Data Lake Architecture - A Comprehensive Design Document

Medical Data Processing Company

Tracker

Revision, Sign off Sheet and Key Contacts

Change Record

Date	Author	Version	Change Reference
08/19/2024	Phuong Cao	0.1	Initial draft

Reviewers / Approval

Name	Version Approved	Position	Date
	1.0	Udacity Reviewer Enterprise Data Lake Architect	

Key Contacts

Name	Role	Team	email
	Data Architect	Medical Data Processing	

Table of Contents

1. Purpose
    1.1 Summary
    1.2 Target Audience
    1.3 In-scope
    1.4 Out of-scope
2. Requirements
    2.1 Requirements of Data Lake solution
    2.2 Existing Technical Environment
    2.3 Current Data Volume
    2.4 Business Requirements
    2.5 Technical Requirements
3. Data Lake Architecture design principle
4. Assumptions
5. Data Lake Architecture for Medical Data Processing Company
6. Design Considerations and Rationale
    6.1 Ingestion layer
    6.2 Storage layer
    6.3 Processing layer
    6.4 Serving layer
    6.5 Data Catalog, Governance and Security
7. Conclusion
8. References

1. Purpose

1.1 Summary

The "Medical Data Processing" company has experienced hyper growth over the past 3 years and current technology stack cannot catch up with the volume of data increasing and spike issue on server processing that lead to a single point of failure on server that hosting critical customer data. The CTO desired the data lake solution will address the company's current and future challenges so this material is the design document that provides the Data-lake Architecture Design solution proposal for this problem.

1.2 Target Audience

"Medical Data Processing" company :

- C-level
- Data Engineers and Database Administrators
- Data and Business Analysts
- Data Scientist

1.3 In-Scope Items

- Requirements, Assumptions, Design of the data architecture.

1.4 Out-of-Scope Items

- System Implementation, Data Governance and Stewardship, Machine Learning.

2. Requirements

2.1 Requirements of Data Lake Solution

The Data Lake Architecture should provides:

- The High availability, reliability and resiliency system.
- The system easily in scale that meet the current growth in data volume and processing performance.
- Removing the data silos within the company and provide the single source of truth.

2.2 Existing Technical Environment

- 1 Master SQL DB Server
- 1 Stage SQL DB Server
    - 64 core vCPU
    - 512 GB RAM
    - 12 TB disk space (70% full, ~8.4 TB)
    - 70+ ETL jobs running to manage over 100 tables
- 3 other smaller servers for Data Ingestion (FTP Server, data and API extract agents).
- Series of web and application servers (32 GB RAM Each, 16 core vCPU)

2.3 Current Data Volume
- Data coming from over 8K facilities.
- 99% zip files size ranges from 20 KB to 1.5 MB.
- Edge cases - some large zip files are as large as 40 MB.
- Each zip files when unzipped will provide either CSV, TXT, XML records
- In case of XML zip files, each zip file can contain anywhere from 20-300 individual XML files, each XML file with one record.

- Average zip files per day: 77,000
- Average data files per day: 15,000,000
- Average zip files per hour: 3500
- Average data files per hour: 700,000
- Data Volume Growth rate: 15-20% YoY

2.4 Business Requirements
- Improve uptime of overall system
- Reduce latency of SQL queries and reports.
- System could be reliable and fault tolerance.
- Architect should scale as data volume and velocity increase.
- Improve business agility and speed of innovation through automation and ability to experiment with new frameworks.
- Embrace open source tools, avoid proprietary solutions which can be lead to vendor lock-in.
- Meta-data driven design - a set of common scripts should be use to process different types of incoming data sets rather than building custom scripts to process each type of data source.
- Centrally store all of the enterprise data and enable easy access.

2.5 Technical Requirements
- Ability o process incoming files on the fly (instead of nightly batch loads today).
- Separate the meta-data, data and compute/processing layers.
- Ability to keep unlimited historical data.
- Ability to scale up processing speed with increase in data volume.
- System should be sustain small number of individual node failures without any downtime.
- Ability to perform change data capture (CDC), UPSERT support on a certain number of tables.
- Ability to drive multiple use cases from the same data-set, without the need to move the data or extract the data.
    - Ability to integrate with different ML frameworks such as TensorFlow.
    - Ability to create dashboards using tools such as PowerBI, Tableau or Microstrategy.
    - Generate daily, weekly, nightly reports using scripts or SQL.
- Ad-hoc data analytics, interactive querying capability using SQL.

3. Data Lake Architecture design principles

Data lakes are the cornerstones of modern big data architecture and centralized repositories of Huge of data volumes in its original format - its be structured , semi-structured or unstructured at scale that stored before a specific use cases has been defined. It enables businesses to create visualizations and dashboards and perform big-data processing, machine learning and real-time analytic to make informed business decisions. The four essential principles for effectively architecture the data lake.

Keep it simple:

- The more complex on data lake is, the more difficult it will be to use. Keep it as simple as possible so that everyone on team can easily understand it.

Centralized Data Storage:

- There's only one point to discover everything about you is that centralize everything in one place. When store all of data in one place, it's much easier to access and analyze. Also this can help you make better decisions about your business. Additionally, storing all your data in one place makes it easier to keep track of and comply with regulations.

Keep your architecture open:

- Keeping data accessible means avoiding vendor lock-in, or over-reliance on a single tool or database (and the gatekeepers that come with it). BY Storing the data in open formats like Arvo and Parquet which are standard, well-known and accessible by different tools.

Scalability:

- Data Lakes can be expanded effortlessly to hold huge amounts of data from diverse sources and be able to scale up or down as needed, depending on the amount of data you're storing and processing.

Data Governance and Security:

- The very important consideration one when designing your data lake is security. Make sure that the data in data lake is safe and secure, and that only authorized users can access it.
- The data governance model should be define early on. The most popular option is the "centralized governance model". With this model, all the data in the data lake is centrally managed by a single team with responsibility of the ensuring that the data is properly organized and labeled, and that only authorized users can access it.

Storage technology:

- The type of storage technology choose will affect the performance of data lake involves selecting scalable and cost-effective solutions for storing large amounts of data "On-Premises or Cloud Storage".

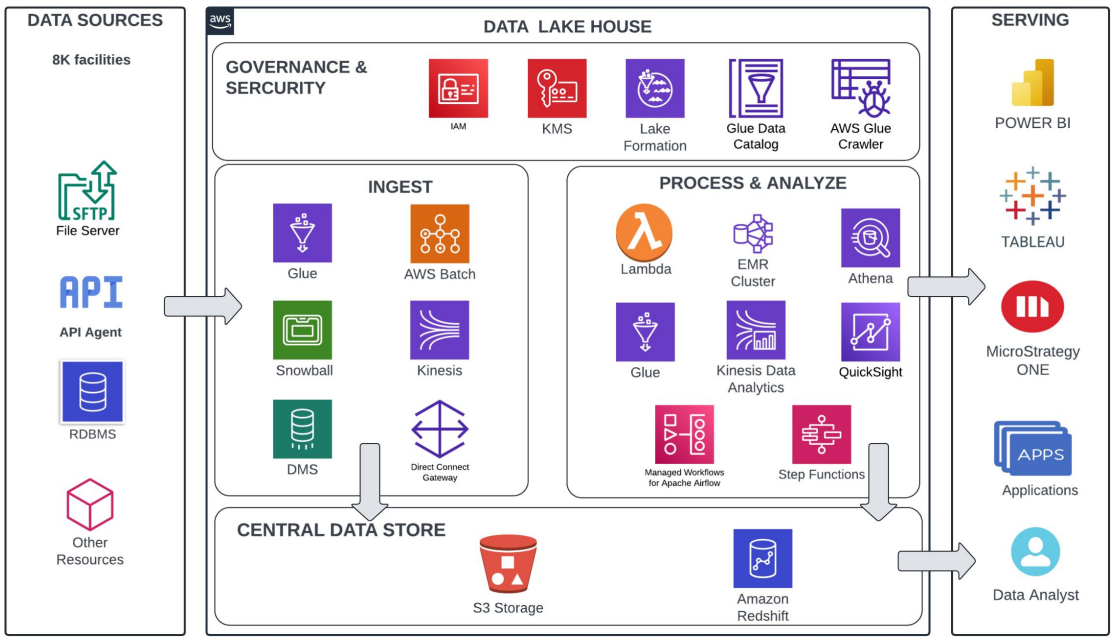
4. Assumptions

- The on-premise data lake requires both hardware and software management requires physical infrastructure with advanced investment beside limits capacity and scaling abilities. So Data lake infrastructure is preferred with cloud-based rather than on-premise.
- All of data should move to cloud-based storage.
- The Maser, Slave server and other (FTP Server, data and API extract agents) should move to cloud to utilize the flexibility and reliability, increased performance and efficiency, and helps to lower IT costs.

5. Data Lake Architecture for Medical Data Processing Company

Data Lake that hosting a data lake in the cloud brings all the benefits of cloud computing to data storage and processing. Cloud data lakes offer scalability, flexibility, accessibility, and cost-effectiveness. The core components of data lake architecture includes 6 key layers.

- Centralized storage with scalable and cost-effective solutions for storing large amounts of data as per the business needs.
- Data ingestion is the process of importing data into the data lake from various sources to the lake before undergoing further processing.
- Processing and analyzing at big data scale involves transforming raw data into meaningful insights using various frameworks, tools and programming languages.
- Data Catalog by organizing and managing to make data discoverable and understandable.
- Governance and Security involves managing data quality, compliance, and lifecycle and make sure protecting sensitive data through encryption, authentication, and access control.
- Serving the processed data to Data visualization and business intelligence tools and consumer applications …



6. Design Considerations and Rationale

6.1 Storage Layer:

The data storage layer is where the ingested data resides and undergoes transformations to make it more accessible and valuable for analysis. This layer is divided into different storage section for ease of management efficiency includes: **Raw , Transformation, Processed** data store. However these data section should be store in the centralized data store.

- Raw data store: the ingested data lands in what is commonly referred to as the raw or landing zone. At this stage, the data is in its native format (CSV, TXT, XML …). The raw data store acts as a repository where data is staged before any form of cleansing or transformation.

- Transformation data store: After residing in the raw zone, data undergoes various transformations. This section is highly versatile, supporting both batch and stream processing.

- Processed data store: Post-transformation, the trusted data can be moved to another zone known as the refined or conformed data zone. Additional transformation and structuring may occur to prepare the data for specific business use cases. The refined data is what analysts and data scientists will typically interact with. It's more accessible and easier to work with, making it ideal for analytics, business intelligence, and machine learning tasks.

To satisfy the 20% YoY Data Growth rate of system so the cloud storage (Amazon S3, Azure Blob Storage, Google Cloud Storage) should be chosen that provides the unlimited scalability ratter than on-primes solution. Amazon S3 storage solution selection is a better choice in terms of the cost and performance and easy to use then other considered such as the cloud storage from Microsoft and Google. It is as the primary persistent data store provide 99.999999999% (11 9's) of durability and easy to Backup and Restore Critical Data that meet Recovery Time Objectives (RTO), Recovery Poin- Amazon t Objectives (RPO), and compliance requirements with S3's robust replication features by replication functionality, data protection with AWS Backup, and various AWS Partner Network solutions.

- The essential component of a data lake built on Amazon S3 is the Data Catalog . The metadata repository stores metadata information (data about data) that encapsulates the different properties, history, origin, versions, and other information about a data asset in highly structured fields – used primarily for tracking, classification, and analysis. and provides an interface to query all assets stored in data centralized lake S3 buckets. The Data Catalog is designed to provide a single source of truth about the contents of the data lake.
- Amazon S3 is an object storage service that perfectly to store vary of format type :

- JSON, CSV, XML, text files or images .. that stored upon ingestion in the raw data zone.

- Open source format types: Apache Parquet and ORC that are standard, well-known and accessible by vary tools that store in Transformation and processed data zone.

- S3 is secure, private, and encrypted by default that protect data with unmatched security, data protection, compliance and access control capabilities and also supports numerous auditing capabilities to monitor access requests to S3 resources.

When SQL-based to analyze structured and semi-structured data across data is needed, Redshift is added to the storage layer as the warehousesing that is a fast and fully managed with the best price performance at any scale. Redshift makes all simple and cost-effective to analyze all your data efficiently using your existing business intelligence tools with provisioned cluster and Serverless options. Beside Amazon **Redshift Spectrum** enables to run queries against exabytes of unstructured data in Amazon S3, with no loading or ETL required.

6.2 Ingestion Layer:

Data ingestion is the process of collecting data from multiple sources and transferred to the lake raw without processing into data lake (Amazon S3). Data can be ingested in batches or in real-time or Stream Processing.

- The beginning is data migration to cloud. AWS provide **Direct Connect** and **Snowball** solution. These support to lift up and shift hundreds of terabytes of data quickly into AWS using Amazon-provided secure appliances for secure transporf data into or out of AWS.
- The raw CSV, TXT, XML records that provided by thousand of medical facility from FTP server could be loading to S3 by vary of tools and cloud native services like as **Amazon Lambda** and scheduling by Amazon **EventBridge**. For the large data size could be ingested by **AWS Batch** or **AWS glue job** .
- The current traditional RDBMS (Oracle, SQL Server, and MySQL) also easily import and sync up by AWS Database Migration Service (DMS) that native change data capture (CDC) support.
- Serverless Streaming Data Service - Amazon **Kinesis** and **Kinesis Firehose** enable Collect, process and analyze real-time, streaming data scaling up to 10,000 MB/s throughput with a single API call.

Open-source tools including Apache Sqoop, Flume, Kafka and Nifi was considered however select to use managed services Amazon provided rather than the self-managed ones.

Apache Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data , events (etc...) from various sources to a centralized data store. Flume is a highly reliable, distributed, and configurable tool. It is principally designed to copy streaming data (log data) from various web servers to HDFS. Flume leverages Hadoop so Flume jobs are internally converted into MapReduce. Apache Flume architecture includes three components: Source, Channel, and Sink. A **Flume Source** is the component of Flume Agent which consumes data (events) from data generators like a web server and delivers it to one or more channels. Flume source receives an event from a data generator, it stores it on **one or more Flume Channels** and stores them till **Flume Sinks** consume them.

Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases (MySQL, Oracle, DB2) . Sqoop also works on top of Hadoop HDFS and a Sqoop job it gets converted to a MapReduce job when submitted.

Apache Kafka is an open source distributed event streaming platform consisting of servers and clients that communicate via a high-performance TCP network protocol with horizontally scalable, fault-tolerant, fast framework that is used by thousands of companies, for high performance data pipelines and streaming analytics for building real time data pipelines.

Apache NiFi was built to automate the movement of data between systems. It provides real-time control that makes it easy to manage the movement of data between any source and any destination.

6.3 Processing and Analytics Layer:

Processing and analytics are the most importance layer in a data lake that allow various roles in your organization like data scientists, data developers, and business analysts to access data with their choice of analytic tools and frameworks. This includes open source frameworks such as Apache Hadoop, Presto, and Apache Spark, and commercial offerings from data warehouse and business intelligence vendors. Many of tools that provides by AWS provider that help to that processing and analytics jobs was done with at least management effort and cost.

AWS Lambda run code without provisioning or managing servers in response to triggers such as changes in data, shifts in system state, or actions by users. Lambda can be directly triggered by AWS services such as Amazon S3, DynamoDB, Amazon Kinesis Data Streams, Amazon Simple Notification Service (Amazon SNS), and Amazon CloudWatch, enabling you to build a variety of real-time data processing systems: Real-time file, Real-time stream processing, Extract, transform, load (ETL), Cron Job…

Amazon Kinesis includes four pieces of the Kinesis platform: Amazon Kinesis Data Streams, Amazon Data Firehose, Amazon Managed Service for Apache Flink, Amazon Kinesis Video Streams that enable us to process, analyze or deliver real-time streaming data.

Amazon EMR provides cloud big data platform for processing vast amounts of data using open source tools such as Apache Spark, Apache Hive, Apache HBase, Apache Flink, Apache Hudi, and Presto EMR is easy to set up, operate, and scale your big data environments by automating time-consuming tasks like provisioning capacity and tuning clusters can run petabyte-scale analysis at less than half of the cost of traditional on-premises solutions and over 3x faster than standard Apache Spark.

AWS Glue is a serverless data integration service that makes it easy to discover, prepare, and combine data for analytics, machine learning, and application development. It provides both

visual and code-based interfaces to make data integration easier with all of the capabilities needed for data integration. Glue allow automatic schema discovery, using AWS Glue crawlers.

Amazon Athena is a good tool for interactive one-time, ad-hoc SQL queries against data on Amazon S3. Athena is serverless, so there is no infrastructure to setup or manage, and can start analyzing data immediately with full ANSI SQL support and works with a variety of standard data formats, including CSV, JSON, ORC, Apache Parquet, and Apache Avro. Athena integrates with Amazon QuickSight for easy visualization.

Amazon QuickSight is a scalable, serverless, embeddable, machine learning-powered business intelligence (BI) service built for the cloud easy for all employees within an organization to build visualizations, perform ad hoc analysis, and quickly get business insights from their data, anytime, on any device.

The align open source tools such as: Hadoop MapReduce, Apache Pig, Hive and Spark as the processing tools were not select due the large of effort on self-management.
Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. Java-based MapReduce is basically a processing method and a model for a distributed computing program. A MapReduce program usually executes in three stages: Map , Shuffle and Reduce stage.

- Map stage - in this, the input data is split into fixed sized pieces known as input splits. Each input split is passed through a mapping function to produce output values.
- Shuffle stage - the output values from the map stage is consolidated in the next stage, which is the shuffle stage.
- Reduce stage - the output from the shuffle stage is are combined to return a single value and is stored in the HDFS.

A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework controls every aspect of data-passing, including scheduling tasks, monitoring them and re-executes the failed tasks. The majority of computing is done on nodes having data stored locally on drives, which lowers network traffic. After the assigned tasks are finished, the cluster gathers and minimizes the data to create the necessary results, then delivers it directly to the Hadoop servers.

Apache Pig is an open-source Apache library that runs on top of Hadoop for analyzing large data sets that consists of a high-level language for expressing data analysis programs. The library takes SQL-like commands written in a language called Pig Latin that would be MapReduce programs.

Apache Hive is an open-source, data warehouse, and analytic package that runs on top of a Hadoop cluster used to efficiently store and process large datasets. Hive is ability to query large datasets, leveraging MapReduce with a SQL-like interface without write really long and complicated MapReduce code in Java so that Hive SQL queries will be converted to MapReduce code behind the scenes.

Apache Spark is a multi-language engine in Scala, Python and Java for executing data engineering, data science, and machine learning on single-node machines or clusters. Howevers, Spark has several notable differences from Hadoop MapReduce that it has an optimized directed acyclic graph (DAG) execution engine and actively caches data in-memory, which can boost performance, especially for certain algorithms and interactive queries. Spark could do real-time as well as batch data processing use cases and provides 100 times faster than MapReduce in performance.

Ensure that multiple long-running ETL jobs run in order and complete successfully, without the need for manual orchestration we also consider to leverage on-hands orchestration services like as AWS Step Functions or Amazon Managed Workflows for Apache Airflow.

AWS Step Functions is a serverless orchestration service designed to facilitate the creation of visual workflows, enabling seamless coordination of AWS Lambda functions and other AWS resources.
Amazon Managed Workflows for Apache Airflow (Amazon MWAA) help use to orchestrates your workflows using Directed Acyclic Graphs (DAGs) written in Python. It can managing the fulfillment of orders, processing data in any complex sequence of tasks. It provides many plugins to run and monitor your DAGs that schedule, execute your jobs steps.

6.4 Serving Layer:

The serving layer consists of data that created by the process & analyze layer for querying by downstream consumers such as data scientists, data analysts, and other business users to leverage various tools, like Power BI, Tableau, and more to use their client apps to access the data stored in the data lake and all of its metadata. So that the data stored in this should be cleaned, enriched, and transformed so it can act as the "single source of truth" that users can trust.

All users can utilize the data in the lakehouse to carry out the analytics tasks they need, like building dashboards, visualizing data, running SQL queries, running machine learning jobs, and more. AWS offers a comprehensive suite of cloud-native solutions to allow consumers to make these done in most effectively like Amazon Redshift as data warehousing for OLAP reports or utilize Athena and Redshift Spectrum for SQL querying on demand on S3 directly with the central sharing data catalog. The open source frameworks such as Apache Hadoop, Presto, and Apache Spark, and commercial offerings from data warehouse and business intelligence vendors.

These allow to run analytics without the need to move your data to a separate analytics system that enable to break data silos and provide the discoverability, innovavation opportunity in business. Additionally, artificial intelligence (AI) and machine learning popular tools are available to build and integrate to support data in any specific business such as predictive analytics and deep learning by the AI services include Amazon Polly for text-to-speech, Amazon Lex for natural language processing and conversational bots beside that Amazon Rekognition for image identification and classification then the outputs can be served to other consumer.

6.5 Data Catalog, Governance and Security:

One decided to completely migrate all of data in to data lake for centralized storage three import aspect must be defined in advance. They are how to guarantee the meta data management, data quality and security.

Data Catalog is a collection of metadata, combined with data discovery and management tools, and it provides an inventory of data assets across all your data sources. It helps data consumers discover, understand, and consume data more productively. Also empowers data analysts and users to work in self-service mode to discover trustworthy data quickly. AWS Glue Data Catalog is the answer that provides a unified metadata repository across a variety of data sources and data formats, integrating with Amazon EMR as well as Amazon RDS, Amazon Redshift and Redshift Spectrum, Amazon Athena, AWS Lake Formation and any application compatible with the Apache Hive metastore . Glue Data Catalog can store and find metadata, keep track of data in data silos, and use that metadata to query and transform the data also provides comprehensive audit and governance capabilities with schema change tracking and data access controls, allowing you to audit changes to data schemas. Data Governance involves managing data quality, compliance, and lifecycle policy requirements regarding data usage ETL (Extract,

Transform, Load) is an important function to ensure that it is moved and understood properly. AWS Glue is an ETL engine that can be used to understand data sources, prepare data, and load it reliably to data stores. S3 Lifecycle configuration rules for objects that have a well-defined lifecycle to handle when data should be archived or deleted.

Data Security protect sensitive data through encryption, authentication, and access control that could be by leverage AWS Lake Formation and AWS Identity and Access Management (IAM) in the centralized catalog. Lake Formation, the centralized catalog can selectively share data resources Databases, tables, rows or column names must be specified and a resource can be shared to an specific organization, department or business by two methods Named resource method and tag-based access control. AWS KMS service provide a security layer for object that stored on S3 by encrypted by default include in rest and in transit.

7. Conclusion

Cloud-based Data lake Architecture are about much more than storing data at scale in a cost-efficient way. This provide a data source for analytics, resulting in valuable insights and better business decisions. Meta-data driven is the key to put all of business data in the one place under the centralized management provides company with more context than ever before, allowing our business to get a step ahead of the competition.

8. References

https://vitalflux.com/data-lake-design-principles-best-practices/
https://www.geeksforgeeks.org/data-lake-architecture-system-design/#best-practices-for-implementing-data-lake-architecture
https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/building-data-lake-aws.html
https://www.spiceworks.com/tech/devops/articles/what-is-metadata/
https://codilime.com/blog/what-is-cloud-data-lake-definition-and-use-cases/
https://www.cloudquery.io/blog/what-is-the-modern-data-stack
https://aws.amazon.com/products/

https://aws.amazon.com/step-functions/
https://aws.amazon.com/managed-workflows-for-apache-airflow/

https://flume.apache.org/
https://data-flair.training/blogs/flume-architecture/

https://kafka.apache.org/intro
https://aws.amazon.com/what-is/apache-kafka/?nc1=h_ls

https://nifi.apache.org/documentation/v1/

https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-sqoop.html

https://docs.aws.amazon.com/whitepapers/latest/big-data-analytics-options/introduction.html

https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html#Overview
https://www.geeksforgeeks.org/map-reduce-in-hadoop/

https://www.shiksha.com/online-courses/articles/what-is-mapreduce

https://pig.apache.org/
https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-pig.html

https://hive.apache.org/
https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hive.html

https://spark.apache.org/
https://aws.amazon.com/what-is/apache-spark/?nc1=h_ls

https://www.geeksforgeeks.org/data-lake-architecture-system-design/#4-data-cataloging
https://docs.aws.amazon.com/whitepapers/latest/best-practices-building-data-lake-for-games/data-cataloging.html