Web: Inceptez.com Mail: info@inceptez.com Call: 7871299810, 7871299817

# Big Data, Data Ingestion and Processing

In the era of the Internet of Things and Mobility, with a huge volume of data becoming available at a fast velocity, there must be the need for an efficient Analytics System.

Also, the variety of data is coming from various sources in different formats, such as sensors, logs, structured data from an RDBMS, etc. In the past few years, the generation of new data has drastically increased. More applications are being built, and they are generating more data at a faster rate.

Earlier, Data Storage was costly, and there was an absence of technology which could process the data in an efficient manner. Now the storage costs have become cheaper, and the availability of technology to transform Big Data is a reality.

## What is Big Data Technology?

- **Everything** – Means every aspect of life, work, consumerism, entertainment, and play is now recognized as a source of digital information about you, your world, and anything else we may encounter.
- Quantified – Means we are storing those "everything" somewhere, mostly in digital form, often as numbers, but not always in such formats. The quantification of features, characteristics, patterns, and trends in all things is enabling Data Mining, Machine Learning, statistics, and discovery at an unprecedented scale on an unprecedented number of things. The Internet of Things is just one example, but the Internet of Everything is even more impressive.
- Tracked – Means we don't directly quantify and measure everything just once, but we do so continuously. It includes - tracking your sentiment, your web clicks, your purchase logs, your geolocation, your social media history, etc. or tracking every car on the road, or every motor in a manufacturing plant or every moving part on an airplane, etc. Consequently, we see the emergence of smart cities, smart highways, personalized medicine, personalized education, precision farming, and so much more.

## Advantages of Big Data

- Smarter Decisions
- Better Products
- Deeper Insights
- Greater Knowledge

- Optimal Solutions
- Customer-Centric Products
- Increased Customer Loyalty
- More Automated Processes, more accurate Predictive and Prescriptive Analytics
- Better models of future behaviors and outcomes in Business, Government, Security, Science, Healthcare, Education, and more.

## D2D Communication Meets Big Data

- Data-to-Decisions
- Data-to-Discovery
- Data-to-Dollars

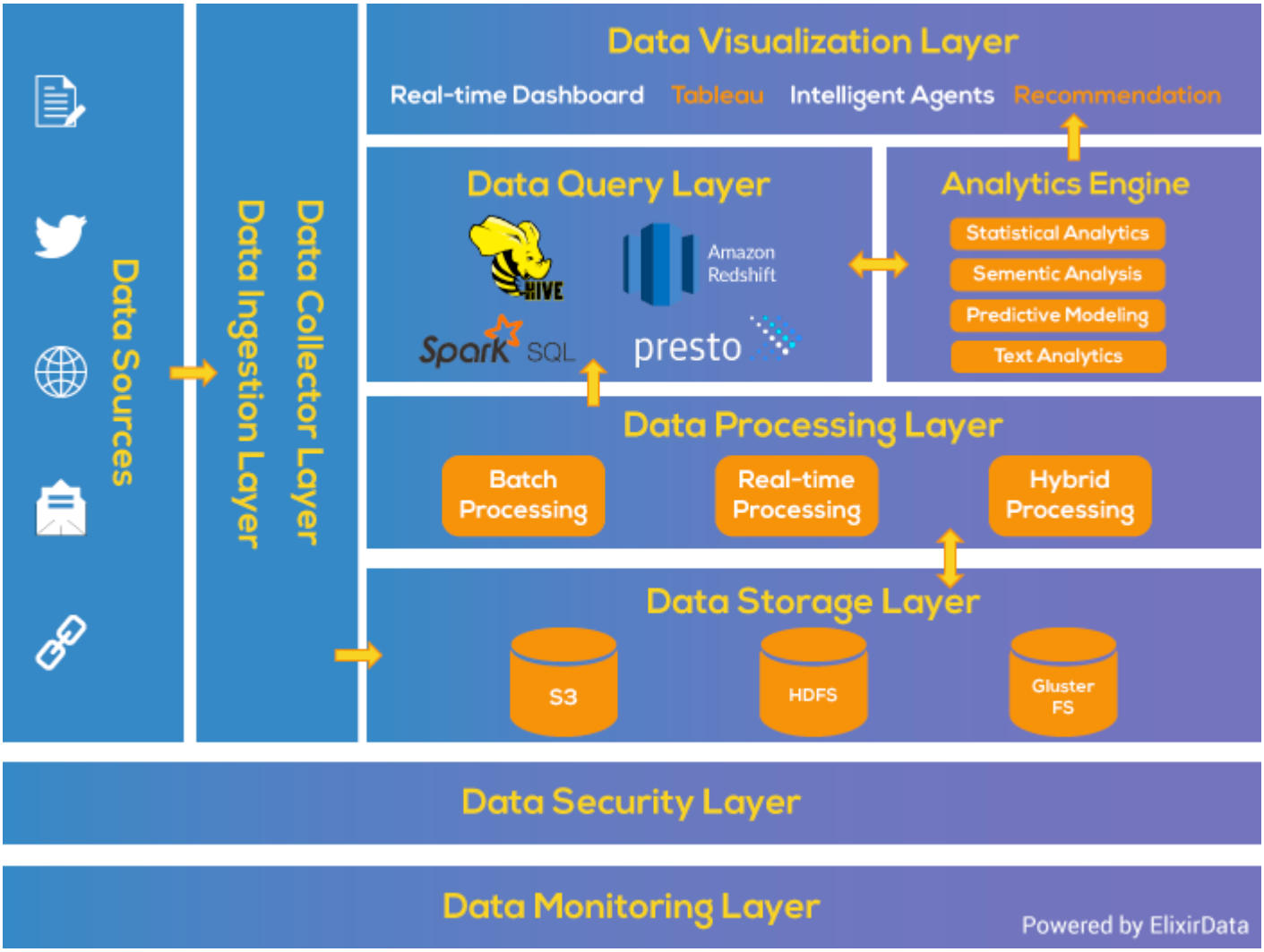
## 10 Vs of Big Data


---

# Big Data Architecture & Patterns

The Best Way to a solution is to "Split The Problem."Big Data Solution can be well understood using Layered Architecture. The Layered Architecture is divided into different Layers where each layer performs a  particular function.

This Architecture helps in designing the Data Pipeline with the various requirements of either Batch Processing System or Stream Processing System. This architecture consists of 6 layers which ensure a secure flow of data.




- ## Data Ingestion Layer

This layer is the first step for the data coming from variable sources to start its journey. Data here is prioritized and categorized which makes data flow smoothly in further layers.

- ## Data Collector Layer

In this Layer, more focus is on the transportation of data from ingestion layer to rest of data pipeline. It is the Layer, where components are decoupled so that analytic capabilities may begin.

- ## Data Processing Layer

In this primary layer, the focus is to specialize the data pipeline processing system, or we can say the data we have collected in the previous layer is to be processed in this layer. Here we do some magic with the data to route them to a different destination, classify the data flow and it's the first point where the analytic may take place.

- ## Data Storage Layer

Storage becomes a challenge when the size of the data you are dealing with, becomes large. Several possible solutions can rescue from such problems. Finding a storage solution is very much important when the size of your data becomes large. This layer focuses on "where to store such a large data efficiently."

- ## Data Query Layer

This is the layer where active analytic processing takes place. Here, the primary focus is to gather the data value so that they are made to be more helpful for the next layer.

- ## Data Visualization Layer

The visualization, or presentation tier, probably the most prestigious tier, where the data pipeline users may feel the VALUE of DATA. We need something that will grab people's attention, pull them into, make your findings well-understood.

---

# Big Data Ingestion Architecture

Data ingestion is the first step for building Data Pipeline and also the toughest task in the System of Big Data. In this layer we plan the way to ingest data flows from hundreds or thousands of sources into Data Center. As the Data is coming from Multiple sources at variable speed, in different formats.

That's why we should properly ingest the data for the successful business decisions making. It's rightly said that "If starting goes well, then, half of the work is already done."

---

## What is Ingestion in Big Data?

**Big Data Ingestion involves connecting to various data sources, extracting the data, and detecting the changed data.** It's about moving data - and especially the unstructured data - from where it is originated, into a system where it can be stored and analyzed.

We can also say that Data Ingestion means taking data coming from multiple sources and putting it somewhere it can be accessed. It is the beginning of Data Pipeline where it obtains or import data for immediate use.

Data can be streamed in real time or ingested in batches, When data is ingested in real time then, as soon as data arrives it is ingested immediately. When data is ingested in batches, data items are ingested in some chunks at a periodic interval of time. Ingestion is the process of bringing data into Data Processing system.

Effective Data Ingestion process begins by prioritizing data sources, validating individual files and routing data items to the correct destination.

## Challenges in Data Ingestion

As the number of IoT devices increases, both the volume and variance of Data Sources are expanding rapidly. So, extracting the data such that it can be used by the destination system is a significant challenge regarding time and resources. Some of the other problems faced by Data Ingestion are -

- When numerous Big Data sources exist in the different format, it's the biggest challenge for the business to ingest data at the reasonable speed and further process it efficiently so that data can be prioritized and improves business decisions.
- Modern Data Sources and consuming application evolve rapidly.
- Data produced changes without notice independent of consuming application.
- Data Semantic Change over time as same Data Powers new cases.
- Detection and capture of changed data - This task is difficult, not only because of the semi-structured or unstructured nature of data but also due to the low latency needed by individual business scenarios that require this determination.

That's why it should be well designed assuring following things -

- Able to handle and upgrade the new data sources, technology and applications
- Assure that consuming application is working with correct, consistent and trustworthy data.
- Allows rapid consumption of data
- Capacity and reliability - The system needs to scale according to input coming and also it should be fault tolerant.
- Data volume - Though storing all incoming data is preferable; there are some cases in which aggregate data is stored.

## Data Ingestion Parameters

**Data Velocity -** Data Velocity deals with the speed at which data flows in from different sources like machines, networks, human interaction, media sites, social media. The movement of data can be massive or continuous.

**Data Size -** Data size implies enormous volume of data. Data is generated from different sources that may increase timely.

**Data Frequency (Batch, Real-Time) -** Data can be processed in real time or batch, in real time processing as data received on same time, it further proceeds but in batch time data is stored in batches, fixed at some time interval and then further moved.

**Data Format (Structured, Semi-Structured, Unstructured) -** Data can be in different formats, mostly it can be the structured format, i.e., tabular one or unstructured format, i.e., images, audios, videos or semi-structured, i.e., JSON files, CSS files, etc.

## Big Data Ingestion Key Principles

To complete the process of Data Ingestion, we should use right tools for that and most important that tools should be capable of supporting some of the fundamental principles written below -

**Network Bandwidth -** Data Pipeline must be able to compete with business traffic. Sometimes traffic increases or sometimes decreases, so Network bandwidth scalability is biggest Data Pipeline challenge. Tools are required for bandwidth throttling and compression capabilities.

**Unreliable Network -** Data Ingestion Pipeline takes data with multiple structures, i.e., images, audios, videos, text files, tabular files data, XML files, log files, etc. and due to the variable speed of data coming, it might travel through the unreliable network. Data Pipeline should be capable of supporting this also.

**Heterogeneous Technologies and System -** Tools for Data Ingestion Pipeline must be able to use different data sources technologies and different operating system.

**Choose Right Data Format -** Tools must provide data serialization format, that means as data comes in the variable format so converting them into single format will provide an easier view to understand or relate the data.

**Streaming Data -** It depends upon business necessity whether to process the data in batch or streams or real time. Sometimes we may require both processing. So, tools must be capable of supporting both.

# Data Serialization in Big Data

Different types of users have various types of data consumer needs. Here we want to share variable data, so we must plan how the user can access data in a meaningful way. That's why a single image of variable data optimize the data for human readability.

Approaches used for this are -

### Apache Thrift

It's an RPC Framework containing Data Serialization Libraries.

### Google Protocol Buffers

It can use the specially generated source code to easily write and read structured data to and from a variety of data streams and using a variety of languages.

### Apache Avro

The more recent Data Serialization format that combines some of the best features which previously listed. Avro Data is self-describing and uses a JSON-schema description. This schema is included with the data itself and natively support compression. Probably it may become a de facto standard for Data Serialization.

---

# Big Data Ingestion Tools

### Apache Flume Architecture

Apache Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.

It has a straightforward and flexible architecture based on streaming data flows. It is robust and faults tolerant with tunable reliability mechanisms and many failovers and recovery mechanisms.

It uses a simple, extensible data model that allows for an online analytic application.

### Functions of Apache Flume

**Stream Data -** Ingest streaming data from multiple sources into Hadoop for storage and analysis.

**Insulate System -** Buffer storage platform from transient spikes, when the rate of incoming data exceeds the rate at which data can be written to the destination

**Scale Horizontally** - To ingest new data streams and additional volume as needed.

### Apache Nifi Overview

Apache Nifi provides an easy to use, the powerful, and reliable system to process and distribute data. **Apache NiFi supports robust and scalable directed graphs** of data routing, transformation, and system mediation logic. Its functions are -

- Track data flow from beginning to end
- Seamless experience between design, control, feedback, and monitoring
- Secure because of SSL, SSH, HTTPS, encrypted content.

### Integrating Elasticsearch with Logstash

Elastic Logstash is an open source, server-side data processing pipeline that ingests data from a multitude of sources simultaneously transforms it, and then sends it to your “stash, " i.e., Elasticsearch.

It easily ingests from your logs, metrics, web applications, data stores, and various AWS services and done in continuous, streaming fashion. It can Ingest Data of all Shapes, Sizes, and Sources.

---

# Big Data Pipeline Architecture


In this Layer, more focus is on transportation data from ingestion layer to rest of Data Pipeline. Here we use a messaging system that will act as a mediator between all the programs that can send and receive messages.

Here the tool used is Apache Kafka. It's a new approach in message-oriented middleware.

## Getting Started with Big Data Pipeline

- Data Pipeline the main component of Data Integration. **All transformation of data happens in Data Pipeline**.
- It is a Python-based tool that streams and transforms real-time data to service that need it.
- Data Pipeline Automate the movement and transformation of data. Data Pipeline is a Data Processing engine that runs inside your application.
- It is used to transform all the incoming data in a standard format so that we can prepare it for analysis and visualization. Data Pipeline is built on Java Virtual Machine (JVM).
- So, a Data Pipeline is a series of steps that your data moves through. The output of one step in the process becomes the input of the next. Data, typically raw data, goes on one side, passes through a series of steps.
- The steps of a Data Pipeline can include cleaning, transforming, merging, modelling and more, in any combination.

## Big Data Pipeline Functions

### Data Ingestion

Data Pipeline Helps in bringing data into your system. It means taking unstructured data from where it is originated into a system where it can be stored and analyzed for making business decisions

### Data Integration

Data Pipeline also helps in bringing different types of data together.

### Data Organization

Organizing data means an arrangement of data; this arrangement is also made in Data Pipeline.

### Data Refining

It's also one of the processes where we can enhance, clean, improve the raw data.

### Data Analytics

After improving the useful data, Data Pipeline provides us with the processed data on which we can apply the operations on raw data and can make business decisions accurately.

## Need Of Big Data Pipeline

A Data Pipeline is software that takes data from multiple sources and makes it available to be used strategically for making business decisions.

Primarily reasons for the need of data pipeline is because it's tough to monitor Data Migration and manage data errors. Other reasons for this are below -

- **Business Decisions** - Critical Analysis is only possible when combining data from multiple sources. For making business decisions, we should have a single image of all the data coming.
- **Connections** - All the time data keeps on increasing, new data came and old data modified, so, each new integration can take anywhere from a few days to a few months to complete.
- **Accuracy** - The only way to build trust with data consumers is to make sure that your data is auditable. One best practice that's easy to implement is never to discard inputs or intermediate forms when altering data.
- **Latency** - The fresher your data, the agiler your company's decision-making can be. Extracting data from APIs and databases in real-time can be difficult, and many target data sources, including large object stores like Amazon S3 and analytics databases like Amazon Redshift, are optimized for receiving data in chunks rather than a stream.
- **Scalability** - Data can be increased or decreased with time we can't say for on Monday data will come less and rest of days comes a lot for processing. So, usage of data is not uniform. What we can do is making our pipeline so scalable that able to handle any amount of data coming at variable speed.

## Big Data Pipeline Use Cases

Data Pipeline is useful to some roles, including CTOs, CIOs, Data Scientists, Data Engineers, BI Analysts, SQL Analysts, and anyone else who derives value from a unified real-time stream of user, web, and mobile engagement data. So, use cases for data pipeline are given below -

- For Business Intelligence Teams
- For SQL Experts
- For Data Scientists
- For Data Engineers
- For Product Teams

---

## Apache Kafka Overview

It is used for building real-time data pipelines and streaming apps. It can process streams of data in real-time and store streams of data safely in a distributed replicated cluster.

Kafka works in combination with Apache Storm, Apache HBase and Apache Spark for real-time analysis and rendering of streaming data.

- Building Real-Time streaming Data Pipelines that reliably get data between systems or applications
- Building Real-Time streaming applications that transform or react to the streams of data.

## Apache Kafka Use Cases

- Stream Processing
- Website Activity Tracking
- Metrics Collection and Monitoring
- Log Aggregation

## Apache Kafka Features

- One of the features of Apache Kafka is durable Messaging.
- Apache Kafka relies heavily on the file system for storing and caching messages: rather than maintain as much as possible in memory and flush it all out to the filesystem, all data is immediately written to a persistent log on the filesystem without necessarily flushing to disk.
- Apache Kafka solves the situation where the producer is generating messages faster than the consumer can consume them in a reliable way.

## Apache Kafka Architecture

Apache Kafka System design act as Distributed commit log, where incoming data is written sequentially on disk. There are four main components involved in moving data in and out of Apache Kafka -

- **Topics -** Topic is a user-defined category to which messages are published.
- **Producers -** Producers post messages to one or more topics
- **Consumers -** Consumers subscribe to topics and process the posted messages.
- **Brokers -** Brokers that manage the persistence and replication of message data.

---

# Big Data Processing Layer


In the previous layer, we gathered the data from different sources and made it available to go through rest of pipeline.

In this layer, our task is to do magic with data, as now data is ready we only have to route the data to different destinations.

In this main layer, the focus is to specialize Data Pipeline processing system or we can say the data we have collected by the last layer in this next layer we have to do processing on that data.

## Big Data Batch Processing System

A simple batch processing system for offline analytics. For doing this tool used is Apache Sqoop.

### What is Apache Sqoop?

It efficiently transfers bulk data between Apache Hadoop and structured datastores such as relational databases. Apache Sqoop can also be used to extract data from Hadoop and export it into external structured data stores.

Apache Sqoop works with relational databases such as Teradata, Netezza, Oracle, MySQL, Postgres, and HSQLDB.

### Functions of Apache Sqoop

- Import sequential data sets from mainframe
- Data imports
- Parallel Data Transfer
- Fast data copies
- Efficient data analysis
- Load balancing

---

## Near Real-Time Processing System

A pure online processing system for online analytics. For this type of processing **Apache Storm** is used**.** The Apache Storm cluster makes decisions about the criticality of the event and sends the alerts to the warning system (dashboard, e-mail, other monitoring systems).

### What is Apache Storm?

It is a system for processing streaming data in real time. It adds reliable real-time data processing capabilities to Enterprise Hadoop. Storm on YARN is powerful for scenarios requiring real-time analytics, machine learning and continuous monitoring of operations.

### 6 Key Features of Apache Storm

- **Fast** – It can process one million 100 byte messages per second per node.
- **Scalable** – It can do parallel calculations that run across a cluster of machines.
- **Fault-tolerant** – When workers die, Storm will automatically restart them. If a node dies, the worker will be restarted on another node.
- **Reliable** – Storm guarantees that each unit of data (tuple) will be processed at least once or exactly once. Messages are only replayed when there are failures.
- **Easy to operate** – It consists of Standard configurations that are suitable for production on day one. Once deployed, Storm is easy to work.
- **Hybrid Processing system -** This consist of Batch and Real-time processing System capabilities. For this type of processing tool used is Apache Spark and Apache Flink.

### What is Apache Spark?

Apache Spark is a fast, in-memory data processing engine with elegant and expressive development APIs to allow data workers to efficiently execute streaming, machine learning or SQL workloads that require fast iterative access to data sets.

With Spark running on Apache Hadoop YARN, developers everywhere can now create applications to exploit Spark's power, derive insights, and enrich their data science workloads within a single, shared data set in Hadoop.

# Real-Time Processing System

### What is Apache Flink?

Apache Flink is an open-source framework for distributed stream processing that Provides results that are accurate, even in the case of out-of-order or late-arriving data. Some of its features are -

- It is stateful and fault-tolerant and can seamlessly recover from failures while maintaining exactly-once application state.
- Performs at large scale, running on thousands of nodes with excellent throughput and latency characteristics.
- It's streaming data flow execution engine, APIs and domain-specific libraries for Batch, Streaming, Machine Learning, and Graph Processing.

### Apache Flink Use Cases

- Optimization of e-commerce search results in real-time
- Stream processing-as-a-service for data science teams
- Network/Sensor monitoring and error detection
- ETL for Business Intelligence Infrastructure

---

# Big Data Storage Layer


Next, the major issue is to keep data in the right place based on usage. We have relational Databases that were a successful place to store our data over the years.

But with the new big data strategic enterprise applications, you should no longer be assuming that your persistence should be relational.

We need different databases to handle the different variety of data, but using different databases creates overhead. That's why there is an introduction to the new concept in the database world,

---

# Big Data Storage Tools

### HDFS : Hadoop Distributed File System

- HDFS is a Java-based file system that provides scalable and reliable data storage, and it was designed to span large clusters of commodity servers.
- HDFS holds a huge amount of data and provides easier access.
- To store such massive data, the files are stored on multiple machines. These files are stored redundantly to rescue the system from possible data losses in case of failure.
- HDFS also makes applications available for parallel processing. HDFS is built to support applications with large data sets, including individual files that reach into the terabytes.
- It uses a master/slave architecture, with each cluster consisting of a single NameNode that manages file system operations and supporting DataNodes that manage data storage on individual compute nodes.
- When HDFS takes in data, it breaks the information down into separate pieces and distributes them to different nodes in a cluster, allowing for parallel processing.
- The file system also copies each piece of data multiple times and distributes the copies to individual nodes, placing at least one copy on a different server rack
- HDFS and YARN from the data management layer of Apache Hadoop.

### Features of HDFS

- It is suitable for the distributed storage and processing.
- Hadoop provides a command interface to interact with HDFS.
- The built-in servers of name node and data node help users to quickly check the status of the cluster.
- Streaming access to file system data.
- HDFS provides file permissions and authentication.

---

### Amazon S3 Storage Service

- Amazon Simple Storage Service (Amazon S3) is object storage with a simple web service interface to store and retrieve any amount of data from anywhere on the internet.
- It is designed to deliver 99.999999999% durability, and scale past trillions of objects worldwide.
- Customers use S3 as primary storage for cloud-native applications; as a bulk repository, or "data lake," for analytics; as a target for backup & recovery and disaster recovery; and with serverless computing.
- It's simple to move large volumes of data into or out of S3 with Amazon's cloud data migration options.
- Once data is stored on Amazon S3, it can be automatically tiered into lower cost, longer-term cloud storage classes like S3 Standard - Infrequent Access and Amazon Glacier for archiving.

---

# Big Data Query Layer

It is the layer where active analytic processing takes place. This is a field where interactive queries are necessaries, and it's a zone traditionally dominated by SQL expert developers. Before Hadoop, we had an insufficient storage due to which it takes long analytics process.

At first, it goes through a Lengthy process, i.e., ETL to get a new data source ready to be stored and after that, it puts the data in database or data warehouse. But now, data analytics became essential step which solved problems while computing such a large amount of data.

Companies from all industries use big data analytics to -

- Increase revenue
- Decrease costs
- Increase productivity

# Big Data Analytics Query Tools

- **Apache Hive Architecture**

Apache Hive is data warehouse infrastructure built on top of Apache Hadoop for providing data summarization, ad-hoc query, and analysis of large datasets.

Data analysts use Hive to query, summarize, explore and analyze that data, then turn it into actionable business insight.

It provides a mechanism to project structure onto the data in Hadoop and to query that data using a SQL - like a language called HiveQL (HQL).

### Features of Apache Hive

- Query data with a SQL - based language.
- Interactive response times, even over massive datasets.
- It's scalable as data variety and volume grows, more commodity machines can be added, without a corresponding reduction in performance Works with traditional data integration and data analytics tools.

- **Apache Spark SQL**

Spark SQL includes a cost-based optimizer, columnar storage, and code generation to make queries fast.

At the same time, it scales to thousands of nodes and multi-hour queries using the Spark engine, which provides full mid-query fault tolerance.

Spark SQL is a Spark module for structured data processing. Some of the Functions performed by Spark SQL are -

- The interfaces provided by Spark SQL provide Spark with more information about the structure of both the data and the computation being performed.
- Internally, Spark SQL uses this extra information to perform additional optimizations.
- One use of Spark SQL is to execute SQL queries.
- Spark SQL can also be used to read data from an existing Hive installation.

- **Amazon Redshift**

Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. We use Amazon Redshift to load the data and run queries on the data.

We can also create additional databases as needed by running a SQL command. Most important we can scale it from hundred gigabytes of data to a petabyte or more.

It enables you to use your data to acquire new insights for your business and customers. The Amazon Redshift service manages all of the work of setting up, operating and scaling a data warehouse.

These tasks include provisioning capacity, monitoring and backing of the cluster, and applying patches and upgrades to the Amazon Redshift engine.

- **Presto - SQL Query Engine For Big Data**

Presto is an open source distributed SQL query engine for running interactive analytic queries against data sources of all sizes ranging from gigabytes to petabytes.

Presto was designed and written for interactive analytics and approaches and the speed of commercial data warehouses while scaling to the size of organizations like Facebook.

**Presto Capabilities**

- Presto allows querying data where it lives, including Hive, Cassandra, relational databases or even proprietary data stores.
- A single Presto query can combine data from multiple sources, allowing for analytics across your entire organization.
- Presto is targeted at analysts who expect response times ranging from sub-second to minutes.
- Presto breaks the false choice between having fast analytics using an expensive commercial solution or using a slow "free" solution that requires excessive hardware.

**Who Uses Presto?**

- Facebook uses Presto for interactive queries against several internal data stores, including their 300PB Data Warehouse. Over 1,000 Facebook employees use Presto daily to run more than 30,000 queries that in the complete scan over a petabyte each per day.
- Leading internet companies including Airbnb and Dropbox are using Presto.

---

# Data Lake and Data Warehouse

## What is Data Warehouse?

A Data Warehouse is a subject-oriented, Integrated, Time-varying, non-volatile collection of data in support of management's decision-making process.

So, a Data Warehouse is a centralized repository that stores data from multiple information sources and transforms them into a standard, multidimensional data model for efficient querying and analysis.

## Difference Between Big Data and Data Warehouse

While comparing, we found that a big data solution is a technology and that data warehousing is an architecture. They are two very different things.

Technology is just that – a means to store and manage large amounts of data. A data warehouse is a way of organizing data so that there are corporate credibility and integrity.

When someone takes data from a data warehouse, that person knows that other people are using the same data for other purposes. There is a basis for reconcilability of data when there is a data warehouse.

## What is Data Lake?

It is a new type of cloud-based enterprise architecture that structures data in a more scalable way that makes it easier to experiment with it.

With data lake, incoming data goes into the lake in a raw form or whatever form data source providers, and there we select and organize the data in a raw form. There are no assumptions about the schema of the data; each data source can use whatever scheme it likes.

It's up to the consumers of that information to make sense of that data for their purposes. The idea is to have a single store for all of the raw data that anyone in an organization might need to analyze.

Commonly people use Hadoop to work on the data in the lake, but the concept is broader than just Hadoop.

### Capabilities of Data Lake

- To capture and store raw data at scale for a low cost
- To store many types of data in the same repository
- To perform transformations on the data
- To define the structure of the data at the time, it is used, referred to as schema

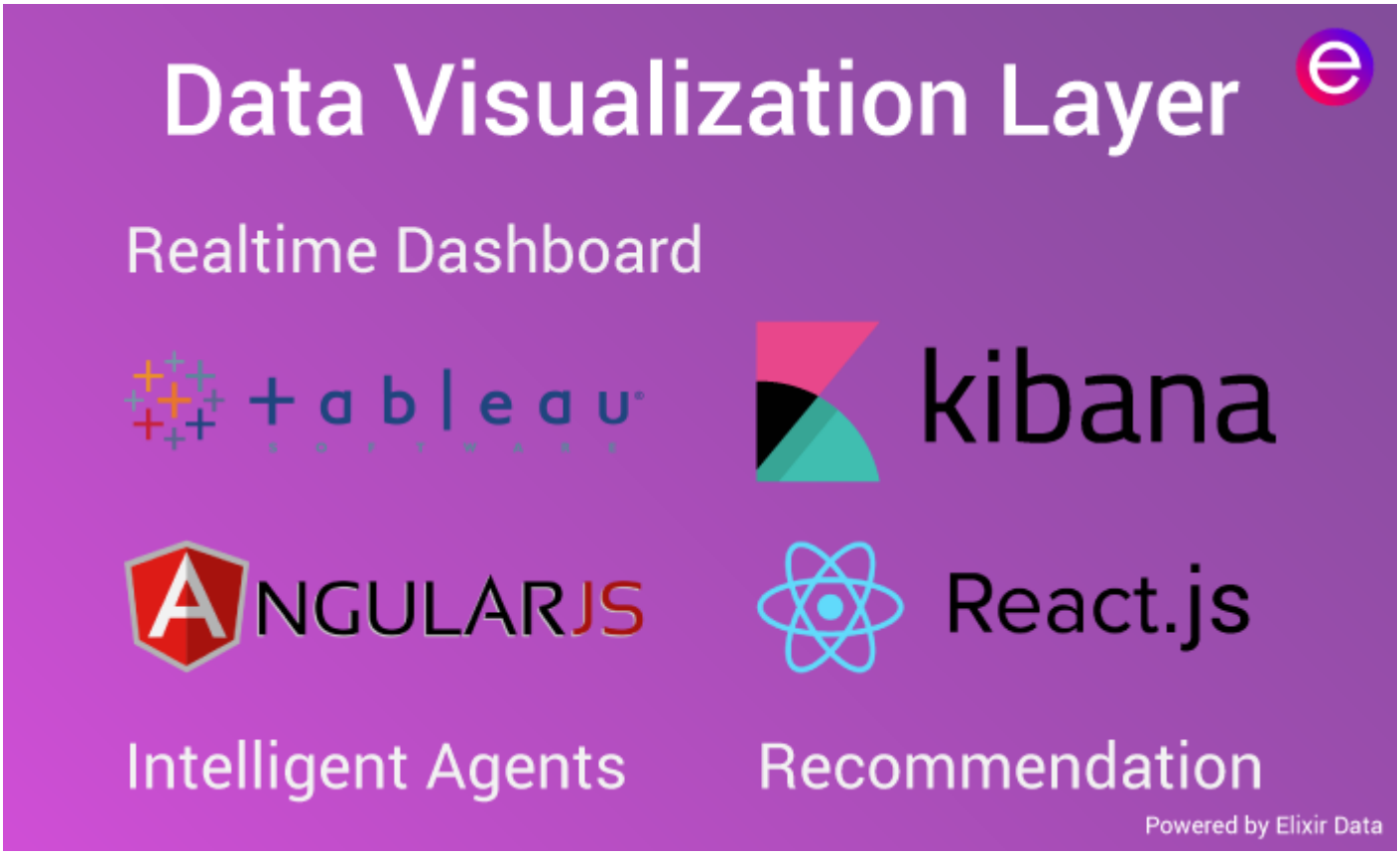
## Data Lake vs Data Warehouse

- With Data Lake incoming data goes into the lake in the raw form and then, we select and organize the data in a raw form. In Data Warehouse Data is cleaned and organized into single consistent schema before putting them into a warehouse and then the analysis is done on the warehouse data.
- Data lakes retain all data. Not only the data that is in use but also data that it might use in the future. On the other hand, when a data warehouse is being developed, considerable time is spent in analyzing different data sources, along with understanding business processes and profiling of data. Data is kept in its raw form and is only transformed when it is ready to be used.
- In Data Lake all data in a data lake is stored in its natural form. Also, the data is always accessible to someone in need of it. In Data Warehouses difficulty faced when trying to induce a change in them. A lot of time is spent during development to get the structure of the warehouse right. Although a good warehouse design is capable of adapting to change.

---

## Real-Time Data Monitoring, Data Visualization, Big Data Security

This layer focus on Big Data Visualization. We need something that will grab people's attention, pull them in, make your findings well-understood. That's why it provides full Business Infographics. Because your findings from your data need the annotation and the bold canvas.




## Data Visualization Layer

The data visualization layer often is the thermometer that measures the success of the project. This is the where the data value is perceived by the user. While it's designed for handling and storing large volumes of data, Hadoop and other tools have no built-in provisions for data visualization and information distribution, leaving no way to make that data easily consumable by end business users.

## Tools For Building Data Visualization Dashboards

### Custom Dashboards for Data Visualization

Custom dashboards are useful for creating unique overviews that present data differently, For example, you can -

- Show the web and mobile application information, server information, custom metric data, and plugin metric data all on a single custom dashboard.
- Create dashboards that present charts and tables with a uniform size and arrangement on a grid.
- Select existing New Relic charts for your dashboard, or create your charts and tables.

## Real-Time Visualization Dashboards

Real-Time Dashboards save, share, and communicate insights. It helps users generate questions by revealing the depth, range, and content of their data stores.

- Data Visualization dashboards always change as new data arrives.
- In Zoomdata, you have the flexibility to create a data analytics dashboard with just a single chart and then add to it as needed.
- Dashboards can contain multiple visualizations from multiple connections side by side.
- You can quickly build, edit, filter, and delete dashboards and move and resize them and then share them or integrate them into your web application.
- You can export a dashboard as an image or as a file configuration like JSON.
- You can also make multiple copies of your dashboard.

## Data Visualization with Tableau

- Tableau is the richest data visualization tool available in the market. With Drag and Drop functionality.
- Tableau allows users to design Charts, Maps, Tabular, Matrix reports, Stories and Dashboards without any technical knowledge.
- Tableau helps anyone quickly analyze, visualize and share information. Whether it's structured or unstructured, petabytes or terabytes, millions or billions of rows, you can turn big data into big ideas.
- It connects directly to local and cloud data sources, or import data for fast in-memory performance.
- Make sense of big data with easy-to-understand visuals and interactive web dashboards.

## Exploring data sets With Kibana

- A Kibana *dashboard* displays a collection of saved visualizations. You can arrange and resize the visualizations as needed and save dashboards, so they are reloaded and shared.
- Kibana act as analytics and visualization platform that builds on Elasticsearch to give you a better understanding of your data.
- Application Performance Monitoring is one key area to implement in projects to ensure proper and smooth operations from day 1. APM solutions provide development and operations team with near real-time insights on how the applications and services are performing in production, allowing for a proactive tune of services, as well as for early detection of possible production issues.
- It gives you the freedom to select the way you give shape to your data. And you don't always have to know what you're looking for.
- Kibana core ships with the classics: histograms, line graphs, pie charts, sunbursts, and more. They leverage the full aggregation capabilities of Elasticsearch.
- The Kibana interface is divided into four main sections:

- Discover
- Visualize
- Dashboard
- Settings

## Introduction to Intelligence Agents

- An intelligent agent is a software that assists people and acts on their behalf. Intelligent agents work by allowing people to delegate work that they could have done, to the agent software.
- Agents can perform repetitive tasks, remember things you forgot, intelligently summarize complex data, learn from you and even make recommendations to you.
- An intelligent agent can help you find and filter information when you are looking at corporate data or surfing the Internet and don't know where the right information is.
- It could also customize information to your preferences, thus saving you the time of handling it as more and more new information arrived each day on the Internet.
- An agent could also sense changes in its environment and responds to these changes.
- An agent continues to work even when the user is gone, which means that an agent could run on a server, but in some cases, an agent runs on the user systems.

## Recommendation Systems

- Recommender systems provide personalized information by learning the user's interests from traces of interaction with that user. For a recommender system to make predictions about a user's interests, it has to determine a user model.
- A user model contains data about the user and should be represented in such a way that the data can be matched to the items in the collection.
- The question is, what kind of data can be used to construct a user profile.
- Obviously, the items that users have seen in the past are important, but other information such as the content of the items, the perception of users of the items or information about users themselves could also be used.
- Most recommender systems focus on the task of information filtering, which deals with the delivery of elements selected from an extensive collection that the user is likely to find interesting or useful.
- Recommender systems are unique types of information filtering systems that suggest items to users. Some of the largest e-commerce sites are using recommender systems and apply a marketing strategy that is referred to as mass customization**.**
- A content-based filtering system often uses many of the same techniques as an information retrieval system (such as a search engine), because both systems require a content description of the items in their domain. A recommender system also requires the modelling of the user's preferences for a longer period which is not needed in an information retrieval system.
- There are several techniques that can be used to improve recommender systems in different ways.

# Big Data Security and Data Flow


Security is the primary task of any work. Security should be implemented at all layers of the lake starting from Ingestion, through Storage, Analytics, Discovery, all the way to Consumption. For proving security to data pipeline, few steps are there that are:-

- **Big Data Authentication**

Authentication will verify user's identity and ensure they are who they say they are. Using the **Kerberos protocol** provides a reliable mechanism for authentication.

- **Access Control**

It is the next step to secure data, by defining which dataset can be consulted by the users or services. Access control will restrict users and services to access only that data which they have permission for; they will access all the data.

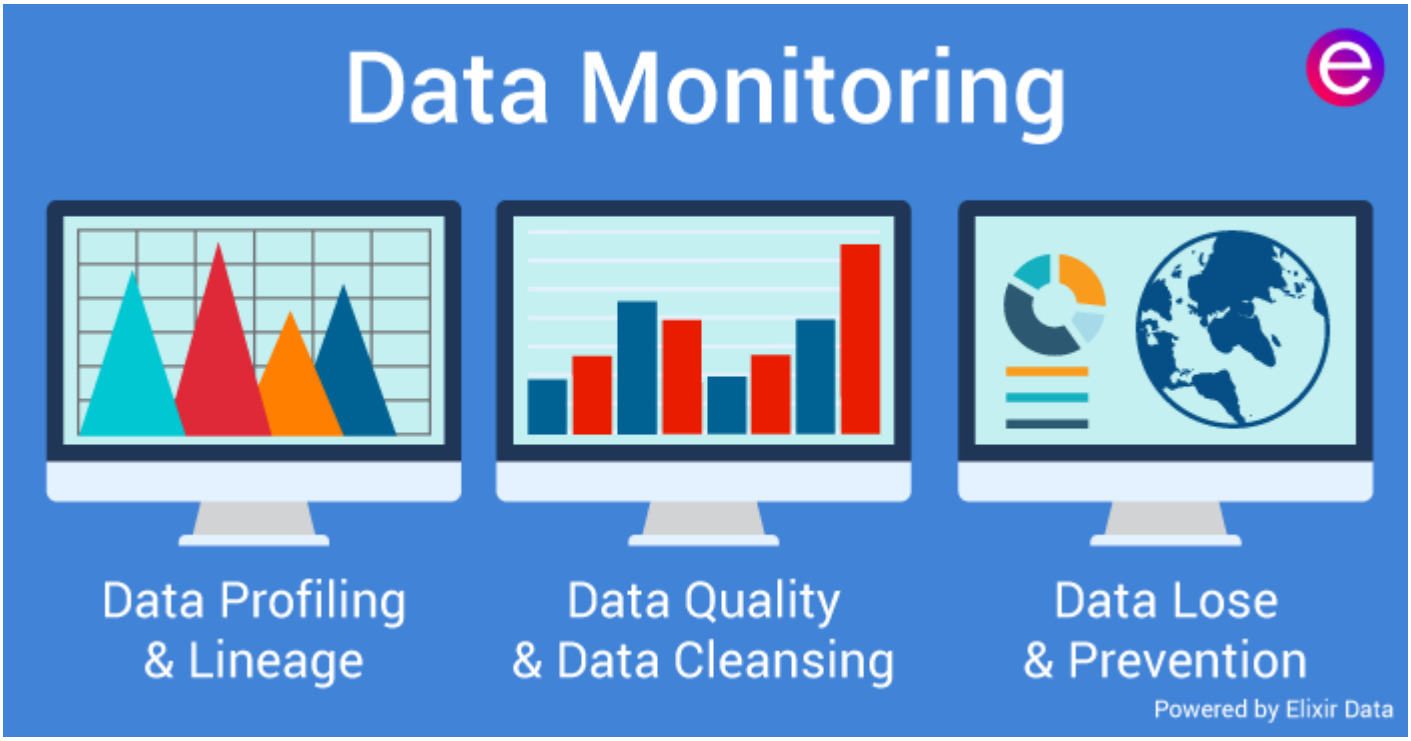
- **Encryption and Data Masking**

Encryption and data masking are required to ensure secure access to sensitive data. Sensitive data in the cluster should be secured at rest as well as in motion. We need to use proper Data Protection techniques which will protect data in the cluster from unauthorized visibility.

- **Auditing data access by users**

Another aspect of data security requirement is Auditing data access by users. It can detect the log on & access attempts as well as the administrative changes.

# Real-Time Data Monitoring




Data In enterprise systems is like food – it has to be kept fresh. Also, it needs nourishment. Otherwise, it goes wrong and doesn't help you in making strategic and operational decisions. Just as consuming spoiled food could make you sick, using "spoiled" data may be bad for your organization's health.

There may be plenty of data, but it has to be reliable and consumable to be valuable. While most of the focus in enterprises is often about how to store and analyze large amounts of data, it is also essential to keep this data fresh and flavorful.

So we can do this? The solution is for monitoring, auditing, testing, managing, and controlling the data. Continuous monitoring of data is an important part of the governance mechanisms.

Apache Flume is useful for processing log data. Apache Storm is desirable for operations monitoring Apache Spark for streaming data, graph processing, and machine learning. Monitoring can happen in data storage layer. It includes following steps for data monitoring:-

- **Data Profiling and lineage**

These are the techniques to identify the quality of data and the lifecycle of the data through various phases. In these systems, it is important to capture the metadata at every layer of the stack so it can be used for verification and profiling.Talend, Hive, Pig.

- **Data Quality**

Data is considered to be of high quality if it meets business needs and it satisfies the intended use so that it's helpful in making business decisions successfully. So, understanding the dimension of greatest interest and implementing methods to achieve it is important.

- **Data Cleansing**

It means implementing various solutions to correct the incorrect or corrupt data.

- **Data Loss and Prevention**

Policies have to be in place to make sure the loopholes for data loss are taken care of. Identification of such data loss needs careful monitoring and quality assessment processes.

# Typical Hadoop Cluster

### Hadoop and HBase clusters have two types of machines:

Masters -- HDFS NameNode, YARN ResourceManager, and HBase Master.

Slaves -- HDFS DataNodes, YARN NodeManagers, and HBase RegionServers.

The DataNodes, NodeManagers, and HBase RegionServers are co-located or co-deployed for optimal data locality.

In addition, HBase requires the use of a separate component (ZooKeeper) to manage the HBase cluster.

### Hortonworks recommends separating master and slave nodes because:

Task/application workloads on the slave nodes should be isolated from the masters.

- Slaves nodes are frequently decommissioned for maintenance.

For evaluation purposes, it is possible to deploy Hadoop using a single-node installation (all the masters and slave processes reside on the same machine).

For a small two-node cluster, the NameNode and the ResourceManager are both on the master node, with the DataNode and NodeManager on the slave node.

Clusters of three or more machines typically use a single NameNode and ResourceManager with all the other nodes as slave nodes. A High-Availability (HA) cluster would use a primary and secondary NameNode , and might also use a primary and secondary ResourceManager .

Typically, a medium-to -large Hadoop cluster consists of a two-level or three-level architecture built with rack-mounted servers. Each rack of servers is interconnected using a 1 Gigabyte Ethernet (GbE) switch. Each rack-level switch is connected to a cluster-level switch (which is typically a

larger port-density 10GbE switch). These cluster-level switches may also interconnect with other cluster-level switches or even uplink to another level of switching infrastructure.

### Partitioning Recommendations for Slave Nodes

- Hadoop Slave node partitions: Hadoop should have its own partitions for Hadoop files and logs. Drives should be partitioned using ext3, ext4, or XFS, in that order of preference. HDFS on ext3 has been publicly tested on the Yahoo cluster, which makes it the safest choice for the underlying file system. The ext4 file system may have potential data loss issues with default options because of the "delayed writes" feature. XFS reportedly also has some data loss issues upon power failure. Do not use LVM; it adds latency and causes a bottleneck.
- On slave nodes only, all Hadoop partitions should be mounted individually from drives as "/grid/[0-n]".
- Hadoop Slave Node Partitioning Configuration Example:
    - /root - 20GB (ample room for existing files, future log file growth, and OS upgrades)
    - /grid/0/ - [full disk GB] first partition for Hadoop to use for local storage
    - /grid/1/ - second partition for Hadoop to use
    - /grid/2/ - ...

**Table 1.1 Sizing Recommendations**

| Machine Type | Workload Pattern/ Cluster Type | Storage[1] | Processor (# of Cores) | Memory (GB) | Network |
|---|---|---|---|---|---|
| **Slaves** | Balanced workload | Twelve 2-3 TB disks | 8 | 128-256 | 1 GB onboard, 2x10 GBE mezzanine/external |
| | Compute-intensive workload | Twelve 1-2 TB disks | 10 | 128-256 | 1 GB onboard, 2x10 GBE mezzanine/external |
| | Storage-heavy workload | Twelve 4+ TB disks | 8 | 128-256 | 1 GB onboard, 2x10 GBE mezzanine/external |
| **NameNode** | Balanced workload | Four or more 2-3 TB RAID 10 with spares | 8 | 128-256 | 1 GB onboard, 2x10 GBE mezzanine/external |

| Machine Type | Workload Pattern/ Cluster Type | Storage[1] | Processor (# of Cores) | Memory (GB) | Network |
|---|---|---|---|---|---|
| **ResourceManager** | Balanced workload | Four or more 2-3 TB RAID 10 with spares | 8 | 128-256 | 1 GB onboard, 2x10 GBE mezzanine/external |

## Typical Workload Patterns For Hadoop

Disk space, I/O Bandwidth (required by Hadoop), and computational power (required for the MapReduce processes) are the most important parameters for accurate hardware sizing. Additionally, if you are installing HBase, you also need to analyze your application and its memory requirements, because HBase is a memory intensive component. Based on the typical use cases for Hadoop, the following workload patterns are commonly observed in production environments:

### Balanced Workload

If your workloads are distributed equally across the various job types (CPU bound, Disk I/O bound, or Network I/O bound), your cluster has a balanced workload pattern. This is a good default configuration for unknown or evolving workloads.

### Compute Intensive

These workloads are CPU bound and are characterized by the need of a large number of CPUs and large amounts of memory to store in-process data. (This usage pattern is typical for natural language processing or HPCC workloads.)

### I/O Intensive

A typical MapReduce job (like sorting) requires very little compute power. Instead it relies more on the I/O bound capacity of the cluster (for example, if you have lot of cold data). For this type of workload, we recommend investing in more disks per box.

### Unknown or evolving workload patterns

### Challenges - Tuning job characteristics to resource usage

Relating job characteristics to resource requirements can be complex. How the job is coded or the job data is represented can have a large impact on resource balance. For example, resource cost can be shifted between disk IOPS and CPU based on your choice of compression scheme or parsing format. Per-node CPU and disk activity can be traded for inter-node bandwidth depending on the implementation of the Map/Reduce strategy.

## Security policies

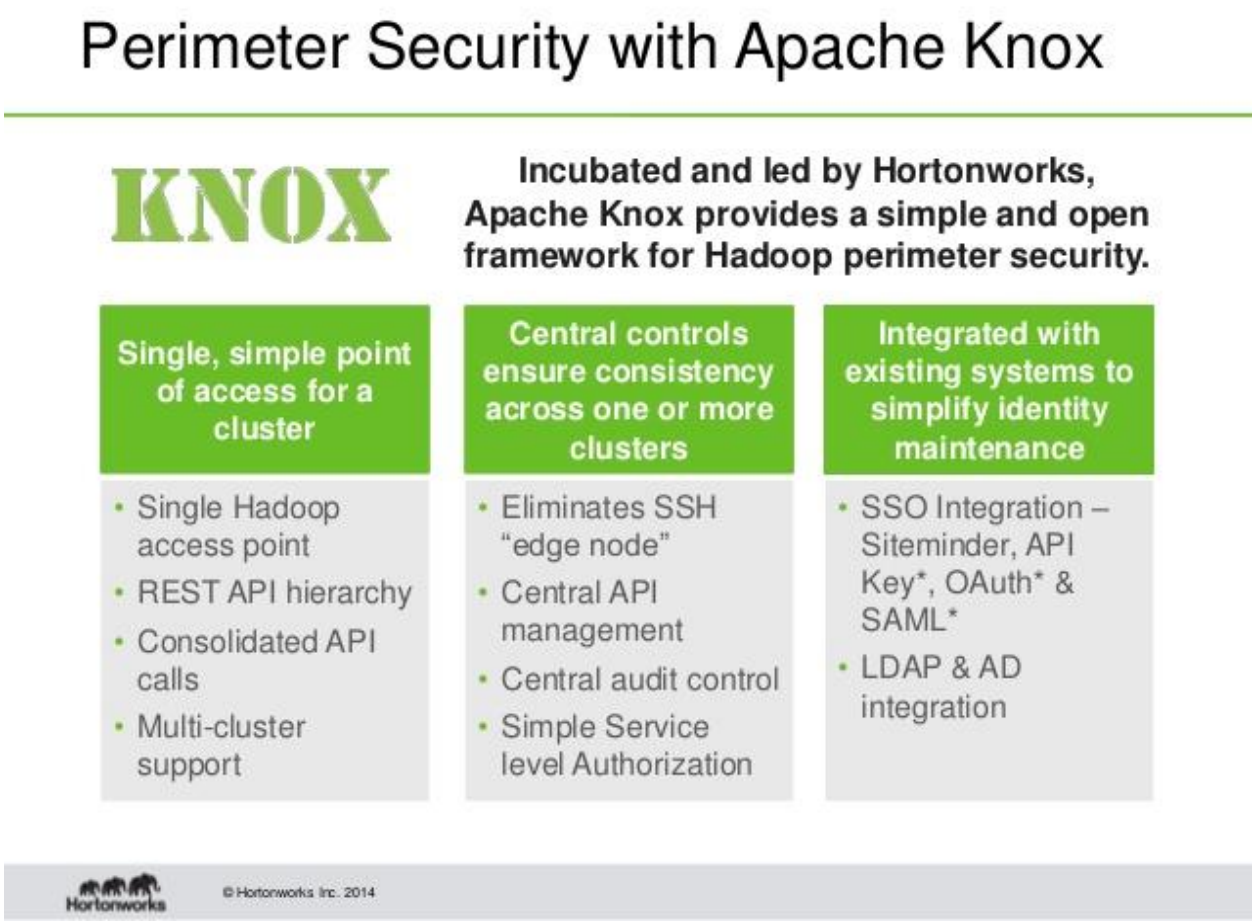
**Apache Atlas:**

**Atlas** is a scalable and extensible set of core foundational governance services – enabling enterprises to effectively and efficiently meet their compliance requirements within **Hadoop** and allows integration with the whole enterprise data ecosystem.

**Kerberos:**

Hadoop uses Kerberos as the basis for strong authentication and identity propagation for both user and services. Kerberos is a third party authentication mechanism, in which users and services rely on a third party - the Kerberos server - to authenticate each to the other.

**Knox Gateway:**




s

## Ranger:

### What does Apache Ranger offer for Apache Hadoop and related components?

**Apache Ranger** (https://hortonworks.com/hadoop/**ranger**/) is a centralized security administration solution for Hadoop that enables administrators to create and enforce security policies for HDFS and other Hadoop platform components.

### What projects does Apache Ranger support today

Apache Ranger supports fine grained authorization and auditing for following Apache projects:

- Apache Hadoop
- Apache Hive
- Apache HBase
- Apache Storm
- Apache Knox
- Apache Solr
- Apache Kafka
- YARN

# BigData Jobs Roles and Responsibilities:

## Chief data officer

### Responsibilities

a. Work with executives, data owners, and data stewards to achieve data accuracy and process requirement goals for all internal and external customers and create data management strategies.
b. Spearhead the data management activities performed by the EIM program, the business data stewards, and data service providers.
c. Establish data policies, standards, organization, and enforcement of EIM concepts as established by the organization.
d. Oversee the monitoring of data quality efforts within the organization and provides a central authority for the resolution on data management issues that cannot be resolved by the data governance council.
e. Establish data vendor management strategy and provide oversight to support implementation by the EIM program and coordinates with the IT organization through the CIO/CTO.
f. Lead the creation of program business definitions and data management goals and principles for execution by the EIM program.
g. Responsible for enterprise information/data management budget and data-related systems initiatives.

## Data analyst

### Responsibilities

a. Coordinate with customers and staff and provide support to all data analysis.

b. Perform data analysis on all results and prepare presentations for clients.
c. Perform audit on data and resolve business-related issues for the customer base.
d. Coordinate with engineering and product management team and ensure accuracy on all deliverables and prepare summaries.
e. Perform data analysis and facilitate in delivery to all end users.
f. Supervise all client issues and coordinate with managers and supervisors and facilitate in deliverables.
g. Monitor and organize all client invoices and perform all timely assessment for all payment issues.
h. Administer all data for customer invoices and provide company metrics.
i. Monitor and resolve all customer invoice data issues and coordinate with various vendors and manage all previous balance.
j. Organize all consumption anomalies and determine defects for data and prepare appropriate resolutions.
k. Supervise process management tools and ensure compliance to all cycle guidelines.
l. Maintain and document library of invoices and resolve all issues in same.
m. Perform internal audit and prepare all invoices and determine quality improvement processes.

# Big data visualizer

## Responsibilities

a. Provide value-add analysis to Business through the use of visualization software to guide analysis, drawing implications from the analysis, and synthesizing into clear communications.
b. Understand how data flows within various systems to provide input on requirements for databases to ensure data is organized properly for

reporting/analytics.
c. Work closely with Data Quality teams to ensure data integrity & completeness.
d. Develop business requirements to drive functional specifications for reporting applications.
e. Work with business and cross-functional teams to thoroughly document reporting processes and systems.
f. The acquisition, management, and documentation of data (including geospatial data).
g. Work with clients/client service teams to plan and carry out data analyses.
h. Participate in proposal writing, client deliverables, and research papers.
i. Create visualizations from data / GIS data analysis for inclusion in proposals, reports, papers, and multi-media projects.

# Big data solutions architect

## Responsibilities

a. Guide the full lifecycle of a Hadoop solution, including requirements analysis, platform selection, technical architecture design, application design and development, testing, and deployment.
b. Provide technical and managerial leadership in a team that designs and develops path-breaking large-scale cluster data processing systems.
c. Help Xtremeinsights customers develop strategies that maximize the value of their data.
d. Help Xtremeinsights establish thought leadership in the big data space by contributing white-papers, technical commentary to the community.

# Big data engineer

## Responsibilities

a. Gather and process raw data at scale (including writing scripts, web scraping, calling APIs, write SQL queries, etc.).
b. Work closely with our engineering team to integrate your amazing innovations and algorithms into our production systems.
c. Process unstructured data into a form suitable for analysis – and then do the analysis.
d. Support business decisions with ad hoc analysis as needed.

# Big data researcher

## Responsibilities

a. Extract data from a variety of relational databases, manipulate, explore data using quantitative, statistical and visualization tools
b. Inform the selection of appropriate modeling techniques to ensure that predictive models are developed using rigorous statistical processes
c. Establish and maintain effective processes for validating and updating predictive models
d. Analyze, model, and forecast health service utilization patterns/ trends and create the capability to model outcomes of what-if scenarios for novel health care delivery models
e. Collaborate with internal business, analytics, and data strategy partners to improve efficiency and increase the applicability of predictive models into the core software products
f. Help manage the innovation cycle of conducting analyses, generating

insights, advocating for the integration of new concepts into existing client tools, helping to translate ad-hoc analyses into scalable software solutions.