SFL Scientific

What is a Data Lake?

- What are the benefits?
- How does a Data Lake differ from a Data Warehouse?
- How are they related?
- What are the benefits?

Data Lakes offer a flexible, scalable, and cost effective data integration solution

A **Data Lake** is a **centralized** and **scalable** repository which serves as a **landing zone** for all **incoming data streams**.

Data Lakes enable **persistent storage** of **raw data** in any format such as **unstructured** (images, video, audio), **semi-structured** (XML, JSON, HTML) or **structured** (csvs, relational data).

Data Lakes are **cost effective** since data is stored raw data and doesn't require processing.

Data Lake providers:

Microsoft Azure Data Lake *

amazon web services S3

databricks

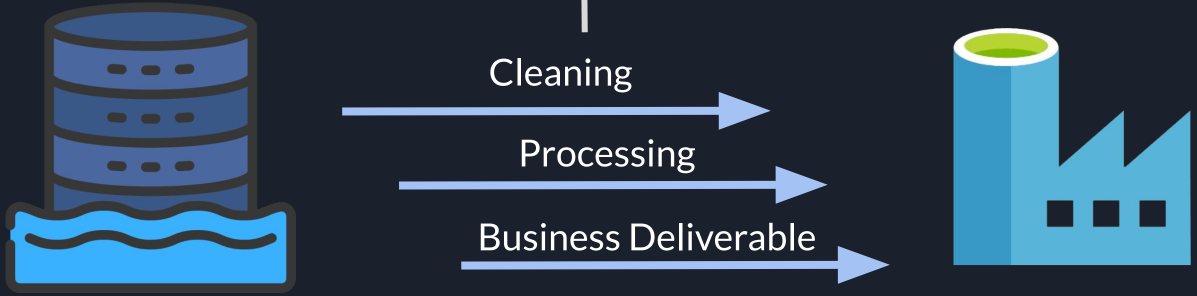
Google Cloud Platform

Data Lake

- Raw Data
- Schema free and highly flexible
- Difficult to query
- Data Engineer & Data Scientists
- Lower cost solution

Data Warehouse

- Processed and transformed data
- Structured schemas
- Optimized for query performance
- Marketing, Finance, and Business Intelligence
- Higher cost solution



Serverless Architecture

- **What are the benefits?**
- **What are the drawbacks?**

Serverless architectures provide a dynamic infrastructure solution that relieves the worry of managing resources

Serverless architectures **DO NOT** refer to compute without servers.

An **infrastructure paradigm** that enables developers to build and deploy applications **without managing** underlying servers.

This enables developers to **focus on code, system design, and functions** and removes the need to consider resource scaling and infrastructure management.

Serverless architecture is a **core concept of cloud computing** and **enables resources to scale** based on the dynamic needs of your business.

Events such as database changes, file uploads, or website requests **trigger the functionality** of the underlying microservices and applications.

Benefits

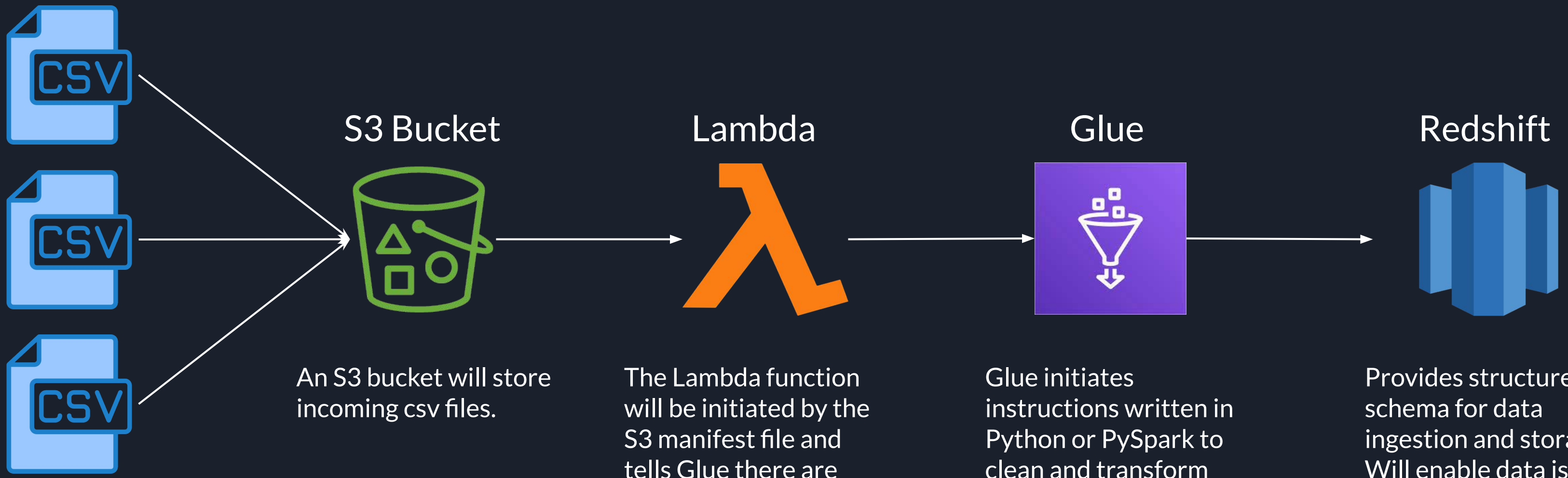
- Horizontal Scale & Redundancies
- Cost-Efficient
- Reduced Overhead
- Event-Driven Design
- Dynamic and Resource Efficient
- Quick Deployments

Drawbacks

- Requires an internet connection
- 3rd Party Dependency
- Challenges with Testing and Debugging
- Learning Curve
- Cold Starts
- Vendor Lock-In

AWS ETL Pipeline

Serverless AWS ETL Pipeline



An S3 bucket will store incoming csv files.

The Lambda function will be initiated by the S3 manifest file and tells Glue there are new files ready for processing.

Glue initiates instructions written in Python or PySpark to clean and transform the data files before loading into the data store.

Provides structured schema for data ingestion and storage. Will enable data is ready for consumption for business objectives.

The ETL Pipeline designed above will receive CSV files from various sources and use an S3 Bucket as an incoming data store for these raw data files. Upon CSV ingestion, the Lambda Function will notify AWS Glue there is a new file ready to be processed. Glue will kick off a pre-defined Python recipe to transform the data so it is usable and structured. Then, it can be stored in Redshift where it can be used to achieve its business deliverable.

Modern MLOps

Tools and Systems for end-to-end ML product management

MLOps describes the systems and tools to manage the lifecycle of your AI & ML products

The AI & ML product lifecycle consists of four keys areas: **Development, Deployment, Monitoring, and Management**. MLOps considers the requirements at each stage.

During **development**, Data Scientists need **managed data**, **development environments,** and clearly **communicated business objectives** of what they are aiming to deliver.

Deployment of models are critical and require **controlled release environments**, **automated testing**, and continuous integration and deployment **(CI/CD)** to ensure models will perform as expected.

Once models are in production, **monitoring is paramount**. Keeping track of both **model performance** and the delivery of **business objectives** will ensure business deliverables are being met.

Model management ensures throughput can **scale** with demand, that model **version controls** are in place, and there is sufficient **documentation** for technical support and customer understanding.

MLOps Tool Suite

Communication: Tools such as Slack and GChat ensure team members can share ideas freely and stay aligned on business objectives.

Development: Jupyter & DataBricks Notebooks give Data Scientists the tools needed explore data and test theories.

Data Management: Data Lakes and data registries such as Alation ensure data assets available for use are defined and accessible.

Environment Management: Docker and Terraform are tools that can ensure consistent deployment environments so models perform as expected and minimize dependency concerns.

CI/CD: Jenkins or GitHub Actions provide build cycles for testing code changes prior to being released into production.

Monitoring: Tableau, PowerBI, and AWS Quicksight provide dashboards that can provide on demand performance regarding model metrics and business KPIs.

Orchestration: Airflow, Luigi, and AWS Glue can offer data pipeline orchestration services to manage data processes.

Scaling: Cloud providers such as AWS, Azure, and GCP provide serverless architectures to ensure your infrastructure can scale with your product's demands.

Product Evolution: MLFlow offers a model registry to ensure updated models are delivering improvements and rollbacks can be clearly determined.

Documentation: Jira and Confluence provide the tools to communicate findings and procedures, while managing tasks and deliverables of your development teams.