.
Thus, we see that the trends for Enterprise Data Lake 3.0 is clearly tilted to a Cloud Managed Infrastructure with Databricks as the key distributed processing engine.
There are technologies like Snowflake/Redshift/Azure SQL Warehouse which basically challenge this new normal for Enterprise Data Lake 3.0 strategy. However, the overall paradigm of Data Lake is coming full circle over the last decade. The major benefits of Enterprise Data Lake 3.0 are:

- Significant cost savings due to Elastic On-demand compute infrastructure
- Elastic Storage capability using AWS S3 or Azure Data Lake Storage as backend with a cost comparable to on-premise storage
- Zero Infrastructure Management, (Fully Cloud Managed Infrastructure) thus eliminating the infrastructure management.

How can Enterprises migrate towards Data Lake 3.0?

A. Data Migration

Enterprises today have petabytes of data on-premise clusters and moving all this data to Cloud is not a simple task. Moving petabytes of data over to cloud from on-premise is not easy. Additionally, ensuring the integrity of data transfer of every file, validating the data transfer process and providing re-start ability, error handling for connection failures etc. needs careful thinking. Some of the other challenges involve the time taken to migrate the data, parallelizing the data transfer etc.

In spite of best efforts, we see that manual data migration is not a viable option, given that there will be continuous data loads in the on-prem infrastructure and identifying delta changes that need to be incrementally migrated.

Sometimes, data migration will lead to excess load on the on-prem cluster, thus bringing down the cluster completely and causing production outage, so throttling the load on the cluster should be considered.

Another important consideration is the partition strategy and partition keys. Some of the data stored in the Enterprise Data Lake would have been optimized using a different partition strategy; getting this partition strategy on the cloud, and verifying if these strategies are going to hold good, needs experience. Certain tools on the cloud have a completely different philosophy on data replication and grouping compared to the on-prem model. This needs to be carefully evaluated and then a partition strategy for the Cloud has to be evolved.

Selective data migration is also required as there will be a lot of data in the on-prem Data Lake that won't be required to be moved to the Cloud.

Data Security risk is enhanced during the migration process. Static Clusters won’t have direct access to the cloud. This would require the data to be copied to intermediate storage unless you have a solution like MLens, which directly moves data from secured clusters to cloud. This is a very tricky problem as usage of intermediate storage creates two problems. One, the data at rest is at risk during this intermediate hop, and the data at motion is at risk for non-encrypted transmission. Migrating data in compressed, encrypted format is a must as data security challenges exist during these data migration initiatives.

It is not just the data that needs to be moved, Enterprise Data Lake is already built with fine-grained security constructs on- premise, providing precise access to the business users.

When we move to the Cloud, this poses a significant challenge. Many a time, this challenge is completely overlooked, and the data is landed into object storage without much design consideration. This results in serious rework, once the data migration is completed and security becomes an afterthought.

Some of the sensitive data needs to be masked as we test these solutions in the development environment.

B. Workload Migration

Once the data and security metadata associated with the data is migrated, then comes the hard part of this journey. How do we migrate the various workloads from the on-premise model to the cloud model? This is not a one-to-one porting and will involve significant effort, if not done with proper strategy and planning.

Each of the various workloads and tools used on-premise has its corresponding equivalent on the Cloud. So, the first step in this journey is to define how these technologies are going to line up between on-premise execution and cloud execution.

For example, an Impala Query when migrated to a Databricks cluster doesn't port one-to-one. There are a significant number of queries that will be ported using automation tools, however, many of the analytical queries supported in Impala don't have equivalents in Spark SQL and thus, need to be rewritten. UDF conversion is another important exercise as multiple UDF implemented in Impala or Hive has to be migrated over. Identifying these queries upfront and understanding the magnitude of migration involved, helps in the right estimation and migration strategy.

Many of the existing data lakes are implemented with Hive Queries or PIG scripts along with MapReduce code orchestrated through tools like Oozie. All of these require a clear migration path and approach to automated migration of these orchestration and job dependency.

The tool mapping from on-premise to cloud technology is another significant effort as each tool has its own challenges and would need significant rework if not approached properly. Choosing the right tool on the cloud without vendor lock-in is critical for successful Data Lake implementations.

Once the technology mappings are completed, one must perform a migration assessment to define the magnitude of the changes. Development of automation tools for migration or using some of the existing migration tools like MLens could be one strategy that can be deployed.

Approach for Migration

Based on our experience working with multiple data lake migration projects on their journey to Enterprise Data Lake 3.0, we recommend a '5 phase' approach to this migration.

The diagram below depicts the 5 phases for moving from Traditional Data Lake to Enterprise Data Lake 3.0. Irrespective of the Target Technologies on the cloud or the cloud provider (AWS/Azure/GCP), we see this approach providing a predictable outcome with the least risk for migration. Having such a framework for migration forces the Migration Team to discover all the hidden challenges upfront and complete a predictable migration.

Discovery and Planning are two important phases for successful migration. Identifying all the challenges for migration upfront and preparing a comprehensive plan for migration is the key to success. A tool-based approach is recommended for this migration planning

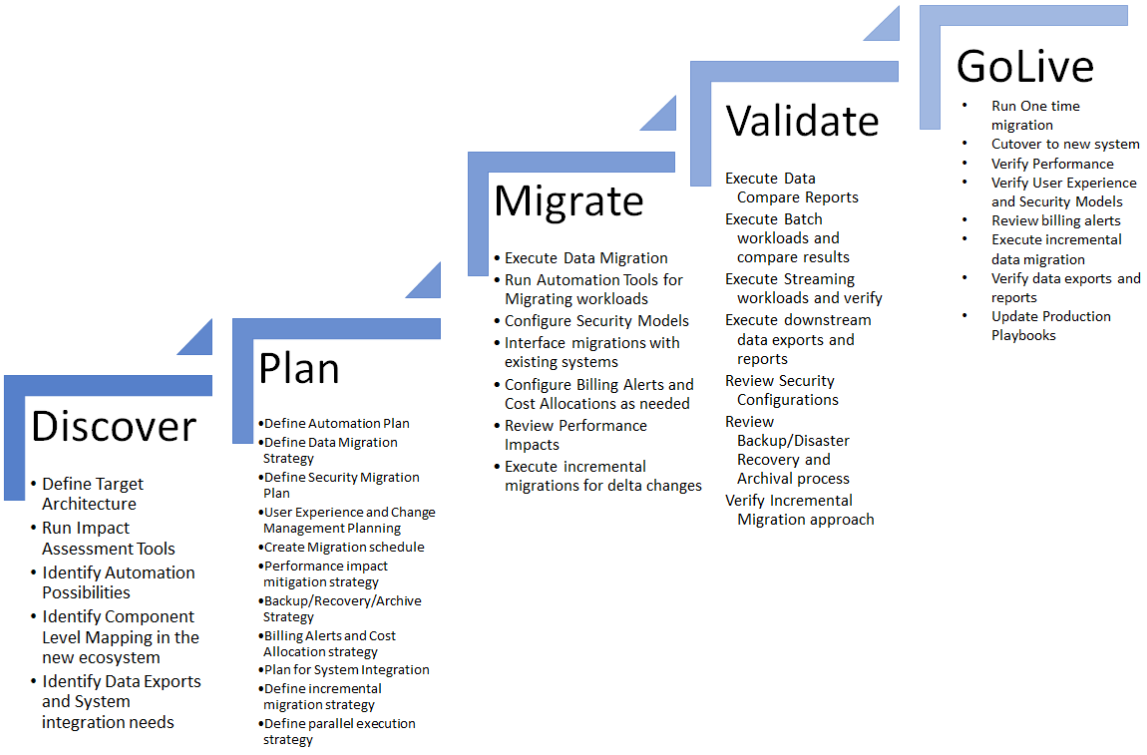

Fig. 4 – Approach for Migration to Serverless Data Lake

Conclusion

Migrating to Enterprise Data Lake 3.0 needs a structured approach with greater understanding of entire Technology landscape and its pitfalls. Getting it right the first time is key to realizing the benefits of Data Lake 3.0. The key takeaways are:

- A Structured approach with the right balance of automation along with a qualified team that brings the right skills with respect to security, data migration and serverless implementation.
- Selecting a cloud neutral Serverless Data Lake with a strong metadata driven orchestration that encapsulates business rules to be modified with ease in agile model
- Planning the migration leveraging an impact assessment tool that identifies the level of automation that can be deployed.
- Deployment of a High-Speed Data Migration Tool that can perform incremental data migration along with the corresponding metadata (ACLs etc.)
- Automation of the comparison reports both for data sets and workload performance to ensure that there are no performance bottlenecks after migration

How can Knowledge Lens help in migrating to Enterprise Data Lake 3.0?

Knowledge Lens has been helping our customers on their Big Data Analytics journey right from its inception. Our deep expertise in Big Data Engineering and Data Science has helped enterprises quickly adopt to the Enterprise Data Lake 3.0. We take pride in our world class MLens Migration Toolkit that has been helping enterprises not only in their Big Data Migration journey but also helping them with Backup, Disaster Recovery and Data Masking solution.

MLens → Migration Toolkit for Enterprise Data Lake 3.0 Journey

MLens (Migration Lens) is designed and purpose built for migrating enterprises to Serverless Data Lake Architecture. It provides a range of toolsets that help in this migration:

- ✓ High Speed Data Migration Tool
- ✓ Migration Impact Assessment Toolkit
- ✓ Automated Migration Assistant for Impala, Hive, Drill and SparkSQL
- ✓ Migration toolkit for Orchestration tools Oozie to Airflow, Azure Data Factory or AWS Data Pipeline.
- ✓ Migration Compare Tool that compares Data, Metadata, Workload performance and datasets.

These toolkits are built specifically for enterprises to move to Serverless architecture.

For more details and demo of these toolkits, reach out to us via
mailto:sales@knowledgelens.com or
visit our website at www.knowledgelens.com