****

dAnalytics

Installation Document

Release 0.1

DRAFT

**Version**: 1.0

**Last Modified:** June 24, 2015

Prepared by:

TurningPoint Global Solutions

1355 Piccard Drive, Suite 250

Rockville, MD. 20850

**APPROVALS**

The undersigned acknowledge that they have reviewed the Design Document and agree with the information presented within this document. Changes to this Design Document will be coordinated with, and approved by the undersigned, or their designated representatives.

| Signature: |  | Date: |
| --- | --- | --- |
| Print Name: | David Hong |  |
| Title: | Project Lead |  |
|  |  |  |

**REVISION HISTORY**

| **Version** | **Date** | **Organization/Point of Contact** | **Description of Changes** |
| --- | --- | --- | --- |
| 0.1 | 06/21/2015 | TurningPoint DevelopmentTeam | Initial Draft Version |
| 0.2 | 06/24/2015 |  | Updatedd based on the review with the DevOps team. |

Table of Contents

[APPROVALS i](#_Toc422950619)

[REVISION HISTORY ii](#_Toc422950620)

[1. INTRODUCTION 4](#_Toc422950621)

[1.1. Purpose 4](#_Toc422950622)

[1.2. Intended Audience 4](#_Toc422950623)

[2. Infrastructure as a Service (IaaS) 5](#_Toc422950624)

[3. Deployment on Amazon Web Services (AWS) 5](#_Toc422950625)

[4. Pre Requesties to deploy dAnalytics 6](#_Toc422950626)

[5. Installation Steps 6](#_Toc422950627)

[5.1. Installation Order 6](#_Toc422950628)

[5.2. Create dAnalytics Database 6](#_Toc422950629)

[5.3. Apache Hadoop, PIG and Scoop Configuration 7](#_Toc422950630)

[i. Hadoop Configurations 7](#_Toc422950631)

[ii. PIG and HADOOP Integration 9](#_Toc422950632)

[iii. PIG & Sqoop commands 9](#_Toc422950633)

[5.4. Rest API 9](#_Toc422950634)

[5.5. Web Application 10](#_Toc422950635)

[6. Pre Requesties Installation Steps 10](#_Toc422950636)

[6.1. MariaDB 10](#_Toc422950637)

[6.2. Apache Http Server 10](#_Toc422950638)

[6.3. Apace Hadoop (Requires Java 7) 10](#_Toc422950639)

[6.4. PIG 11](#_Toc422950640)

[6.5. Scoop 11](#_Toc422950641)

INTRODUCTION

The OpenFDA is an innovation project that aims at creating easy access to the public data, to create a new level of openness and accountability, to ensure the privacy and security of public FDA data, and ultimately to educate the public and save lives.

The FDA launched its very first openFDA challenge to the developer community to take advantage of the following datasets and explore the range and extent of its impact for 1) research and 2) consumers.

* **Adverse events data.** FDA’s publicly available drug adverse event and medication error reports, and medical device adverse event reports.
* **Recalls data.** Enforcement report data, containing information gathered from public notices about certain recalls of FDA-regulated products.
* **Labeling data.** Structured Product Labeling (SPL) data for FDA-regulated human prescription drug, OTC drug and biological product labeling.

TurnintPoint has taken up the challenge of data analytics and thus the **dAnalytics** was born.

Purpose

The Intallation Document (ID) describes the steps to install and configure dAnalytis for each of the system’s components.

Intended Audience

The intended audience for this document includes DevOps engineers and deployment team. Linux, Java and database application deployment skills. The document is made available to the open source community who can make use of the publically available code base to extened it.

1. Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) is a form of cloud computing that provides virtualized computing resources over the Internet.  dAnalytics software meets all the requirements that needs for an software application to be deployed on any of the IaaS platforms. As part of this project we have choose Amazon Web Services based services that AWS is providing and its pricing.

1. Deployment on Amazon Web Services (AWS)

For better performance and scalability dAnalytics components are deployed on different iinstances. dAnalytics software components and configured on three Elastic Compute Cloud (EC2) instances.

AWS configuration details

* **EC2 Server 1**: Apache HTTP Web Server and Rest API is deployed and configured. Web application will invoke the Rest APIs and render the results in graphical user interface representation to the end user via web browser.
* **EC2 Server 2**: Hadoop Master node is deployed and configured on this server. Hadoop nodes will be active only when there is active data load.
* **EC3 Server 3**: MariaDB and Hadoop Slave node is deployed and configured. Hadoop will be used to load the data from the external source into MariaDB.

Deployment architecture diagram.

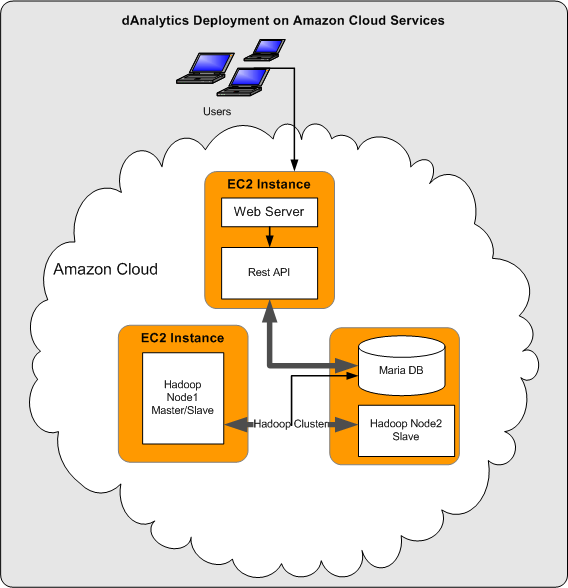


Figure 1.0

1. Pre Requesties to deploy dAnalytics

Below softwares and components are required to successfully deploy dAnalytics web application. Before installing the dAnalaytics web application, install and configure all pre requesties.

* Java – Version 1.8
* Java – Version 1.7
* Apache Hadoop
* PIG
* Apache Sqoop
* MariaDB
* Apache HTTP Server

For steps to install any of the above softwares, refer to section 5.0 Pre Requesties Installation below in the document.

Installation Steps

Download dAnalytics binary from the github using the below link, extract the contents of danalayatics.zip. The below steps would need these files to be moved the target servers based on the components being installed and configured.

<https://github.com/TPRockville/dAnalytics/binary>

Installation Order

* 1. Create dAnalytics Database
  2. Apache Hadoop, PIG and Scoop Configuration
  3. Run PIG commands
  4. Run Scoop commands
  5. Deploy and configure Rest API
  6. Deploy and configure Web Application

Create dAnalytics Database

Execute the below scripts on MariaDB to create the database schema for dAnalytics.

* Execute drugSummary.sql for creating the database and run required (DDL) Data Defination Language statements

.

$cat drugSummary.sql |mysql

* Execute populateDrugPeaks.sql to create the poocedure which will be used later.

$cat populateDrugPeaks.sql |mysql

Apache Hadoop, PIG and Scoop Configuration

Hadoop Configurations

* Configure the Hadoop environment
  + vi /opt/hadoop/conf/hadoop-env.sh
    - Add export JAVA\_HOME and set it to JDK home.
    - Add export HADOOP\_OPTS=-Djava.net.preferIPv4Stack=true.
* Configure the core-site.xml file in the conf folder.
  + vi /opt/hadoop/conf/core-site.xml
  + Add the below configuration

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://<hostname>:9000</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>

* Configure the hdfs-site.xml file in the conf folder.
  + vi /opt/hadoop/conf/hdfs-site.xml
  + Add the below configuration

<configuration>

<property>

<name>dfs.data.dir</name>

<value>/opt/hadoop-data/dfs/data</value>

</property>

<property>

<name>dfs.name.dir</name>

<value>/opt/hadoop-data/dfs/name</value>

</property>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>

* Configure the mapred-site.xml file in the conf folder.
  + vi /opt/hadoop/conf/mapred-site.xml
  + Add the below configuration

<configuration>

<property>

<name>mapred.job.tracker</name>

<value><Master HostName>:9001</value>

</property>

<property>

<name>mapred.tasktracker.map.tasks.maximum</name>

<value>4</value>

</property>

</configuration>

* Configure on which machine your namenode/jobtracker has to run on.
  + vi /opt/hadoop/hadoop/conf/masters
    - <hostname>
* Configure on which machine your datanode/tasktracker has to run on.
  + vi /opt/hadoop/hadoop/conf/slaves
    - <hostname>
* Now export HADOOP\_HOME=<install root of hadoop> and PATH=$PATH:$HADOOP\_HOME/bin

And export HADOOP\_COMMON\_HOME=<Install directory>

* Change the ownership of the Hadoop folder, if the username is not root.
  + chown –R <username> /opt/hadoop
* Copy the /opt/hadoop folder to the machines where you want to configure data nodes.
  + scp –r /opt/hadoop username@ip:/opt/Hadoop
* Format Namenode
  + /opt/hadoop/bin/hadoop namenode –format
* Start and Stop Hadoop
  + start-all.sh
  + stop-all.sh

PIG and HADOOP Integration

* Add the below in ~/.bash\_rc
  + export HADOOP\_PREFIX=<Root directory of Hadoop>
  + export HADOOP\_CONF\_DIR=<Conf directory of Hadoop>

PIG & Sqoop commands

* Put the input csv file into hdfs using hadoop fs –put <<INPUT.CSV>> . This will put the INPUT.CSV into /user/root/<<INPUT.CSV>> in hdfs.
* Copy the Pig folder from <https://github.com/TPRockville/dAnalytics/tree/master/Engg/Src/Batch/>, place it in any folder (user root as Hadoop in configured).
* Compile the <https://github.com/TPRockville/dAnalytics/tree/master/Engg/Src/Batch/Pig/PigUdf> and place the jar in the same folder above. I am attaching the compiled jar for your convenience.
* Run the shell <https://github.com/TPRockville/dAnalytics/blob/master/Engg/Src/Batch/Pig/pigscripts.sh> with parameter as path of the input csv in hdfs as /user/root/<<INPUT.CSV>>
* Run sqoop commands from the attached file. We need to change the mariadb configuration for each script.

Example:

sqoop export --connect jdbc:mysql://192.168.20.50:3306/drug\_complete\_ds --username root --password root123 --table drug\_list --columns "drug\_id,drug\_name"  --export-dir /user/hadoop/CompleteDS/druglist/part-\* --verbose

Run Spike Identification script

* Connect to MariaDB

$mysql

* Choose drug\_summary database.

$ use drug\_summary

* Execute stored procedure populate find spikes and populate spike data.

$ run populateDrugBasedSpike(‘5’);

Rest API

dAnalytics Rest API are built on Spring Boot. Spring Boot has embedded Tomact Server and this requires Java 8.

1. Copy openfda-service.tar from the downloaded binary file.
2. Transfer openfda-service.tar to the target server
3. Unzip the contents in openfda-service.tar on the target server
4. Configuring the Rest API
   1. Navigate to conf folder (under openfda-service) to update application.properties
      1. Register the http port under server.port

Example: server.port=8080

* + 1. Register Server IP/Domain name server.address

Example: server.address= localhost

* + 1. Register the JDBC parameters spring.datasource.url=jdbc:mariadb://<db\_server\_name>:<port>/<schema\_name>?user=<username>&password=<password>&useUnicode=true&characterEncoding=UTF8
  1. Navigate to **bin** folder under **openFDA-service** directory, and execute

$nohup openfda-service &

1. Verify if the deployment is successful by invoking the Healthcheck endpoint. Click on the link or copy and paste the link on the web browser.

http://<server.address>:<server.port>/jderive/health.

Web Application

1. Copy danalyticsapp.tar from the downloaded binary file.
2. Transfer danalyticsapp.tar to the target server
3. Unzip the contents in danalyticsapp.tar on the target web server.

Example : Copy the extracted contents from danalyticsapp.tar to /var/www/ folder.

1. Restart Apache Server (sudo service httpd restart).

Pre Requesties Installation Steps

MariaDB

* Create MariaDB.repo under /etc/yum.repos.d/
* Edit MariaDB.repo and copy the below content, save and exit.

[mariadb]

name = MariaDB

baseurl = http://yum.mariadb.org/10.0/centos7-amd64

gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB

gpgcheck=1

* Execute $sudo yum install mariadb

Apache Http Server

Run the below command to install Apache HTTP Server and SSL components to configure web requests to listen on secured socket.

* Execute **$sudo yum install httpd**
* Execute **$sudo yum install ssl**

Apace Hadoop (Requires Java 7)

* Download Hadoop 1.2.1 tar file
  + wget <http://www.webhostingreviewjam.com/mirror/apache/hadoop/common/hadoop-1.2.1/hadoop-1.2.1-bin.tar.gz>
* Extract to /opt/hadoop location
  + tar -xzf hadoop-1.2.1.tar.gz –C /opt/hadoop
* Set a password-less login to all the machines
  + Generate an RSA key for that user
    - ssh-keygen -t rsa -P ""
  + Copy the generated keys to authorized keys
    - cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys
  + Give Execute permission to the authorized\_keys
    - chmod 755 $HOME/.ssh/authorized\_keys
  + Check whether it is able to login without password
    - ssh localhost
  + Copy the generated keys to other machines where password-less login is required
    - ssh-copy-id -i $HOME/.ssh/id\_rsa.pub username@ip

PIG

* Download pig-0.11.1.tar file

wget http://archive.apache.org/dist/pig/pig-0.11.1/pig-0.11.1.tar.gz

* Extract to /opt/pig location
  + tar -xzf pig-0.11.1.tar.gz –C /opt/pig
* Now export PIG\_HOME=<install root of pig> and PATH=$PATH:$PIG\_HOME/bin
* To run Pig
  + Local mode
    - pig –x local
  + MapReduce mode
    - pig or pig –x mapred

Scoop

* Download sqoop-1.4.6 file
  + wget http://mirror.nexcess.net/apache/sqoop/1.4.6/sqoop-1.4.6.bin\_\_hadoop-1.0.0.tar.gz
* Extract to /opt/sqoop location
  + tar -xzf sqoop-1.4.6.bin\_\_hadoop-1.0.0.tar.gz –C /opt/
* Change the user permissions and owner if it’s not the same
  + sudo chmod -R 755 sqoop-1.4.6.bin\_\_hadoop-1.0.0/
  + sudo chown -R <username> sqoop-1.4.6.bin\_\_hadoop-1.0.0/
* Now export SQOOP\_HOME=<install root of sqoop> and PATH=$PATH:$SQOOP\_HOME/bin