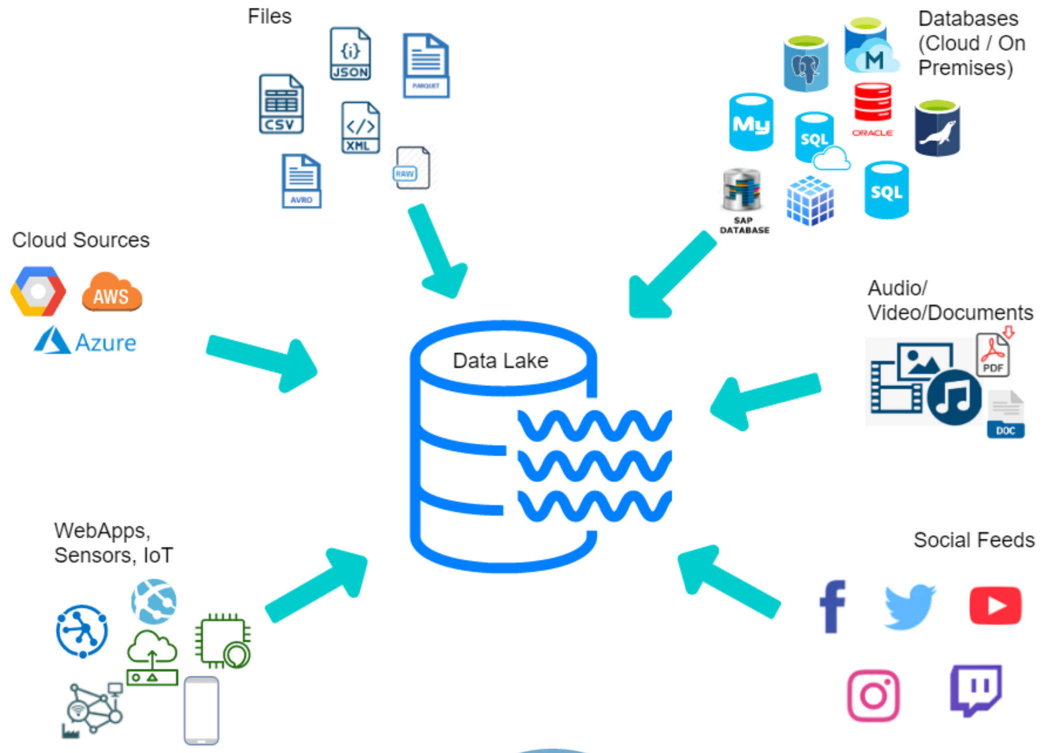
ETL Python Exercise

Section 2

<u>What is a Data Lake?</u>

A data lake is a central storage repository that stores large amounts of raw, granular data from several sources. This data can be structured, semi-structured, or unstructured data, and the data could comprise of several terabytes or petabytes. It is a place to store every type of data in its native format with no constraints on format or size. It offers high data quantity to increase analytic performance and native integration.

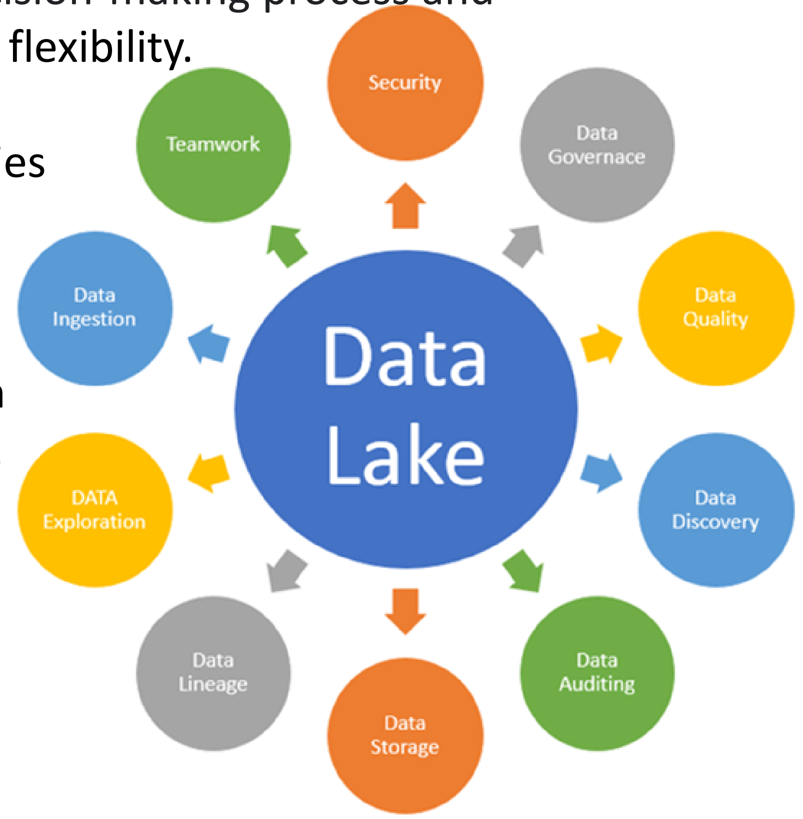
Data lake is a great alternate to data warehouse. Flat architecture and object storage feature, data lake makes it easier to locate and read data using the unique identifiers and metatags from different regions.



Section 2 continued…

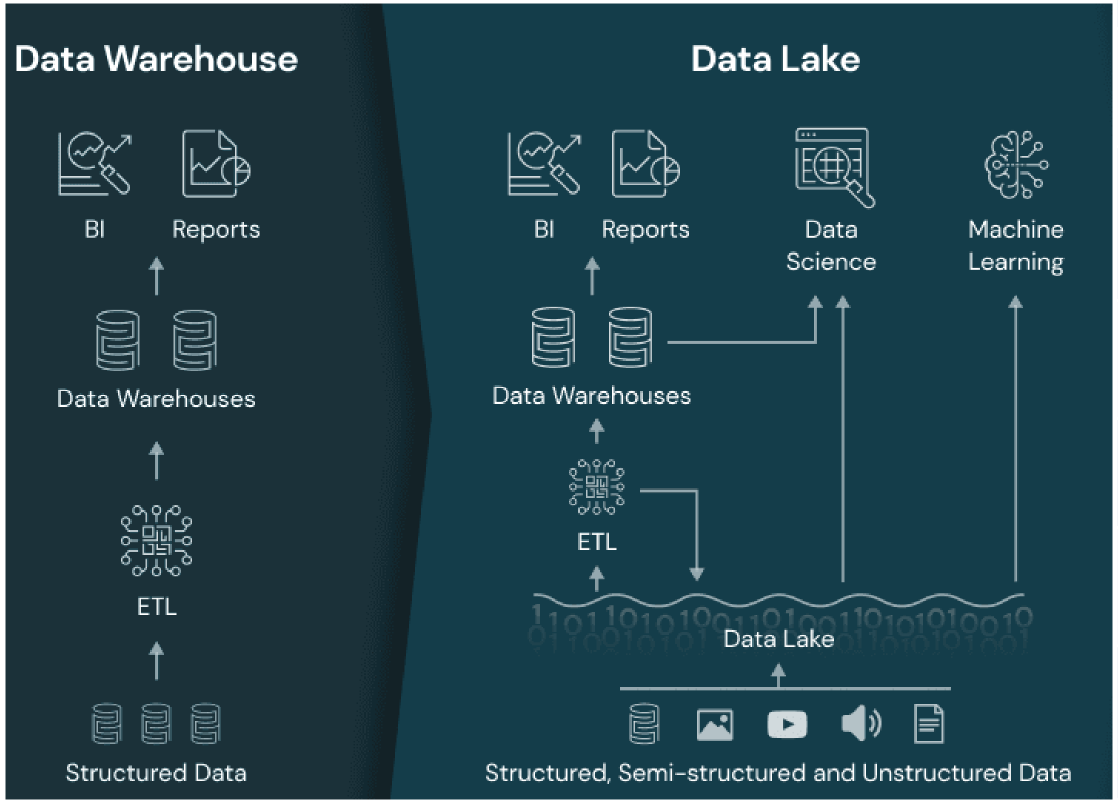
<u>Benefits</u>

- Data Lake offers business agility by offering key abilities that can improve decision-making process and analytics. Data lake also offer ease of Implementation in the form of schema flexibility.
- Data lakes can improve scalability and performance of a business and simplifies business management by helping in real-time decision analytics and adapts to changes easily.
- It serves as a single data platform and excels in its ability to derive value from unlimited data types stored all types of structured and/or unstructured form.
- It provides simplicity of data storage i.e., eliminating the need for data modeling at the time of storing them.
- Data lake provides various options and language support for analysis.



Section 2 continued…

<u>How does it differ from a data warehouse?</u>



Data Lake	Data Warehouse
A data lake is a vast pool of raw data.	A data warehouse is a repository for structured, and filtered data.
Data Lake has a flat architecture.	Data Warehouse architecture is hierarchal where data is stored in files and folders.
Data lake tends to ingest data very quickly and prepare it later as people access it.	With a data warehouse, on the other hand, you prepare the data very carefully upfront before you ever let it in the data warehouse
Data Lakes are highly agile and can be configured and reconfigured as needed.	Data warehouses are less agile and have fixed configuration.
Data lakes are designed for low-cost storage.	Data Warehouses are expensive and time-consuming.
Data Lakes use Extract-Load-Transform process.	Data Warehouse use Extract-Transform-Load process.
Schema is defined after data is stored for high agility and ease of data capture but needs work at the end of the process.	Schema is defined before data is stored. Needs work at the start of the process.

Section 2 continued…

Serverless Architecture

- Serverless computing is the abstraction of servers, infrastructure, and operating systems. Serverless architecture is an approach to software design that allows developers to build and run services without having to manage the underlying infrastructure. With a serverless development model, developers can build, deploy and run applications without having to manage servers. Once deployed, a serverless application will respond to traffic and automatically scale up or down as needed.

Examples:

AWS Lambda, Amazon SNS , Microsoft Azure Functions, Google Cloud Functions are all well-known examples of serverless services offered by the cloud providers.

Section 2 continued…

Pros and Cons of Serverless Architecture

Pros:

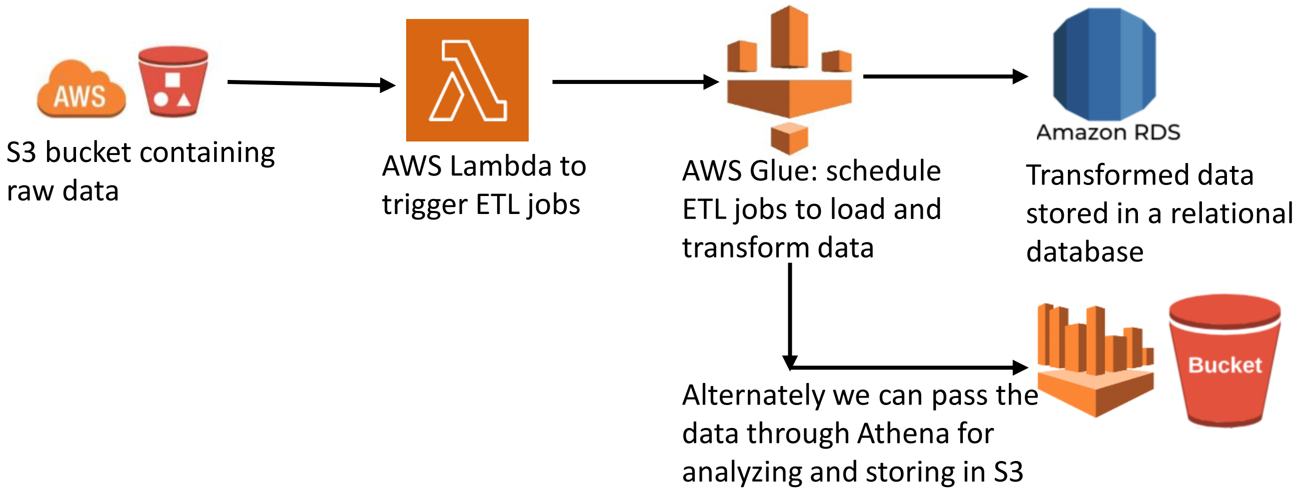
- No management of servers - Serverless computing runs on servers managed by cloud service providers.
- Self-maintaining. Clients don't need to perform server maintenance.
- Quicker time to market by allowing quick deployments and updates
- Highly available and scalability. Applications based on serverless architecture have endless potential for scalability.
- Flexible event-driven scaling (Auto-scaling)
- Micro billing- Users only pay for the execution time only.

Cons:

- Serverless computing introduces new security concerns. Data is shared with Cloud Vendors and privacy can be breached by giving control of your servers to third-party providers.
- Serverless architectures are not efficient for long-running processes like batch Jobs. You may end up paying more for computation time than when paying for a reserved instance.
- Performance impact raised by cold-start scenario when invoking the hosted function for the first time.

Section 2 continued…

<u>ETL pipeline for Section 1 using serverless AWS services</u>



S3 bucket containing raw data

AWS Lambda to trigger ETL jobs

AWS Glue: schedule ETL jobs to load and transform data

Transformed data stored in a relational database

Alternately we can pass the data through Athena for analyzing and storing in S3

S3 Bucket: S3 is an AWS service that provides object storage through a web service interface. For our scenario, S3 will be used for storing raw data (csv files)

AWS Lambda: AWS Lambda is an event-driven, serverless computing platform. In our case, it will be used for triggering AWS Glue ETL jobs whenever file is uploaded in S3.

AWS Glue: 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. In our case, Glue will perform the ETL and Transformation of raw data.

Amazon RDS: It is a relational database service that is super simple to setup and operate. PostgreSQL, MySQL, Maria DB, Oracle, SQL Server, and Amazon Aurora.

S3 Bucket/Athena: Transformed data will be stored in S3 Bucket or it can be dumped in MySQL compatible database i.e., Athena.

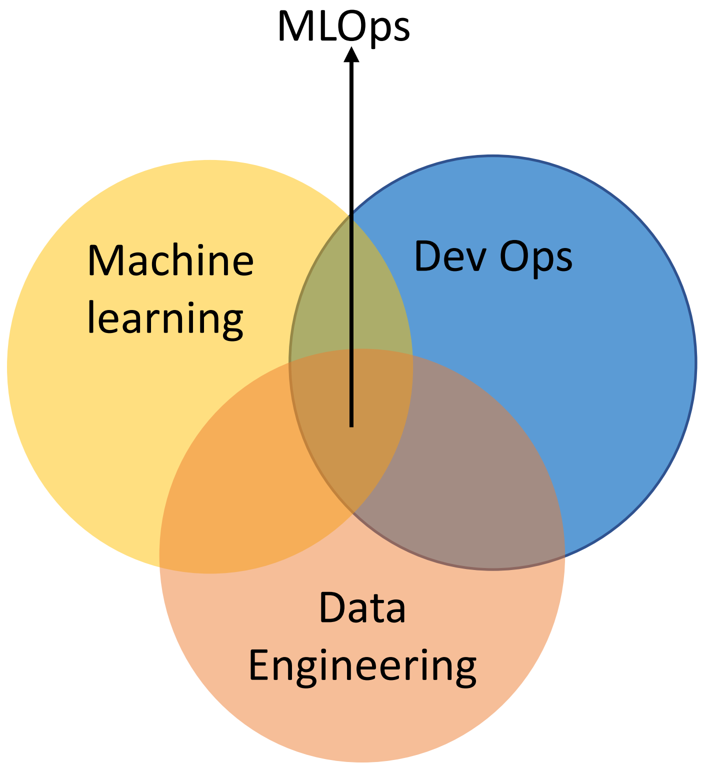
Section 2 continued…

<u>Describe modern MLOps</u>

MLOps is an ML engineering culture and practice that aims at unifying ML system development (Dev) and ML system operation (Ops). It is the fusion of traditional DevOps processes in the context of data science and machine learning.

MLOps is sort of a culture, framework or set of rules that allow us to perform and implement machine learning more efficiently. It includes many features of DevOps such as automation and testing but is focused on Machine Learning and Data Science by improving deployment and model monitoring.

MLOps is the combination of traditional DevOps processes in data science and machine learning environment. Combining the learnings and methodologies from DevOps with the concept of Data and ML yields an operating model termed MLOps or DataOps.



Section 2 continued…

Approach of Organizations

By adopting an MLOps approach, data scientists and machine learning engineers can collaborate and increase the pace of model development and production, by implementing continuous integration and deployment (CI/CD) practices with proper monitoring, validation, and governance of ML models.

Productionizing machine learning is difficult. The machine learning lifecycle consists of many complex components such as data ingest, data prep, model training, model tuning, model deployment, model monitoring, explainability, and much more. It also requires collaboration and hand-offs across teams, from Data Engineering to Data Science to ML Engineering.

As you would expect, it requires stringent operational rigor to keep all these processes synchronous and working in tandem. MLOps encompasses the experimentation, iteration, and continuous improvement of the machine learning lifecycle. MLOps can contribute efficiency, scalability, and risk reduction for the organizations.

Thank you!