**What is a Data Lake? Explain its benefits, how it differs from a data warehouse, and how it might benefit a client.**

A Data Lake is a large reservoir of data that stores raw, unprocessed data in structured, semi-structured and unstructured formats.

Key features of a data lake:

* Data lakes retail ALL data
  + Including the data that gets filtered out from the data warehouse and sometimes the raw data that is never used.
* Data lakes support all data types
  + Stores data in its raw form and transform it only when we’re ready to use it
* Datalakes support all users
  + Accessible to all teams; each team uses the data as they deem fit.
  + Data Scientists could use the raw data
  + Data Analysts could use a more structured version to derive insights
* Data lakes adapt easily to change
  + Since the data is in raw for, each team can make changes per their requirements, come up with a solution, explore data in novel ways to answer the question at hand
* Data lakes provide faster insights
  + As it contains all data and data types, because it enables users to access data before it has been transformed, cleansed and structured it enables users to get to their results faster than the traditional data warehouse approach.

A Data warehouse on the other hand has data in a structured fashion to provide meaningful insights stored in a historic perspective.

Many industries like healthcare, transportation, and education have unstructured data, data lakes can be beneficial for such clients to provide real-time insights and the flexibility of data storage facilitates answering different questions.

**Explain serverless architecture. What are its pros and cons?**

Serverless Architecture is a way to build and run applications and services without having to maintain on premise infrastructure, that is, not worrying about managing or operating the servers or runtimes.

Advantages of Serverless Architecture:

* Server management is not necessary
* Reduces cost, as developers are only charged for the server space they use
* Serverless architectures are inherently scalable
* Quick deployments and updates are possible
* Code can run closer to the end user, decreasing latency

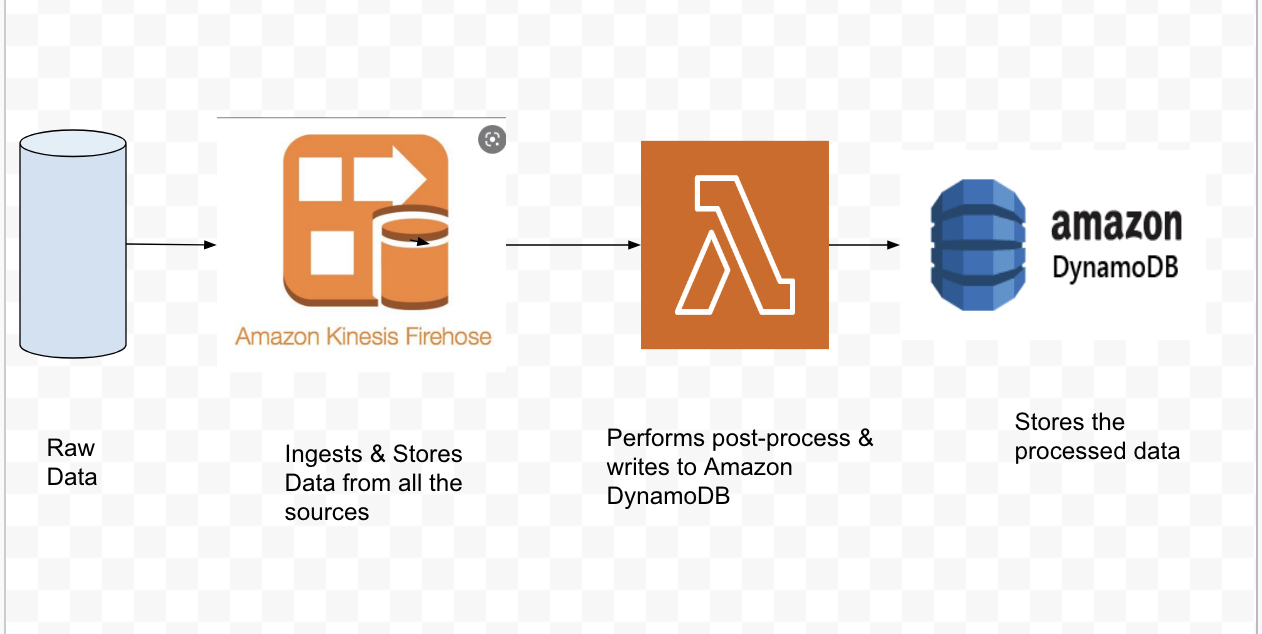
Disadvantages of Serverless Architecture:

* Testing and debugging become more challenging
* Serverless computing introduces new security concerns
* Serverless architectures are not built for long-running processes
* Performance may be affected

**Please provide a diagram of the ETL pipeline from Section 1 using serverless AWS services. Describe each component and its function within the pipeline.**

The ETL model I have provided is a simple one. It extracts data from a single source which is a CSV file and performs transformations on this file. And then loads this data in a database(both Relational/ NoSQL)

An equivalent ETL pipeline using AWS services would look like:



*Image Source: Google*

Amazon Kinesis Firehose: Amazon Kinesis Data Firehose is the easiest way to reliably load streaming data into data lakes, data stores, and analytics services. It can also batch, compress, transform, and encrypt your data streams before loading, minimizing the amount of storage used and increasing security.

Amazon Lambda: AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers, creating workload-aware cluster scaling logic, maintaining event integrations, or managing runtimes. With Lambda, you can run code for virtually any type of application or backend service - all with zero administration

Amazon DynamoDB: Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-active, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications.

We can also use Amazon Redshift or Amazon S3 for storing the processed data depending on the use case.

**Describe the process of deploying infrastructure as code on AWS or Azure and why this is a good method of architecture delivery for client**

Infrastructure as a Code (IaC) is the process of deploying and managing the infrastructure through a descriptive model. The configuration modules are treated as a source code and stored in a version control system.

This solves the problem of environmental drift. We have configurations of several environments in a configuration file and the same build of source code after development and testing is deployed on various environments configured in the model. That is, the environment remains the same and we keep changing the builds deployed on the environment.

**Describe modern MLOps and how organizations should be approaching management from a tool and system perspective; what tools do you recommend? Where would you provision such services in your assessment?**

MLOps is the process of deploying and maintaining the machine learning models in a production environment reliably and continuously.

MLOps is essentially practicing the Continuous Deployment of DevOps in Machine Learning, hence the name MLOps.

It is used for Continuous Integration, Continuous Deployment and Continuous Training of the Machine Learning Models in real time.

Amazon Sage Maker is a Machine Learning model used by many companies to prepare, train and deploy models, continuously and quickly by using the several built-in features for ML.

In Section 4, I have used a basic pretrained model- half plus two. I could have also used facenet, which is a TensorFlow implementation for facial recognition. The Facenet model takes in face images and provides a vector as the output. For such a model, MLOps would have been a good option to train and test.

**References:**

<https://www.bluegranite.com/blog/bid/402596/top-five-differences-between-data-lakes-and-data-warehouses>

<https://www.cloudflare.com/learning/serverless/why-use-serverless/>

[https://aws.amazon.com/{service}/](https://aws.amazon.com/%7Bservice%7D/)