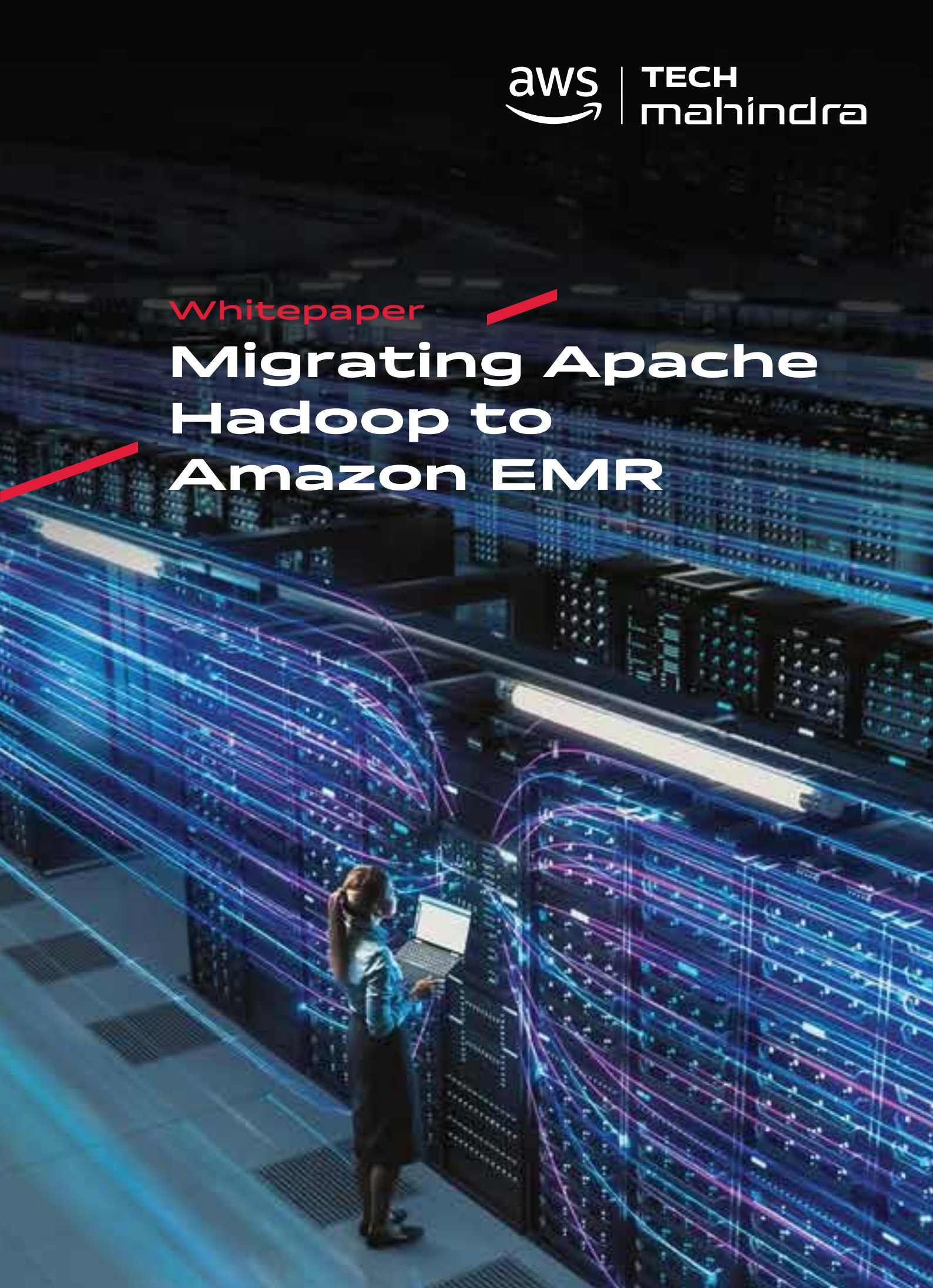


Whitepaper

Migrating Apache Hadoop to Amazon EMR



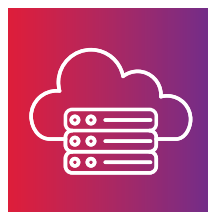
Abstract

The migration of Apache Hadoop systems to Amazon EMR has become increasingly popular for organizations seeking to modernize their data processing capabilities. However, the migration process can be complex and daunting without proper guidance. This whitepaper aims to provide a comprehensive guide for organizations who want to migrate Apache Hadoop systems to Amazon EMR. It also delves into architectural patterns, steps, and data process flow needed for the migration.

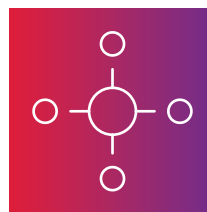
Key Takeaways



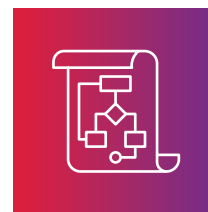
Current issues with Hadoop systems



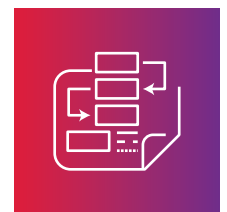
Amazon EMR advantages



High-level data flow for migrating a Hadoop to Amazon EMR



Architecture patterns for Hadoop to Amazon EMR migration



Step by step migration

Introduction

Apache Hadoop is an open-source framework for processing large data across a cluster of compute nodes. While being a powerful big data framework, Hadoop is also known to have some challenges and limitations. Here are a few potential hurdles that you may encounter while working with the Hadoop systems:

- **Cost:** While Hadoop is generally more cost-effective than traditional data warehousing solutions, it can still be expensive to operate and maintain, especially at scale. To store your datasets on Hadoop distributed file system (HDFS), you must plan and pay for 3x the capacity to ensure data redundancy and fault tolerance.
- **Complexity:** Hadoop is a complex system that requires a deep understanding of distributed computing and data management principles. It can be difficult to learn and gain expertise in Hadoop systems, especially for those who are new to big data technologies.



- **Scalability:** While Hadoop is designed to handle large amounts of data, it may not scale well for very large datasets with certain types of layouts or very high levels of concurrency. When data grows, you must plan and procure additional nodes in advance to store the data. Since the storage and compute are tightly coupled, you cannot scale down your nodes when your cluster is idle.
- **Performance:** Hadoop can be slower than other big data processing technologies for certain types of queries or workloads. So, it might not be the best option for systems that require real-time processing or low latency.
- **Integration:** Hadoop is not conventionally designed to work with traditional data warehouses and business intelligence tools, making it challenging to integrate and interoperate with other systems in your business units.

Apart from these limitations, there are many factors to consider such as security, data governance and access control. It is important to carefully evaluate the benefits and drawbacks of Hadoop systems for your specific use case requirements to determine whether it is the right choice for your organization.

Amazon EMR is the managed service for high volume petabyte data processing, interactive analytics, and machine learning using open-source tools like Spark and Presto.

The advantages of using AWS EMR over Hadoop are:

Cost savings:

Running your big data workloads at scale on EMR can be more cost-effective than running them on-premises or in a different cloud environment. EMR offers several cost optimizations and savings plans such as the ability to use EC2 spot instances that offer up to 90% discounts over the on-demand instances, Graviton2 support¹ and performance-optimized² runtime engines³. EMR offers managed scaling⁴ that automatically increases or

decreases the number of compute units in your cluster based on your workloads. So you only pay for the resources that you actually use and not for any idle servers. EMR also offers EMR file system (EMRFS)⁵ to efficiently read and write very large datasets to S3, saving you the storage costs associated with the 3x data replication in HDFS. With EMR, you only pay for the resources you use, such as EC2 instances, storage, and network bandwidth. This allows you to



scale up or down as needed, reducing costs compared to a traditional, static Hadoop cluster. It allows you to automatically scale your cluster based on workload demands, reducing the cost of underutilized resources. We can decouple storage and compute easily here so by keeping data in Amazon S3, we can spin up EC2 instances for required low compute capacity to save cost effectively.

Simplified maintenance:

Migrating to EMR can simplify the maintenance and management of your big data applications. EMR automates operations such as provisioning, bootstrapping, and configuring the cluster hardware and software. EMR also offers a new option called EMR Serverless⁷ that allows you to run open-source big data frameworks like Spark and Hive without having to configure, manage, maintain, or scale any hardware. EMR Serverless also takes care of security, OS patching and software updates.

Integration with other AWS services:

EMR integrates seamlessly with larger AWS ecosystem. For example, EMR integrates with S3 for data storage, Amazon Redshift⁸ for data warehousing, AWS Glue data catalog⁹ for metadata operations and Amazon

Sagemaker¹⁰ for interactive data processing and ML training, which makes it easy to build and run your big data applications and other workloads on a wide variety of AWS services based on your use case requirements.

Flexibility:

EMR Instance Fleets¹¹ offers a wide variety of provisioning options for both on-demand and spot EC2 instances. You can choose up to 30 instance types per fleet when you create your EMR cluster to ensure capacity. EMR also offers deployment options such as EMR on EKS¹² (Kubernetes) and EMR Serverless. You can switch from one deployment option to another without having to change your application code since all the options use the same underlying data processing engines.

Security:

EMR provides several options to secure your cluster and data. For data protection, EMR offers at-rest and in-transit encryption. You can enable Kerberos authentication with or without cross-realm trust to Active Directory. EMR integrates with Apache Ranger and AWS Lake Formation for fine-grained, row and column level authorization. You can audit API calls through Amazon CloudTrail or Ranger. With respect to infrastructure security, AWS has a strong track record for



security and access control. As a managed service, EMR is designed to meet the security and access control requirements of the enterprise customers. EMR is also SOC, PCI, FedRAMP and HIPAA compliant.

Resiliency:


EMR can access the same data from S3 and same metadata from AWS Glue Data Catalog or Amazon RDS from all the subnets in a VPC. So, in an unlikely event of an Availability Zone failure, you can easily launch a new EMR cluster in a different subnet and re-initiate your data processing jobs. FedRAMP and HIPAA compliant.

While migrating to EMR offers you all these benefits and more, the migration process itself can be complex and time-consuming, and requires significant planning and resources. It is important to carefully evaluate the benefits and drawbacks of this migration before deciding whether it is the right choice for your organization.

Here is a high-level data flow for migrating a Hadoop system to AWS EMR:

- 1. Data collection:** The first step in the migration process is to collect and organize the data that you want to move to EMR. This may involve extracting data from on-premises systems or other cloud environments and transforming it into a format that can be processed by EMR.
- 2. Data storage:** The next step is to store the data in a location that is accessible to EMR. EMR is designed to work with S3 as its primary storage via EMRFS. Hence, a standard migration typically involves transferring your on-premises HDFS data to S3.
- 3. Data processing:** Once the data is migrated to S3, you can use EMR to process the data using Hadoop, Spark, or any other big data framework. EMR allows you to spin up a cluster of EC2 instances to run your big data workloads.
- 4. Data analysis:** After the data has been processed, you can use tools such as Amazon Athena or Amazon Redshift to analyze and query the data. You can also use other AWS services, such as Amazon QuickSight, to visualize and explore the data.
- 5. Data distribution and visualization:** Finally, you can use EMR or other AWS services to disseminate the processed and analyzed data to the appropriate stakeholders or systems. This may involve transferring the data back to the on-premises systems or other cloud environments or making it available to users through a dashboard or an API. Amazon QuickSight can be used for

Architecture Patterns for Hadoop to EMR Migration



Patterns	(1) Re-host	(2) Lift and Shift	(3) Rip and Replace
Description	Re-host the application to cloud	Migrate to cloud with minor or no modification and refactoring for the workloads using cloud native services	Redesign and rewrite the solution
Target Platform	Same as on premise, but hosted on cloud	Data platform and data processing use cloud native solution	Data platform and data processing use best of breed solution.
Examples	Rehost Cloudera Hadoop, MongoDB to AWS	Migrate HDFS data to S3, move data processing to AWS native services (EMR)	Build new solution on AWS. Redesign new data processing using best of breed solution.
Duration	3 to 6 months	6 to 12 months	6 to 12 months
Advantages	Quick Migration Minimal risk Cost saving	<ul style="list-style-type: none"> • Simplicity • Low Risk • Low Disruption 	Improved functionality Increased efficiency Security Enhancements

We have 4 main steps to migration Hadoop to EMR:

- Data Migration
- Hive Migration
- Hadoop workload (Jobs) migration
- Post Migration

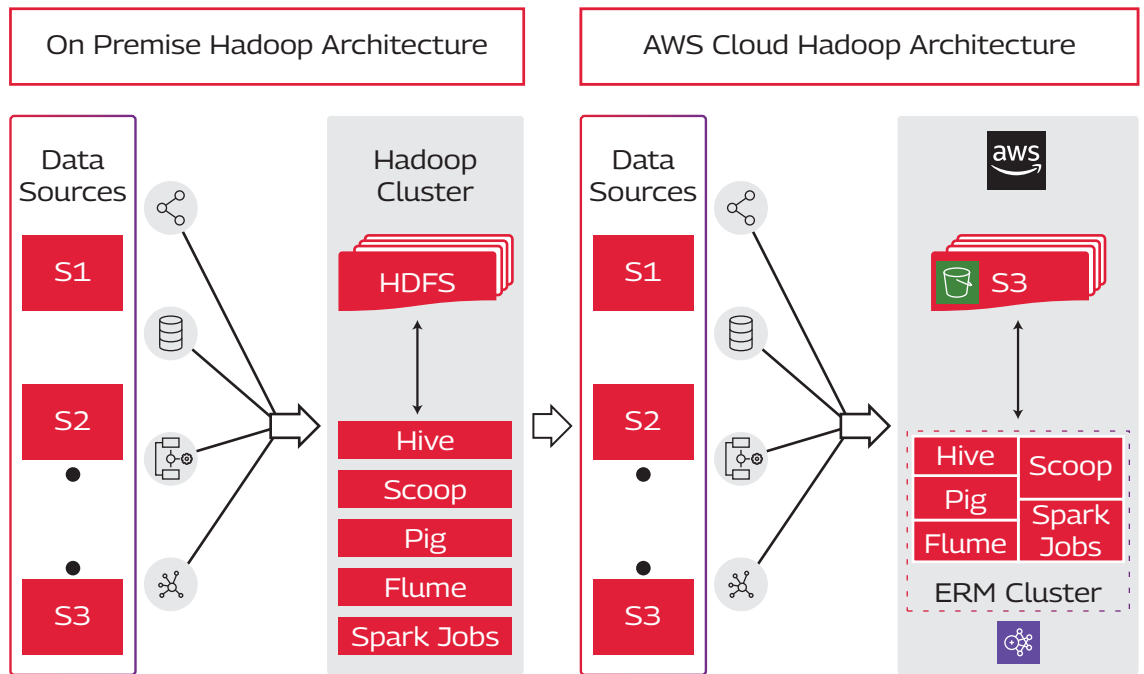


Figure 1: Migration Overview

Data Migration:

For data migration from Hadoop to AWS, we can use tools like distCP or sqoop. Extracted data can be stored on S3 which is scalable and durable storage solution. AWS Snowball is a secure and fast data transfer solution that can be used to copy large amounts of HDFS data from on-premises to AWS for history load. With Snowball, you can transfer petabyte-scale data sets in a matter of days, compared to months when transferring data over the network. Snowball uses a physically secure storage device that is shipped to your location, where you copy the data onto the device, and then ship it back to AWS for import into S3.

Push Method



Pull Method

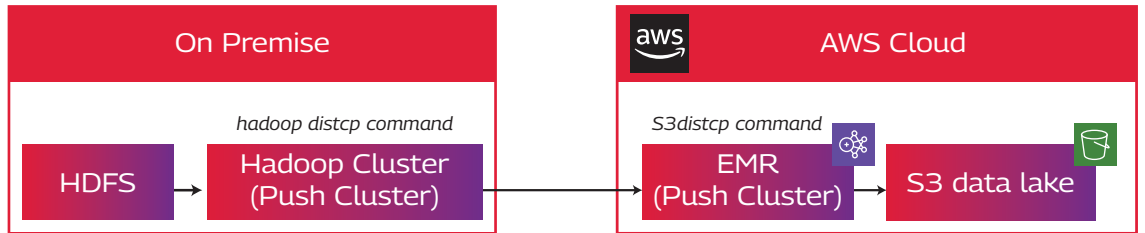


Figure 2: Data Migration Methods

In case of push method, we can have hadoop cluster on which distCP commands will be executed; and in case of pull method, we will have EMR clusters on which these commands will pull data from on-premises.

Hive Migration:

This is one time migration where Amazon RDS or AWS Glue can be used as the Hive metastore in AWS. We need to export data from Hive and import to RDS/Glue data catalog.

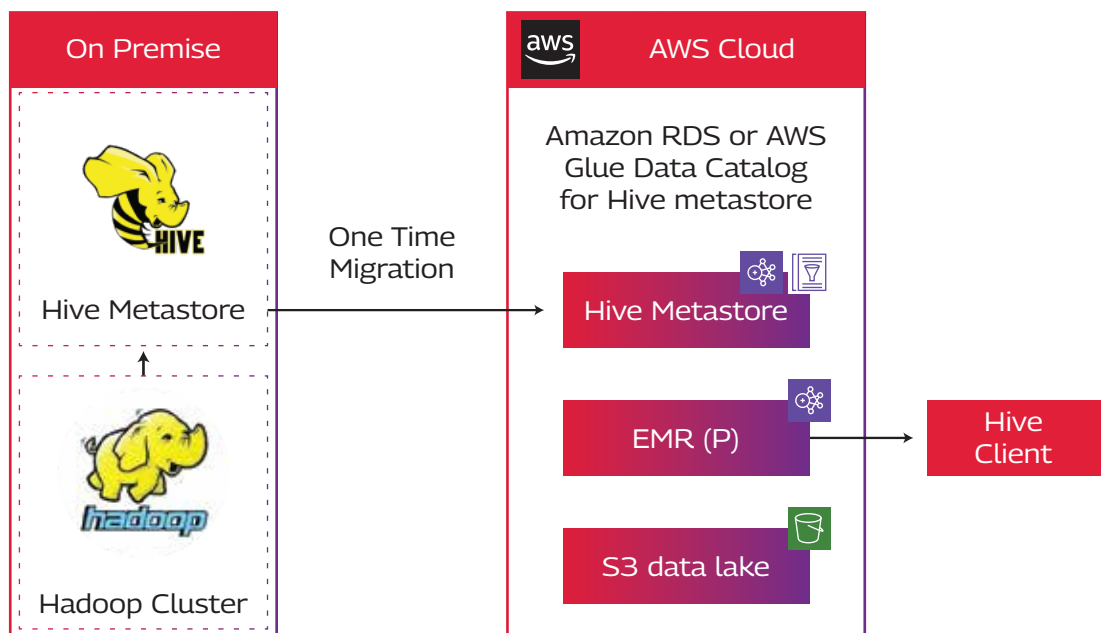


Figure 3: Hive metastore migration

Jobs Migration:

Below are the important points to be taken care during jobs migration.

- Choose appropriate size and configuration for EMR clusters
- Ensure dependent data and metadata in HDFS/Hive is migrated
- Migrate code to repoint the data sources from `hdfs://` to `s3://`
- Cloud storage is immutable. So, if any incompatible operations such as merge exist in the code, the code and logic must be rewritten

Post Migration:

We need to make sure data is validated and reconciled during and post migration process. We can use Amazon Cloudwatch and Cloudtrail to view and customize jobs and resources. As required, scale and resize EMR clusters when data volume changes. There will be continuous process of optimization of the application.

Conclusion:

Migrating Apache Hadoop to Amazon EMR can provide significant benefits to organizations looking to improve their big data processing capabilities. Amazon EMR offers a scalable and cost-effective solution that eliminates the need for complex infrastructure management and allows users to focus on data analysis and insights. Overall, Amazon EMR can provide a powerful platform for organizations seeking to leverage big data analytics to gain insights and drive business value. By following the guidelines and recommendations outlined in this whitepaper, organizations can achieve a seamless and successful migration to Amazon EMR and take full advantage of its capabilities for big data processing and analysis.

Authors:



Satyawan Kadalag

Satyawan Kadalag is Principal solution Architect at Tech Mahindra. having total IT experience of 16 years. He's responsible for the consultation and design of customers' cloud solution architectures. He has extensive experience in legacy and cloud databases.



Piyush Patra

Piyush Patra is a Partner Solutions Architect at Amazon Web Services. He helps partners with their Analytics journeys supporting them for achieving differentiation, enabling their technical and sales teams on AWS native analytics services, setting up partners for success by helping design solutions adhering to recommended best practices. In his spare time, he loves to cook for friends and family and explore different cuisines.



Pankaj Bajaj

Pankaj Bajaj is Global Practice head for Data-on-Cloud and Visualization COE at Tech Mahindra. having total IT experience of 22 years. He is responsible for Pre-Sales, Solutioning, Competency development & Platform Development on Various Hyperscale's including AWS.



Veena Vasudevan

Veena Vasudevan is a Senior Partner Solutions Architect and an EMR specialist at AWS focusing on Big Data and Analytics. She helps customers and partners build highly optimized, scalable and secure solutions, modernize their architectures, and migrate their Big Data workloads to AWS.



Sathish Arumugam


Sathish Arumugam is a Sr. Partner Solution Architect at Amazon Web Services. Sathish is a Containers TFC AoD and AWS Data Analytics Specialty certified Solutions Architect. He helps partners and customers with the AWS Well Architected best practices in

their cloud transformation journey and the business critical workloads hosted on the AWS cloud. In his spare time, he loves to spend time with his family and pursue his passion for Cricket.

Apache Hadoop to EMR migration can be complex and costly. To have optimized, streamlined migration solution which adheres to AWS standard Well architected pillars, please contact us.

References:

1. *Amazon EMR now provides up to 35% lower cost and up to 15% improved performance for Spark workloads on Graviton2-based instances. (n.d.). Amazon Web Services, Inc.*
<https://aws.amazon.com/about-aws/whats-new/2020/10/amazon-emr-provides-lower-cost-improved-performance/>
2. *Run Apache Spark workloads 3.5 times faster with Amazon EMR 6.9 | Amazon Web Services. (2023, January 30). Amazon Web Services.*
<https://aws.amazon.com/blogs/big-data/run-apache-spark-workloads-3-5-times-faster-with-amazon-emr-6-9/>
3. *Introducing Amazon EMR Managed Scaling - Automatically Resize Clusters to Lower Cost | Amazon Web Services. (2022, August 11). Amazon Web Services.*
<https://aws.amazon.com/blogs/big-data/introducing-amazon-emr-managed-scaling-automatically-resize-clusters-to-lower-cost/>
4. *EMR File System (EMRFS) - Amazon EMR. (n.d.).*
<https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-fs.html>

- 
5. *EMR File System (EMRFS) - Amazon EMR. (n.d.-b).*
<https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-fs.html>
 6. *Open Source Big Data Analytics | Amazon EMR Serverless | Amazon Web Services. (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/emr/serverless/>*
 7. *Using Amazon Redshift integration for Apache Spark with Amazon EMR - Amazon EMR. (n.d.).*
<https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-spark-redshift.html>
 8. *Use the AWS Glue Data Catalog as the metastore for Spark SQL - Amazon EMR. (n.d.).*
<https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-spark-glue.html>
 9. *Use the AWS Glue Data Catalog as the metastore for Spark SQL - Amazon EMR. (n.d.-b). <https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-spark-glue.html>*
 10. *Use the AWS Glue Data Catalog as the metastore for Spark SQL - Amazon EMR. (n.d.-b). <https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-spark-glue.html>*
 11. *Configure instance fleets - Amazon EMR. (n.d.).*
<https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-instance-fleet.html>
 12. *What is Amazon EMR on EKS? - Amazon EMR. (n.d.).*
<https://docs.aws.amazon.com/emr/latest/EMR-on-EKS-DevelopmentGuide/emr-eks.html>

