**Hadoop**

HDFS – Huge data, batch processing needs

MapReduce – Like cat – grep – sort – unique count. Computing framework. HDFS + YARN

Pig – Dataflow language/script. PigLatin is a high level MapReduce. Simplifies the MapReduce job if you have many transformations. Quick counts, aggress on huge data volume

Sqoop – Transfer data from relational to Hadoop. Incremental updates. Writes data back to relational db from the Hadoop cluster. For huge amount of legacy data to put on Hadoop

Oozie – Workflow scheduler system. Schedule MapReduce, Pig, Sqoop, Hive jobs

Flume – Aggregates huge amount of data

HBase – Runs on Hadoop. NoSQL. Enables realtime, random access to data via key/value pairs. Complex, nested schema. Real time processing (not batch). Scaleability for columns

* like BigTable

Hive – DataWarehousing. SQL-like interface HiveQL. Under the hood it uses MapReduce. Not transactional (updates), rather OLAP. Analytics, reports, partitioning concept subpartitions as Buckets. These are individual files each

* like BigQuery

Avro – Data serialization system. Serialize data structures into fast, binary format. Use it in Hive and Pig. JSON defined datatypes (schema)

Kafka – Messaging system (publisher/subscriber). Stream processor to process. Connectors to databases. Stores messages in Key/Value in partitions. Two types: queue (consumers query the queue), publish (record is broadcast to consumers). Queue: consumer reads what it wants therefore can scale, publish: data goes to everybody. Stores data (decouples it)

* like Pub/Sub

Nifi – Data routing, transformation. Processors (written for specific purpose) with attributes (input.output). Processors for all sorts of purposes. Connectors via attributes (channeling). All communications via attributes/properties. Visual drag&drop interface. High level design, no need for programming.

* like DataFlow

Ambari – Managing, monitoring, provisioning Hadoop clusters

Case:

1. Daily log Files were transported to *HDFS*
2. MR jobs parsed these log files and output files in *HDFS*
3. Create Hive tables with partitions and locations pointing to *HDFS* locations
4. Create Hive query scripts (call it *HQL* if u like as diff from *SQL*) that in turn ran MR jobs in the background and generated aggregation data
5. Put all these steps into an *Oozie* workflow - scheduled with Daily *Oozie* Coordinator

**GCP**

BigTable – Runs on GCP. NoSQL. Key/value pairs. Stores and queries quickly

* like HBase

BigQuery – DataWarehousing. Analytic functions. Standard SQL-like. Load jobs, query jobs

* like Hive

Pub/Sub - Messaging system (topic/publisher/subscriber)

* like Kafka

DataFlow – Apache Beam dataflow system. Using a pipeline and ParDo-s (Map) and DoFn (custom classes to process in custom way). Can do Stream processing form Pub/Sub as well. Low level programming of your dataflow

* like Nifi