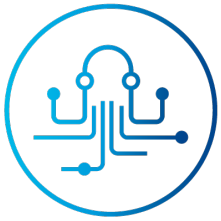

doolytic

Technical White Paper

August, 2016

Smart data discovery for the data scientist in all of us

doolytic

Contents

1 - Introduction

doolytic merges Hadoop and Elastic functionalities in a single easy-to-use graphical interface, leveraging the community efforts on both platforms, enabling Citizen Data Scientists to perform data discovery directly from their datalakes. doolytic's powerful yet flexible tools put high performance analytical and data sampling capabilities directly in the hands of users desiring to perform smart data discovery on any type of big data.

doolytic Main Features:

- Big data discovery with analytics and search functionalities on top of multiple platforms including Elastics & Hadoop Data Lakes
- Concurrent access to billions of records and lightening quick response
- Big Data blending, enrichment, aggregations on top of Hadoop Data Lakes through visual wizards
- Masks Hadoop/Spark complexity for self-service dataset creation
- Easy interactive dashboards for business users allow big data analytics distribution enterprise wide

The product is based completely on JSON, NOSQL with computation occurring on back-end nodes.

In addition to doolytic's native Web interface, a series of connectors for standard BI tools are available. All connectors function without ODBC/JDBC drivers and allow the preservation of existing BI platform investment.

2 - High level architecture

doolytic requires an Elastic cluster for big data discovery and dashboarding. Hadoop (Pig, Hive), Spark and R are not mandatory but do enable complex data manipulation, the creation of new datalake enriched datasets, massive exports, statistical analyses, etc.

Elastic main features:

- Distributed, scalable, and highly available
- Fail-safe
- Real-time search and analytics capabilities
- Sophisticated RESTful API
- Open source

Elastic is built to be always available and to scale according to needs: both scaling up and scaling out are supported.

As depicted in Figure 1, doolytic supports an end-to-end integration with Elastic, Hadoop and Spark R for performing advanced analytics computations.

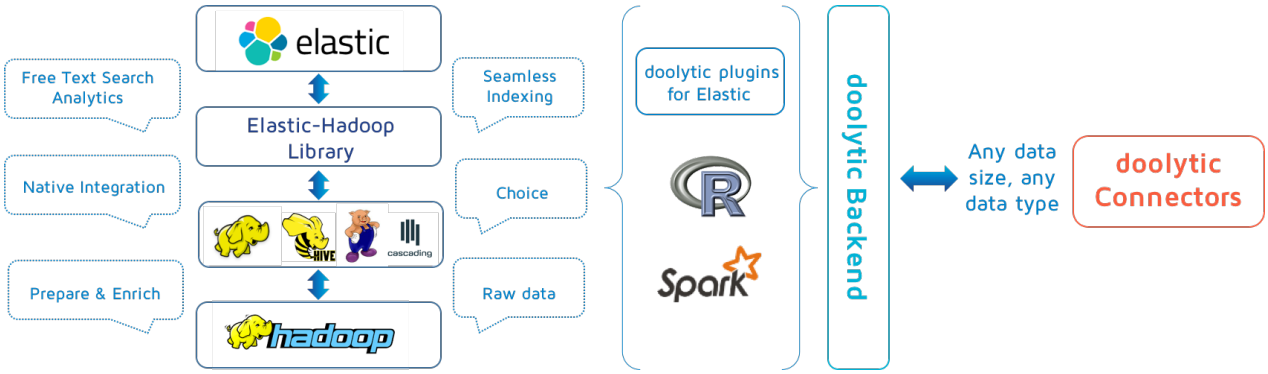

Figure 1. doolytic high level architecture

doolytic uses a Relation-Action™ approach whereby a system of link/star schematics generation between doolytic datalakes is employed. This system works in a declarative mode without generation of data in order to maintain disk space on the datalake unaltered and maintain performance levels for data access.

3 – Deployment

doolytic is designed to run on a scaleout cluster of low-cost commodity servers, which can run on premise or on public clouds (AWS EC2, Azure). doolytic can be deployed on both physical and virtual machines.

The minimum cluster configuration consists of 3 nodes. Using master election, each node is configured to be coordinator and worker with replication making the cluster failure resistant. Users can submit a query to any node

3.1 – Scenario 1

In this case, the datalake comprises an Elastic cluster, **without** Hadoop integration. Tools like Talend, Informatica, Logstash, custom programs, Message Queue systems (see Figure 2), etc. can be used for the ingestion phase.

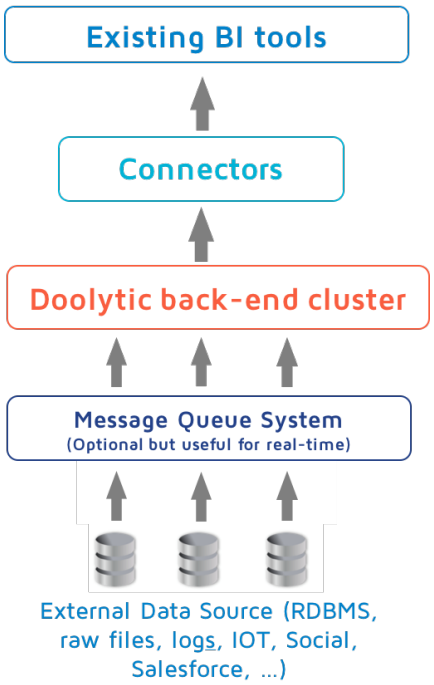

Figure 2. Data ingestion using Message Queue systems

3.2 – Scenario 2

Figure 3 depicts a scenario where the data lake consists of two clusters (namely, Elasticsearch and Hadoop with Pig, Hive and Spark). All nodes of both clusters must be co-located in the same network.

In this scenario, two kinds of hardware deployment are possible:

Both clusters share the same hardware

Advantages:
Smaller cluster size, less network traffic

Disadvantages:
Larger hardware requirements, longer recovery time, shared resources

Clusters run on different hardware

Advantages:
Fewer hardware requirements, more performance, lower recovery time

Disadvantages:
More expensive, more servers to manage

Ingestion can be managed with specific ETL tools for Hadoop such as Pig, Hive, Spark (and SparkSQL), Talend, Pentaho Data Integration, Informatica and others.

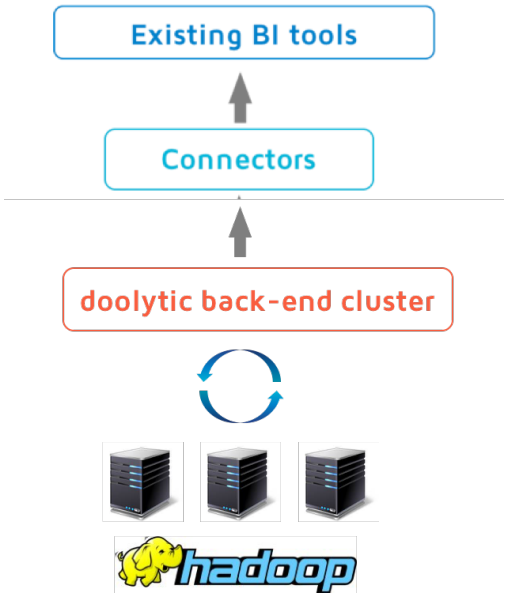

Figure 3. doolytic with Elastic and Hadoop

3.3 – Connectors

doolytic currently provides extensions for Qlik Sense (see Figure 4) and QlikView to interact with datalakes. The extensions for Qlik interact with the backend via REST APIs, offering functionality including:

- Complete data discovery on datalake datasets
- No Qlik App reload needed
- doolytic does NOT rely on QlikView Direct Discovery
- Data are NOT loaded into the Qlik memory (data processing managed by back-end nodes)
- On-the-fly aggregation and charting
- Pushing data into Qlik variables
- The Qlik "current selection" is translated into Rest API calls to retrieve data
- Qlik and doolytic models linked by field naming conventions

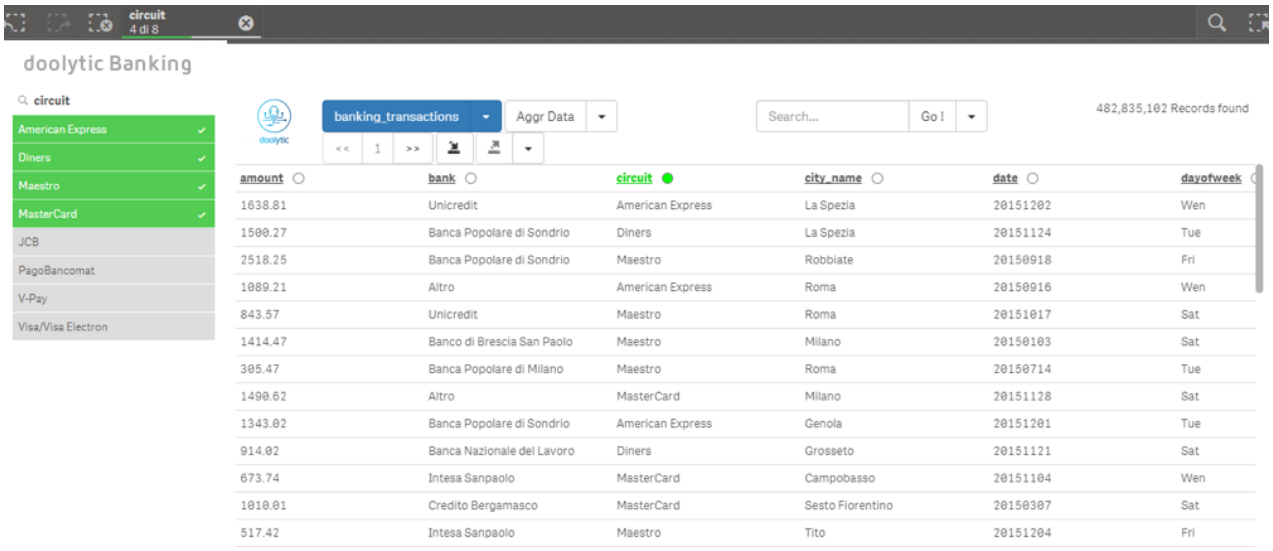

Figure 4. doolytic extension in Qlik Sense

4 – Security & Auditing

doolytic offers enhanced security featuring Single Sign-On with Kerberos and LDAP. Auditing is accomplished with complete monitoring of logs by the supervisor, who can audit:

- Record count for single operations/queries
- Time of logging for single users
- Most used query clustering for single users
- Response time analysis for each operation/query
- Real-time charting for all monitoring needs on historic data
- Scoring for data/tables interrogated by users with top classifications

5 – Additional Features

- **Query Engine** joining different wizards for the generation of advanced queries on the doolytic datalake and Hadoop
- **New Table Wizard** for creating new tables from existing tables on both Hadoop and doolytic clusters. This function permits the enrichment of datalakes through the addition of new data from any source and the possibility of inserting custom formulas
- **Join Table Wizard** for creating tables by joining existing tables with the possibility of adding, merging and stringing fields
- **Import Data Wizard** for importing data from hdfs (Hadoop) to doolytic datalake
- **Import File Wizard** for importing different file types from hdfs (Hadoop) to doolytic datalake
- **Export Wizard** for exporting from doolytic datalake to csv, xls, hdfs
- **Query Wizard** for creating SQL (SparkSql, Hive, Impala, etc) queries to run on hdfs (Hadoop) with the possibility of generating tables on hdfs and/or doolytic