Data Pipelines

In this lesson, we will learn about Data Pipelines.

WE'LL COVER THE FOLLOWING

What Are Data Pipelines?

> *Data pipelines* are the core component of a data processing infrastructure. They facilitate the efficient flow of data from one point to another & also enable the developers to apply filters on the data streaming-in in real-time.

Today's enterprise is data-driven. That makes data pipelines key in implementing scalable analytics systems.

Features Of Data Pipelines

Speaking of some more features of the data pipelines -

- These ensure smooth flow of data.
- Enables the business to apply filters and business logic on streaming data.
- Avert any bottlenecks & redundancy in the data flow.
- Facilitate parallel processing of data.
- Avoid data being corrupted.

These pipelines work on a set of rules predefined by the engineering teams &

the data is routed accordingly without any manual intervention. The entire

flow of data extraction, transformation, combination, validation, converging of data from multiple streams into one etc. is completely automated.

Data pipelines also facilitate parallel processing of data via managing multiple streams. I'll talk more about the distributed data processing in the upcoming lesson.

Traditionally we used *ETL* systems to manage all the movement of data, but one major limitation with them is they don't really support handling of real-time streaming data. Which is possible with new era evolved data processing infrastructure powered by the data pipelines.

What Is ETL?

If you haven't heard of ETL before, it means Extract Transform Load.

Extract means fetching data from single or multiple data sources.

Transform means transforming the *extracted* heterogeneous data into a standardized format based on the rules set by the business.

Load means moving the *transformed* data to a data warehouse or another data storage location for further processing of data.

The *ETL* flow is the same as the *data ingestion* flow. It's just the entire movement of data is done in batches as opposed to streaming it, through the data pipelines, in real-time.

Also, at the same time, it doesn't mean the *batch processing* approach is obsolete. Both real-time & batch data processing techniques are leveraged based on the project requirements.

You'll gain more insight into it when we will go through the *Lambda* & *Kappa* architectures of distributed data processing in the upcoming lessons.

In the previous lesson, I brought up a few of the popular data processing tools such as *Apache Flink*, *Storm*, *Spark*, *Kafka* etc. All these tools have one thing in common they facilitate processing of data in a cluster, in a distributed environment via data pipelines.

This Netflix case study is a good read on how they migrated batch ETL to Stream processing using Kafka & Flink

What is distributed data processing? How does it work? We are gonna look into it in the next lesson. Stay tuned.