****

dAnalytics

Deployment Document

Release 0.1

**Version**: 1.0

**Last Modified:** July 6, 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. |
| 0.3 | 06/25/2015 |  | Updated David Hongs comments |
| 0.4 | 06/25/2015 |  | Included Github link to download dAnalytics binary files. |
| 0.5 | 06/29/2015 |  | Included steps to deploy from source file. |
| 0.6 | 07/06/2015 |  | Included steps to deploy from docker hub. |

Table of Contents

[APPROVALS i](#_Toc423947696)

[REVISION HISTORY ii](#_Toc423947697)

[1. INTRODUCTION 5](#_Toc423947698)

[1.1. Purpose 5](#_Toc423947699)

[1.2. Intended Audience 5](#_Toc423947700)

[2. Platform 6](#_Toc423947701)

[3. Deployment on Amazon Web Services (AWS) 6](#_Toc423947702)

[3.1. AWS configuration details 6](#_Toc423947703)

[3.2. Deployment architecture diagram. 7](#_Toc423947704)

[4. Pre Requesties to deploy dAnalytics 7](#_Toc423947705)

[4.1. Pre Requesites List 8](#_Toc423947706)

[4.2. Pre Requesties Installation Steps 8](#_Toc423947707)

[i. MariaDB 8](#_Toc423947708)

[MariaDB a opensource relational database is used to store the summarized data using a start schema to provide the data through Rest APIs and for dAnalytics UI. 8](#_Toc423947709)

[ii. Apache Http Server 8](#_Toc423947710)

[iii. Apace Hadoop (Requires Java 7) 9](#_Toc423947711)

[iv. Apache PIG 9](#_Toc423947712)

[PIG is used for data summarize the adverse report data and store in a start schema for adverse report analysis. 9](#_Toc423947713)

[v. Apache Scoop 10](#_Toc423947714)

[Apache Scoop is used to efficiently transfer the summarized data into the mariaDB relational database. 10](#_Toc423947715)

[5. Installation Steps 10](#_Toc423947716)

[5.1. Installation Order 11](#_Toc423947717)

[5.2. Creating dAnalytics Database schema 11](#_Toc423947718)

[5.3. Apache Hadoop, PIG and Scoop Configuration 11](#_Toc423947719)

[i. Hadoop Configurations 11](#_Toc423947720)

[ii. PIG and HADOOP Integration 13](#_Toc423947721)

[iii. Download Data from FDA Adverse Event Reporting System (FAERS) 13](#_Toc423947722)

[Download the quarterly data files provide by FEARS. The files listed on FAERS are raw data extracted in XML format. Below is the link to download the AdverseDrug Effects data. 13](#_Toc423947723)

[http://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/ucm082193.htm 13](#_Toc423947724)

[iv. PIG & Sqoop commands 14](#_Toc423947725)

[5.4. Rest API 14](#_Toc423947726)

[i. Deploy as container from Docker hub 14](#_Toc423947727)

[ii. Deploy using the binaries downloaded from Github 15](#_Toc423947728)

[5.5. Web Application 15](#_Toc423947729)

[i. Deploy as container from Docker hub 15](#_Toc423947730)

[ii. Deploy using the binaries downloaded from Github 15](#_Toc423947731)

[6. Sanity Test 16](#_Toc423947732)

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.

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 withLinux, 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. Platform

Infrastructure as a Service (IaaS) platform, a form of cloud computing that provides virtualized computing resources over the Internet is the platform of choice for the dAnalytics project. IaaS platform provides the performance and scalabitly that meet dAnalytics requirements for current and future needs. In particular, the dAnalytics project will be deployed in the Amazon Web Services based services.

1. Deployment on Amazon Web Services (AWS)

For better performance and scalability dAnalytics components are deployed on three different Servers. AWS servers are called Elastic Compute Clound (EC2) instances, listed below are the details of the configuration of the 3 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.

|  |  |  |
| --- | --- | --- |
| Server Name | Hardware Configuration | Software Components deployed and configured |
| EC2 Server 1  (Presentation Layer) | t2.micro  100GB disk space  2 GB RAM | Apache HTTP Web Server  Apache Tomcat Web Container |
| EC2 Server 2  (Business Layer) | m3.xlarge  250GB disk space  8GB RAM | Apache Hadoop 1.2 |
| EC3 Server 3  Persistent Data Layer | m3.xlarge  250GB disk space  8GB RAM | MariaDB 10.0  Apache Hadoop 1.2 |

Deployment architecture diagram.

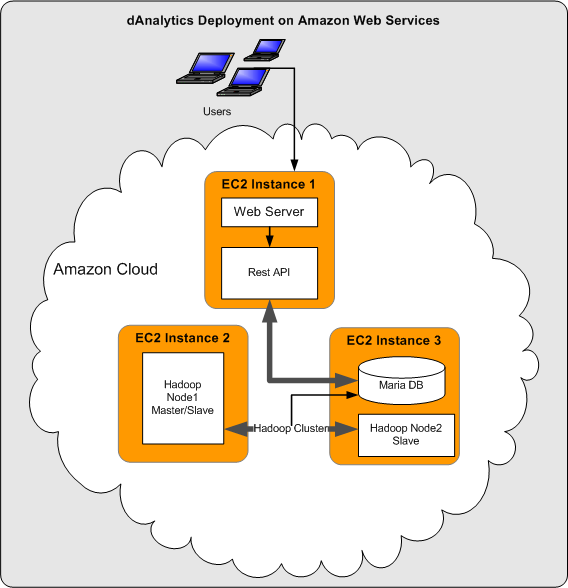


Figure 1.0

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 requesites.

Pre Requesites List

|  |  |  |
| --- | --- | --- |
| Software | Software Version | Required on Server(s) |
| Java | 1.8 | EC2 Server 1 |
| Java | 1.7 | EC2 Server 2 and EC2 Server 3 |
| Apache Hadoop | 1.2 | EC2 Server 2 and EC2 Server 3 |
| Apache Pig |  | EC2 Server 2 and EC2 Server 3 |
| Apache Scoop |  | EC2 Server 2 and EC2 Server 3 |
| Maria DB | 10.0 | EC2 Server 2 |
| Apache HTTP Server | 2.x | EC2 Server 1 |

Pre Requesties Installation Steps

MariaDB

MariaDB a opensource relational database is used to store the summarized data using a start schema to provide the data through Rest APIs and for dAnalytics UI.

* 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)

Apache Hadoop is used to process the large set of adverse reports data that is provided by FDA into dAnalytics relational database.

Follow the below instructions to download and install Apache Hadoop.

* 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

Apache PIG

PIG is used for data summarize the adverse report data and store in a start schema for adverse report analysis.

Follow the below instructions to download and install 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

Apache Scoop

Apache Scoop is used to efficiently transfer the summarized data into the mariaDB relational database.

Follow the below instructions to download and install 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

Installation Steps

Download dAnalytics binary from the github using the below link, extract the contents of danalytics-full.tar. 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/tree/master/Engg /Release/binaries/danalytics.tar](https://github.com/TPRockville/dAnalytics/tree/master/Engg%20/Release/binaries/danalytics.tar)

Extract danalytics.tar on your local server, it will have the below components in it.

|  |  |
| --- | --- |
| Rest API | danalytics-api.tar |
| Web application | danalytics-ui.tar |
| Data Processor | danalytics-dataprocessor.tar |
| Database scripts | danalytics-dbscripts.tar |

Installation Order

All the below steps needs to be executed to have a fully functional dAnalytics. Hadoop Components are used to prepare and load the data into MariaDB. These services will be in passive mode when there data load is not happening.

* 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

\*\* To deploy from dAnalytics source files, execute steps 1 to 4 of the above installation order and follow the below instructions (Skip Step 5 and 6 of above installation order).

1. Follow the instructions of <dAnalytics Home>/Engg/Src/REST/README.md to Build and Deploy dAnalytics REST API.
2. Follow the instruction of <dAnalytics Home>/Engg/Src/UI/README.md to Build and Deploy dAnalytics Web Application.

Creating dAnalytics Database schema

1. Transfer danalytics-api.tar to the target server
2. extract the contents of danalytics-api.tar on the target server
3. Execute the below scripts on MariaDB dAnalytics.

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

$ cat drugSummary.sql | mysql

* Execute populateDrugPeaks.sql to create the stored procedure.

$ 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>

Download Data from FDA Adverse Event Reporting System (FAERS)

Download the quarterly data files provide by FEARS. The files listed on FAERS are raw data extracted in XML format. Below is the link to download the AdverseDrug Effects data.

<http://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/ucm082193.htm>

By using dAnalytics “Data Processor” convert the the XML file to CSV format. Below are the steps to convert the XML files.

Extract the danalytics-dataprocessor.tar

Execute the below java comman to convert the files, Argument 1 : Name of CSV output file name and Argument 2 : the xml file that neds to converted

java -cp :data-loader.jar:/home/hadoop/data\_loader/csvgenerator/lib/\*com.tpgsi.parser.XMLParser <Argument 1 : CSV file> <Argument 2 : XML file>

PIG & Sqoop commands

Copy the files from that was converted from XML file s in step 5.3 I as use the below command to load the files.

* 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

Rest API

Deploy as container from Docker hub

1. Pull the image from Docker Hub to a machine having docker deamon running using the below command.

$docker pull tprockville/da-rest:latest

1. Check the images list in the docker instance by using the below command. Note the <IMAGE ID>

$docker images

1. Check the process to see if tprockville/da-rest:latest is running already by using the below command. Make a note of the <CONTAINER ID>

$docker ps -as

1. Kill the process incase it is already running by using the below command.

$docker kill -f <CONTAINER ID>

1. Remove running process from the docker insatance, if required (optional)

$docker rm -f <CONTAINER ID>

1. Run the image to create a container

$docker run -it -d -p 8080:8080 tprockville/da-rest:latest

1. Check the logs. This will get into virtual container bash.

$docker exec -it <CONTAINER ID> bash

$cat ./logs/jDerive.log

Deploy using the binaries downloaded from Github

Folow the below steps to install dAnalytics Rest API component.

1. Transfer danalytics-api.tar to the target server
2. extract the contents of danalytics-api.tar on the target server
3. 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 &

Web Application

Deploy as container from Docker hub

1. Pull the image from Docker Hub to a machine having docker deamon running using the below command.

$docker pull tprockville/da-ui:latest

1. Check the images list in the docker instance by using the below command. Note the <IMAGE ID>

$docker images

1. Check the process to see if tprockville/da-ui:latest is running already by using the below command. Make a note of the <CONTAINER ID>

$docker ps -as

1. Kill the process incase it is already running by using the below command.

$docker kill -f <CONTAINER ID>

1. Remove running process from the docker insatance, if required (optional)

$docker rm -f <CONTAINER ID>

1. Run the image to create a container

$docker run -it -d -p 8080:8080 tprockville/da-ui:latest

Deploy using the binaries downloaded from Github

Folow the below steps to install dAnalytics UI component on web server (Apache HTTP Server).

1. Transfer danalytics-ui.tar to the target server
2. Extract the contents of danalytics-ui.tar on the “Document Root” folder of apache http server.

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

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

Sanity Test

Execute the below steps to check if the dAnalytics application is up and running.

1. Open a browser and access the dAnalytics web application, http(s)://<server.address>:<server.port>/index.html
2. The browser would show the Adverse Event graph.
3. Select a drug in the filter critera section and submit the web request.
4. The graph on the browser should plot the adverse events for the selected drug.