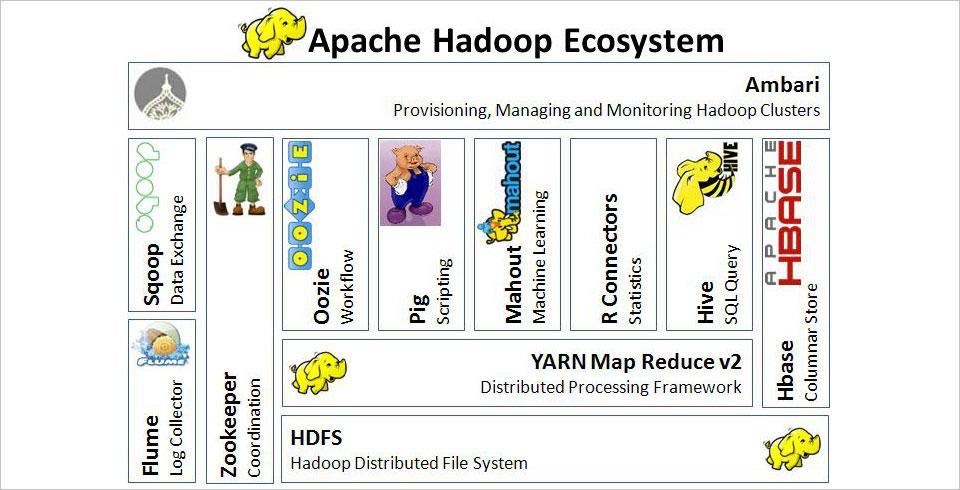
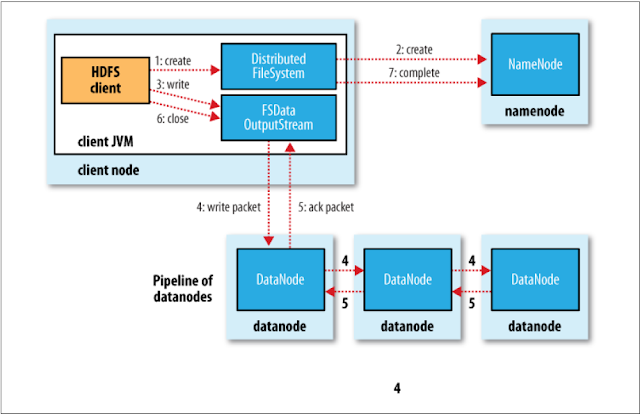
14 Day Big Data Challenge

Day 2 – HDFS

1. Why Hadoop Ecosystems for Big Data?
   * Scalable: Petabytes (10ˆ15 Bytes) of data on thousands of nodes
   * Economical: Commodity components only
   * Reliable: fault tolerant (e.g. hardware failure)



1. HDFS – Hadoop Distributed File System
   * HDFS is the storage layer for Hadoop
   * Data files are split into blocks and distributed to **DataNodes**
   * Each block is replicated on multiple nodes (default 3x)
   * **NameNode** stores metadata
2. Master (NameNode)
   * Manages filesystem namespace
   * File metadata
   * Mapping file to list of blocks
   * Authorization & authentication
   * Mapping of DataNode to list of blocks
   * Monitor DataNode health
   * Replicate missing blocks
3. Slave (DataNode)
   * Handle block storage on multiple volumes and block integrity
   * Clients access the blocks directly from DataNodes
   * Periodically send heartbeats and block reports to NameNode
   * Blocks are stored as underlying OS’s files



1. Data “Ingest”and storage
   * Hadoop typically ingests data from many sources and in many formats:
     1. Traditional data management systems (databases)
     2. Logs and other machine generated data (event data)
     3. Imported files
2. Data “Ingest” tools
   * HDFS – Direct file transfer
   * Apache Sqoop (“SQL-to-Hadoop”)
     1. High speed import to HDFS from relationship database (and vice versa)
     2. Supports many data storage systems (e.g., Netezza, Mongo, MySQL, Teradata, Oracle)
     3. Sqoop uses JDBC (but does have some features for others)
        1. JDBC = Java Database Connectivity
   * Apache Flume
     1. Distributed service for ingesting streaming data
     2. Ideally suited for event data from multiple systems (e.g., log files)
   * Kafka
     1. A high throughput, scalable messaging system
     2. Distributed, reliable publish-subscribe system
     3. Integrates with Flume and Spark Streaming