Cloud Data Lakes

You know you are working with big data when you can not use a single computer but you have to use a distributed system.

Understanding How hardware works can help solve what is big data and what is not big data.
CPU => Brain of the Computer, it is the fastest, 2.5Ghz means that the computer can process 2.5 billion operations per second
Memory => Where data is temporarily stored, it is the 2nd fastest
Disk storage =.> Long term storage support,

A distributed system or cluster is a bunch of connected machines, and these machines are called nodes.
The main difference between Parallel computing and distributed computing is that in Parallel computing, all processors have access to a shared memory that they can use to share information, while in distributed each node has its own memory

Hadoop and Spark
Hadoop => Data storage, an ecosystem of tools for big data storage and data analysis. Hadoop is an older system than Spark but is still used by many companies. The major difference between Spark and Hadoop is how they use memory. Hadoop writes intermediate results to disk whereas Spark tries to keep data in memory whenever possible. This makes Spark faster for many use cases.
Hadoop MapReduce => Data processing a system for processing and analyzing large data sets in parallel.
Hadoop YARN => Resource manager a resource manager that schedules jobs across a cluster. The manager keeps track of what computer resources are available and then assigns those resources to specific tasks
Hadoop Distributed File System (HDFS) => Utilities a big data storage system that splits data into chunks and stores the chunks across a cluster of computers.

Apache Hive, Pig => SQL for Hadoop.
Apache Storm, Flink => Streaming

Spark contains libraries for data analysis, machine learning, graph analysis, and streaming live data. There are three options for working on a cluster with Spark, standalone mode, Mesos, and Yarn.

The technique works by first dividing up a large dataset and distributing the data across a cluster. In the map step, each data is analyzed and converted into a (key, value) pair. Then these key-value pairs are shuffled across the cluster so that all keys are on the same machine. In the reduce step, the values with the same keys are combined together.

MapReduce => Map, Shuffle, Reduce

There are two modes to run spark => Local mode, and shared mode
Spark uses functional programming because functional programming is perfect for distributed systems. In distributed systems, your functions should not have side effects on variables outside their scope. Spark requires you to write pure functions, that are functions that do not have an effect on anything outside of themselves.

When you run your functions Spark builds a DAG(Directed Acyclic Graph), in Spark multi-step combos are called stages, Spark uses a method called **Lazy evaluation** to preprocess data, in anonymous functions(Lambda), the left side is your input, the right side is what you put in.

Ways of handling the data
Imperative programming => Python and Spark data frames(this cares about the How question),
Declarative programming => SQL(this cares about the What question)
Usually, Imperative programming concepts run as a level of abstraction behind declarative programming.

To submit the Spark script you have to use spark-submit, and if you ever forget where it is. run "which spark-submit".

When loading from s3, you can load everything directly from one bucket, you just have to ensure that they all have the same schema.
Since AWS is a binary object store, it can store all types of format, while HDFS will usually require a certain file format, the popular choices are Avro and Paroquet.

Spark Lazy evaluation will put you in trouble.
You can use Accumulators to access variables globally.
Spark broadcast variables are secure read-only variables that get distributed and cached to different worker nodes, this is useful when the driver sends a packet of

information to worker nodes. Broadcast join is a way of joining a large table and a small table in Spark. A Broadcast join is like a Map side join in MapReduce.

Important Ports in Spark

- Port 8888 => This is necessarily used by the Jupyter notebook
- Port 4040 => This shows active spark jobs
- Port 7077 => This port is used by Spark to speak to active worker nodes.
- Port 8080 => This is the port used by Spark UI to display active jobs.

Functions in Spark

There are two types of functions in Spark

- Transformation functions
- Actions

Code Optimization => Skewed data, this means that due to non-optimal partitioning that the data is heavy on a few partitions. This can be seriously problematic. Usually, we tackle this by running on spark job to understand the distribution of your data, this single spark job can help avoid large discrepancies in the data distribution. There are two other main approaches to fixing the Spark data distribution problem.

- Changing the distribution of the data
- Partitioning the data into smaller chunks

Three ways to ensure the data is properly skewed,

- Use alternative columns that are more normally distributed
- Make composite keys
- Partition by the Number of spark workers.

The Data Lake is the new type of Data warehouse that evolved to cope with things like,

- Variety of data formats and structuring

- The agile and ad-hoc nature of data exploration activities needed by new roles like the data scientist
- The widespread spectrum data transformation needed by Advanced analytics like Machine learning, graph analytics, and recommender systems.

In Data Lake systems, you don’t do ETL, you do ELT.

Data Lake Options

- Storage (HDFS), Processing (Spark) => AWS EMR (HDFS + Spark)
- Storage(S3), Processing (Spark) => AWS EMR (Spark)
- Storage(S3), Processing(Serverless) => AWS Athena

Athena is that self-managed system that allows you to do all your data processes, run queries without creating additional resources.