*Creating a Thriving Data Lake*

*Using*

*Open Source Technologies*

|  |  |
| --- | --- |
| Ram Raju, Krishna Nagalla  Stinger Ghaffarian Technologies, Inc.  Greenbelt, Maryland  [rraju@sgt-inc.com](mailto:rraju@sgt-inc.com), knagalla@sgt-inc.com | Sahil Khanna  University of Maryland  College Park, Maryland  sahilkhanna.umd@gmail.com |
|  |  |

# Abstract

The concept of a Data Lake is emerging as a popular paradigm to organize and build the next generation of data analytics systems, to harness big data from a variety of sources – structured, semi-structured and unstructured data within organizations, but there are also some concerns and questions for large enterprises with regard to implementation of Data Lakes. The paper discusses the concept of Data Lakes, the design of a prototype data lake, the deployment of a sample data lake using Open Source Technologies in a Public Cloud infrastructure, the utilization of this Data Lake, to realize sophisticated analytics capabilities, and a discussion about the benefits of, and the challenges posed by Data Lakes.

# Background

The central goal of a Data Lake is to act as a data repository for vast amounts of data stored in raw, processed and refined formats, along with a variety of data analytics tools, to support sophisticated analytics. Data in the data lake is collected from the many, and ever-increasing number of, data sources in organizations. The data can then be transformed and distributed to downstream systems as required. A data lake approach can reduce data silos, help support big data initiatives, store huge data sets quickly and process various data types.

Traditional approaches, such as data warehouses, require processing of the data set at the time of storage using traditional Extract-Transform-Load (ETL) approaches, which involve both cost, and almost inevitable inadequacies to the staggering rate of data changes. These traditional ETL approaches provide structured and reined data which is easier to analyze and query – but increasingly, this benefit is outweighed by the high cost and time-to-market within the evolving enterprise. This drawback of having to process all the data through ETL processes, led to the emergent of Data Lakes. A Data Lake achieves data storage without ETL processing and provides more flexible and responsive data analytics, better suited to the modern enterprise. This paper describes the design and implementation of a prototype Data Lake in a Public Cloud infrastructure using Open Source Technologies.

The prototype Data Lake described in this paper, is deployed on the Amazon Web Services EC2 environment. The design offers an illustration for how users can realize powerful data analysis using structured, unstructured and semi-structured data in an Open Source Data Lake, and how analysts can easily use the distributed processing paradigm to achieve scalable and fault tolerant data analytics. The project accomplishes efficient data processing for wide variety of business problems and offers tools to execute queries, searches, processing streams and visualizing data. It showcases the use of scalable, state-of-the-art open source data analytics technologies to analyze aviation delay data, and stream data from Twitter. Using Apache Spark, Apache Pig, Apache Hive, Apache Kafka, Apache Oozie and Apache Sqoop, the project demonstrates how analytics can quickly and easily build powerful data pipelines and statistical models.

# Objectives

* Design and Deploy a sample Data Lake to provide fault tolerant, scalable, reliable and fast data analytics system which fulfill typical data warehouse requirements such as security, monitoring, governance, analytics, management and ingestion adaptors.

* Demonstrate how modern, open source technology frameworks and the public cloud, can be effectively used to provide big data analytics capabilities at low cost and with rapid speed to solve complex problems in the various domains. The use of interactive interfaces and automatic data processing can greatly simplify the technical aspects of analytics, allowing analysts to focus on the actual analysis.

# Datalake Concept

## Definition

“If you think of a Datamart as a store of bottled water, cleansed and packaged and structured for easy consumption then the Data Lake is a large body of water in a more natural state. The contents of the Data Lake stream in from a source to fill the lake, and various users of the lake can come to examine, dive in, or take samples.” - James Dixon, CTO, Pentaho Corporation

A Data Lake is a single store of all data in the enterprise ranging from raw data (which implies exact copy of source system data) to transformed data which is used for various tasks. A Data Lake supports both storage and analysis which includes reporting, visualization, analytics, statistics and machine learning. The Data Lake includes structured data from relational databases (rows and columns), semi-structured data (CSV, logs, XML, JSON), unstructured data (emails, documents, PDFs) and even binary data (images, audio, video) thus creating a centralized data store accommodating all forms of data, readily available to analysts across the organization.

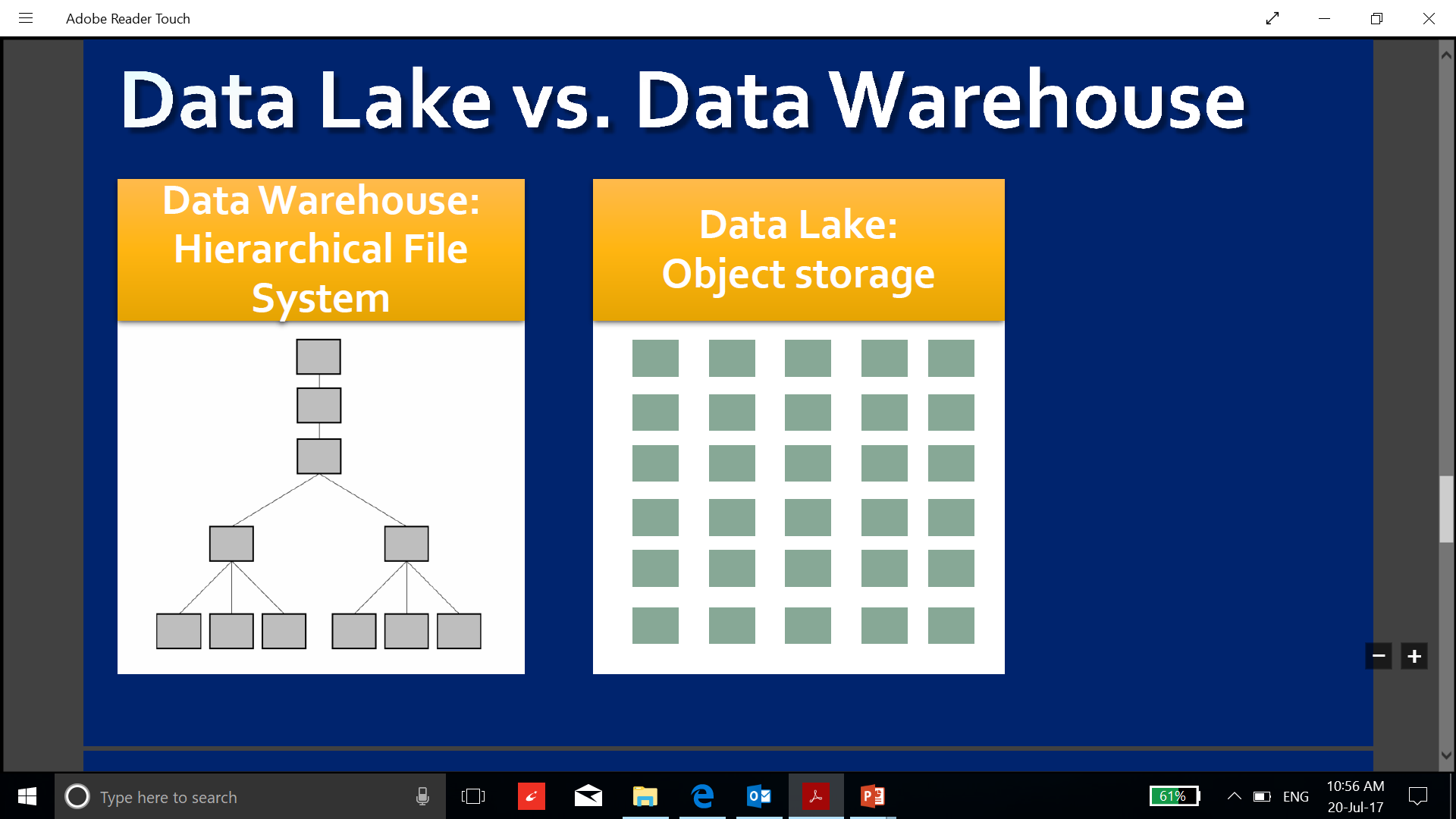


Figure 1 Data Lake Concept Diagram

## Characteristics of the Data Lake

* It stores all data including raw data, processed data and refined data
* It supports all data types: structured, semi structured, unstructured
* It works on the concept of schema on read which offers low barrier on collection and control
* It consists of strong data architecture and data management components
* It is “Self-service” model for analysts
* It contains a hybrid mix of storage technology (not just Hadoop) and analytical tools
* It supports “Big Data” characteristics: Volume, Velocity and Variety
* It is highly agile, it can be configure as required

## Data lake Technologies

Datalake concept can be implemented using many technologies. The technologies parts of our research were Sqoop, Hadoop Distributed File System, Yarn, Map Reduce, Pig, Hive, Tez, Oozie, Zookeeper, Spark, Ranger, Atlas, Kafka, HBase, Solr, Zeppelin, Apache Falcon, Apache Flume, Ipython Notebook, Kibana and Elastic Search. After the research, we selected the Hortonworks distribution (HDP) to implement our Data Lake design. Hortonworks data platform is an open source project which is a combination of various big data tools. The HDP distribution provides tools to implement, install, configure, monitor, secure and govern the Data Lake, while maintaining a commitment to “pure” open source approach.

# Datalake high level Architecture

**Data Ingestion**

**Stream Data**

**Stationary Data**

**Data Management**

**Data Analytics**

**Governance/Security/Monitoring**

**Refined Data**

**Processed Data**

**Raw Data (Landing zone)**

Figure 2 Data Lake High Level Architecture

The Datalake design is divided into four segments:

## Data Ingestion

This segment handles the data transfer between the Data Lake and External Systems. In this project, we considered two major data types: Streaming data (i.e. logs, news, tweets) and Stationary data (i.e. csv data, json data, database tables). These data types are integrated with the Datalake using Data ingestion adaptors.

## Storage

Storage segment is divided into 3 zones: raw data zone, processed data zone and refined data zone. All the incoming data is first transferred into raw data zone using data ingest adaptor. After the transfer, the data filtering and data wrangling operation are performed on the raw data and output is stored in processed data zone for data analytics. The reports generated after the analysis of the processed data is stored in refined data zone.

Refined Data Zone

- Heavy governance

- Trusted, curated data

- Lineage, data quality

Processed Data Zone

- Minimal governance

- Make sure there is no sensitive data

Landing (Raw) Data Zone

- Minimal governance

- Make sure there is no sensitive data

Figure 3 Benefits of Zones

## Data Management and Control

This capability in the Data Lake includes Data life cycle management functions such as allocating application resources, managing data replications, designing data pipelines, data security, meta-data management and scheduling life cycles is achieved using this segment. All the segments are controlled, secured, monitored and governed using data control segment. This segment achieves centralized security, governance and monitoring. It implements automatic installation and configuration of the Data Lake tools on a large set of cluster nodes.

## Data Analytics

Many tools are incorporated to provide data analytics capabilities such as data search, functional programming, querying, data visualization and data modelling. These operations can be implemented on raw data, processed data and refined data.

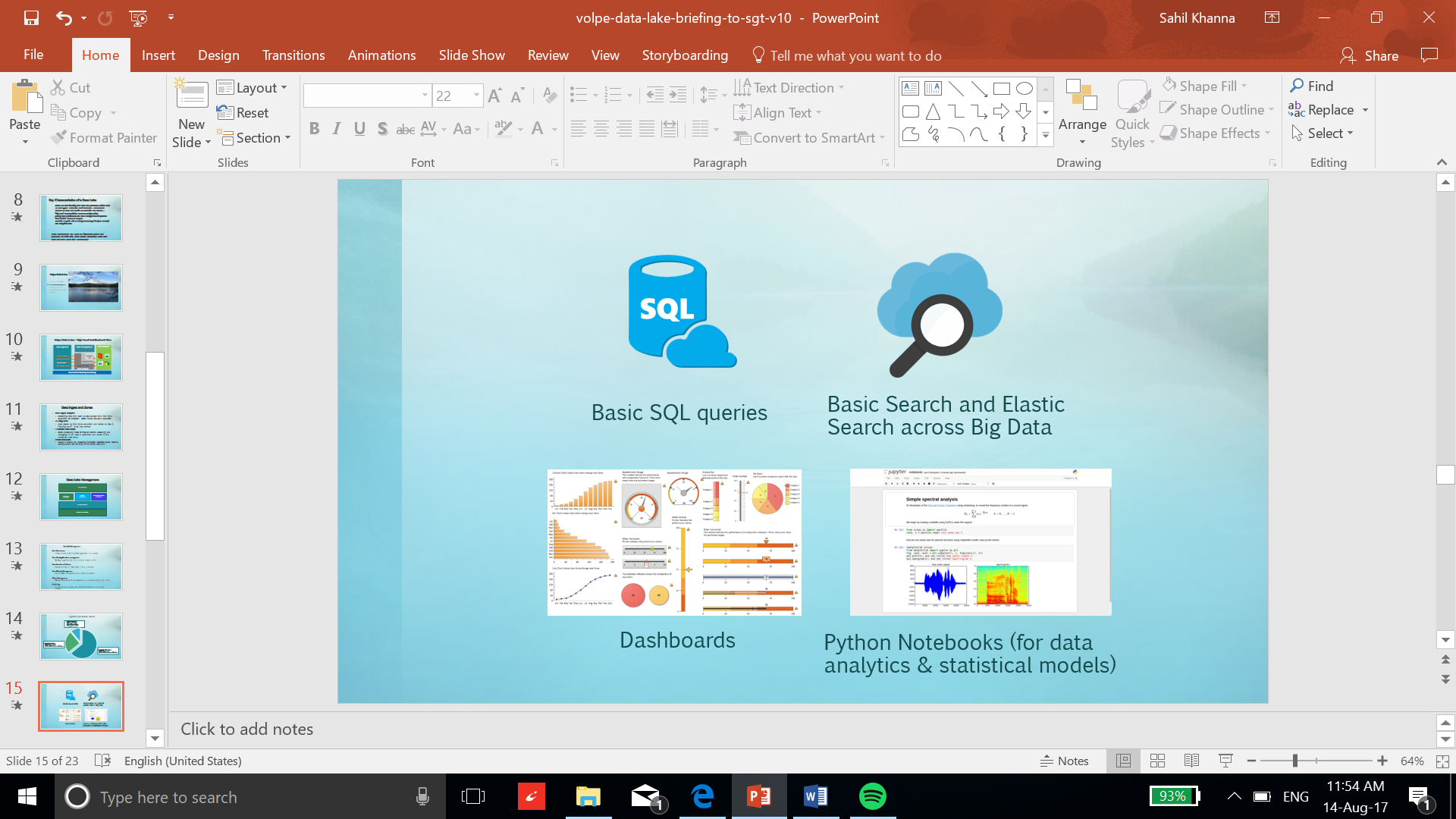
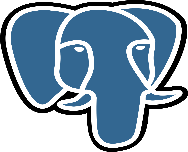


Figure 4 Analytics Capabilities of Data Lake

## Datalake technical Architecture

This section describes the mapping of the high level architecture described in the preceding section to the technology elements in the data lake solution.

Zookeeper



PostgreSQL

Stream

Jupyter Notebook

HDFS

YARN

Kafka

Pig

Hive

Map Reduce

Spark

Solr

Tez

Security Monitoring Governance

Ranger Ambari Atlas

**Scheduling**

Oozie

**Data Flow**

Sqoop

Figure 5 Datalake Designed Architecture

## Data Ingestion

To implement data ingestion layer, we implemented 2 ingest adaptors; Sqoop (to transfer structured data) and Kafka (to transfer stream data). Each is described in greater detail below:

### [Image result for kafka](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiTr4fU9tnVAhUkyoMKHW0NDDgQjRwIBw&url=https://twitter.com/apachekafka&psig=AFQjCNHDT-vyPhwfbTDAEIccR4hs9AvjHQ&ust=1502910125954019)Kafka

Apache Kafka is a fast, scalable, durable, and fault-tolerant publish-subscribe messaging system. It lets you publish and subscribe to streams of records, store streams of records in a fault-tolerant way and process streams of records as they occur.

A picture containing parking, sign, meter

Description generated with very high confidence

Figure 6 Kafka Process Diagram

### Sqoop

Apache Sqoop efficiently transfers bulk data between Apache Hadoop and structured datastores such as relational databases. Sqoop helps offload certain tasks (such as ETL processing) from the EDW to Hadoop for efficient execution at a much lower cost. Sqoop can also be used to extract data from Hadoop and export it into external structured datastores.

## Storage

Storage segment is implemented using Hadoop distributed file system with the additional provision to integrate PostgreSQL with the Data Lake.

### [Image result for hdfs](https://www.snaplogic.com/snaps/hdfs-read-write)Hadoop distributed file system

HDFS is a Java-based file system that provides scalable and reliable data storage. It was designed to span large clusters of commodity servers. HDFS has demonstrated production scalability of up to 200 PB of storage and a single cluster of 4500 servers, supporting close to a billion files and blocks. It offers fault tolerant, reliable data storage using high availability, scalability and replication properties.

## Data Management

The Data management layer was implemented using Yarn, Zookeeper and Oozie to manage life cycle scheduling, make data pipelines, manage resources and applications.

### Yarn

YARN provides resource management and a central platform to deliver consistent operations, security, and data governance tools across Hadoop clusters. YARN also extends the power of Hadoop to incumbent and new technologies found within the data center so that they can take advantage of cost effective, linear-scale storage and processing.

### [Image result for apache zookeeper](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwisgdO199nVAhVB7YMKHViYA6QQjRwIBw&url=https://hub.docker.com/_/zookeeper/&psig=AFQjCNFBhk_zcKemMS6JfpfcsAzclogLsg&ust=1502910329778866)Zookeeper

Apache Zookeeper provides operational services for a Hadoop cluster. Zookeeper provides a distributed configuration service, a synchronization service and a naming registry for distributed systems. Distributed applications use Zookeeper to store and mediate updates to important configuration information.

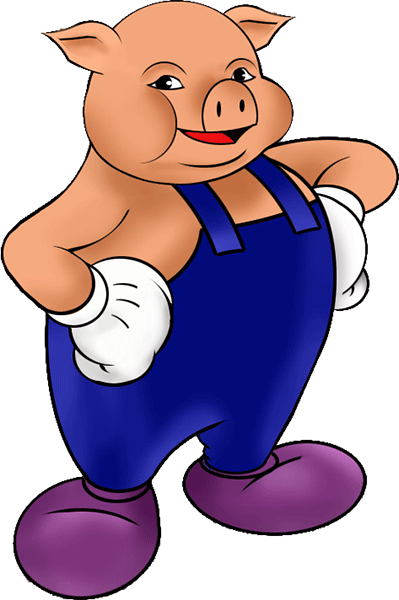
### [Image result for apache oozie logo](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwj747ne99nVAhWF64MKHY11ANwQjRwIBw&url=https://cwiki.apache.org/confluence/display/OOZIE/Oozie%2BLogos%2Band%2BImages&psig=AFQjCNEXG77jrJEydE6L1Em1ooB7vY7Uwg&ust=1502910410683967)Oozie

Apache Oozie is a Java Web application used to schedule Apache Hadoop jobs. Oozie combines multiple jobs sequentially into one logical unit of work. It is integrated with the Hadoop stack, with YARN as its architectural center, and supports Hadoop jobs for Apache MapReduce, Apache Pig, Apache Hive, Apache Spark and Apache Sqoop. Oozie can also schedule jobs specific to a system, like Java programs or shell scripts.

## Data Analytics

Tools incorporated to offer wide variety of data analytics operations are Pig, Hive, Spark, Solr, Jupyter Notebook and Map Reduce.

### **Pig**



Apache Pig allows Apache Hadoop users to write complex MapReduce transformations using a simple scripting language called Pig Latin. Pig translates the Pig Latin script into MapReduce so that it can be executed within YARN for access to a single dataset stored in the Hadoop Distributed File System (HDFS).

### [Image result for apache hive](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwj02q60uNfVAhVC-lQKHfUYBFwQjRwIBw&url=https://en.wikipedia.org/wiki/Apache_Hive&psig=AFQjCNHhycSbvj-CZZr3WwjexO-HfVkB6w&ust=1502824697979615)Hive

Hadoop was built to organize and store massive amounts of data of all shapes, sizes and formats. Because of Hadoop’s “schema on read” architecture, a Hadoop cluster is a perfect reservoir of heterogeneous data—structured and unstructured—from a multitude of sources. Data analysts use Hive to query, summarize, explore and analyze that data, then turn it into actionable business insight

### [Image result for apache spark logo](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjCh7-F-NnVAhWh6oMKHX-1BEQQjRwIBw&url=https://commons.wikimedia.org/wiki/File:Apache_Spark_logo.svg&psig=AFQjCNF-7HJuGHPwS35OLwlbUhvr3diA1A&ust=1502910495812016)Spark

Apache Spark is a fast, in-memory data processing engine with elegant and expressive development APIs to allow data workers to efficiently execute streaming, machine learning or SQL workloads that require fast iterative access to datasets. With Spark running on Apache Hadoop YARN, developers everywhere can now create applications to exploit Spark’s power, derive insights, and enrich their data science workloads within a single, shared dataset in Hadoop.

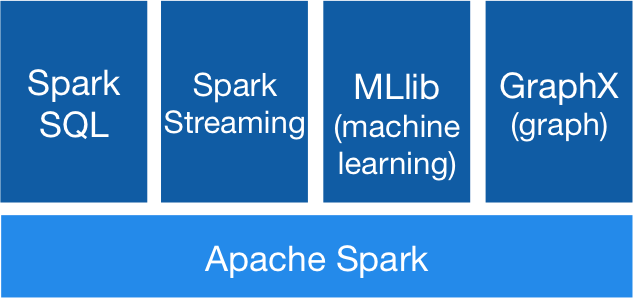
[](https://spark.apache.org/)

Figure 7 Spark Modules Architecture

### [Image result for apache solr](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiPyO-EudfVAhWJxVQKHfnLCKMQjRwIBw&url=https://en.wikipedia.org/wiki/Apache_Solr&psig=AFQjCNFIw4K1OygijCKLFvyREsUqFaBfuA&ust=1502824865714771)Solr

Apache Solr is the open source platform for searches of data stored in HDFS in Hadoop. Solr powers the search and navigation features of many of the world’s largest Internet sites, enabling powerful full-text search and near real-time indexing. Whether users search for tabular, text, geo-location or sensor data in Hadoop, they find it quickly with Apache Solr.

### [Related image](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjOltCr-NnVAhWoy4MKHUAXCCIQjRwIBw&url=http://www.cosam.calpoly.edu/news/cal-poly-helps-make-gravitational-wave-calculations-available-all&psig=AFQjCNFDDkzGQAxAF0d1AOR6gihKW4Uv6Q&ust=1502910567616767)Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more

### [Image result for mapreduce logo](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiPofDG-NnVAhUM9IMKHeK1DjoQjRwIBw&url=https://bigishere.wordpress.com/2016/06/22/mapreduce-the-heart-of-hadoop/&psig=AFQjCNEe__zqq9kKUXg4dZz7m5-gR82O_A&ust=1502910634069557)Map Reduce

MapReduce is the original framework for writing applications that process large amounts of structured and unstructured data stored in the Hadoop Distributed File System (HDFS). Apache Hadoop YARN opened Hadoop to other data processing engines that can now run alongside existing MapReduce jobs to process data in many ways at the same time.

## Data control

Data control, security, monitoring and governance is achieved using Apache Ambari, Apache Atlas and Apache Ranger. These tools provide a capability to make security policies, monitor health of the Data Lake, change configuration using web interface, install various tools automatically and interactive governance of data such as tracking data lineage or exchanging metadata.

### Apache Atlas

Atlas is designed to exchange metadata with other tools and processes within and outside of the Hadoop stack, thereby enabling platform-agnostic governance controls that effectively address compliance requirements

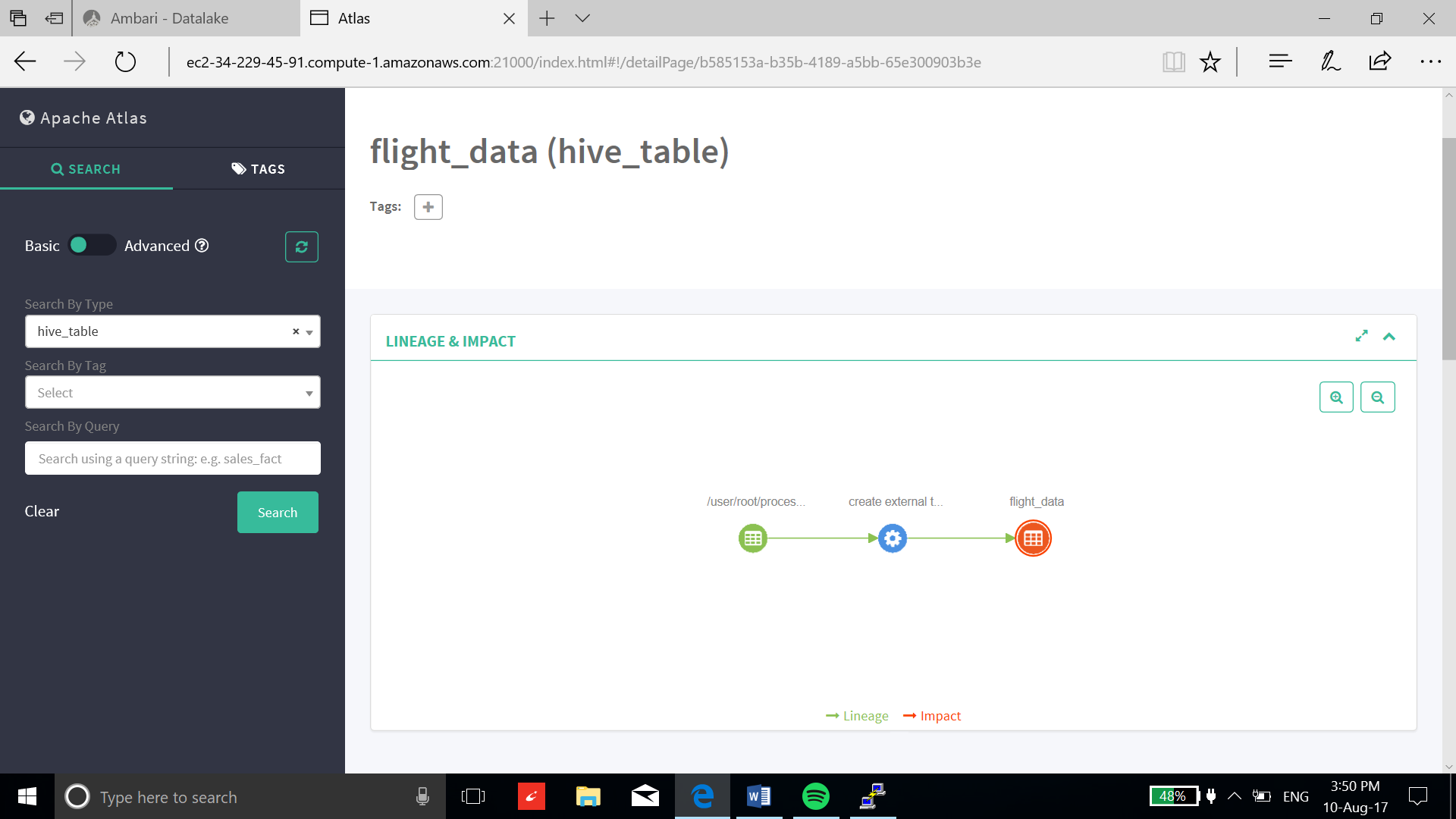


Figure 8 Using Apache Atlas User Interface Tracking Data Lineage

### Apache Ambari

A completely open source management platform for provisioning, managing, monitoring and securing Apache Hadoop clusters. Apache Ambari takes the guesswork out of operating Hadoop. Apache Ambari, as part of the Hortonworks Data Platform, allows enterprises to plan, install and securely configure HDP making it easier to provide ongoing cluster maintenance and management, no matter the size of the cluster.

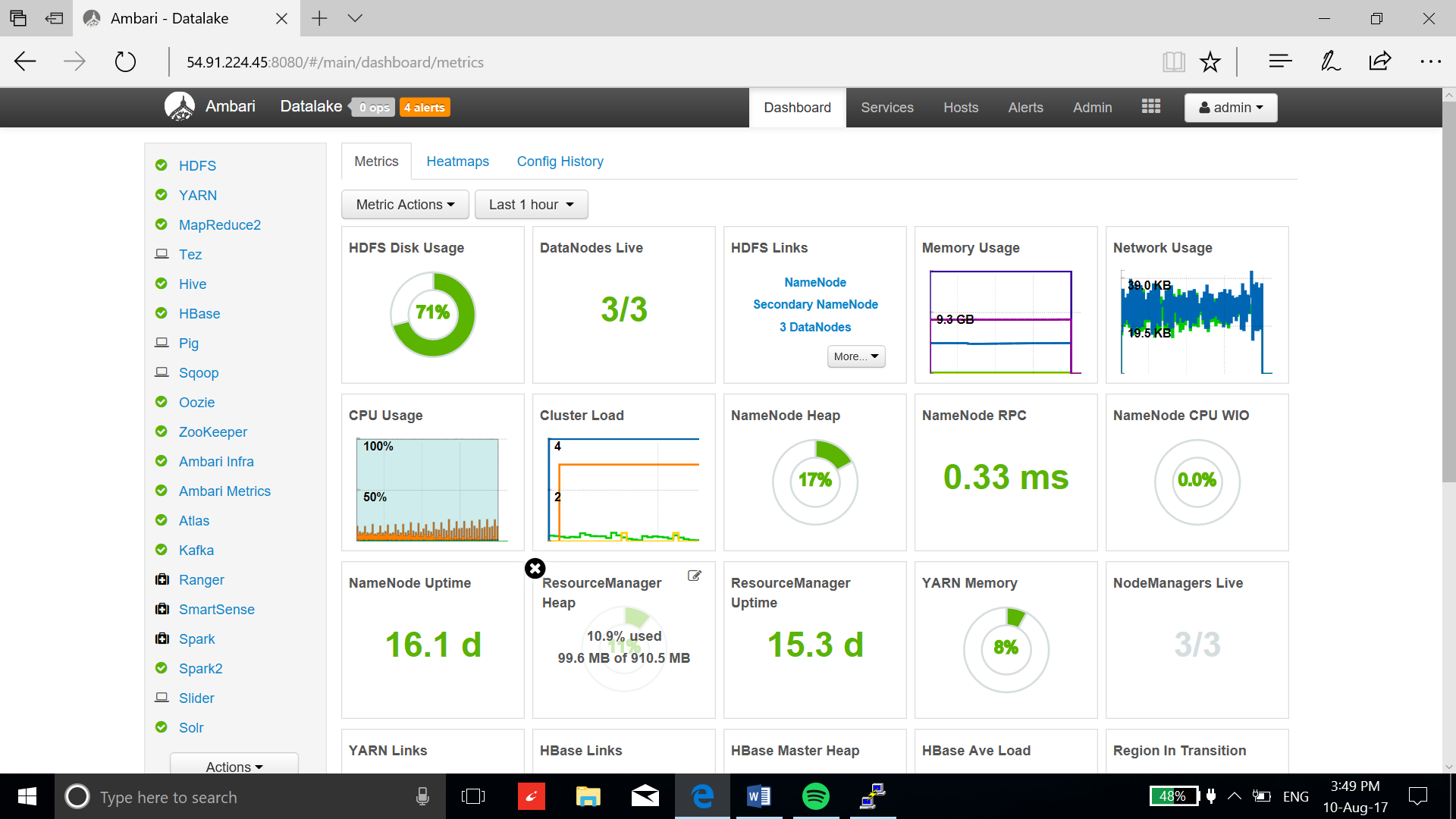


Figure 9 Monitoring Data Lake using Apache Ambari Dashboard

### Apache Ranger

Apache Ranger delivers a comprehensive approach to security for a Hadoop cluster. It provides a centralized platform to define, administer and manage security policies consistently across Hadoop components.

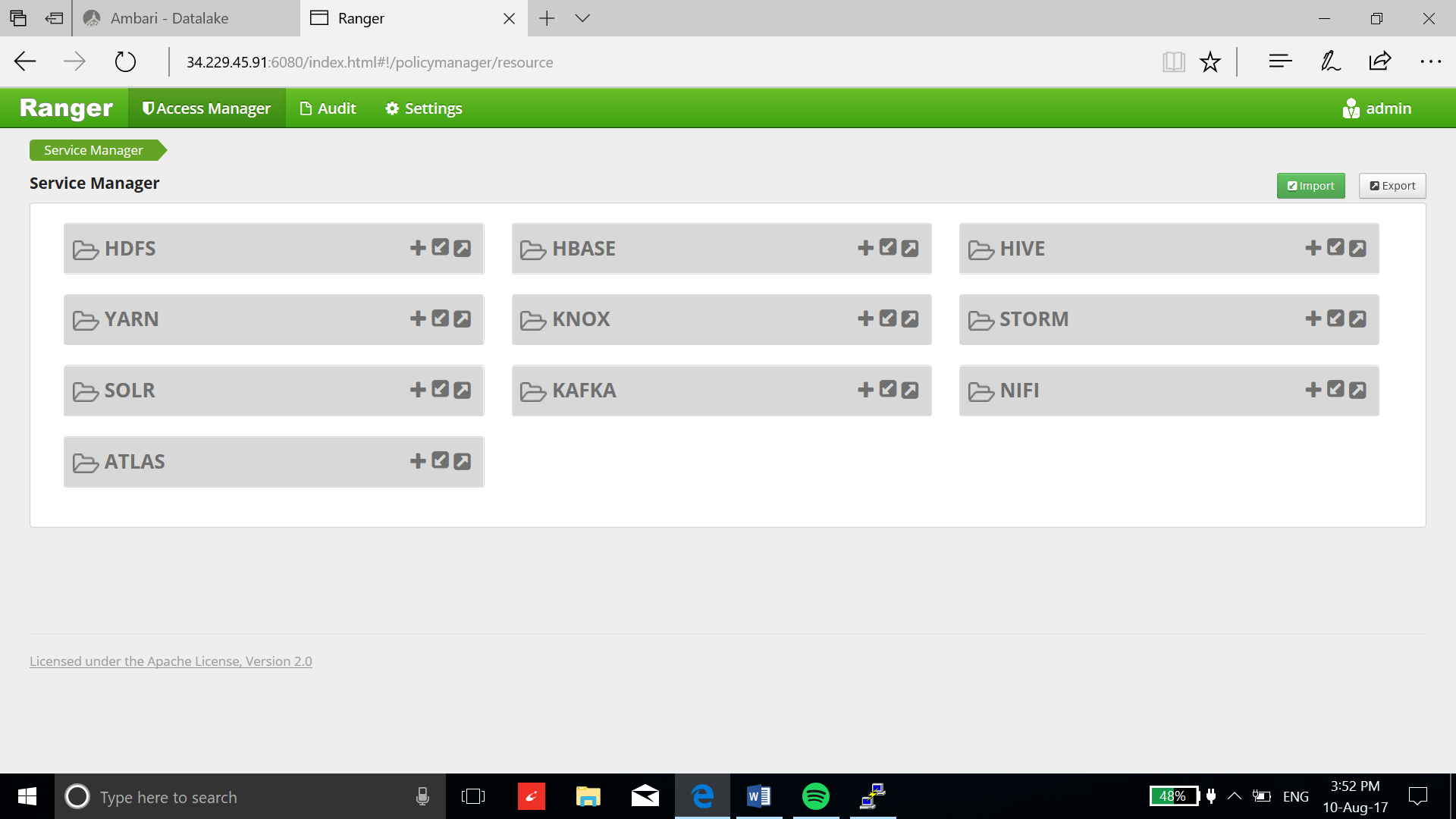


Figure 10 Apache Ranger Dashboard for Centralized Control

# Benefits of using the Datalake

* Fast, Schema-less Storage
* Capability to make data-pipelines for automatic data processing
* Wide variety of analytics tools to design data models
* Integration with traditional data warehouse
* Interactive data governance and data security
* Easy monitoring, configuring and controlling
* Scalable and fault tolerant

# Challenges of Data Lake

## ****Understand the purpose and limitations of the technology****

Hadoop environment provides a vendor independent tool-chain for data storage, management and analytics in a scalable fashion – it can manage infinite amount of data and with add-ons makes real-time and data science use-cases available for the enterprise. Many organizations do not have Big Data. They just load their RDBMS content and get surprised that a 20-node enterprise Hadoop setup’s performance is subpar compared with a single PostgreSQL instance because the purpose and capabilities are different.

### Proposed Solution: Insightful research on the various tools and data types required by company to solve its business problems.

## Security

When building a data lake our goal is to have all data at one place and make it available for various teams. Therefore, planning and deployment of the security framework should be a well-defined sub-project.

### Proposed Solution: Centralized security provision to make policies based on users and group to give access based on files, folders, databases, tables and columns.

## Data governance

An RDBMS enforces a schema, also many rules, e.g. you cannot add a string to an integer field. A data lake does not really have such enforcement mechanisms by default – you can put anything, anywhere. You should setup policies, procedures, processes, responsible personnel, documentation, a technology enforcing those and rich documentation to avoid chaos. However, constructing such a framework is a tough task.

### Proposed Solution: Interactive governance to work with meta-data, data lineage and replications.

## Complexity

Acquiring knowledge of various big data technologies is not an easy task. All the tools are built on different architecture and works with different commands set. Learning these commands and operate various tools requires time.

### Proposed Solution: Provisioned user interfaces and single Api to utilize all the tools.

# Installation process

The installation of this prototype Datalake was implemented using Apache Ambari which provides a user-friendly user interface and quickly implement the installation process over large cluster. The installation can be implemented on hundreds of nodes automatically using Apache Ambari-server and Apache Ambari-agents. All the process to install and configure the Data Lake is documented for the future installation. The same process can be used to implement Data Lake on the private network without any changes in installation.

# Data pipelines

Data Lake provides capability to serialize and schedule tasks which can be utilized to make data pipelines. This capability achieves automatic and periodic data processing without human intervention. Data pipelines are implemented using Apache Oozie which integrates different scripts into a single task. This task can also be scheduled using Apache Oozie.

To implement a data pipeline, users require to write scripts using tools such as Pig, Hive, Spark, Sqoop and Map Reduce. Along with the scripts, Oozie needs “job.properties” and “workflow.xml” file to implement a data pipeline. “Job.properties” file is used to provide tasks specific locations such as Input directory, Output directory, Node manager, Resource manager and scripts addresses. “Workflow.xml” file executes different script in a sequential order. Using workflow.xml, user can serialize and parallelize thousands of tasks which can be scheduled based on a period to automate the process. The capability to integrate various tools into one data flow process makes the Data Lake implementation robust to big data analysis.

# Jupyter Notebook Interface

Analysis is performed using a Jupyter Notebook along with Python libraries, Spark and Kafka.

The notebook has “cells” which can contain code, text, images or data — allowing for a comprehensive analytical report to be built in a single, intuitive browser based interface. This way analysts can get to work as soon as the cluster is working simply by opening the browser-based interface. No additional back-end configuration is necessary.

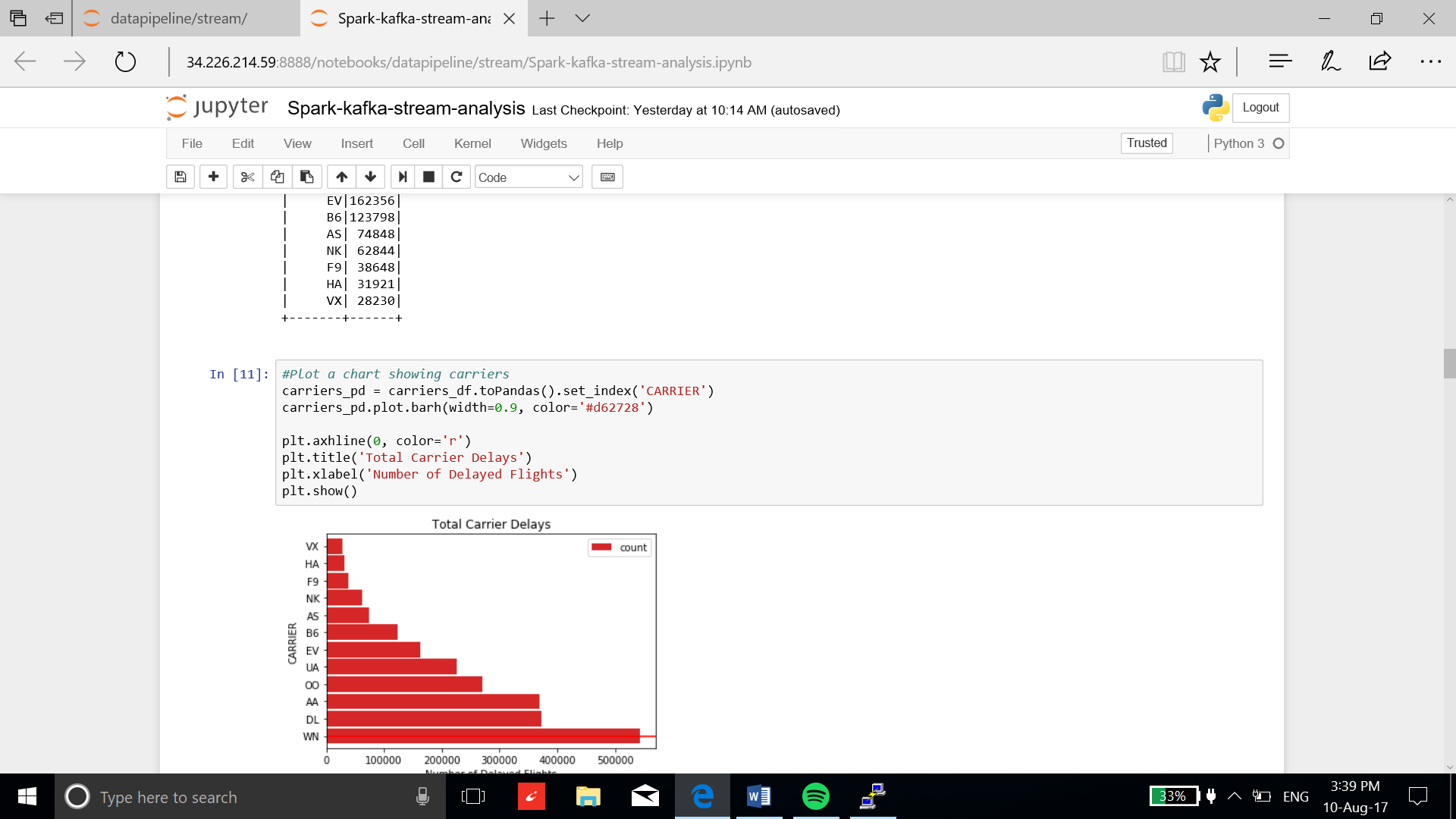


Figure 11 Jupyter Notebook Interface

## Stream Analytics Using Kafka and Python libraries

In this example, I have integrated the twitter stream with the Data Lake using Apache Kafka. The twitter stream is processed using python libraries and tweets’ texts are stored in processed data zone for future analytics. I have also performed the sentiment analysis on tweets using Python library and stored the results in refined data zone. This example illustrates the Data Lake capability to perform different kind of operations on the stream data.

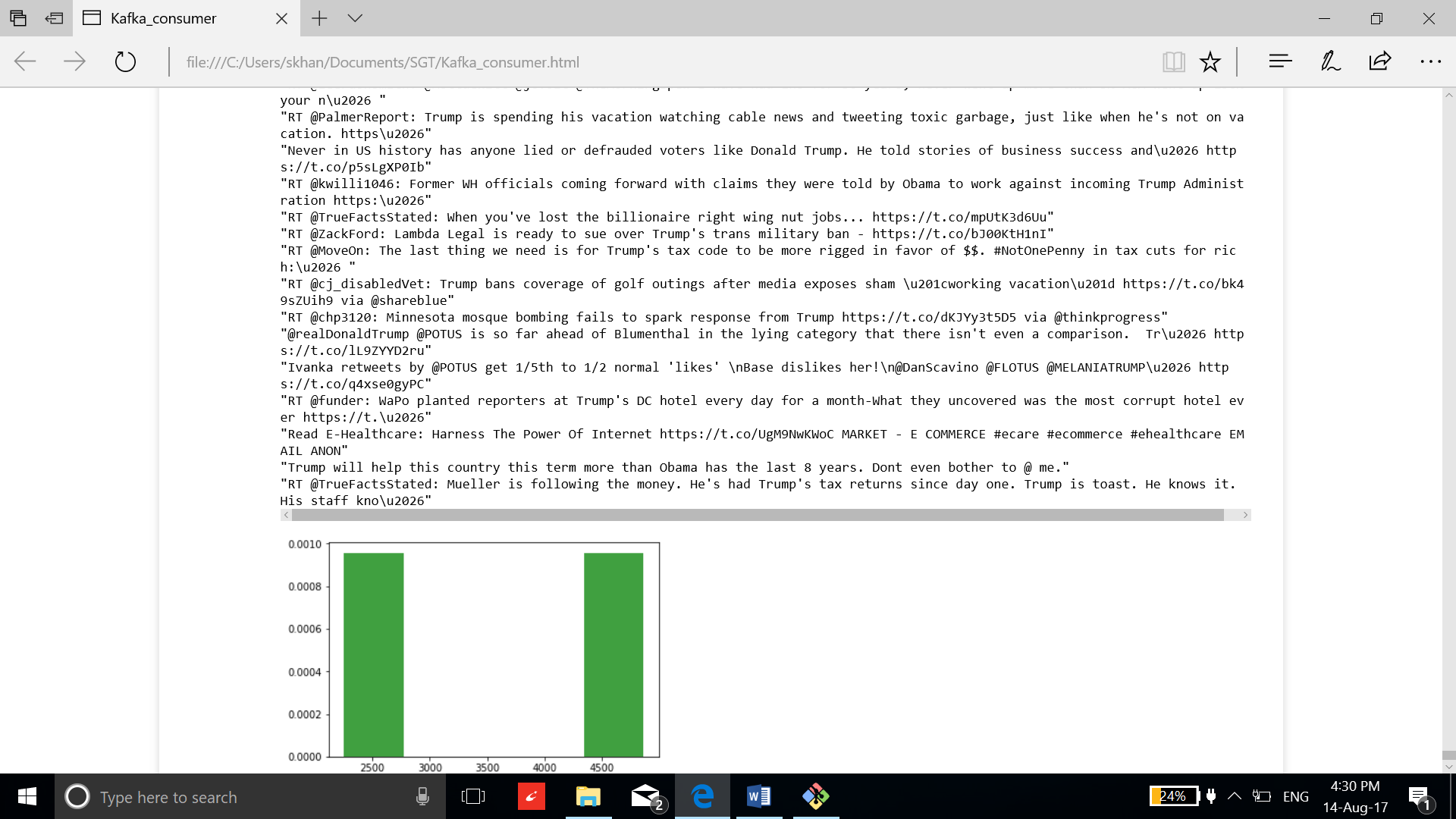
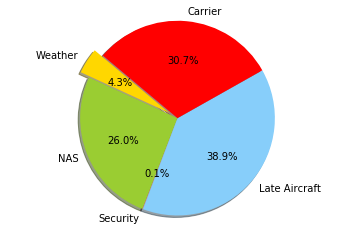


Figure 12 Processed tweets and Sentiment analysis result as bar graph

## Descriptive Analytics Using Spark SQL module

Below are some examples of preliminary analysis of the data set. These were generated in Jupyter Notebook using Python code and libraries, and Spark modules. The sample source code to generate these visualizations is contained in Jupyter Notebook, and can be made available to readers upon request.

**Question:** Which airline carriers had the most delays in 2017? Why? And when?

A screenshot of a cell phone

Description generated with high confidence

Figure 14 Carrier's Number of Delay Flights in 2017

Figure 13 Percentage Cause of Flight Delay

A close up of a logo

Description generated with high confidence

Figure 15 Number of Delayed Flights per Month

Another set of visualizations that examines delays by airport appears below. Note that the analyses below and in the preceding example are not normalized and present the cumulative summary statistics. Therefore, a particular airport or airline having significantly higher delays relative to another airport or airline might be due to the sheer volume of flights being served. A future iteration of the analysis will factor flight volumes and other considerations. These simple examples illustrate how quick insights can be gleaned with very little effort.

**Question:** Which airports had the most delays? When and why?

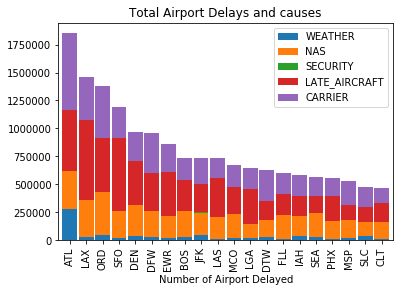


Figure 16 Total Delayed Flights due to Different Reasons Figure 17 Delays Representation based on Carriers

#### A close up of a sign Description generated with very high confidence

Figure 18 Delay from ORD airport to BNA airport Figure 19 Delays Representation Based on Destination

# Conclusion

The aim of the paper is to demonstrate the concept of data lake, importance of data lake in big data era, the implementation of a data lake design, the technologies involved in data lake and the usage of data lake to solve various kind of business problems. While providing a detailed description on various technologies implemented in this project, I tried to highlight some of the challenges faced by enterprises to implement data lake and how we can use different tools to mitigate these challenges.

# Templates and Sample Code

All documents and scripts can be found on the following links:

* Documents: <https://github.com/SahilKhanna129/Datalake>
* Scripts: <https://github.com/SahilKhanna129/Datalake-dataflow-pipelines>
* Videos: <https://drive.google.com/drive/folders/0B2kOllGgl10Gc2RXVEw0bDVwNFk>

# References

* “Open Big Data”, Available: <http://www.open-bigdata.com/top-7-challenges-of-building-a-data-lake/>
* Dharnikota R. (2016), “5 Challenges to Accelerate Data Lake Adoption”, Available: <http://data-informed.com/5-challenges-to-accelerate-data-lake-adoption/>
* Hortonworks Ecosystem, Available: <https://hortonworks.com/ecosystems/>
* Apache Projects, Available: <https://projects.apache.org/projects.html?category#big-data>