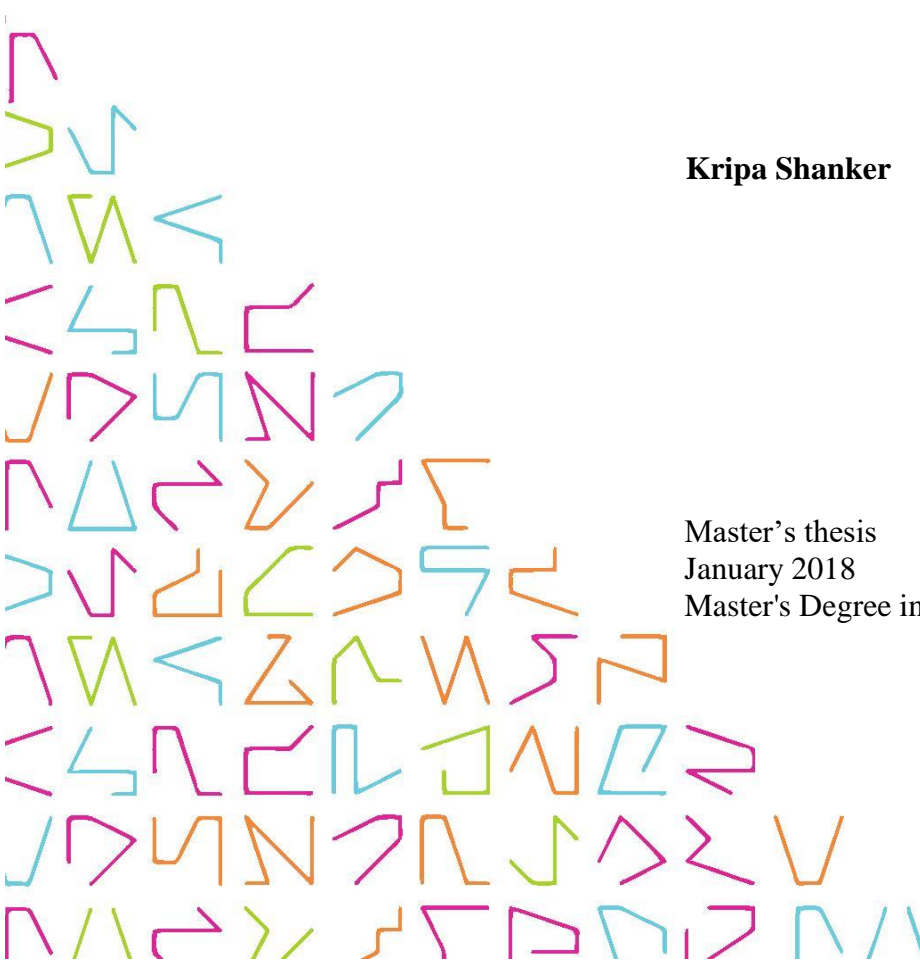


# **Big Data Security Analysis And Secure Hadoop Server**

**Kripa Shanker**

Master's thesis  
January 2018  
Master's Degree in Information Technology



## ABSTRACT

Tampereen Ammattikorkeakoulu  
Tampere University of Applied Sciences  
Master's Degree in Information Technology

Kripa Shanker

Big data security analysis and secure hadoop server

Master's thesis 62 pages, appendices 4 pages  
January 2018

---

Hadoop is a so influential technology that's let us to do incredible things but major thing to secure informative data and environment is a big challenge as there are many bad guys (crackers, hackers) are there to harm the society using this data. Hadoop is now used in retail, banking, and healthcare applications; it has attracted the attention of thieves as well. While storing sensitive huge data, security plays an important role to keep it safe. Security was not that much considered when Hadoop was initially designed.

Security is an important topic in Hadoop cluster. Plenty of examples are available in open media on data breaches and most recently was RANSOMEWARE which get access in server level which is more dangerous for an organizations. This is best time to only focus on security at any cost and time needed to secure data and platform. Hadoop is designed to run code on a distributed cluster of machines so without proper authentication anyone could submit code and it would be executed. Different projects have started to improve the security of Hadoop.

In this thesis, the security of the system in Hadoop version 1, Hadoop version 2 and Hadoop version 3 is evaluated and different security enhancements are proposed, considering security improvements made by the two mentioned projects, Project Apache Knox Gateway, Project Apache Ranger and Apache Sentry, in terms of encryption, authentication, and authorization.

This thesis suggests some high-level security improvements using hardening server implementing all best security practices and Kerberos. Hadoop Kerberos-based authentication is currently getting used widely. I am also going to use ElasticSearch method to analyze server errors log data to catch culprit of system and Server parameters, critical system alert notification with NetData.

**Key words:** ElasticSearch, Encryption methodology, Kibana, Big Data, LogStash, NetData.

## CONTENTS

1.	HADOOP INTRODUCTION.....	6
1.1	Overview of Hadoop.....	6
1.2	Hadoop Architecture.....	9
1.2.1.	Hadoop V.1.....	9
1.2.2.	Hadoop V.2.....	12
1.2.3	Hadoop V.3.....	15
1.3.	File System for Hadoop.....	16
1.3.1.	HDFS overview.....	17
1.3.2	Features of the HDFS.....	20
2.	BIG DATA TYPES.....	21
2.1.	Big data as per data source .....	22
2.1.1.	Fast Data.....	22
2.1.2.	Dark Data.....	22
2.1.3.	Lost Data.....	22
2.1.4	New Data.....	23
2.2.	Big data as per structures.....	23
2.2.1.	Structured data .....	23
2.2.2.	Unstructured Data.....	24
2.2.3.	Semi-structured data.....	24
2.2.4.	Big Data as per storage.....	25
2.2.5	Big Data as a Service (BDaaS) .....	25
3.	TYPES OF SECURITY.....	27
3.1	Approach to secure Hadoop echo system .....	27
3.2.	Hadoop Security Projects.....	27
3.2.1.	Apache Knox Gateway.....	28
3.2.2.	Apache Sentry.....	32
3.2.3.	Apache Ranger.....	34
3.2.4.	Apache Accumulo.....	35
3.2.5.	Apache Rhino.....	35
3.2.6	Apache Eagle.....	36
3.3	Information Security Compliance Regulations.....	37
4.	HADOOP SERVER LEVEL SECURITY .....	38
4.1.	Best practices to hardening Linux Server.....	38
4.2.	Server Logging.....	40
4.3.	Auditing Hadoop Servers.....	42
4.4	Configure Audit Daemon to audit Linux / UNIX server.....	42
5.	ELASTICSEARCH, LOGSTASH AND KIBANA FOR SECURITY ....	46
5.1.	Installation of Elasticsearch, Logstash and Kibana.....	48
5.1.1.	Installation of Elasticsearch.....	48
5.1.2.	Installation of Logstash.....	51
5.1.3	Installation of Kibana.....	52
6.	MONITORING HADOOP SERVER WITH NETDATA .....	56
6.1.	Introduction.....	56
6.2.	Interface of NetData .....	56
6.3.	Monitored Server Parameters.....	56
6.4	How to Setup / Install NetData.....	57

7. CONCLUSION.....	64
LIST OF REFERENCES.....	66
APPENDICES.....	68
Appendix 1. Project usage important Linux commands.....	69
Appendix 2. Well known securities methods.....	69
Appendix 3. Levels in hadoop security.....	70
Appendix 4. Complete layered and functional security of hadoop server....	70

### List of Figures

FIGURE 1: Apache hadoop ecosystem
FIGURE 2: Decide preventive security measures
FIGURE 3: Hadoop 1.0 architecture
FIGURE 4: Hadoop 1.x components in-detail architecture
FIGURE 5: architectural difference between hadoop 1.0 and 2.0
FIGURE 6: MapReduce Vs yarn architecture
FIGURE 7: Hadoop 2.x architecture
FIGURE 8: Hortonworks hadoop 3.0 architecture
FIGURE 9: HDFS federation architecture
FIGURE 10: HDFS architecture
FIGURE 11: 7 V's of big data
FIGURE 12: BDaaS offerings in the cloud into one of 4 types
FIGURE 13: Apache Knox gateway for hadoop ecosystem
FIGURE 14: Apache Sentry data and server
FIGURE 15: Apache Eagle Hadoop security architecture
FIGURE 16: Elasticsearch architecture model
FIGURE 17: Logstash Architecture model
FIGURE 18: Kibana Architecture model
FIGURE 19: Kibana dashboard
FIGURE 20: Netdata dashboard
FIGURE 21: Netdata server monitoring menu
FIGURE 22: Hadoop server memory graph
FIGURE 23: Hadoop server disks graph
FIGURE 24: Hadoop server kernel graph
FIGURE 25: Hadoop server IPv6 networking graph
FIGURE 26: Complete layered and functional security of hadoop server

## ABBREVIATIONS AND TERMS

ACL	Access control list.
HDFS	Apache Hadoop Distributed File System.
IDE	Integrated Development Environment.
YARN	Apache Hadoop resource- Yet another Resource Negotiator.
N/W	Network
S/W	Software
UPN	Uniform Principle Name
URI	Uniform Resource Identifier
PB	Peta Bytes
EB	Exabyte
MTDS	Multi-terabyte data-sets
HUE	A browser-based desktop interface for interacting with Hadoop
EAB	Elementary Attack Behavior
HIPPA	Health Insurance Portability and Accountability Act
PCI	Payment Card Industry
DSS	Data Security Standard
FISMA	Federal Information Security Management Act
URL	Uniform Resource Locator
DoS	Denial-of-Service
DLP	Data Leakage Protection
HLA	Homomorphic Linear Authenticators
IaaS	Infrastructure as a Service
AD	Active Directory
LDAP	Lightweight Directory Access Protocol
IDPS	Intrusion Detection and Prevention Systems
LUN	Logical Unit Number
MD5	Message Digest 5
PaaS	Platform as a Service
PKI	Public Key Infrastructure
SHA	Secure Hash Algorithm
SOA	Service Oriented Architecture
BDBaaS	Big Data Base as a Service
TKS	Ticket Granting Service

## 1. INTRODUCTION

In this technology-driven world, computers have penetrated all walks of our life, which allows more of our personal and corporate data available electronically for anyone to access. Unfortunately, the technology that provides so many benefits is the same that can be used for destructive purposes.

Hadoop is a highly scalable, flexible platform to support all manner of analytics-focused, data-centric applications. But it was not developed with security or data governance in mind. Security and data governance are critical components to any data platform that is trying to break into the enterprise mainstream. The Hadoop community both the open source community and vendor community – have made significant efforts in recent times, but still more work is needed. It is critical that the Hadoop community comes together to solve Hadoop's security challenges because these are precisely the issues that are and will continue to prevent many practitioners from deploying Hadoop to support industrial scale production-grade workloads and mission-critical applications.

For organizations that store sensitive data in the Hadoop ecosystem, such as proprietary or personal data that is subject to regulatory compliance (HIPPA, PCI, DSS, FISAM, etc.), security is the most important aspect.

The various pillars of security are Identity and Access, Network security, Data security, Application security and last but not the least monitoring and quick fixes. This thesis concentrates primarily on Hadoop server security via Apache KNOX project and encryption.

### 1.1 Overview of Hadoop

First, we have to know more about Hadoop and its ecosystems to understand better and implement security at every layer.

Apache foundation's Hadoop is an open-source framework, that has a capability of processing larger amounts of structured, unstructured and semi structured data sets in a distributed fashion across clusters & nodes of commodity computers & hardware using a simplified programming model. Hadoop provides a reliable shared storage and

analysis system which is implemented to scale from a single cluster to thousands of different servers. The two staple components required in any Hadoop cluster are a Hadoop Distributed File System (HDFS) and Hadoop MapReduce. These two will be referred as basic building blocks for any Hadoop setup, but there are many other tools and applications or modules available for use.

The below diagram gives a brief understanding of the Hadoop Ecosystem.

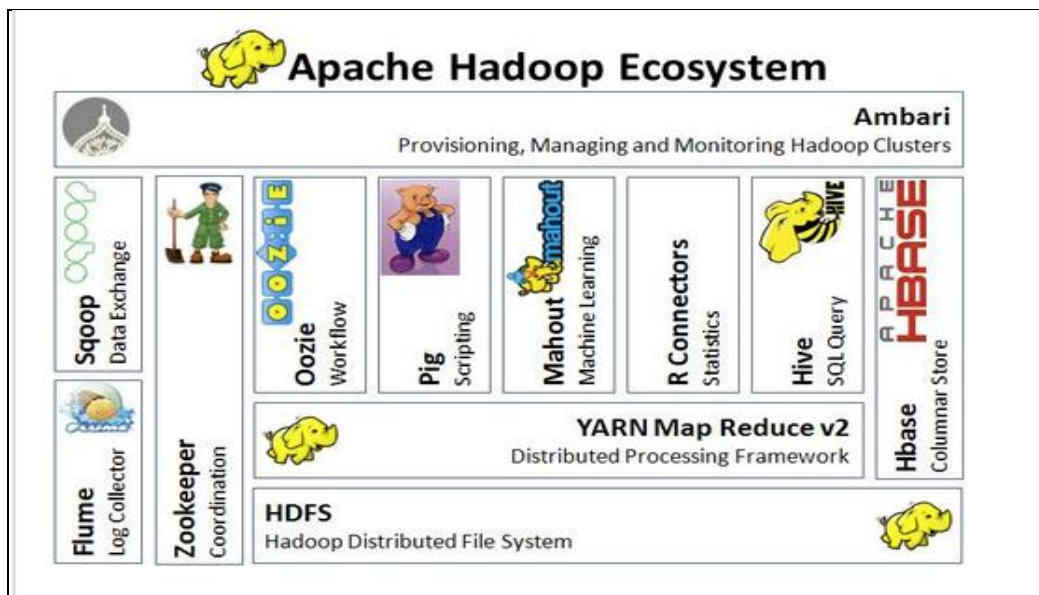


FIGURE 1. Apache hadoop ecosystem ([www.ossmentor.com/2015/07/what-is-hadoop-ecosystem.html](http://www.ossmentor.com/2015/07/what-is-hadoop-ecosystem.html))

### Hadoop Threats and Response to secure Hadoop ecosystem

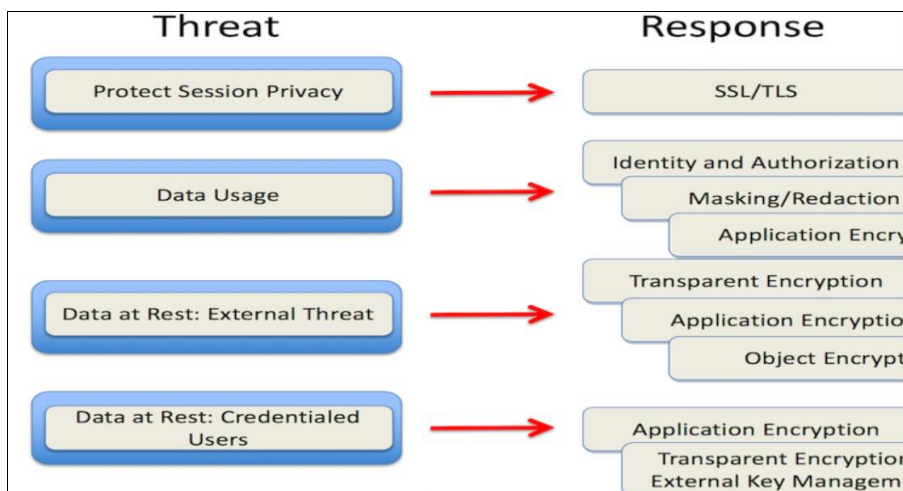


FIGURE 2. Decide preventive security measures (Securing\_hadoop\_final\_v2.pdf, p17)

**Hadoop ecosystem important Modules** description as per (Tom White, 2009)

Module	Explanation/Use
<b>HDFS</b>	Hadoop's File Share that can be local or shared depending on your setup
<b>MapReduce</b>	Hadoop's Aggregation/Synchronization tool enabling highly parallel processing
<b>Hive</b>	It is a SQL query window which is equal to Microsoft Query Analyzer.
<b>Pig</b>	A dataflow scripting tool equal to a Batch processing or job or a simple ETL processor.
<b>Flume</b>	Collector/Facilitator of Log file information
<b>Ambari</b>	Web-based Admin tool for managing, provisioning and monitoring Hadoop Cluster
<b>Cassandra</b>	High-Availability, Scalable, Multi-Master database platform
<b>Spark</b>	Programmatic based compute engine allowing for ETL, machine learning and more
<b>ZooKeeper</b>	Coordinator service for all your distributed processing
<b>Oozie</b>	Workflow scheduler managing Hadoop jobs
<b>Hbase</b>	It will store, search and automatically share the table across multiple nodes
<b>Sqoop</b>	It is a command line tool that control mapping between tables and storage layer
<b>NoSQL</b>	This enable them to store and retrieve data from NoSQL Databases like MongoDB
<b>Mahout</b>	This is designed to handle various algorithms in Hadoop clusters
<b>Solr/Lucene</b>	Tool for indexing large blocks of unstructured text
<b>Avro</b>	Is a serialization system that bundle the data together with schema
<b>SQL</b>	Structured Query language use in RDBMS databases like Oracle, DB2,MySQL



## 1.2 Hadoop Architecture

### 1.2.1 Hadoop V.1

Hadoop 1.x major components are: HDFS and MapReduce. They are also known as the “Two Pillars” of Hadoop 1.x. As per (Sachin P Bappalige, blog) the below diagram

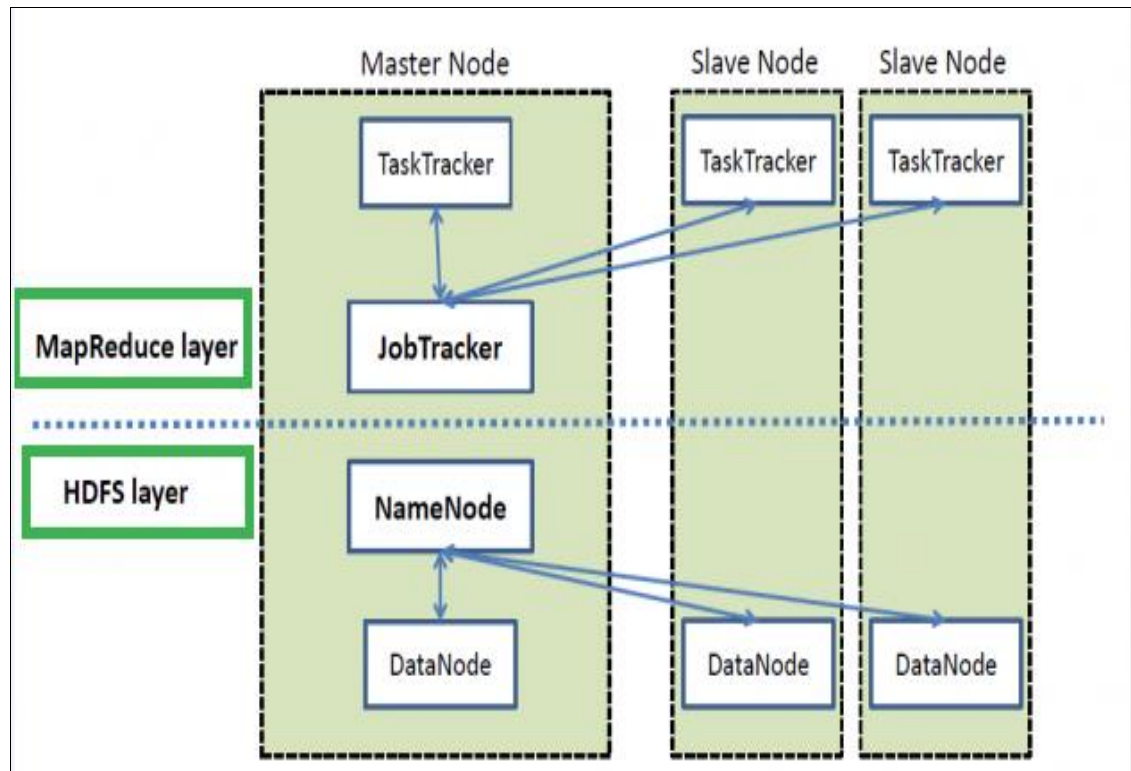


FIGURE 3. Hadoop 1.0 architecture ([www.opensource.com](http://www.opensource.com), hadoop bigdata info)

HDFS component is further divided into two sub-components:

**NameNode:** This sub-component of HDFS executes file system namespace operations and determines the mapping of blocks to DataNodes.

**DataNode:** This sub-component is responsible for serving read and write requests from the file system's clients.

#### **MapReduce:**

Hadoop MapReduce is a distributed data processing programming model. It is a software framework to write applications to process huge amounts of data (multi-

terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a very reliable, fault-tolerant manner.

A MapReduce job usually splits the input data-set into independent chunks which are then processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then submitted as inputs to the reduce tasks. Both the input and the output of the job will be stored in a file-system. The framework takes care of scheduling tasks, monitoring them and then re-executing the failed tasks.

Typically, the compute nodes and the storage nodes are the same, i.e. the MapReduce framework that helps in processing and the Hadoop Distributed File System (i.e. storage) are running on the same set of nodes. This configuration allows the framework to effectively schedule tasks on the nodes where data is already present, resulting in high aggregate bandwidth across the cluster.

The MapReduce framework consists of a master JobTracker and a slave TaskTracker per cluster-node. The master's task is scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks that are re-directed by the master.

Minimally, applications specify the input/output locations and supply the map and reduce functions via implementations of appropriate interfaces and abstract classes. Other job parameters along with these comprise the job configuration. The Hadoop job client then submits the job (which is a .jar/executable file etc.) and configuration to the JobTracker which then takes up the responsibility of distributing the software/configuration to the slaves, schedules and monitors the tasks, provides the status & diagnostic information to the client (i.e. job client).

Although the Hadoop framework is implemented in Java<sup>TM</sup>, MapReduce applications may not be written in Java.

- The utility Hadoop Streaming allows users to create and run jobs with any executables *like shell utilities* as the mapper and/or the reducer.
- Hadoop Pipes is a SWIG which is compatible C++ API to implement MapReduce applications that are non JNI<sup>TM</sup> based.

## Inputs and Outputs

The MapReduce framework operates extensively on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, possibly of various types.

The <key, value> classes should be serializable by the framework and hence need to adopt the Writable interface. In addition, the key classes have to implement the Writable Comparable interface to assist sorting by the framework.

Input and Output types of a MapReduce job:

(Input) <k1, v1> -> **map** -> <K2, v2> -> **combine** -> <K2, v2> -> **reduce** -> <k3, v3> (output)

## Hadoop 1.x Components In-detail Architecture

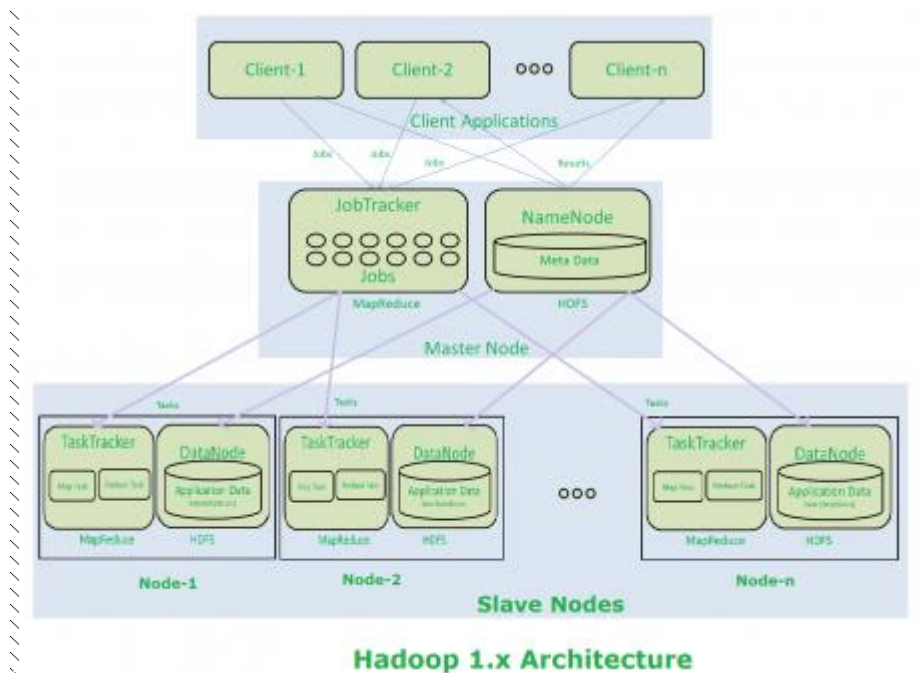


FIGURE 4. Hadoop 1.x components In-detail architecture (www.journaldev.com, 8808)

### Description of above diagram architecture:

1. Clients (one or more) submit their work to Hadoop System.

2. When Hadoop System receives a Client Request, first it is received by a Master Node.
3. Master Node's MapReduce component "Job Tracker" is responsible for receiving Client Work and divides it into manageable independent Tasks and assigns them to Task Trackers.
4. Slave Node's MapReduce component "Task Tracker" receives those Tasks from "Job Tracker" and performs those tasks by using MapReduce components.
5. Once all Task Trackers finished their job, Job Tracker takes those results and combines them into final result.
6. Finally Hadoop System will send that final result to the Client.

**Hadoop 1.x has the following Limitations/Drawbacks:**

1. This is limited to Batch Processing of larger amount of Data, which already exists Hadoop System.
2. It was not designed for Real-time Data Processing.
3. It is not suitable for Data Streaming.
4. Max of 4000 nodes can only be supported per Cluster.
5. JobTracker which is a single component to perform many activities such as Resource Management, Job Scheduling and re-scheduling, Job Monitoring, etc.
6. JobTracker is the single point of failure.
7. It does not support Multi-tenancy Support.
8. It supports only one Name Node and One Namespace per Cluster.
9. It does not support Horizontal Scalability.
10. It runs only Map/Reduce jobs.
11. It follows Slots concept in HDFS to allocate Resources (Memory, RAM, and CPU). It has static Map and Reduce Slots. That means once it assigns resources to Map/Reduce jobs, they cannot be re-used even though some slots are idle.

**1.2.2 Hadoop V.2**

Hadoop 1.x has many limitations or drawbacks. Main drawback of Hadoop 1.x is that MapReduce Component in its Architecture. This means it will support only

MapReduce-based batch or data processing applications. This release has resolved most of the Hadoop 1.x limitations by using new architecture.

### Basic architectural difference:

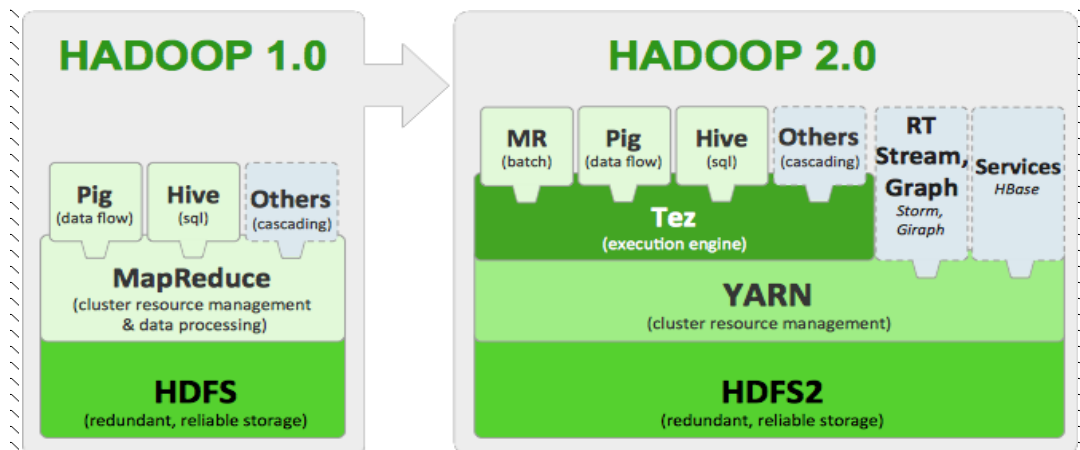


FIGURE 5. Architectural difference between Hadoop 1.0 and 2.0([www.dineshonjava.com/hadoop-architecture/](http://www.dineshonjava.com/hadoop-architecture/))

The Apache Hadoop framework is composed of the hadoop common, HDFS, hadoop Yarn and hadoop MapReduce. HDFS and MapReduce components are originally derived from google MapReduce and Google file system (Tom White 2009, 31).

### New Features of Hadoop 2.0:

1. Hadoop 2.x Architecture has one extra and new component that is: YARN (Yet another Resource Negotiator). YARN splits up the two major functionalities of overburdened JobTracker (resource management and job scheduling/monitoring) into two separate daemons that's are a global Resource Manager and Per-application Application Master.

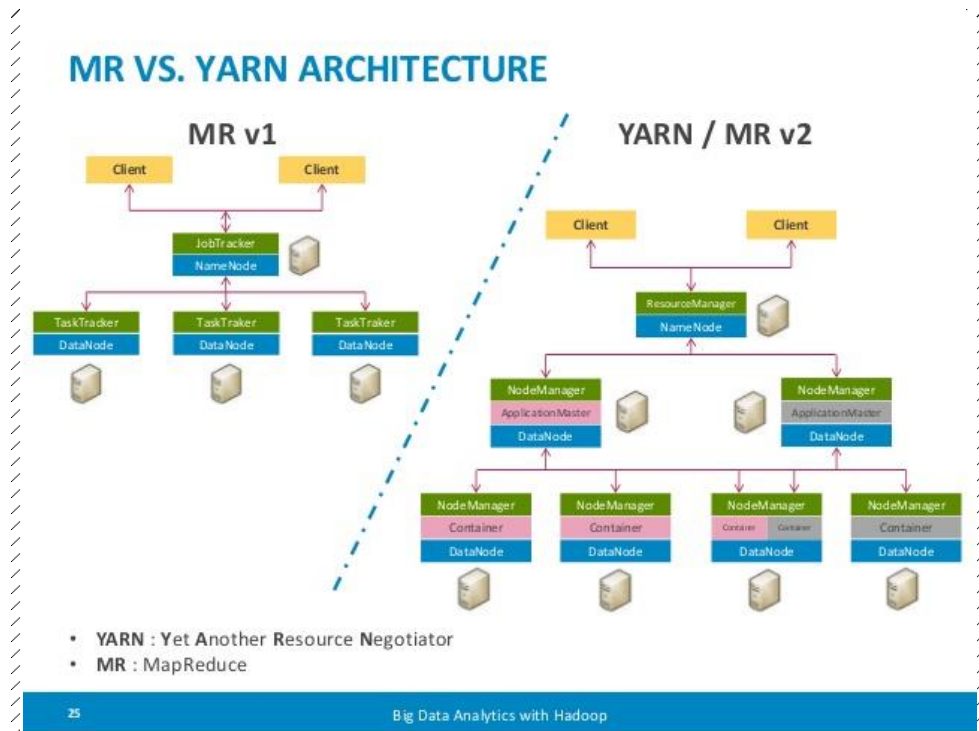


FIGURE 6. MR Vs Yarn architecture (Philippe Julio, hadoop-architecture, slideshare)

2. Hadoop 2.0 offers additional support and compatibility for file system snapshots. They are the sub trees of a specific file system or point-in-time images of complete file system.
3. Hadoop 2.0 users and applications of the likes of Pig, Hive, or HBase are capable of identifying different sets of files that require caching.
4. Hadoop 2.0 offers a solution for the problem on hand. To rescue, the High Availability feature of HDFS will allow any of the two redundant name nodes to run in the same cluster.
5. These name nodes may run in active/passive as assigned with one operating as the primary name node, and the other as a hot standby one.

## Hadoop 2.0 Architecture

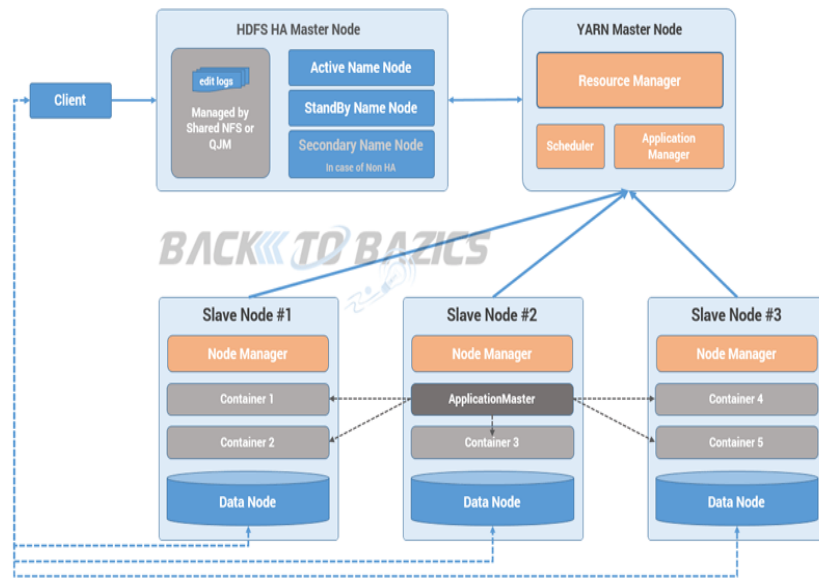


FIGURE 7. Hadoop 2.x architecture (understanding-hadoop2-architecture, backtobasics.com)

### Hadoop 2.x has the following Limitations/Drawbacks:

1. Fault Tolerance is handled through replication leading to storage and network bandwidth overhead.
2. YARN timeline service introduced in Hadoop 2.0 has some scalability issues.
3. HDFS Balancer in Hadoop 2.0 caused skew within a DataNode because of addition or replacement of disks
4. Java and Hadoop tasks, the heap size needs to be set through two similar properties

MapReduce. { map,reduce }.java.Opts and mapreduce. { map,reduce }.memory.mb

### 1.2.3 Hadoop V.3

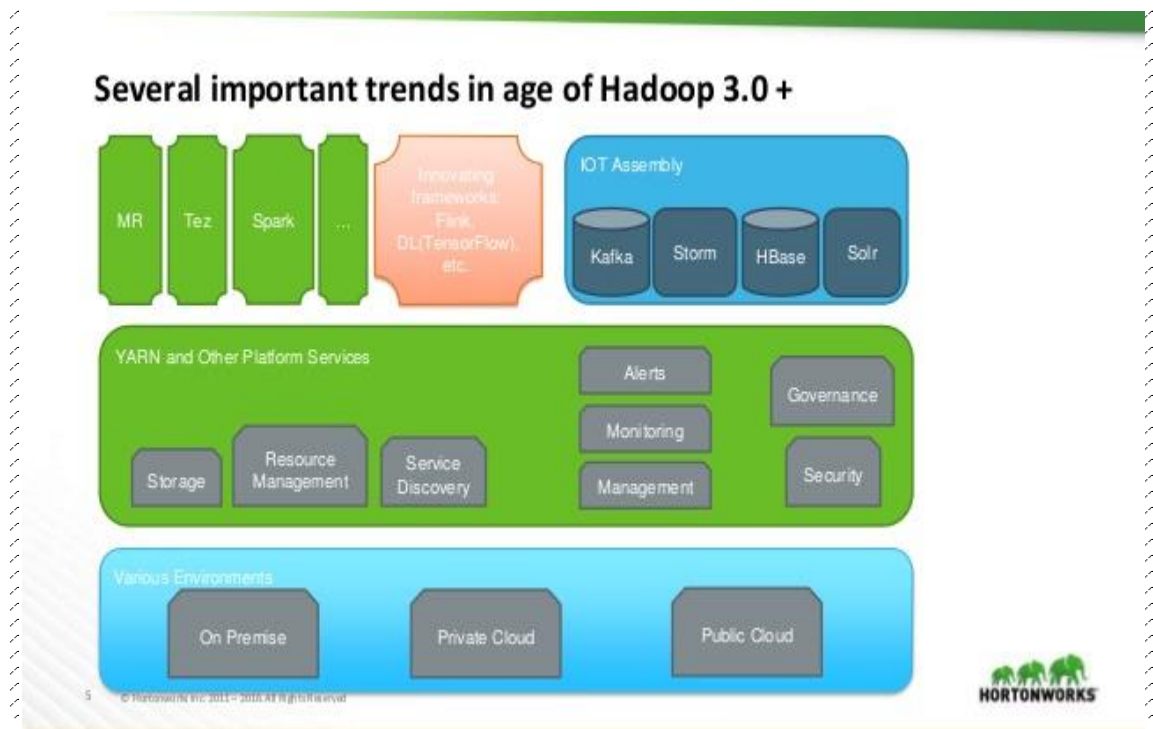


FIGURE 8. Hortonworks hadoop 3.0 architecture (Hadoop Summit, slideshare)

#### Benefits:

- YARN Timeline service has been enhanced with ATS v2 which has an improvement in the scalability and reliability.
- HDFS improves fault tolerance as it supports for Erasure Coding.
- This supports 2 or more NameNodes.
- In order to fix many bugs, resolve compatibility issues and change in some existing installation, shell scripts have been rewritten.
- The latest Hadoop-client-api and Hadoop-client-runtime artifacts the Hadoop's dependencies into a single jar.
- Support for Opportunistic Containers and Distributed Scheduling
- The map output collector in MapReduce was added with the native java implementation.
- For shuffle-intensive jobs that improves the performance over 30%.



- Now, Hadoop also supports integration with Microsoft Azure Data Lake & Aliyun Object Storage System. This can be an alternative to Hadoop-compatible file system.
- A lot many changes have been made to heap management for Hadoop daemons and MapReduce tasks.

### 1.3 File Systems for Hadoop

Big Data, the term that refers to large and complex data sets which needs to be processed by specially designed hardware and software tools.

The data sets are typically of the order of Tera or Exabyte in size. Big Data requires the storage of a huge amount of data. It is a necessity for advanced storage infrastructure; a need to have a storage solution which is designed in a “scale out” fashion on multiple servers. As per (Fei-Hu 2016, 19) Lustre and HDFS are leading technologies.

File Systems	Description
<b>Quantcast File System</b>	High-performance, fault-tolerant, distributed file system
<b>HDFS</b>	Distributed file system providing high-throughput access
<b>Ceph</b>	Unified, distributed storage system
<b>Lustre</b>	File system for computer clusters
<b>GlusterFS</b>	Scale-out NAS file system
<b>PVFS</b>	Designed to scale to petabytes of storage

HDFS is the most popular and used file system in Hadoop.

#### 1.3.1 HDFS overview

The Hadoop is a distributed file system designed to run on commodity hardware. It is almost same like existing distributed file systems. However, there are significant differences from other distributed file systems. HDFS is highly fault-tolerant and is designed to be deployed on commodity (low-cost) hardware. HDFS provides high throughput access to application data and is suitable for applications that contain huge data sets. HDFS ease a few POSIX requirements to allow streaming access to file system data.

The main goal of HDFS is detection of faults and quick, automatic recovery from them. HDFS has been designed to be easily transferable from one platform to another. HDFS provides interfaces for applications to move themselves closer to the data located. A usual file in HDFS will be in gigabytes to terabytes in size. Thus, HDFS is tuned to support large files, and to provide high aggregate data bandwidth and scale to hundreds of nodes in a single cluster. HDFS is specifically designed for batch processing rather than interactive use by users. The emphasis is on high throughput of data access instead of low latency of data access.

The HDFS Federation Architecture's pictorial representation is given below:

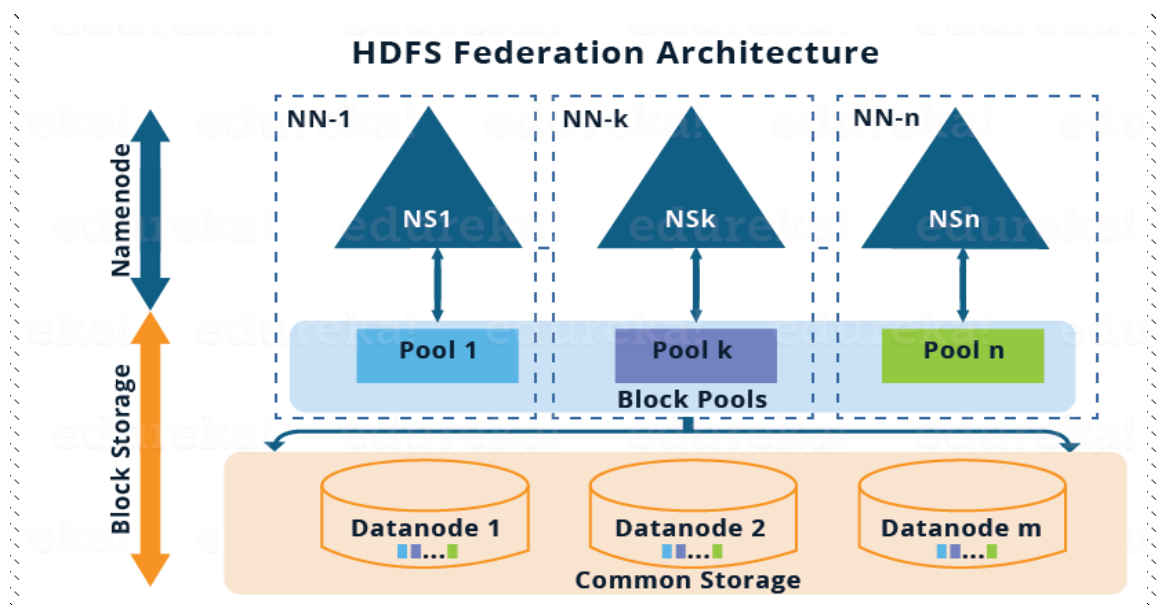


FIGURE 9. HDFS federation architecture (What-is-HDFS-federation, quora)

The originally released white paper by Google describing distributed file system which is also called as "Map Reduce Methodology" is the base technology to almost all the big data solutions provided by the number of companies.

The described method was developed as an open-source implementation in Yahoo's Apache project. The SW products hereby born were called Apache Hadoop, Hadoop's Distributed File System (HDFS) and Hadoop MapReduce.

Later, Apache Pig which is the 1st high-level non-SQL framework for Hadoop, Apache Hive which is Hadoop's 1st SQL access framework, Apache Spark which is an

evolution of MapReduce boosting performance /speed significantly and many more also added as the ecosystem components. Apache HBase is a further development of basic HDFS and is a fault tolerant for huge data including compression & filters.

This list of software technologies is not limited around Apache Hadoop, but it gives a rough understanding on the topic. From the initiation of Apache Hadoop project the evolution lasted over 10 years.

The below image describes Hadoop's distributed file system.

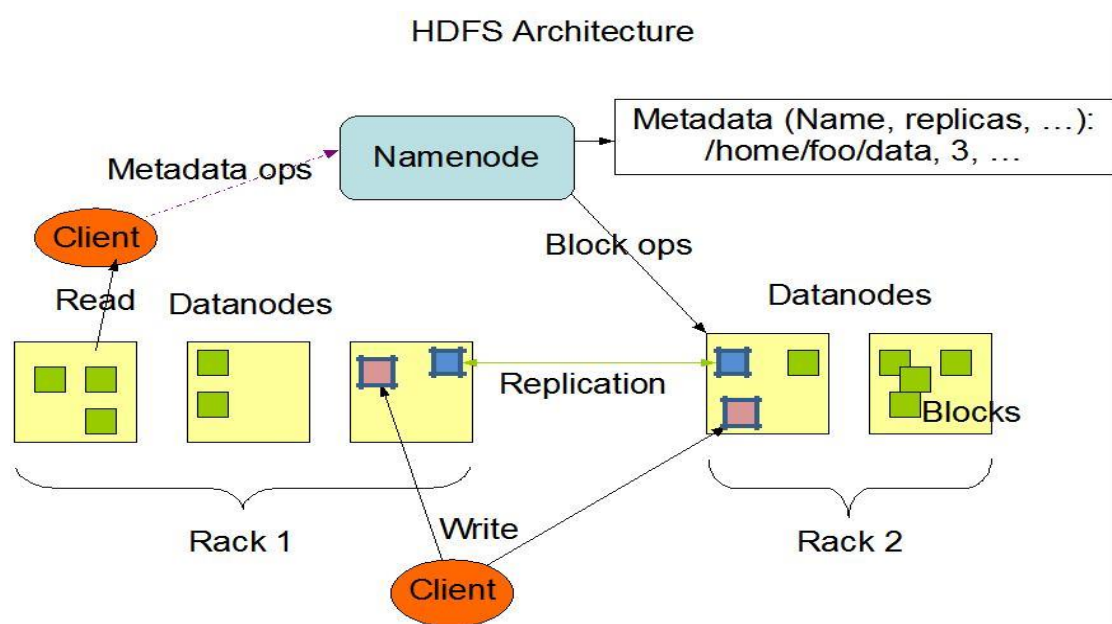


FIGURE 10. HDFS architecture (HDFS architecture guide hadoop1.2.1, page 5)

To protect from external hazards, the data is usually replicated to 3 large blocks (by default 128 MB), and the blocks are located at least in two different physical racks. The racks can be in the local area or in the same premises or even in different geographical areas, which may impact to the cost of the service. Datanodes will have processing elements with several processors and data storages. Namenode is responsible to keep track of the data blocks and their associations to the Datanodes. Namenode also responsible to detect if a Datanode fails to respond or data is lost in one block.

Extensive processing ability will be achieved through slicing the processing into several Datanodes. MapReduce will take responsibility to arrange and slices the computing work as favorable for parallel processing. The number of Datanodes and data blocks can

be increased for even more parallel processing on need basis. So, this is about an enforcement of unlimited number of computer resources accompanied with local storage where latencies of accessing the data is reduced thanks to local copies of data.

### **1.3.2 Features of HDFS**

#### **Rack awareness**

HDFS divides the data into multiple blocks and stores them on different data nodes. It will be rack aware by the use of a script which allows the master node to map the network topology of the cluster. Within the HDFS the default implementation allows us to provide a script that returns the “rack address” of each of a list of IP addresses.

#### **Reliable storage**

HDFS provides scalable, fault-tolerant storage at a minimal cost. The HDFS software detects and compensates for hardware issues, including disk problems and server failure. HDFS stores file across the group of servers in a cluster. Files are divided into the blocks and each block is written to more than one of the servers. The replication provides both fault-tolerance and performance.

#### **High throughput**

HDFS guarantees data availability by continually monitoring the servers in a cluster and the blocks include checksums. When a block is read, the checksum is verified, it will be restored from one of its replicas when the block is damaged. If a server or disk fails, all of the data is stored is replicated to some other nodes or nodes in the cluster from the collection of replicas.

## 2. BIG DATA TYPES

The big data tour begin with data management, reporting, BI, forecasting, fast data and Big Data. Big Data became very popular term that describes any voluminous amount of structured, semi structured and unstructured data that has the potential to be mined for information. It is well known by Seven Vs: Volume, Velocity, Variety, Veracity, Value, Variability and Visualization. As per (Prathap K, blog 7'v) the below will say everything about Seven Vs.

- **Volume:** When compared, the amount of data being created is huge than traditional data sources
- **Variety:** data can be from different sources and is being created by both machines and humans.
- **Velocity:** Since data is being generated by machines, the process will never stop even while we asleep; and it is being generated extremely fast.
- **Veracity:** This describes about quality of the data. The data is sourced from various places for which a need arises to test the veracity/quality of the data.
- **Variability:** if the meaning of data is constantly changing, it can have a huge impact on your data. Example: A coffee shop may offer 6 different blends of coffee, but if you get the same blend every day and it tastes different every day, that is variability.

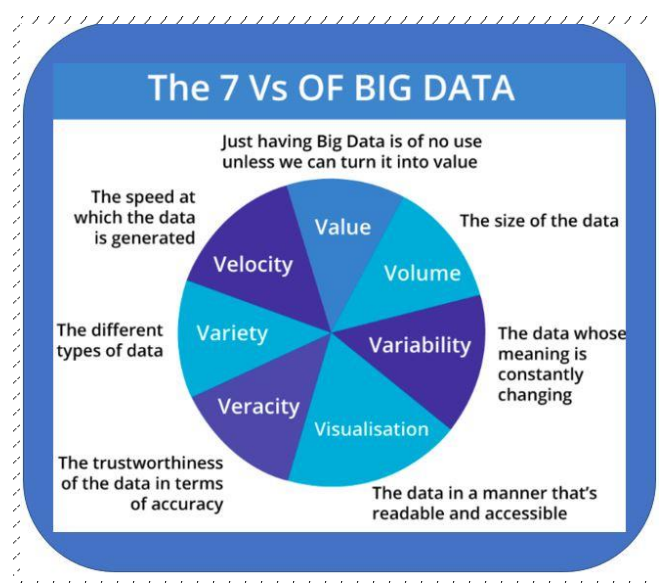


FIGURE 11: 7 V's of big data (www.prathap kudupublog.com)

- **Visualization:** Visualizing large amounts of complex data using charts and graphs is much more effective in conveying meaning than spreadsheets and reports.
- **Value:** After analyzing the huge amounts of data, the organization should get a value from the data.

Big Data has been classified as per their virtue and the below are the all available types of Big data:

## **2.1 Big Data as per data source**

It is estimated that a person generates on an average of 200 GBs of data and the volume is doubling every seven months. As per study, by 2025, we will have 40 EB of human genomic data, or about 400 times more than the 100 petabytes now stored in YouTube (Michael Kanellos 2016, blog [www.forbes.com]).

### **2.1.1 Fast Data**

Fast Data sets are still huge, but the value gyrates around being able to deliver a better answer now: a merely accurate traffic forecast in near real-time is better than a perfect analysis an hour from now. Cameras to detect telltale signals of intoxication have been installed by the west Japan railway system to keep people from falling onto the tracks. Such systems have been adopted by IBM and Cisco and are building their companies according to it. Such Big Data applications have been used by the most of the companies. Backing up systems, monitoring software, historical-to-current patterns, and safety checks will all be needed along with speed.

### **2.1.2 Dark Data**

Video streams, photographs, hand written comment cards, and ingress-egress data from security kiosks: Dark Data is information you have, but can't easily access. There are 245 million surveillance cameras worldwide. Out of which 20% are networked and only 2% are HD capable.

### **2.1.3 Lost Data**

This data also called an Operational Data. This is the data which we find inside of commercial buildings and industrial plants. To name a few, data of manufacturing equipment, chemical boilers, industrial machinery, etc.

It's not technically lost. The problem is that is it often blocked in operational systems. For an instance, McKinsey & Co is the company that estimated that an offshore oil rig might have 30,000 sensors, but it only uses 1% of that data for decision making.

#### **2.1.4 New Data**

New Data contains the information we could get, and want to get, but likely aren't mining now. As per a web source, an approximate of 8.6 trillion gallons gets lost through leaks in pipes worldwide annually which is enough to fill the Hoover Dam. Israel's TaKaDu is taking the initiative step in solving the problem with a complex algorithm that can identify the source of leaks. Israel is leading in Innovations and leader in cyber security.

### **2.2 Big Data as per structures**

Big data is sourced from many different places, and as a result you need to test the veracity or quality of the data. Data can be structured, un-structured and semi-structured.

#### **2.2.1 Structured data**

Structured Data is the data which is stored in databases, in the form of rows and columns and or in an ordered manner. But, as per the research it is identified that it could be around 20% of the total existing data, and is used the most in programming and computer-related activities. This data is also known as RDBMS like Oracle, SQL Server, DB2 and Sybase.

Even the structured data will also be generated by machines and humans. All the data that generates by machines such as data received from sensors, web logs and financial systems is classified as machine-generated data. These include medical devices, GPS data and data of usage statistics captured by servers and applications and the huge amount of data that usually move through trading platforms. Structured data that

generated by humans i.e., all the data a human input into a computer, such as his name and other personal details. As most of nations keep record of their citizens into database like in India is called AADHAR.

### **2.2.2 Unstructured data**

Unlike structured data which resides in the traditional row-column databases, the unstructured data will have no clear format in storage. As per the research analysis done by many researchers the unstructured data will be around 80%, and such unstructured data will be stored and analyzed manually.

Similar to structured, unstructured data can also be classified into machine generated and human generated depending on its source. Machine-generated consists of all the satellite images, the scientific data from various experiments and radar data captured by various technologies.

Unstructured data that is generated by humans is found in abundance across the internet, which includes social media data, mobile data and website content. The pictures we upload to Social media sites such as Facebook, Twitter, Instagram, etc., the videos we upload or watch in YouTube and even the text messages we send all contribute using various chat applets such as WhatsApp, Facebook Messenger, etc. to the gigantic heap that is unstructured data.

### **2.2.3 Semi-structured data**

Unstructured data and semi-structured data have always been unclear since both are divided by a blurred line. This is because the semi-structured data looks like unstructured. Semi-structured data is the information which is not in structured format or not in traditional rows and columns; but contains some operational or organizational properties which make it easier to process. As an example, Logs that being logged as XML formats and NoSQL documents are considered to be semi-structured, since they contain keywords that can be used to process the document easily.



Big Data analysis has a definite business value, as its analysis and processing can help a company achieve cost reductions and growth. So it is essential that you do not wait too long to utilize the potential of this excellent business opportunity.

#### **2.2.4 Big Data as per storage**

The significant requirements of big data storage are that it handles very large amounts of data and keep scaling to keep up with increase/growth, and that it can provide the IOPS (input output operations per second) necessary to deliver data to the tools that helps in analyzing it. The largest web-based operations till date are hyper scale computing environments. But, it is highly feasible that such compute and storage architectures will bleed down into more mainstream enterprises in the coming years.

Scale-out or clustered NAS: This type of storage was a distinct product category, with specialized suppliers such as Isilon and BlueArc. But a measure of the increasing importance of such systems is that both of these have been bought relatively recently by big storage suppliers – EMC and HDS, respectively. It has gone mainstream, and the big change here was with NetApp incorporating true clustering.

Object storage is the other storage format that is built for huge numbers of files. This undertakes the same challenge as scale-out NAS that traditional tree-like file systems become unwieldy when they contain large numbers of files. In Object-based storage each file will have a unique identifier and indexing the data and its location. It's almost same like the DNS way of doing things on the internet than the kind of file system we're used to. This type of storage can scale to very high capacity and large numbers of files in the billions, so are another option for enterprises that want to take advantage of big data. Needless to say, object storage is a less mature technology than scale out network attached storage.

#### **2.2.5 Big Data as a Service (BDaaS)**

There are 4 possible combinations for BDaaS:

- a) PaaS only – focusing on Hadoop.

- b) IaaS and PaaS – focusing on Hadoop and optimized infrastructure for performance.
- c) PaaS and SaaS – focusing on Hadoop and features for productivity and exchangeable infrastructure.
- d) IaaS and PaaS and SaaS – focusing on complete vertical integration for features and performance.

The below diagram will explain all above DBaaS:

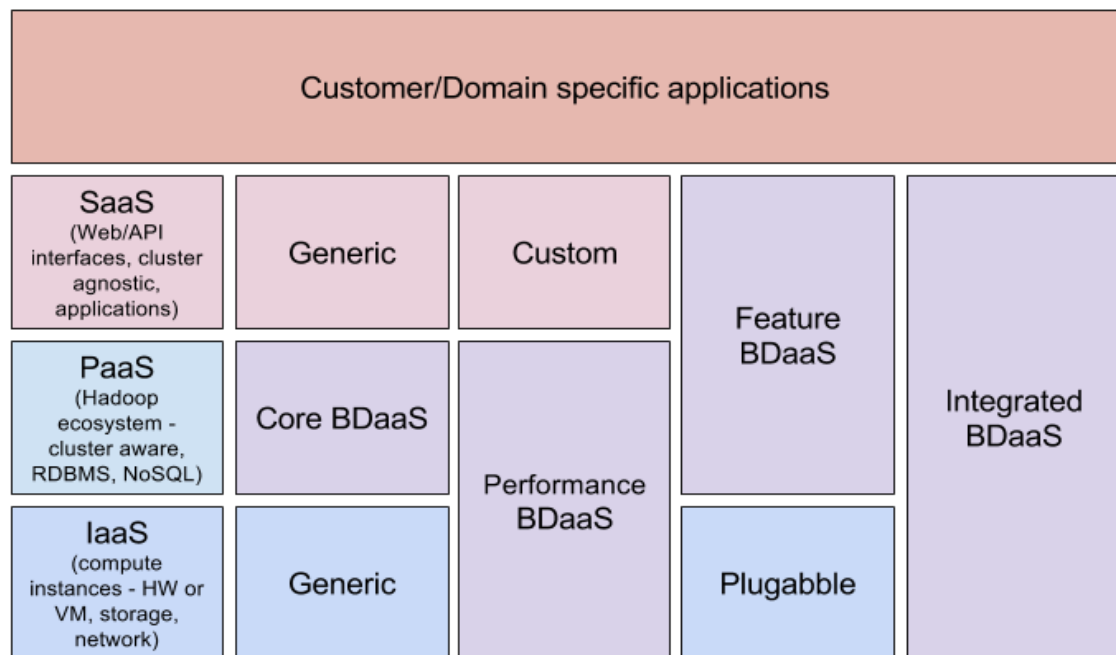


FIGURE 12. BDaaS offerings in the cloud into one of 4 types (www.semantikoz.com)

DBaaS tools provide alike functionality to that of relational database management Systems (RDBMS) like Oracle, SQL Server, Sybase, etc. but store data in the cloud and provide Access without launching virtual machines. There are many tools available for IT infrastructure like IaaS, Paas, MBaaS and DBaaS.

Available famous DBaaS tools are

- Amazon RDS
- Azure DocumentDB
- Oracle

### 3. TYPES OF SECURITY

The following is the way to secure Hadoop infrastructure. We need to apply all below level to secure servers and Data resides on. The target is to avoid any security threats, ensure to implement all below processes. The first three layers for security are called “AAA’s” for security and data protection. AAA is the concept of Identity (Ben Spivey 2015, 22).

<b>Access</b>	Physical (lock & Key) and virtual(Firewalls and VLANS)
<b>Authentication</b>	logins
<b>Authorization</b>	Only permitted users can access
<b>Encryption on rest</b>	Data protection for files on storage
<b>Encryption on networks</b>	Data protection on network
<b>Auditing</b>	Keep track for access and data change
<b>Logging</b>	A security log is used to track security-related information on a computer system
<b>Policy Documentation</b>	Protect against human errors using proper security policy

#### 3.1 Approach to secure Hadoop echo system

- Network based isolation of clustered servers
- HDFS file ownership
- Kerberos
- Combination of Kerberos and HDFS ACL enabled lockdown
- Hbase and accumulo security
- Encryptions
- TLS between server and clients

#### 3.2 Hadoop Security Projects

There are six major Hadoop security projects are currently available: Apache Knox Gateway, Apache Sentry, Apache Argus, Apache Accumulo, Project Rhino and Apache Eagle(Apache.org).

### 3.2.1 Apache Knox Gateway

Apache Knox Gateway intended to improve Hadoop's security from the outside. Apache Knox Gateway provides a REST API gateway for interacting with Hadoop clusters to create a security perimeter between Hadoop and the rest of the world.

Knox Gateway controls and moderates the communication with Hadoop. Knox includes the following support and features:

- LDAP and Active Directory integration,
- Identity federation based on HTTP headers,
- Service-level authorization
- Auditing.

Knox behaves like an exceptional security resolution for the enterprise. By integrating identity management frameworks (IMF) it hides Hadoop hosts and ports behind it. This also streamlines Hadoop access: Instead of connecting to different Hadoop clusters, which all have different security policies, Knox becomes the single entry point for all the Hadoop clusters in the organization. This can be run on server or clusters.

Knox delivers three groups of services:

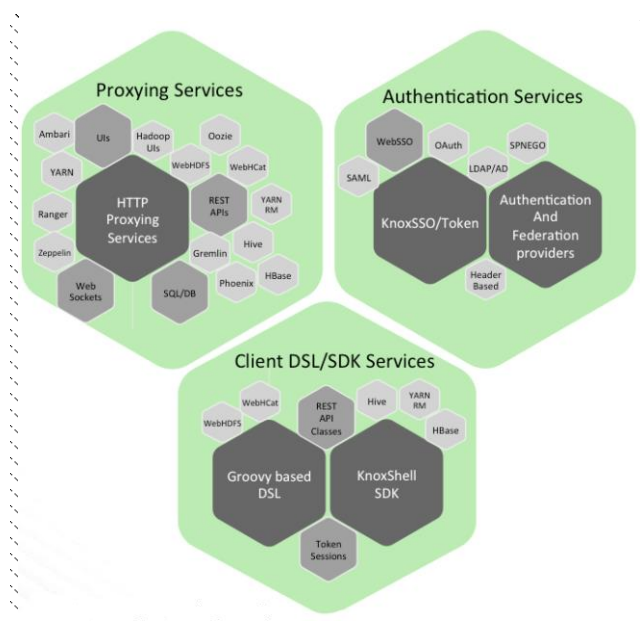


FIGURE 13. Apache Knox gateway for hadoop ecosystem (knox.apache.org)

- **Proxying Services**  
providing access to Apache Hadoop via proxying of HTTP resources is the primary goal of the Apache Knox project.
- **Authentication Services**  
Authentication for WebSSO flow for UIs and REST API access. All available options are LDAP/AD, Header based PreAuth, Kerberos, SAML and OAuth.
- **Client Services**  
Client development is possible with the scripting through DSL or using the Knox Shell classes directly as Software Development Kit.

### **Supported Apache Hadoop Services**

The below mentioned Apache Hadoop ecosystem services have integrations with the Knox Gateway:

Ambari  
WebHDFS (HDFS)  
Yarn RM  
Stargate (Apache HBase)  
Apache Oozie  
Apache Hive/JDBC  
Apache Hive WebHCat (Templeton)  
Apache Storm  
Apache Tinkerpop - Gremlin  
Apache Avatica/Phoenix  
Apache SOLR  
Apache Livy (Spark REST Service)  
Kafka REST Proxy

### **Supported Apache Hadoop ecosystem UIs**

Name Node UI  
Job History UI  
Yarn UI  
Apache Oozie UI  
Apache HBase UI

Apache Spark UI  
Apache Ambari UI  
Apache Ranger Admin Console  
Apache Zeppelin  
Apache NiFi

### **Configuring Support for new services and UIs**

Apache Knox offers a configuration driven method of adding new routing services. This helps for new Apache Hadoop REST APIs to be configured very quickly and easily. Users and developers will be enabled to add support for custom REST APIs to the Knox gateway as well. This capability has been added in the release 0.6.0 and furthers the Knox commitment to extensibility and integration.

### **Authentication**

Collecting credentials is the responsibility of the Providers with the role of authentication presented by the API consumer, validate them and communicate the successful or failed authentication to the client or the rest of the provider chain.

Apart from it, the Knox Gateway provides the Shiro authentication provider. This is a provider that forces the Apache Shiro project for authenticating BASIC credentials against an LDAP user store. OpenLDAP, ApacheDS and Microsoft Active Directory will be supported.

### **Federation/SSO**

For users or customers that require logins to be presented to a limited set of trusted entities within the enterprise, the Knox Gateway may be configured to federate the authenticated identity from an external authentication event.

With the role of federation this will be done through providers. The set of out-of-the-box federation providers include:

***KnoxSSO Default Form-based IDP -***

The usual configuration of KnoxSSO provides a form-based authentication mechanism that enforces the Shiro authentication to authenticate against LDAP/AD with logins or credentials collected from a form-based challenge.

### ***Pac4J -***

Numerous authentication and federation capabilities will be added by the pac4j provider which includes: SAML, CAS, OpenID Connect, Google, Twitter, etc.

### ***HeaderPreAuth-***

A simple process for propagating the identity through HTTP Headers which will specify the username and group for the authenticated user; this has been built with vendor use cases like SiteMinder & IBM Tivoli Access Manager.

### **KnoxSSO**

The KnoxSSO is an integration service that provides a normalized SSO token for representing the user's authentication.

This token is usually used for WebSSO capabilities for involving user interfaces and their consumption of the Apache Hadoop REST APIs.

KnoxSSO summarizes the actual identity provider integration away from involving applications as they only need to be aware of the KnoxSSO cookie. The token is presented by the web browsers such as chrome, IE, Edge, Firefox, etc. as a cookie and applications that are participating in the KnoxSSO integration are able to cryptographically validate the presented token and remain agnostic to the underlying SSO integration.

### **Authorization**

Access decisions made by the providers with the authorization role for the requested resources based on the effective user identity context. This identity context is firm by the authentication provider & the identity assertion provider mapping rules. Evaluation of the identity context's user and group principals against a set of access policies is done

by the authorization provider in order to decide whether access should be granted to the effective user for the resource requested.

In addition to it, the Knox Gateway provides an access control list (ACL) based authorization provider that assesses the rules that comprise of username, groups and IP addresses. These access control lists are bound to and protect resources at the service level. i.e., they protect access to the Apache Hadoop services themselves based on user, group and remote internet protocol address.

### **Audit**

The ability to determine what actions have been taken by whom during or later on some period of time is provided by the auditing capabilities of the Knox Gateway.

The facility has been built on an extension of the Log4j framework and it might be extended by replacing the out of the box implementation with another.

### **3.2.2 Apache Sentry**

Apache Sentry is developed by Cloudera which is a security layer for Hadoop applications such as Hive, Impala, and Solr. It allows administrators to grant or revoke access to servers, databases, and tables and not just in file system (HDFS) level. Lower granularity for columns or cells isn't supported at the moment.

Sentry also allows us to set different privileges for DML Operations such as SELECT, INSERT and TRANSFORM statements and for DDL Operations such as creating and modifying schemas. It even makes multi-tenant administration available for Hadoop, so different policies can be created and maintained by separate administrators for databases and schemas.

Although the project is still under development, it promises to work right out of the box with Apache Hive & Cloudera Impala. This project is a part of the Project Rhino initiative but still deserves focus in its own right.



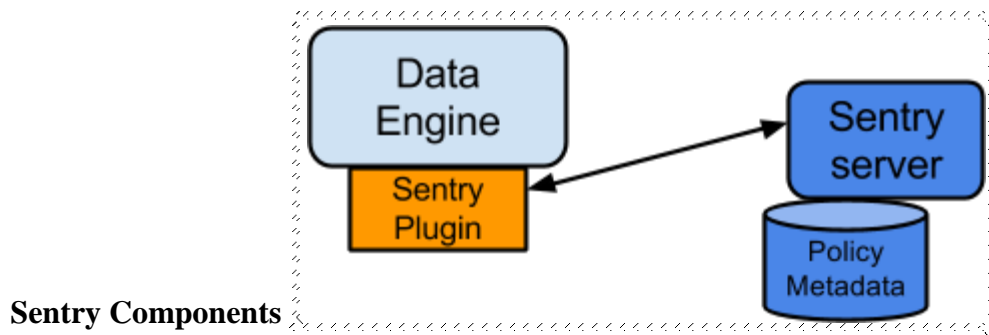


FIGURE 14. Apache sentry data and server (Cloudera enterprise 5.6, documentation)

There are components involved in the authorization process:

- **Sentry Server:** Authorization metadata will be managed the Sentry RPC server. Interface to securely retrieve and manipulate the metadata will be supported;
- **Data Engine:** A data processing application like Hive or Impala that needs to authorize access to data or metadata resources. The data engine loads the Sentry plug-in and all requests from client for accessing resources are intercepted and routed to the Sentry plug-in for parsing;
- **Sentry Plugin:** The Sentry plug-in will run in the data engine. It offers interfaces to manipulate authorization metadata stored in the Sentry server and includes the authorization policy engine that validates access requests using the authorization metadata retrieved from the server.

Key Concepts:

- **Authentication** - Verifying credentials to reliably identify a user
- **Authorization** - Limiting the user's access to a given resource
- **User** - Individual identified by underlying authentication system
- **Group** - A set of users, maintained by the authentication system
- **Privilege** - An instruction or rule that allows access to an object
- **Role** - A set of privileges; a template to combine multiple access rules
- **Authorization models** - Defines the objects to be subject to authorization rules and the granularity of actions allowed. For example, in the SQL model, the objects can be databases or tables, and the actions are SELECT, INSERT,

CREATE and so on. For the Search model, the objects are indexes, collections and documents; the access modes are query, update and so on.

### 3.2.3 Apache Ranger

The Hadoop security project “Ranger” supposed to be named in tribute to Chuck Norris in his "Walker, Texas Ranger" role. The project “Ranger” has its roots in XA Secure, which was later acquired by Hortonworks, then renamed to Argus before settling in at the Apache Software Foundation as Ranger.

Since Apache Ranger which was formerly known as Apache Argus overlaps with Apache Sentry also deals with authorization and permissions. An authorization layer has been added to Hive, HBase, and Knox, and they claim that it has an advantage over Sentry since it includes column level permissions in Hive.

Every part of Hadoop has its own LDAP and Kerberos authentication as well as its own means and rules of authorization (and in most cases totally different implementations of the same). This means you will configure Kerberos or LDAP to each individual part then define those rules in each and every configuration. What Apache Ranger does is to provide a plug-in to each of these parts of Hadoop and a common authentication repository. Also in a centralized location it will allow you to define policies.

Apache Ranger has the following goals:

- Centralized security administration to manage all security related tasks in a central user interfaces or using REST application program interfaces.
- Fine grained authorization to do a particular action and/or operation with Hadoop component/tool and managed through a central administration component/tool.
- Standardize authorization method in all Hadoop components.
- Enhanced support for various authorization methods - Role based access control, attribute based access control etc.
- Centralize auditing of user authorizations and administrative actions within all the components of Hadoop.

### **3.2.4 Apache Accumulo**

Apache Accumulo is not a completely Hadoop security project but it's a distributed key/value store that is based on Google's Big Table and built on top of Apache Hadoop, ZooKeeper and Thrift. This includes authorization at the cell level. Therefore, highly specified access to the data can be granted or restricted at the highest resolution possible: per user and per key/value cell. This product was originally created and contributed to Apache by the NSA. The main features of this product are server side programming, design to scale HDFS instances and cell based access control.

#### **Major Features**

##### ***Server-side programming***

Accumulo has a programming mechanism which is called Iterators that can modify key - value pairs at different points in the data management process.

##### ***Cell-based access control***

Every Accumulo key and value pair has its own security label that limits query results based off user authorizations.

##### ***Designed to scale***

Accumulo runs on a cluster using one or more HDFS instances. Depending on the data stored in Accumulo changes, nodes can be added or removed.

##### ***Stable***

Accumulo has a stable client application programming interface that follows semantic versioning. Each Accumulo release will go through extensive testing.

### **3.2.5 Apache Rhino**

Rhino is an initiative project to bring Hadoop's security up to par by contributing code directly to the relevant Apache security projects.

This initiative led by Intel, views Hadoop as a full stack solution that also includes projects such as ZooKeeper, Hive, Oozie, etc., and wishes to improve security for all.

Project Rhino's goals are to add support for encryption and key management, create a common authorization framework for Hadoop, add single sign-on and token-based authentication, extend HBase support for access control lists up to the cell level, and improve audit logging. Basically, it targets to add all the security features that are missing in Hadoop.

Extending HBase support for access control lists down to the cell level has already been implemented.

### 3.2.6 Apache Eagle

Eagle is a Top Level Project (TLP) of Apache Software Foundation (ASF). Apache Eagle (Eagle is a name of bird which is following data) is an open source analytics solution for identifying security and performance issues instantly on big data platforms. It analyzes data activities, yarn applications, JMX metrics, and daemon logs like system logs, audit logs etc. This tool provides state-of-the-art alert engine to identify security breach, performance issues. Big data platform usually generates larger amount of operational logs and metrics in real time.

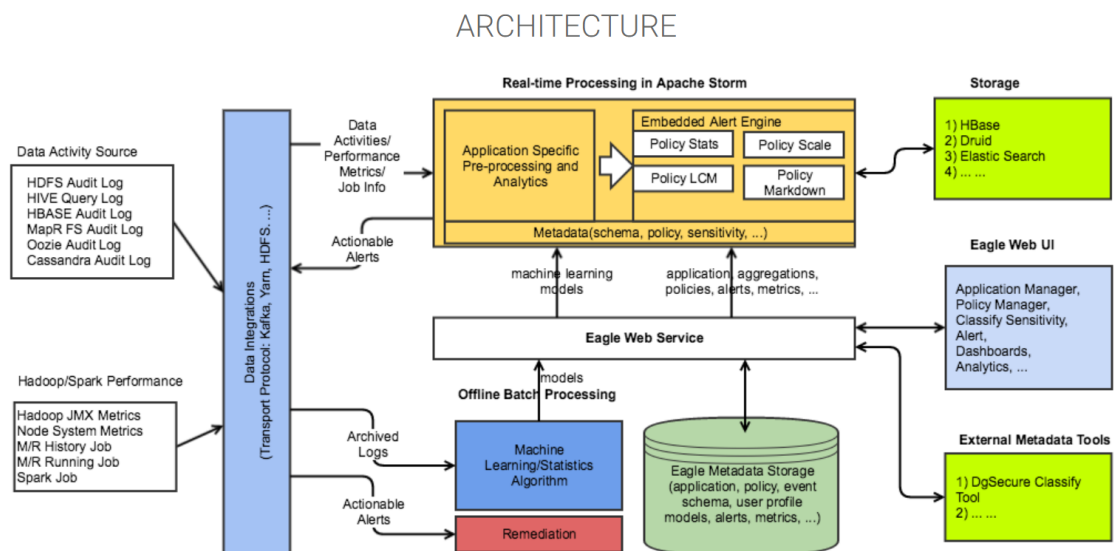


FIGURE 15. Apache eagle hadoop security architecture (<http://eagle.apache.org>)

### **3.3 Information Security Compliance Regulations**

Information security looks to be administered by an ever-changing excess of laws, policies and regulations; each somewhat relevant and apparently originating in a different jurisdiction. Compliance and regulation are different with an organization to organization which depending on their particular activities, their customers and their physical locations.

## 4. SERVER LEVEL SECURITY

Most of Hadoop environment are using different flavor of Linux Operating system like Ubuntu, CentOS, Redhat, Debian and Fedora. First point of security begins with server security using their operating system.

### 4.1 Best practices to harden Linux Server

Securing server is a huge responsibility of System Administrators to protect data, property and time from crackers or scammers. The below are required steps to keep most secure server:

- a) Encrypt transmitted data with password or using keys / certificates.
- b) We should never use these services FTP, Telnet, Rsh and Rlogin to lessen chances to capture files from packet sniffer. For remote login and remote file transfer, the SSH protocol is recommended. We have to secure SSH.
- c) We should only installed required software to reduce vulnerability. Use the RPM package manager such as yum/apt-get/dpkg to review all installed software packages on a system.
- d) We should use only one network service per system.
- e) We have to keep our system up to date by applying latest service patch as soon as are available.
- f) We should always use O/S security extensions like SELinux.
- g) User accounts management is very important to manage security of system. System Admins should implement strong password policy, password ageing, no use of old password, locking logins after unsuccessful login tries, non- root users ID should not set to 0 and root user should be disable.
- h) We must protect Linux servers physical console access, disable the booting from external devices such as DVDs / CDs / USB pen and all persons must pass some sort of security checks before accessing your server.
- i) We should keep record of open network ports.
- j) X Windows on server is not required.
- k) IPTABLES is a user space application program that allows you to configure the firewall. We can also use the TCPWrappers a host-based networking ACL

system to filter network access to Internet. You can prevent many denial of service attacks with the help of Iptables and TCPWrappers.

- l)** Separation of the operating system files (/usr, /home, /tmp etc.) from user files may result into a better and secure system. Create separate partitions for Apache and FTP server roots which have noexec, nodev and nosuid configuration options.
- m)** Disable unwanted SUID and SGID Binaries.
- n)** Files not owned by any user or group can pose a security problem so we have to scrutinize each file and either assign it to a correct user and group or delete it.
- o)** Privilege separation is one of the fundamental security paradigms implemented in Linux and Unix-like operating systems. A special user, called root, has super-user privileges. We should avoid running command as root user so we can grant a normal user SUDO privilege than normal user behave like super-user.
- p)** We should use centralize authentication service, this service keep control over O/S account and authentication validity of users. OpenLDAP server keep sync data for clients and servers.
- q)** We should configure logging and auditing to collect all failed logins and cracking attempts. By default syslog stores data in /var/log/ directory. This helps to identify software misconfiguration which might allow the system to various attacks. I will setup an ElasticSearch to analyze all these logs. We can use audited for system auditing. Writing audit records to the disk is its responsibility. The rules that exist in [/etc/audit. Rules] will be read by this daemon during the startup.
- r)** We should install IDS (Intrusion Detection System) as this system will detect malicious activity such as denial of service attacks (DoS attack), port scans or even attempts to crack into computers by monitoring network traffic. Now-a-days cyber-attacks are so common so this IDS plays an important role to avoid such attacks.
- s)** We should install fail2ban which scans the log files for too many failed login attempts and blocks the IP address which is showing malicious signs. FAIL2BAN is also called DENYHOST. This is so easy to install and configure.

```
# sudo apt-get install fail2ban
```

- t) We should secure Apache/PHP/Nginx server using [httpd.conf] file.
- u) Last but not least a proper system backup so whenever needed to restore should be available.

## 4.2 Server Logging

Ubuntu Linux O/S has a great feature of logging by which we can trace anything happened in the server. System logs are be one of the first resources to trouble-shoot system and application issues. The logs are so informative to find out hint or way to get solution of a problem. Ubuntu system will provide vital information from various system log files. These log files are generally plain American Standard Code for Information Interchange (ASCII) text in a standard log file format, and most of them sit within the system log subdirectory (/var/log). System log daemon (syslogd) will generate most of the logs on behalf of the system and certain programs/applications.

The below mentioned are the different log files that located under /var/log/ directory.

1. **/messages** – Contains global system messages, including the messages that are logged during system startup. Several things get logged in /var/log/messages including mail, cron, daemon, kern, auth, etc.
2. **/dmesg** – Contains kernel ring buffer information. During the system start up, it shows the number of messages on the screen that returns information about the hardware devices that the kernel detects during boot process. Kernel ring buffer will have these messages and it will overwrite the old message with the new one. *dmesg* command will help to view the content of this file.
3. **/auth.log** – Contains system authorization information, including user logins and authentication mechanism that were used.
4. **/boot.log** – Contains information that are logged when the system boots.
5. **/var/log/daemon.log** – will have the information logged by the various background daemons that executes or runs on the system.
6. **/dpkg.log** – Contains information that are logged when a package is installed or removed using *dpkg* command.
7. **/kern.log** – Contains information logged by the kernel. Helpful for you to troubleshoot a custom-built kernel.



8. **/lastlog** – Displays the recent login information for all the users. This is not an ASCII file. You should use lastlog command to view the content of this file.
9. **/mail.log** – Contains the log information from the mail server that is running on the system. For example, sendmail logs information about all the sent items to this file.
10. **/user.log** – Contains information about all user level logs.
11. **/Xorg.x.log** – Log messages from the X.
12. **/alternatives.log** – Information by the update-alternatives are logged into this log file. On Ubuntu, update-alternatives maintains symbolic links determining default commands.
13. **/btmp** – This file contains information about failed login attempts. Use the last command to view the btmp file. For example, “last -f /var/log/btmp | more”.
14. **/cups** – All printer and printing related log messages.
15. **/anaconda.log** – When you install Linux, all installation related messages are stored in this log file.
16. **/yum.log** – Contains information that are logged when a package is installed using yum.
17. **/cron** – Whenever cron daemon (anacron) starts a cron job, it logs the information about the cron job in this file.
18. **/secure** – Contains information related to authentication and authorization privileges. For example, sshd logs all the messages here, including unsuccessful login.
19. **/wtmp & utmp** – Contains login records. Using wtmp you can find out who is logged into the system. who command uses this file to display the information?
20. **/faillog** – Contains user failed login attempts. Use faillog command to display the content of this file.

Apart from the above log files, /var/log directory may also contain the following sub-directories depending on the application that is running on your system.

- **/apache2** – Contains the apache web server access\_log and error\_log.
- **/lighttpd/** – Contains light HTTPD access\_log and error\_log.
- **/conman/** – Log files for ConMan client. conman connects remote consoles that are managed by conmand daemon.

- **/mail/** – This subdirectory contains additional logs from your mail server. For example, sendmail stores the collected mail statistics in /var/log/mail/statistics file.
- **/prelink/** – prelink program modifies shared libraries and linked binaries to speed up the startup process. /var/log/prelink/prelink.log contains the information about the .so file that was modified by the prelink.
- **/audit/** – Contains logs information stored by the Linux audit daemon (auditd).
- **/setroubleshoot/** – SELinux uses setroubleshootd (SE Trouble Shoot Daemon) to notify about issues in the security context of files, and logs those information in this log file.
- **/var/log/samba/** – Contains log information stored by samba, which is used to connect Windows to Linux.
- **/var/log/sa/** – Contains the daily sar files that are collected by the sysstat package.
- **/var/log/sss/** – Use by system security services daemon that manage access to remote directories and authentication mechanisms.

There are very important commands are available to monitor logs and permissions on Linux/Unix files are Grep, touch, tail, head, awk, chmod, chgrp, less and last. There are some other utilities as well like rotatelog, savelog and logger.

### **4.3 Auditing Hadoop Servers**

Auditing of server can be done by many ways of which few I will discuss here

1. Finding from where logins are done
2. Finding user who changed / executed a commands
3. Check User actions

### **4.4 Configure Audit Daemon to audit Linux / UNIX server**

The Linux Audit Daemon (auditd) is a framework to allow auditing events on a Linux system. In Linux system a daemon called auditd is responsible for monitoring individual system calls, and logging them for inspection. By proper configuration of this Daemon,

we can keep track every events that happening in the server and we can find out suspicious event and soon we can take appropriate an action to avoid any bad attempt which can harm our server.

We can monitor the following:

- Audit file access and modification
- Monitoring of system calls and functions
- Detect anomalies like crashing processes
- Set tripwires for intrusion detection purposes
- Record commands used by individual users

**Components for Audit:** Linux Server has mainly three main components to perform audit which are Kernel, Binaries (auditd, auditctl, audispd, aureport, ausearch, autrace, aulast, aulastlog, ausyscall and audev) and files (audit. Rules, auditd.conf)

**Installation:** We can install using apt-get install method in Ubuntu but in other flavors of UNIX is already installed.

```
# apt-get install auditd audispd-plugins
```

After successfully installation, we can configure system auditing by two methods. One is to use a command-line utility called auditctl. The other method is to edit the audit configuration file located at /etc/audit/audit.rules.

The below commands to start/restart/status

```
$ sudo service auditd start/restart/status
```

Once auditd starts running, it will start generating an audit daemon log in /var/log/audit/audit.log as auditing is in progress.

```
$ sudo nano /etc/audit/audit.rules
```

```
# First rule - delete all  
-D
```

```
# increase the buffers to survive stress events. make this bigger for busy systems.
-b 1024

# monitor unlink() and rmdir() system calls.
-a exit,always -S unlink -S rmdir

# monitor open() system call by Linux UID 1001.
-a exit,always -S open -F loginuid=1001

# monitor write-access and change in file properties (read/write/execute) of the
following files.
-w /etc/group -p wa
-w /etc/passwd -p wa
-w /etc/shadow -p wa
-w /etc/sudoers -p wa
# monitor read-access of the following directory.
-w /etc/secret_directory -p r
# lock the audit configuration to prevent any modification of this file.
-e 2
```

To check if a specific file (e.g., /etc/passwd) has been accessed by anyone:

```
$ sudo ausearch -f /etc/passwd
```

To check if a specific directory (e.g., /etc/secret\_directory) has been accessed by anyone, run the following:

```
$ sudo ausearch -f /etc/secret_directory
```

**Lynis:** This is a host-based, open-source security auditing application that can evaluate the security profile and posture of Linux and other UNIX-like operating systems. We can even do automation for this auditing using Lynis. Lynis will perform several tests to determine the log file, available rules and more. For proper intrusion detection,

integration with an Intrusion Detection System (IDS) is key in discover events when they occur and take appropriate actions. Lynis won't perform any system hardening automatically but help to find out what to be done to secure our server.

Lynis's software repository uses the HTTPS protocol, so we'll need to make sure that HTTPS support for the package manager is installed. Use the following command to check:

```
$ dpkg -s apt-transport-https | grep -i status
```

If not installed then need to be install using this command

```
$ sudo apt-get install apt-transport-https
```

To begin with installation process, need to repository's key:

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys  
C80E383C3DE9F082E01391A0366C67DE91CA5D5F
```

Add the Lynis repository to the list of those available to the package manager:

```
$ sudo add-apt-repository "deb [arch=amd64]  
https://packages.cisofy.com/community/lynis/deb/ xenial main"
```

```
$ sudo apt-get install lynis
```

## 5. ELASTICSEARCH, LOGSTASH AND KIBANA TRIO FOR SECURITY

### ElasticSearch

This application will help in monitoring server logs and review server security breaches and threats. ElasticSearch is well known for super-fast search result due to text base search instead of index. ElasticSearch uses Apache Lucene to create and manage this inverted index. There are the ElasticSearch nodes that form the cluster, and often Logstash instances, Kibana instances, Beats agents at clients.

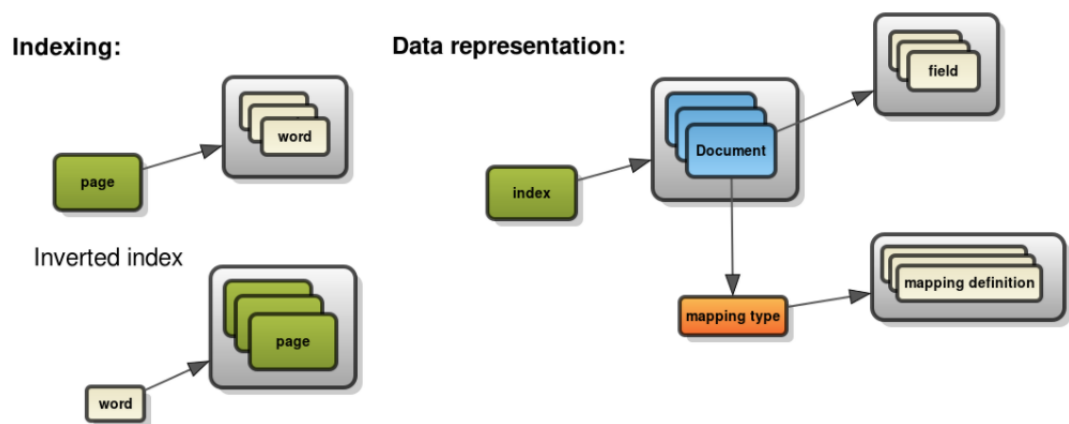


FIGURE 16. Elasticsearch architecture model (<http://www.elasticsearchtutorial.com>)

### Logstash

Logstash is an open source data collection engine which has real-time pipelining capabilities. It can dynamically collect data from various sources. It normalizes the data into desire server. Cleanse and democratize all your data for diverse advanced downstream analytics and visualization use cases.

While Logstash originally drove innovation in log collection, its capabilities extend well beyond that use case. Any type of event can be enriched and transformed with a broad array of input, filter, and output plugins, with many native codecs further simplifying the ingestion process. Logstash accelerates your insights by harnessing a greater volume and variety of data.

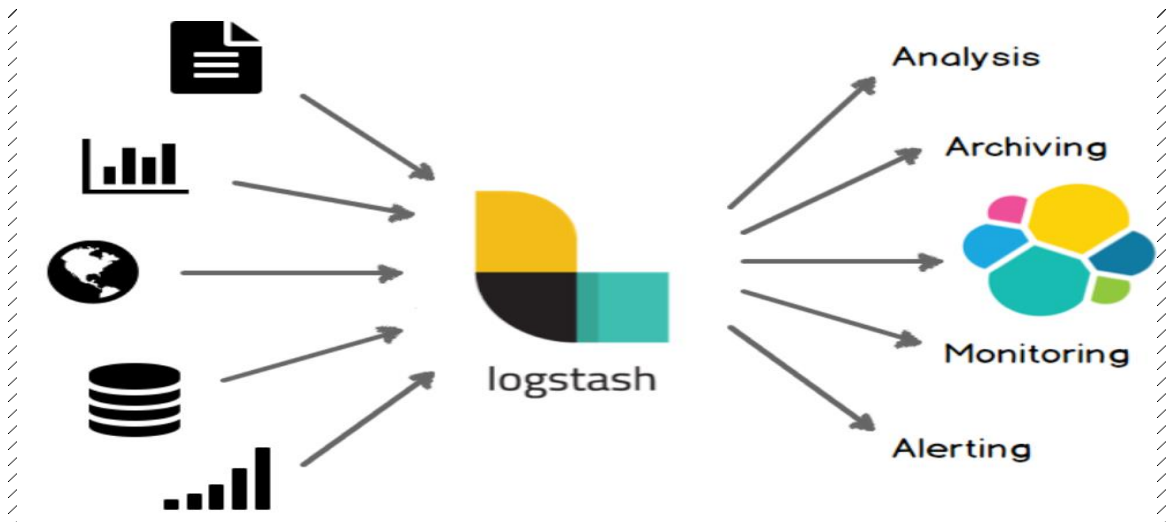


FIGURE 17. Logstash Architecture model ([www.elastic.co/guide/en/logstash](http://www.elastic.co/guide/en/logstash))

## Kibana

Kibana is a frame into the Elasticsearch stack. This enables pictorial exploration and real-time analysis of server logs data. Kibana is used for data exploration, visualization, and dashboard. It gives lot of freedom to visualize data and helpful to secure server by analyzing server logs data.

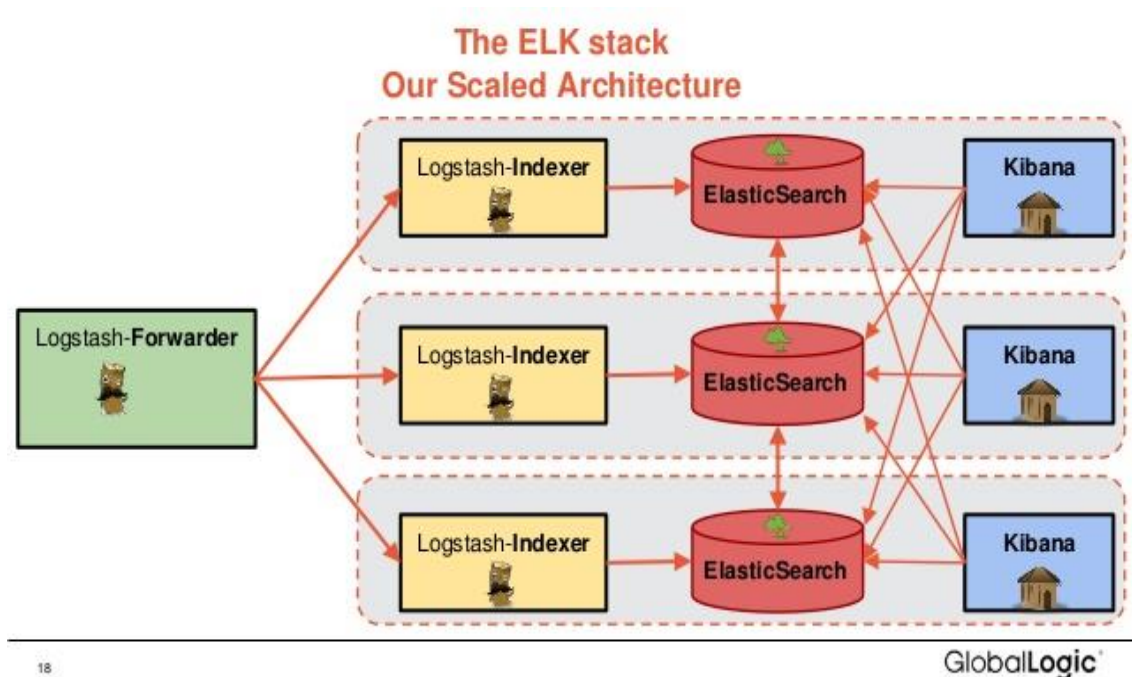


FIGURE 18. Kibana architecture model ([www.slideshare.net/GlobalLogicUkraine](http://www.slideshare.net/GlobalLogicUkraine), p18)

## 5.1 Installation of Elasticsearch, Logstash and Kibana

### 5.1.1 Installation of Elasticsearch

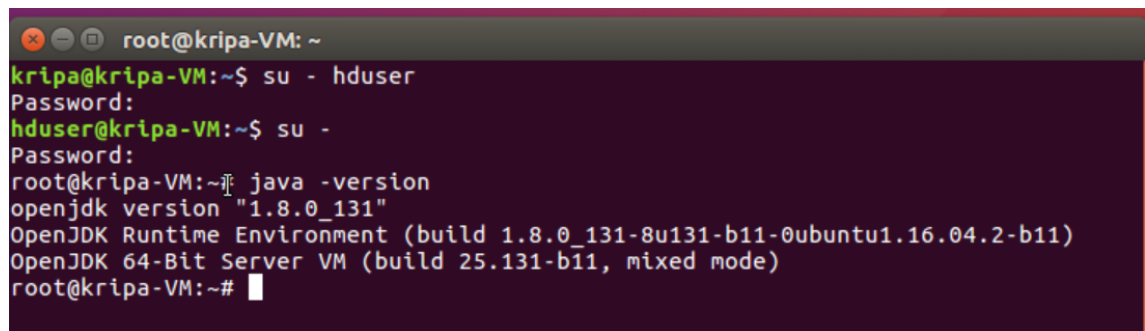
Java is required for the Elasticsearch Installation. Elasticsearch requires Java 8. It is recommended to use the Oracle JDK 1.8.

Install JAVA.

```
$ sudo apt-get install -y oracle-java8-installer
```

After successful java installation, we can verify JAVA version

```
$ java -version
```

A terminal window screenshot showing the process of installing Java 8 and verifying its version. The user starts as root, switches to the 'hduser' account, and then runs 'java -version'. The output shows 'openjdk version "1.8.0\_131"' and 'OpenJDK Runtime Environment (build 1.8.0\_131-8u131-b11-0ubuntu1.16.04.2-b11)'.

```
root@kripa-VM: ~  
kripa@kripa-VM:~$ su - hduser  
Password:  
hduser@kripa-VM:~$ su -  
Password:  
root@kripa-VM:~# java -version  
openjdk version "1.8.0_131"  
OpenJDK Runtime Environment (build 1.8.0_131-8u131-b11-0ubuntu1.16.04.2-b11)  
OpenJDK 64-Bit Server VM (build 25.131-b11, mixed mode)  
root@kripa-VM:~#
```

### Install Elasticsearch

First we need to add the elastic repository key to the server using below command

```
$ sudo wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add
```

—

We also need to install https transport protocol and need to add elasticsearch repository to the 'sources.list.d' directory.

```
$ sudo apt-get install apt-transport-https
```



```
$ echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | sudo tee -a
/etc/apt/sources.list.d/elastic-5.x.list
```

Now need to update and install elasticsearch

```
$ sudo apt-get update
```

```
$ sudo apt-get install -y elasticsearch
```

```
Start/status/stop Elasticsearch service
```

```
$ systemctl daemon-reload
```

```
$ systemctl enable elasticsearch.service
```

```
$ systemctl start elasticsearch.service
```

```
$ systemctl stop elasticsearch.service
```

```
$ systemctl status elasticsearch.service
```

We get below screen and showing Active (running) in green, means elasticsearch started

```
root@kripa-VM: ~
root@kripa-VM:~# systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2017-11-01 07:45:24 GMT; 26min ago
     Docs: http://www.elastic.co
  Process: 1058 ExecStartPre=/usr/share/elasticsearch/bin/elasticsearch-systemd-pre-exec (code=exited, status=0/SUCCESS)
 Main PID: 1065 (java)
    Tasks: 39
   Memory: 2.1G
      CPU: 2min 40.179s
   CGroup: /system.slice/elasticsearch.service
           └─1065 /usr/bin/java -Xms2g -Xmx2g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+AlwaysPre
Nov 01 07:45:23 kripa-VM systemd[1]: Starting Elasticsearch...
Nov 01 07:45:24 kripa-VM systemd[1]: Started Elasticsearch.
Nov 01 07:47:10 kripa-VM elasticsearch[1065]: [2017-11-01T07:47:10,027][WARN ][o.e.c.l.LogConfigurator ] ignoring unsupported logging configuration file [/e
lines 1-15/15 (END)
```

To know the open port for elasticsearch service using below command.

```
$ netstat -plntu
```

Active Internet connections (only servers)

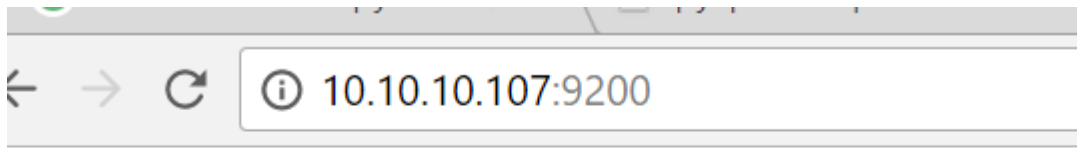
Proto	Q Send	Q Local Address	Foreign Address	State	PID/Program name
Tcp	0	0 127.0.1.1:53	0.0.0.0:*	LISTEN	2631/dnsmasq
Tcp	0	0 0.0.0.0:22	0.0.0.0:*	LISTEN	878/sshd
Tcp	0	0 0.0.0.0:5432	0.0.0.0:*	LISTEN	1031/postgres
Tcp	0	0 0.0.0.0:5601	0.0.0.0:*	LISTEN	734/node
Tcp	0	0 127.0.0.1:3306	0.0.0.0:*	LISTEN	1025/mysqld
Tcp	0	0 127.0.0.1:6379	0.0.0.0:*	LISTEN	965/redis-server 12
tcp6	0	0 :::9300	:::*	LISTEN	1065/java
tcp6	0	0 :::21	:::*	LISTEN	886/vsftpd
tcp6	0	0 :::22	:::*	LISTEN	878/sshd
tcp6	0	0 :::8983	:::*	LISTEN	1789/java
tcp6	0	0 :::5432	:::*	LISTEN	1031/postgres
tcp6	0	0 :::2181	:::*	LISTEN	867/java
tcp6	0	0 :::45669	:::*	LISTEN	867/java
tcp6	0	0 127.0.0.1:7983	:::*	LISTEN	1789/java
tcp6	0	0 :::9200	:::*	LISTEN	1065/java
udp	0	0 0.0.0.0:631	0.0.0.0:*		2983/cups-browsed
udp	0	0 0.0.0.0:5353	0.0.0.0:*		715/avahi-daemon: r
udp	0	0 127.0.1.1:53	0.0.0.0:*		2631/dnsmasq
udp	0	0 0.0.0.0:68	0.0.0.0:*		2609/dhclient
udp	0	0 0.0.0.0:53497	0.0.0.0:*		715/avahi-daemon: r
udp6	0	0 :::33337	:::*		715/avahi-daemon: r
udp6	0	0 :::5353	:::*		715/avahi-daemon: r

root@kripa-VM:~#

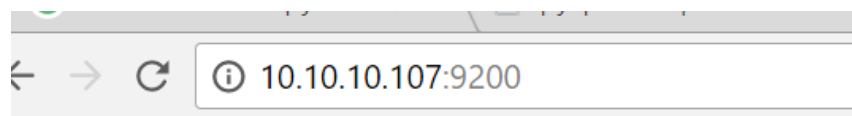
We need to configure Elasticsearch as per our requirement in “elasticsearch.yml” which is available in the location “/etc/elasticsearch” and main parameters like change cluster name, node name and network.host.

We can verify at any internet browser typing

<https://localhost:9200/> or <https://IP Address: 9200>



```
"name" : "Franz Kafka",
"cluster_name" : "elasticsearch",
"cluster_uuid" : "zFulveNVRtCyATQMzEstZA",
"version" : {
  "number" : "5.6.2",
  "build_hash" : "57e20f3",
  "build_date" : "2017-09-23T13:16:45.703Z",
  "build_snapshot" : false,
  "lucene_version" : "6.6.1"
},
"tagline" : "You Know, for Search"
```



```
"name" : "Franz Kafka",
"cluster_name" : "elasticsearch",
"cluster_uuid" : "zFulveNVRtCyATQMzEstZA",
"version" : {
  "number" : "5.6.2",
  "build_hash" : "57e20f3",
  "build_date" : "2017-09-23T13:16:45.703Z",
  "build_snapshot" : false,
  "lucene_version" : "6.6.1"
},
"tagline" : "You Know, for Search"
```

### 5.1.2 Installation of Logstash

Logstash is also a part of ELK Stack. ELK is combination of Elasticsearch, Logstash and Kibana and is a robust open source solution for searching, analyzing and visualizing data. Logstash is a data processing pipeline for managing events and logs

## Requirements for Installation

1. A user with sudo privilege
2. VM with Ubuntu 16.04
3. Oracle –Java8-nstaller

\$ sudo apt install logstash

Set java home in set java- home at below file

cd etc/logstash

nano statup.option

need to start logstash service

\$ sudo systemctl enable logstash

\$ sudo systemctl start logstash

\$ sudo systemctl status logstash

```

root@kripa-VM:/etc/elasticsearc# sudo systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2017-11-01 11:43:00 GMT; 2min 49s ago
     Main PID: 9298 (java)
       Tasks: 14
      Memory: 301.8M
         CPU: 1min 43.798s
    CGroup: /system.slice/logstash.service
            └─9298 /usr/bin/java -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+Disabl

Nov 01 11:43:00 kripa-VM systemd[1]: Started logstash.
lines 1-11/11 (END)

```

### 5.1.3 Installation of Kibana

Well, This open source application also need basic requirement before moving forward to installation process as like other ELK stack application. The basic requirements are An Ubuntu 16.04 VPS, user with sudo level access, Oracle8-java etc, once system has all these requirement then we can install Kibana using below command

\$ sudo apt install kibana

After successful installation of Kibana we need to open the kibana.yml file and restrict the remote access to the Kibana instance like this

```
$ cd /etc/kibana
```

```
$ nano kibana.yml
```

We have change server.host, server.name and elasticsearch.url into above file.

```
$ sudo systemctl daemon-reload
```

```
$ systemctl enable kibana
```

```
$ systemctl start kibana
```

```
$ systemctl status kibana
```

```
root@kripa-VM: /etc/kibana
root@kripa-VM: /etc/kibana# ls -ltr
total 8
-rw-r--r-- 1 root root 4646 Oct 23 10:13 kibana.yml
root@kripa-VM: /etc/kibana# pwd
/etc/kibana
root@kripa-VM: /etc/kibana# ls -ltr
total 8
-rw-r--r-- 1 root root 4646 Oct 23 10:13 kibana.yml
root@kripa-VM: /etc/kibana# nano kibana.yml
root@kripa-VM: /etc/kibana# sudo systemctl daemon-reload
root@kripa-VM: /etc/kibana# systemctl enable kibana
Synchronizing state of kibana.service with SysV init with /lib/systemd/systemd-sysv-install...
Executing /lib/systemd/systemd-sysv-install enable kibana
root@kripa-VM: /etc/kibana# systemctl start kibana
root@kripa-VM: /etc/kibana# systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2017-11-06 07:21:21 GMT; 39min ago
     Main PID: 736 (node)
    CGroup: /system.slice/kibana.service
            └─736 /usr/share/kibana/bin/./node/bin/node --no-warnings /usr/share/kibana/bin/./src/cli -c /etc/kibana/kibana.yml

Nov 06 07:25:46 kripa-VM kibana[736]: {"type":"log","@timestamp":"2017-11-06T07:25:46Z","tags":["warning","elasticsearch","admin"],"pid":736,"message":"No li
Nov 06 07:25:49 kripa-VM kibana[736]: {"type":"log","@timestamp":"2017-11-06T07:25:49Z","tags":["warning","elasticsearch","admin"],"pid":736,"message":"Unabl
Nov 06 07:25:49 kripa-VM kibana[736]: {"type":"log","@timestamp":"2017-11-06T07:25:49Z","tags":["warning","elasticsearch","admin"],"pid":736,"message":"No li
Nov 06 07:25:55 kripa-VM kibana[736]: {"type":"log","@timestamp":"2017-11-06T07:25:55Z","tags":["status","plugin:elasticsearch85.6.3","error"],"pid":736,"sta
Nov 06 07:26:03 kripa-VM kibana[736]: {"type":"log","@timestamp":"2017-11-06T07:26:03Z","tags":["status","plugin:elasticsearch85.6.3","error"],"pid":736,"sta
Nov 06 07:26:13 kripa-VM kibana[736]: {"type":"log","@timestamp":"2017-11-06T07:26:13Z","tags":["warning"],"pid":736,"kibanaVersion":"5.6.3","nodes":{"[["versi
Nov 06 07:26:14 kripa-VM kibana[736]: {"type":"log","@timestamp":"2017-11-06T07:26:14Z","tags":["status","plugin:elasticsearch85.6.3","error"],"pid":736,"sta
Nov 06 07:26:42 kripa-VM kibana[736]: {"type":"log","@timestamp":"2017-11-06T07:26:42Z","tags":["status","plugin:elasticsearch85.6.3","info"],"pid":736,"stat
Nov 06 07:26:42 kripa-VM kibana[736]: {"type":"log","@timestamp":"2017-11-06T07:26:42Z","tags":["status","ui settings","info"],"pid":736,"state":"green","mes
Nov 06 08:01:00 kripa-VM systemd[1]: Started Kibana.
lines 1-17/17 (END)
```

Now we can access Kibana using this link <http://ipaddress:5601>

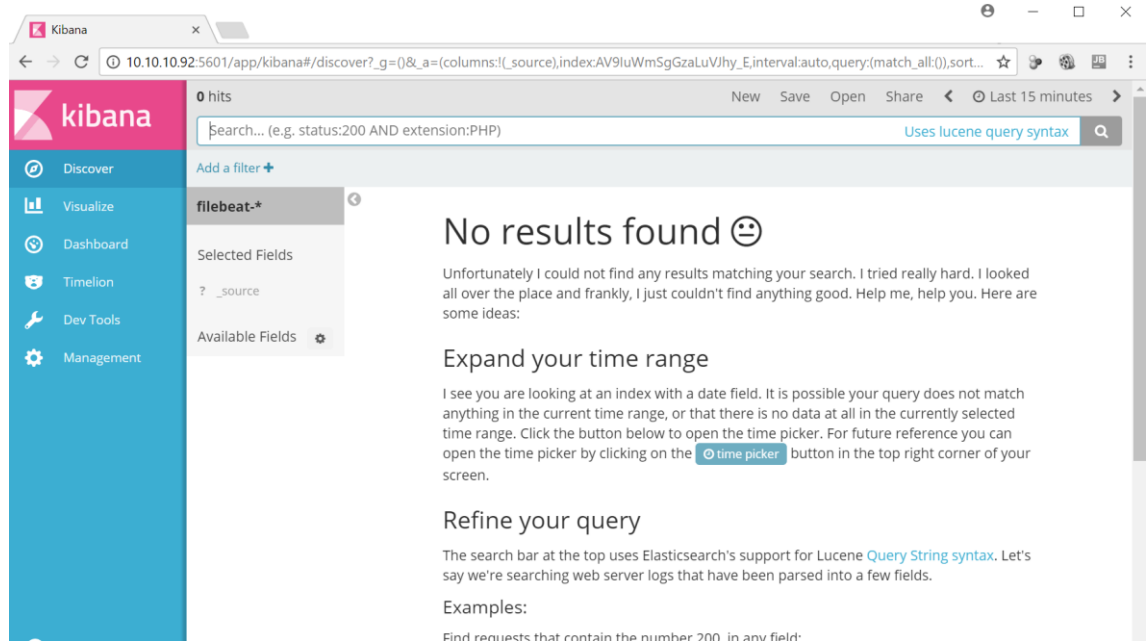


FIGURE 19. Kibana dashboard

## Filebeat

```
$ cd /etc/filebeat
```

```
$ nano filebeat.yml
```

```
filebeat.prospectors:
```

```
- type: log
```

```
  enabled: true
```

```
  paths:
```

```
    - /var/log/*.log
```

The prospector is yields all files in the path `/var/log/*.log`, in a simple way is that Filebeat will yield all files in the directory `/var/log/` that end with `.log`.

```

root@kripa-VM:/etc/filebeat# ls -ltr
total 116
-rw----- 1 root root 4196 Oct  6 20:27 filebeat.yml
-rw-r--r-- 1 root root 20027 Oct  6 20:27 filebeat.template.json
-rw-r--r-- 1 root root 20027 Oct  6 20:27 filebeat.template-es6x.json
-rw-r--r-- 1 root root 25087 Oct  6 20:27 filebeat.template-es2x.json
-rw-r--r-- 1 root root 37819 Oct  6 20:27 filebeat.full.yml
root@kripa-VM:/etc/filebeat# nano filebeat.yml
root@kripa-VM:/etc/filebeat# systemctl status filebeat
● filebeat.service - filebeat
   Loaded: loaded (/lib/systemd/system/filebeat.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: https://www.elastic.co/guide/en/beats/filebeat/current/index.html
root@kripa-VM:/etc/filebeat# systemctl enable filebeat
Synchronizing state of filebeat.service with SysV init with /lib/systemd/systemd-sysv-install...
Executing /lib/systemd/systemd-sysv-install enable filebeat
root@kripa-VM:/etc/filebeat# systemctl status filebeat
● filebeat.service - filebeat
   Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: https://www.elastic.co/guide/en/beats/filebeat/current/index.html
root@kripa-VM:/etc/filebeat# systemctl start filebeat
root@kripa-VM:/etc/filebeat# systemctl status filebeat
● filebeat.service - filebeat
   Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2017-11-07 07:09:00 GMT; 4s ago
     Docs: https://www.elastic.co/guide/en/beats/filebeat/current/index.html
  Main PID: 3439 (filebeat)
    Tasks: 6
   Memory: 13.5M
      CPU: 212ms
  CGroup: /system.slice/filebeat.service
          └─3439 /usr/share/filebeat/bin/filebeat -c /etc/filebeat/filebeat.yml -path.home /usr/share/filebeat -path.config /etc/filebeat -path.data /var/li

Nov 07 07:09:00 kripa-VM systemd[1]: Started filebeat.
lines 1-12/12 (END)

```

## 6. MONITORING HADOOP SERVER WITH NETDATA

### 6.1 Introduction

Netdata is a free, real-time resource monitoring tool with a friendly web front-end. Netdata is an open source, scalable, distributed, real-time, performance and health monitoring tool for Linux Servers. Netdata comes with simple, easy to use and extensible web dashboards that can be used to visualize the processes and services on your system. This tool come with fancy charts which represent utilization of CPUs, Memory, Disk I/O and network bandwidth, Apache, Postfix and many more.

### 6.2 Interface of Netdata

The web front-end is very receptive and needs no Flash plugin. The User Interface doesn't clutter things up with unneeded features, but sticks to what it does. The most commonly needed charts (i.e. CPU, RAM, network, and disk) are at the top. Netdata even allows you to control the chart with play, reset, zoom and resize with the controls on the bottom right of each chart.

There's currently no authentication, so if you're concerned about someone getting information about the applications you're running on your system, you should restrict who has access via a firewall policy. The UI is simplified in a way anyone could look at the graphs and understand.

### 6.3 Monitored Server Parameters

- Total and Per Core CPU usage, interrupts, softirqs and frequency.
- Total Memory, RAM, Swap and Kernel usage.
- Disk I/O (per disk: bandwidth, operations, backlog and utilization).
- Monitors Network interfaces including: bandwidth, packets, errors and drops).
- Monitors Netfilter Linux firewall connections, events, errors.
- Processes (running, blocked, forks and active).
- System Applications with the process tree (CPU, memory, swap, disk reads/writes and threads).
- Apache and Nginx Status monitoring with `mod_status`.



- MySQL database monitoring: queries, updates, locks, issues, threads, etc.
- Postfix email server message queue.
- Squid proxy server bandwidth and requests monitoring.
- Hardware sensors (temperature, voltage, fans, power, humidity, etc).
- SNMP devices.

#### 6.4 How to Setup / Install Netdata

Netdata installation requirements as per ([www.digitalocean.com](http://www.digitalocean.com)):

Before installation system must have the below packages:

Sl.	Package Name	Description
1	Libuuid	part of util-linux for GUIDs management
2	Zlib	gzip compression for the internal Netdata web server
3	libmnl-dev	minimalistic Netlink communication library
4	Gcc	GNU C compiler
5	Make	An utility for Directing compilation
6	Autoconf	automatic configure script build
7	Curl	curl is a command line tool for transferring data with URL
8	Git	Git is popular version control system

Netdata plugins details to monitor system:

Sl.	Package Name	Description
1	Bash	for shell plugins and <b>alarm notifications</b>
2	Iproute(Iproute2)	for monitoring <b>Linux traffic QoS</b>
3	python	for most of the external plugins
4	nodejs	used for monitoring <b>named</b> and <b>SNMP</b> devices
5	lm-sensors	or monitoring <b>hardware sensors</b>
6	netcat	for shell plugins to collect metrics from remote systems
7	python-pymongo	used for monitoring <b>mongo dB</b> databases
8	python-pymysql	python-mysqldb is a lot faster and thus preferred

- A server running Ubuntu 18.04 LTS.
- A non-root user with sudo privileges.

Before running “netdata-installer.sh” Netdata installer, need to check latest update in the server running Ubuntu O/S

```
$ sudo apt-get update
```

After successfully update, need to install required dependent packages and tools

```
$ sudo apt-get install zlib1g-dev uuid-dev libmnl-dev gcc make autoconf autoconf-  
archive autogen automake pkg-config curl
```

I face problem during installation of uuid-dev package due to upper version were already exists so I need to downgrade an installed package to required level with below command and then install the package.

```
$ sudo apt-get install libuuid1=2.20.1-5.1ubuntu20
```

```
$ sudo apt-get install libuuid-dev
```

**These below tools also installed to monitor server performance**

```
$ sudo apt-get install python python-yaml python-mysqldb python-psycopg2 nodejs lm-  
sensors netcat
```

Cloning of the Netdata repository into my home directory (/home/hduser/netdata) by using below command

```
$ git clone https://github.com/firehol/netdata.git --depth=1 ~/netdata
```

Now need to move netdata folder and run installer package to install netdata.

```
$ sudo ./netdata-installer.sh
```



1. Create a group netdata via the Synology group interface. Give it no access to anything.
2. Create a user netdata via the Synology user interface. Give it no access to anything and a random password. Assign the user to the netdata group. Netdata will chuid to this user when running.
3. Change ownership of the following directories:

```
$ chown -R root:netdata /opt/netdata/usr/share/netdata
```

```
$ chown -R netdata:netdata /opt/netdata/var/lib/netdata /opt/netdata/var/cache/netdata
```

```
$ chown -R netdata:root /opt/netdata/var/log/netdata
```

We can also check the status of netdata service with be below command

```
$ systemctl status netdata
```

```
hduser@kripa-VM:~$ systemctl status netdata
● netdata.service - Real time performance monitoring
   Loaded: loaded (/lib/systemd/system/netdata.service; enabled; vendor preset:
   Active: active (running) since Sat 2018-11-24 05:23:17 GMT; 2min 54s ago
     Process: 1241 ExecStartPre=/bin/chown -R netdata:netdata /var/run/netdata (cod
     Process: 1237 ExecStartPre=/bin/mkdir -p /var/run/netdata (code=exited, status
     Process: 1225 ExecStartPre=/bin/chown -R netdata:netdata /var/cache/netdata (c
     Process: 1223 ExecStartPre=/bin/mkdir -p /var/cache/netdata (code=exited, stat
   Main PID: 1245 (netdata)
      Tasks: 18
     Memory: 41.8M
        CPU: 6.393s
   CGroup: /system.slice/netdata.service
           └─1245 /usr/sbin/netdata -P /var/run/netdata/netdata.pid -D -W set gl
             └─1275 bash /usr/libexec/netdata/plugins.d/tc-qos-helper.sh 1
               └─1280 /usr/libexec/netdata/plugins.d/apps.plugin 1
                 └─1288 /usr/bin/python /usr/libexec/netdata/plugins.d/python.d.plugin
hduser@kripa-VM:~$
```

Now we can access Netdata URL from any web browser using `http:// Server_IP:19999/`, which will display every necessary system metrics, even we can enable critical server metrics notifications.

The below is fancy dashboard of Netdata, which give very attractive and eye catchable pictures of every system parameters

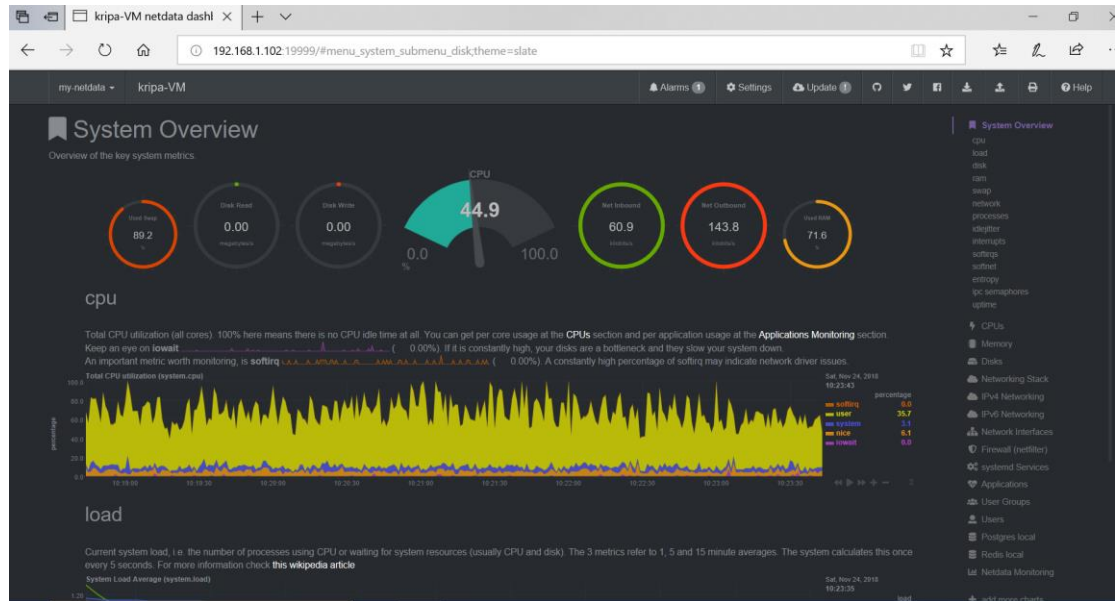


FIGURE 20. Netdata dashboard (from my netdata URL)

The quickest way of navigating around the dashboard is by using the menu tree on the right of the page. This changes focus and color depending on which section of the page you are currently viewing.

Netdata URL can be access from any web browser by using below string

<http://192.168.1.102:19999/>

Where 192.168.1.102 is my virtual IP address and 19999 is open port for netdata. The below picture is displaying all available menu option to monitor server parameters for security and performance. Critical server notification can be also enables to receive email alerts for any system security or performance related which could help to resolve any server related issues.

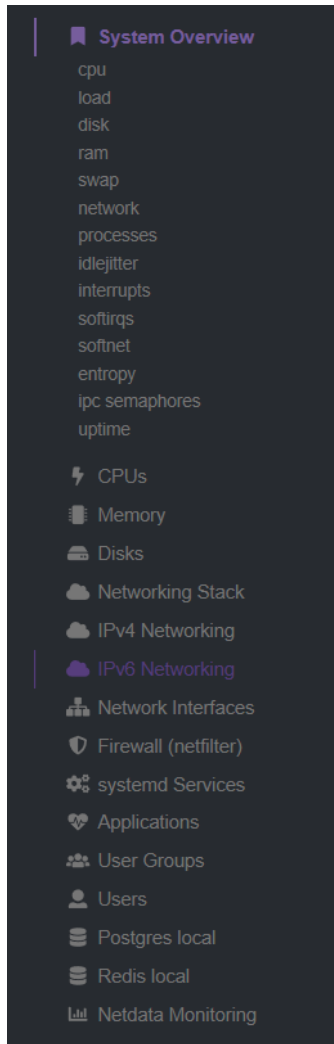


FIGURE 21. Netdata system monitoring menu (from my netdata URL)

Here is few sample diagram of Memory, CPU and networking but this tools provide monitoring of all type of system components.

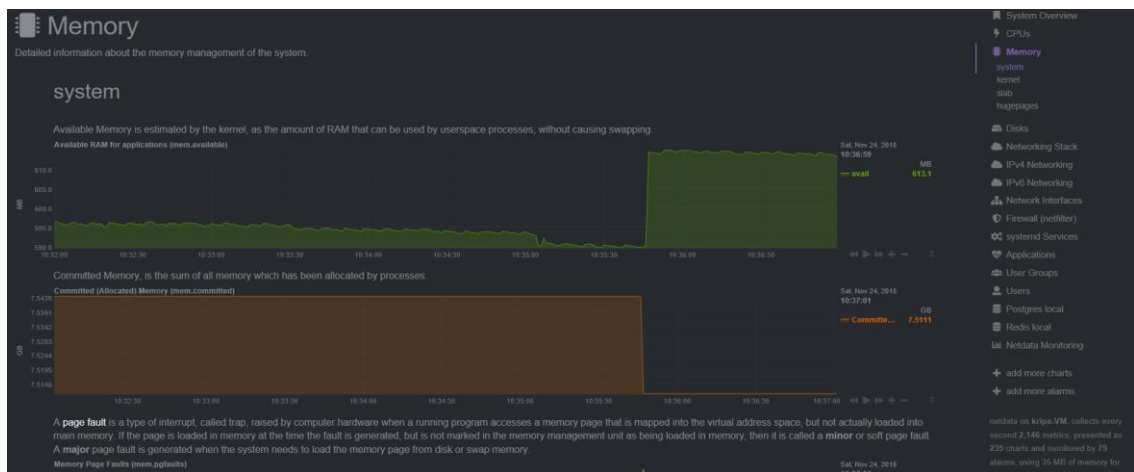


FIGURE 22. Hadoop server memory graph (from my netdata URL)

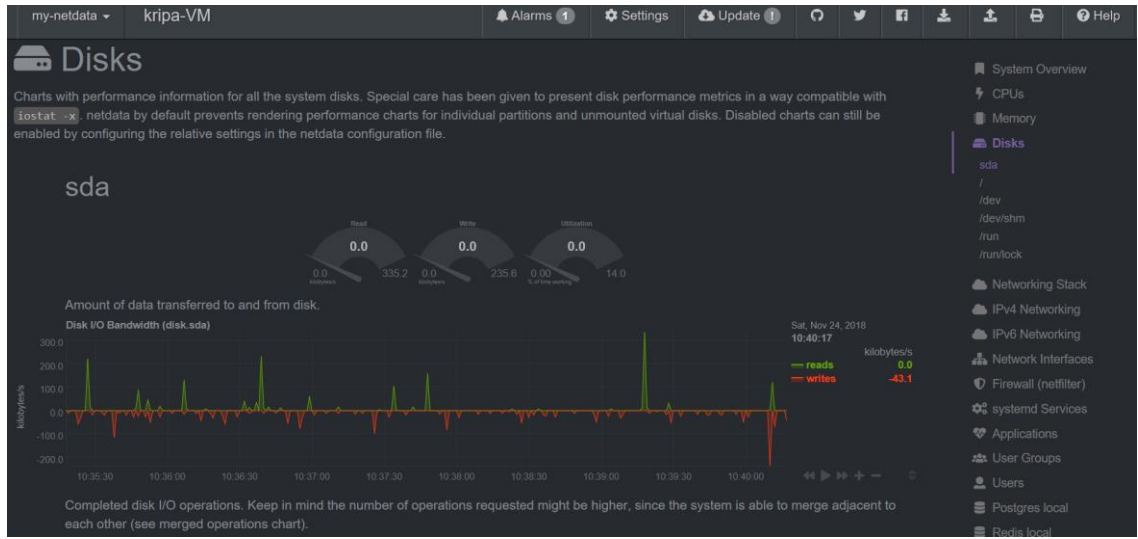


FIGURE 23. Hadoop Server Disks graph (from my netdata URL)



FIGURE 24. Hadoop server kernel graph (from my netdata URL)



FIGURE 25. Hadoop server IPv6 networking graph (from my netdata URL)

## 7. CONCLUSION

This has been a true learning experience. In the beginning the goals appeared as almost overly ambitious to achieve. Even though the final breakthrough was not achieved, a lot of progress has happened, and this work proposes how to continue the work towards an ideal solution.

Big data is by definition big, but a one-size-fits-all approach to security is inappropriate. The capabilities within Hadoop allow organizations to optimize security to meet user, compliance, and company requirements for all their individual data assets within the Hadoop environment. Capabilities like role, user, and group permissions, data masking and multiple encryption and authentication options make it practical to provide different levels of security within a single, large environment. Growing integration support between Hadoop and Active Directory, LDAP and identity management solutions allows organizations to extend their enterprise security solutions so the Hadoop infrastructure doesn't have to be a silo.

Security is one of the fastest-changing aspects of Hadoop. The capabilities are continually enhanced and surpassed. For the latest security updates, check with the Apache project or your Hadoop distributor.



## Hadoop Security Then and Now

Component	Original Hadoop Release	Now Included/Available
<b>Encryption</b>	Not included	DEK encryption automatically applied to data in HDFS and in motion; additional data protection features are specific to each commercial distribution; KMS manages encryption keys; Kerberos is commonly used; additional encryption methods are Hadoop compatible/available.
<b>Authentication</b>	None	Kerberos is the foundation for Hadoop secure mode; Active Directory and LDAP extended to Hadoop; identity management solutions extended to Hadoop.
<b>Access &amp; Permissions</b>	HDFS file permissions	Permissions can be set by individual, group, and role and set for specific data types and files; data masking can be applied to limit data that is accessed.

I have go through all Hadoop versions securities loop holes and its architectures. I main focus is to secure Hadoop server which are running in Ubuntu Linux operating system. I have installed Elasticsearch, Logstash and Kibana Trio to monitor server logs for any suspicious activities.

I also installed Netdata, which having very fancy Dashboard to monitor server each and everything like Ram, CPU, disk and Kernel and give alert notifications as well

## LIST OF REFERENCES

Learning/ Apache. Apache Hadoop Foundation. <http://www.apache.org/>  
<http://openmlp.apache.org/documentation/1.6.0/manual/openmlp.html>

Cloudera security overview.

[https://www.cloudera.com/documentation/enterprise/latest/topics/sg\\_edh\\_overview.html](https://www.cloudera.com/documentation/enterprise/latest/topics/sg_edh_overview.html)

Source of HDFS architecture.

[http://hortonworks.com/apache/hdfs/#section\\_2](http://hortonworks.com/apache/hdfs/#section_2)

Linux Server Hardening

<https://www.cyberciti.biz/tips/linux-security.html>

Hadoop File System

<http://www.linuxlinks.com/article/20130411160837132/QuantcastFileSystem.html>

Hadoop 2.0 Architecture

<http://backtobasics.com/big-data/hadoop/understanding-hadoop2-architecture-and-itsdemons/>

MapReduce

[https://hadoop.apache.org/docs/r1.2.1/mapred\\_tutorial.html](https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html)

Apache Knox. <https://knox.apache.org/>

Apache Ranger. <https://ranger.apache.org/>

Apache Sentry. <https://cwiki.apache.org/confluence/display/SENTRY/Sentry+Tutorial>

7 V's of Big Data. <http://www.prathap kudupublog.com/2018/01/7-vs-of-big-data.html>

Elastic Search. <http://www.elasticsearchtutorial.com/basic-elasticsearch-concepts.html>

Securing Hadoop

[https://securosis.com/assets/library/reports/Securing\\_Hadoop\\_Final\\_V2.pdf](https://securosis.com/assets/library/reports/Securing_Hadoop_Final_V2.pdf)

4 BDaas business models

<https://www.semantiko.com/blog/big-data-as-a-service-definition-classification/>

Hadoop 2.x Back to Basics

<https://backtobasics.com/big-data/hadoop/understanding-hadoop2-architecture-demons/>

Hadoop 1.x: Architecture and major components

<https://www.journaldev.com/8808/hadoop1-architecture-and-how-major-components-works>

Big Data: Hadoop 1.x Vs 2.x architecture

<http://annovate.blogspot.com/2014/07/big-data-hadoop-1x-vs-hadoop-2x.html>

High level architecture of hadoop version 1 and 2

[www.dineshonjava.com/hadoop-architecture/](http://www.dineshonjava.com/hadoop-architecture/)

How to install NetData

<https://www.digitalocean.com/community/tutorials/how-to-set-up-real-time-performance-monitoring-with-netdata-on-ubuntu-16-04>

HDFS federation architecture, [www.quora.com/What-is-HDFS-federation](http://www.quora.com/What-is-HDFS-federation)

MR and Yarn architecture, [www.slideshare.net/PhilippeJulio/hadoop-architecture/25-MR\\_VS\\_YARN\\_ARCHITECTURE\\_MR,p25](http://www.slideshare.net/PhilippeJulio/hadoop-architecture/25-MR_VS_YARN_ARCHITECTURE_MR,p25)

Tom White, “Hadoop: The Definitive Guide”, O’Reilly Media, 2009.

Sachin P Bappalige, ”An introduction to Apache Hadoop for big data”  
(<https://opensource.com/life/14/8/intro-apache-hadoop-big-data>)

D. Borthakur, “The Hadoop distributed file system: Architecture and design,” Hadoop Project Website, vol. 11, p. 21, 2007.

C. Tankard, “Big data security,” Network security, vol. 2012, no. 7, pp. 5–8, 2012.

S. Narayanan, “Securing Hadoop. Packt Publishing Ltd”, 2013.

Michael Kanellos, “The 5 Different Types of Big Data”, [www.forbes.com](http://www.forbes.com), 2016.

Hortonworks, “Hive authorization,” 2013.

Bhushan Lakhe, “Practical Hadoop Security” download online, 2017.

Steve Piper, “Big Data Security for Dummies”, Blue Coat Special Edition, 2015.

CSA (cloud security alliance),”Big Data Security and Privacy Handbook”, 2016.

Ben Spivey, “Hadoop Security Protecting Your Big Data Platform “, 2015.

Fei-Hu, “Big Data Storage Sharing and Security” ,2016.

## APPENDICES

Appendix 1. Project usage important commands.

To find noowner files in system

```
$ find /dir -xdev \( -nouser -o -nogroup \) -print
```

Assign sudo level access to user

```
#usermod -aG sudo username
```

Appendix 2. Well known securities methods.

Accounting	Accounting of logins
Authentication	Logins must be authenticate using AD
Authorization	Put authorize device for logins to access any server or URL
Data Protection	Need to protect Data at high priority as leaking personnel data is crime
Encryption	File level encryption is a must to secure data and directory level access
Perimeter level Security With Apache security projects	Many Security project launch by Apache to secure Hadoop server like Apache sentry, Knox, Ambari, Eagle, etc.

## Appendix 3. Levels in hadoop security

Security Level	Description
Level 0	A fully-functional non-secure cluster which security is zero is called level 0 security. A non-secure cluster is very vulnerable to attacks and must never be used in production.
Level 1	Set up authentication checks to prove that users/services accessing the cluster are who they claim to be. You can then implement some simple authorization mechanisms that allow you to assign access privileges to users and user groups. Auditing procedures to keep a check on who accesses the cluster and how, can also be added in this step. However, these are still very basic security measures. If you do go to production with only authentication, authorization and auditing enabled, make sure your cluster administrators are well trained and that the security procedures in place have been certified by an expert.
Level 2	For more robust security, cluster data, or at least sensitive data, must be encrypted. There should be key-management systems in place for managing encrypted data and the associated encryption keys. Data governance is an important aspect of security. Governance includes auditing accesses to data residing in megastores, reviewing and updating metadata, and discovering the lineage of data objects.
Level 3	<p>At this level, all data on the cluster, at-rest and in-transit, must be encrypted, and the key management system in use must be fault-tolerant. A completely secure enterprise data hub (EDH) is one that can stand up to the audits required for compliance with PCI, HIPAA, and other common industry standards. The regulations associated with these standards do not apply just to the EDH storing this data. Any system that integrates with the EDH in question is subject to scrutiny as well.</p> <p>Leveraging all four levels of security, Cloudera's EDH platform can pass technology reviews for most common compliance regulations.</p>

## Appendix 4. Complete layered and functional architecture of Hadoop security

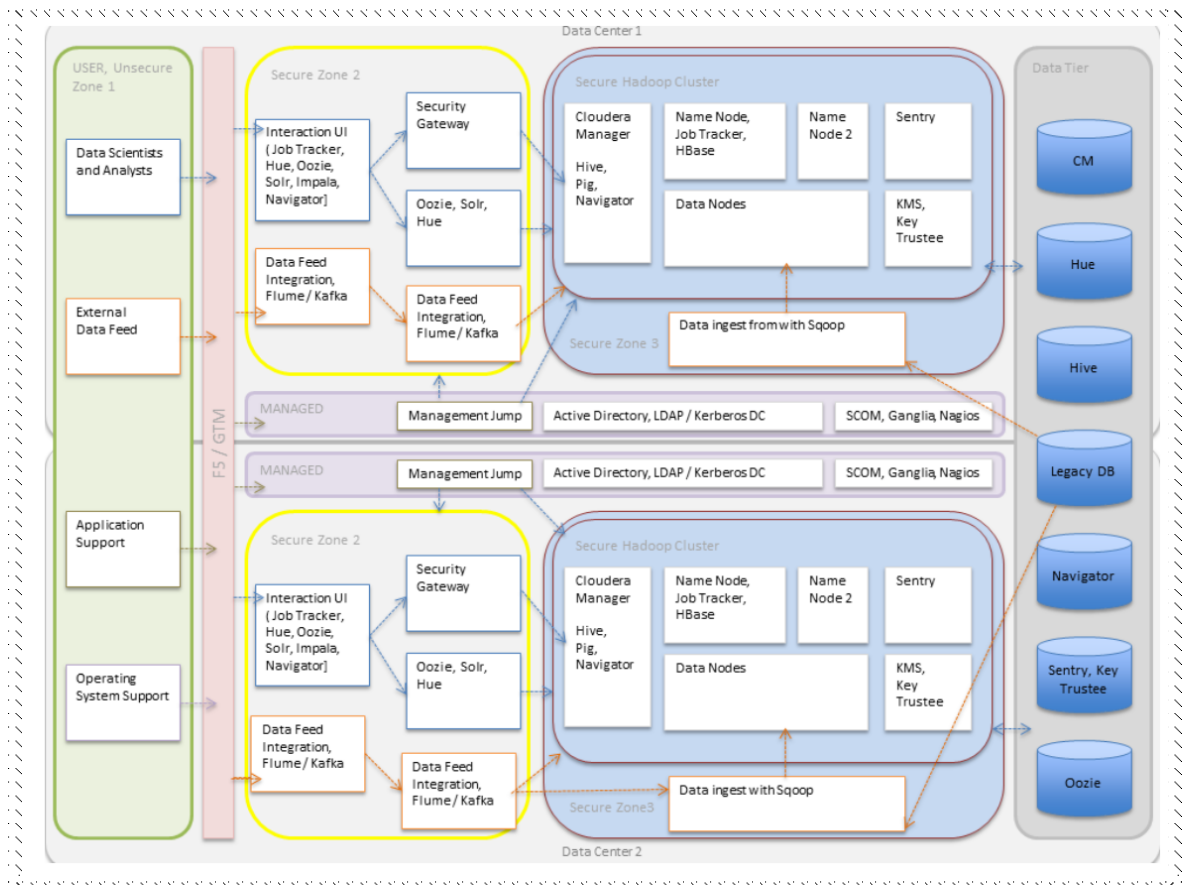


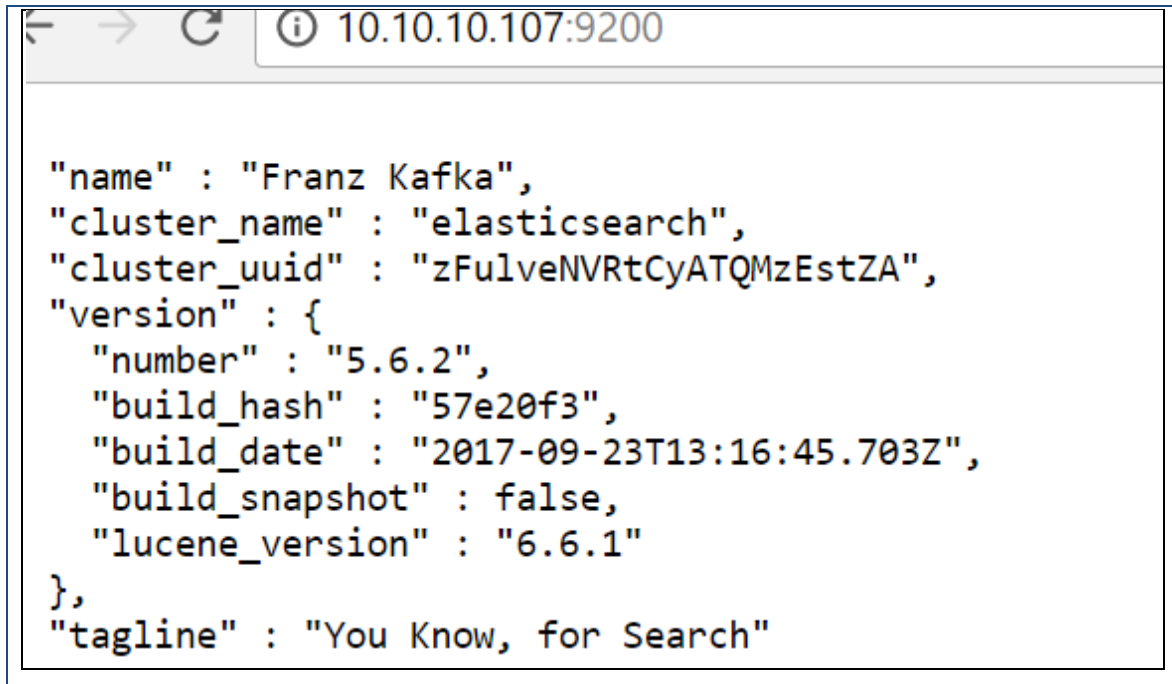
FIGURE 26. Complete layered and functional security of hadoop server (Cloudera security overview, [www.cloudera.com](http://www.cloudera.com))

### How Big Data helps solve and prevent crime

- Perform complex criminal investigation and analysis: Detect, investigate and solve crime through link analysis by connecting people, companies, places and things.
- Use ALL available information, such as financial transactions, criminal records, mobile phone records, telemetry, et cetera. We'll show you how to extract relevant information from these datasources, e.g. by detecting faces from photos and identifying key information in textual documents.
- Detect patterns in criminal behavior and respond quickly by automatically flagging suspicious activity.

- As an example, this demonstration will go through a case of a money laundering investigation. We'll go through the full process of data acquisition, analysis and conclusion

Elasticsearch Browser (www.10.10.10.107:9200)

A screenshot of a web browser window displaying the Elasticsearch status page. The address bar shows the URL '10.10.10.107:9200'. The main content area displays a JSON object with the following fields: 'name' (Franz Kafka), 'cluster\_name' (elasticsearch), 'cluster\_uuid' (zFulveNVRtCyATQMzEstZA), 'version' (5.6.2), 'build\_hash' (57e20f3), 'build\_date' (2017-09-23T13:16:45.703Z), 'build\_snapshot' (false), 'lucene\_version' (6.6.1), and 'tagline' (You Know, for Search).

```
"name" : "Franz Kafka",
"cluster_name" : "elasticsearch",
"cluster_uuid" : "zFulveNVRtCyATQMzEstZA",
"version" : {
  "number" : "5.6.2",
  "build_hash" : "57e20f3",
  "build_date" : "2017-09-23T13:16:45.703Z",
  "build_snapshot" : false,
  "lucene_version" : "6.6.1"
},
"tagline" : "You Know, for Search"
```

**Malware**, or malicious software, are computer programs designed to infiltrate and damage computers and networks. Malware can include viruses, worms, Trojan horses, spyware, adware, and ransomware. Cyber-criminals hide malware in attachments to emails and messages, or as documents or software on websites. Opening or clicking on these attachments can download the malware onto company devices and can allow them to access company information systems, often without the user being aware of it.

**Ransomware** is a type of malware that, once downloaded to a computer or network, locks the users out of the system and/or encrypts certain files and data. The ransomware then notifies the users of a ransom demand that must be paid, or else the data will be deleted or the key to unlock the encryption will be destroyed, eliminating any chance of recovery.

There are many configuration files available in my virtual server for elasticsearch, kibana, logstash and netdata tools for Hadoop server security monitoring.