INSTALLING APACHE ZEPPELIN* ON CLOUDERA DISTRIBUTION OF HADOOP*

Authors:

Karthik Vadla (Big Data Solutions / Intel Corporation)

Abhi Basu (Big Data Solutions / Intel Corporation)

CONTENTS

EXECUTIVE SUMMARY

Data science is not a new discipline. However, with the growth of big data and adoption of big data technologies, the request for better quality data has grown exponentially. Today data science is applied to every facet of life—product validation through fault prediction, genome sequence analysis, personalized medicine through population studies and Patient 360 view, credit card fraud-detection, improvement in customer experience through sentiment analysis and purchase patterns, weather forecast, detecting cyber or terrorist attacks, aircraft maintenance utilizing predictive analytics to repair critical parts before they fail, and many more. Every day, data scientists are detecting patterns in data and providing actionable insights to influence organizational changes.

The data scientist's work broadly involves acquisition, cleanup, and analysis of data. Being a cross-functional discipline, this work involves communication, collaboration, and interaction with other individuals, internal and possibly external to your organization. This is one reason why the "notebook" features in data analysis tools are gaining popularity. They ease organizing, sharing, and interactively working with long workflows. IPython* Notebook is a great example but is limited to usage of Python* language. Apache Zeppelin* is a new web-based notebook that enables data-driven, interactive data analytics, and visualization with the added bonus of supporting multiple languages, including Python*, Scala*, Spark SQL, Hive*, Shell, and Markdown. Zeppelin also provides Apache Spark* integration by default, making use of Spark's fast in-memory, distributed, data processing engine to accomplish data science at lightning speed.

In this paper we describe how to install and configure Apache Zeppelin on the Cloudera Distribution of Apache Hadoop*, providing access to Hadoop and Spark.

INTRODUCTION

This paper demonstrates how easy it is to install Apache Zeppelin* [1] notebook, which is a web-based editor on Cloudera Distribution of Apache Hadoop* [2] and perform data analytics using different interpreters such as Spark* [3], Hive* [4], SQL, and more.

OBJECTIVES

Specifically, the following are the key objectives of this paper:

1. Define the operating system, software stack, and tools.
2. Define the setup and configuration of Apache Zeppelin on a multi-node Hadoop cluster.
3. Run some tests from Zeppelin notebook to validate that the installation of all components was successful.

AUDIENCE

Software developers and technologists can use this document to install the above software as a proof point. We assume familiarity with Linux* (especially CentOS*) commands, installation, and configuration in this document.

SYSTEM SETUP AND CONFIGURATION

Components

Listed below are the specs of our test Hadoop cluster.

HARDWARE (7 NODE CLUSTER) COMPONENTS

Component Type	Part Name	Qty	Installed
Chassis	Chassis-2U Wildcat Pass Black Intel® R2000WTXXX	1	✓
Motherboard	Wildcat Pass Server Board Intel® S2600WT2 1Gb Ethernet ports	1	✓
CPU	Intel® Xeon® DP E5-2699 v3 (code name Haswell-EP) LGA2011 2.3GHz 45MB 145W 18 cores CM8064401739300	2	✓
Heatsink	Included	2	✓
Memory	16GB 2133 Reg ECC 1.2V DDR4 RDIMM Crucial* CT8G4RFS4213 ← 192GB	12	✓
ATA Hard Drive	2TB HDD 3.5in SATA 7200RPM 128MB Seagate* Constellation ES.3 ST2000NM0033 ← 4TB	12	✓
ATA Hard Drive	800GB SSD 2.5in SATA 6Gb/s Intel® SSDSC2BA800G301 DC S3700 Series 7mm	1	✓
Optical Drive	DVDRW – SATA Slim Black Intel® AXXSATADVDRWROM	1	✓
Network Adapter	NIC – 10 Gbe RJ-45 Server I/O Module Dual Port Intel® AXX10GBTWLIOM	1	✓
Network Adapter	On Board	2	✓
Chassis Component	HDD Bay – 2U Hot-Swap Drive Cage Upgrade Kit 8x3.5in A2U8X35S3HSDK	1	✓
Power Supply	Power Supply - 1100W Common Redundant Platinum Efficiency Intel® AXX1100PCRPS	2	✓
Cable	Cable – (2) 950mm Straight SFF8643 Connectors Intel® AXXCBL950HDHD	1	✓
Power Cord	Power Cord – 6ft 14AWG 15A w/3 Conductors (C13/5-15P) Black Monoprice 5292	2	✓
Cable	Cable - (2) 800mm Straight SFF8643 To Straight SFF8643 Connectors Intel® AXXCBL800HDHD	1	✓
Chassis Component	PCIE Riser- 2U Intel® A2UL8RISER2	2	✓

Network Component	Expansion Module LSI Logic Corp 9300-8I SATA/SAS Controller 2 LSI00344	1	✓

SOFTWARE COMPONENTS

Component	Distribution and Version Details
Operating System	CentOS* 6.6 (Developers Workstation version)
Application Software	CDH 5.4.0 (Apache Hadoop* 2.6.0) Apache Spark* 1.3.0
Apache Zeppelin*	Version 0.5

Software Requirements

These installation commands are specific to CentOS (https://www.centos.org/download/). If you do not login as 'root', you must use 'sudo' for all the commands.

- Update CentOS packages (`yum update`)
- Install latest version of Java*, preferably version 1.7 or later (`yum install java-1.8.0-openjdk-devel`)
- Install Git (`yum install git`)
- Install Node.js* and npm (`yum install nodejs npm`)
- Bower (is installed by npm)
- Install Apache Maven* [5] - refer to these steps for installation.

Important Note: When you are working in a corporate environment, you need to set the proxies for Git, Nnpm, and Bower individually along with Maven.

Setting Proxies

- For Git

  git config --global http.proxy http://your.company.proxy:port

  git config --global https.proxy http://your.company.proxy:port

- For npm
  npm config set proxy http://your.company.proxy:8080
  npm config set https-proxy http://your.company.proxy:8080

- For Bower
  Create a file: nano ~/.bowerrc

```
{
        "proxy":"http ://<host>:<port>",
        "https-proxy":"http ://<host>:<port>"
}
```

Building Zeppelin Binaries

- Download and extract the latest version of Apache Zeppelin from GitHub [6]

- Now cd to /incubator-zeppelin-master
- The current versions of CDH, Hadoop, and Spark are:
  - CDH 5.4.0
  - Spark 1.3.0
  - Hadoop 2.6.0
- Maven command to build the Zeppelin (locally):

  ```
  mvn clean package -Pspark-1.3 -Ppyspark -Dhadoop.version=2.6.0-cdh5.4.2 -Phadoop-2.6 –DskipTests
  ```

  OR

  Maven command to build the Zeppelin for YARN (All spark queries are tracked in Yarn history):

  ```
  mvn clean package -Pspark-1.3 -Ppyspark -Dhadoop.version=2.6.0-cdh5.4.2 -Phadoop-2.6 -Pyarn –DskipTests
  ```

  Profiles included:

  Pspark-1.3: Installs spark framework support for Zeppelin

  Ppyspark: Installs all configurations required to run pyspark interpreter in Zeppelin

  Phadoop-2.6: Installs Hadoop version support for Zeppelin

Once the build is successful, continue with the configuration.

General Configuration of Zeppelin

- To access the Hive metastore, copy the hive-site.xml from HIVE_HOME/conf into ZEPPELIN_HOME/conf folder (where HIVE_HOME and ZEPPELIN_HOME refers to the install locations of this software).
- In ZEPPELIN_HOME/conf folder duplicate **zeppelin-env.sh.template** and rename it to **zeppelin-env.sh**
- In ZEPPELIN_HOME/conf folder duplicate **zeppelin-site.xml.template** and rename it to **zeppelin-site.xml**

Yarn Configuration of Zeppelin

If you have built binaries for yarn, set the master property for the Spark interpreter, i.e., **master=yarn-client** via Zeppelin UI (Interpreter tab)

- In Zeppelin /conf directory go to **zeppelin-env.sh** file, uncomment the export HADOOP_CONF_DIR and specify the configuration directory location of the yarn-site.xml file (e.g., export HADOOP_CONF_DIR =/etc/hadoop/conf).

Start Zeppelin

```
./bin/zeppelin-daemon.sh start
```

Note: Sometimes you may not be able to run the above command. In that case, make all scripts in /bin folder executable with the following command:

```
chmod –R 777 .
```

After this try the previous command againto start Zeppelin.

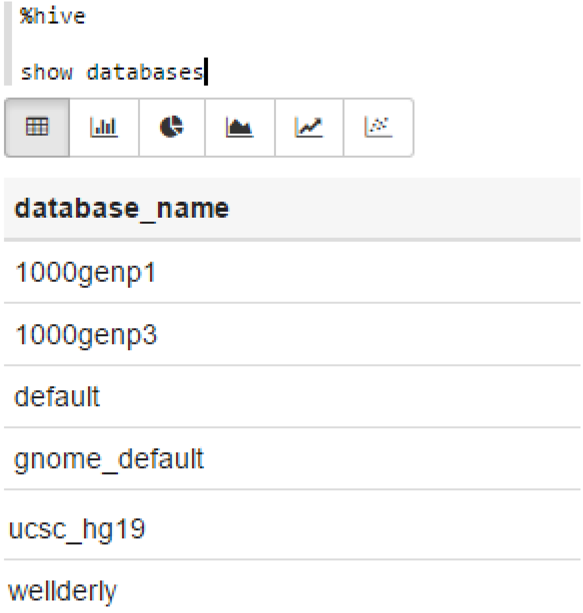
And now you can access your notebook at http://localhost:8080 or http://host.ip.address:8080

Stop Zeppelin

```
./bin/zeppelin-daemon.sh stop
```

TESTING

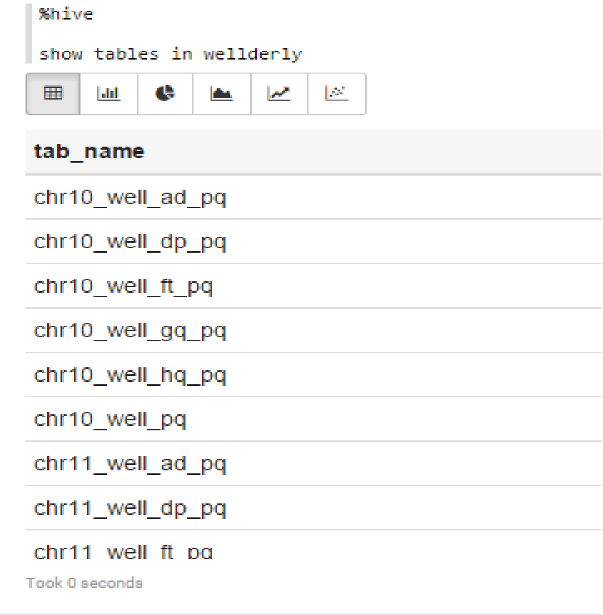
1. Start the Zeppelin application:
   ./bin/zeppelin-daemon.sh start and access **http://localhost:8080** (or IP address of the node it is installed on)
2. If you already have data on the Apache Hive metastore, which is accessible via **hive** commands locally, let's test Zeppelin commands. Use the %hive interpreter to access the Hive metastore and list all available databases. In this example we already have some public genome databases available in our Hive metastore. If you do not have any data in your Hive metastore, you may want to load some data before starting this test or skip to step 4. Now, type these commands in notebook:
   %hive
   show databases
   The code snippet is echoed back and the code execution output is displayed:



```
%hive

show databases
```

database_name
1000genp1
1000genp3
default
gnome_default
ucsc_hg19
wellderly

3. To display tables in a specific database, such as“**wellderly, type these commands in the notebook**
   %hive
   show tables in wellderly
   Again, the code snippet is echoed back and the code execution output is displayed:



```
%hive

show tables in wellderly
```

tab_name
chr10_well_ad_pq
chr10_well_dp_pq
chr10_well_ft_pq
chr10_well_gq_pq
chr10_well_hq_pq
chr10_well_pq
chr11_well_ad_pq
chr11_well_dp_pq
chr11_well_ft_pq

Took 0 seconds

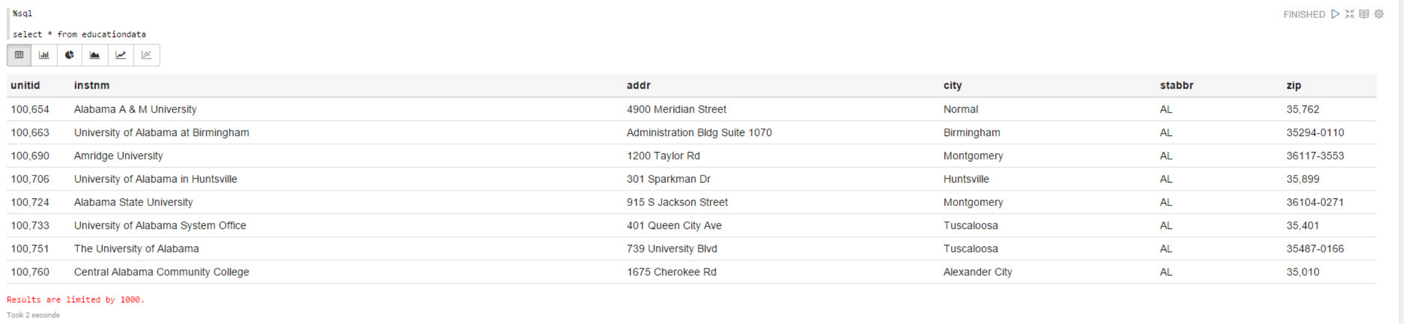
4. Download the test dataset education.csv (http://inventory.data.gov/dataset/032e19b4-5a90-41dc-83ff-6e4cd234f565/resource/38625c3d-5388-4c16-a30f-d105432553a4) and place it in your HDFS [7] location. Using the Scala interpreter register a table using the .csv file in HDFS. Use the code snippet to register the table. Note: Scala interpreter is the default, so nothing needs to be specified in Zeppelin (like %hive) when using Hive.

```
val eduText = sc.textFile("hdfs://your.ip.address /user/hadoop/education.csv")

case class Education(unitid : Integer, instnm : String, addr : String, city : String,
                     stabbr : String, zip : String)

val education = eduText.map(s=>s.split(",")).filter(s=>s(0)!="UNITID").map(
    s=>Education(s(0).toInt,
            s(1).replaceAll("\"", ""),
            s(2).replaceAll("\"", ""),
            s(3).replaceAll("\"", ""),
            s(4).replaceAll("\"", ""),
            s(5).replaceAll("\"", "")
        )
)

// Below line works only in spark 1.3.0.
// For spark 1.1.x and spark 1.2.x,
// use bank.registerTempTable("educationdata") instead.
education.toDF().registerTempTable("educationdata")
```

After that, run the command below:

%sql

select * from educationdata



```
%sql
select * from educationdata
```

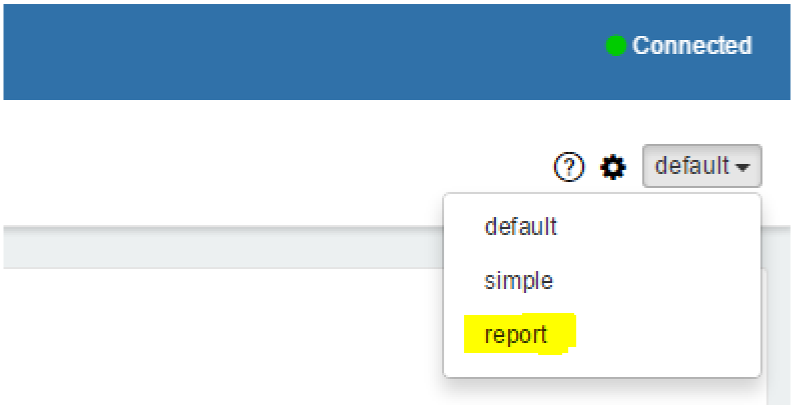
unitid	instnm	addr	city	stabbr	zip
100,654	Alabama A & M University	4900 Meridian Street	Normal	AL	35,762
100,663	University of Alabama at Birmingham	Administration Bldg Suite 1070	Birmingham	AL	35294-0110
100,690	Amridge University	1200 Taylor Rd	Montgomery	AL	36117-3553
100,706	University of Alabama in Huntsville	301 Sparkman Dr	Huntsville	AL	35,899
100,724	Alabama State University	915 S Jackson Street	Montgomery	AL	36104-0271
100,733	University of Alabama System Office	401 Queen City Ave	Tuscaloosa	AL	35,401
100,751	The University of Alabama	739 University Blvd	Tuscaloosa	AL	35487-0166
100,760	Central Alabama Community College	1675 Cherokee Rd	Alexander City	AL	35,010

```
Results are limited by 1000.
Took 2 seconds
```

You now have installed and configured Zeppelin correctly and you have been able to test the installation successfully. Documentation for Zeppelin is available here: https://zeppelin.incubator.apache.org/docs/index.html.

SHARING A NOTEBOOK

1. If you want to share these notebook results with another user, you can simply send the URL of your notebook to that user. (That user must have access to the server node and cluster on which you created your notebook). That user not only can view all your queries, but also run all your queries to view your queries' results.
2. If you want to share only the results without any queries (report-mode), please follow these steps:
    a. Go to right corner of the Zeppelin window, where you see a dropdown list after the settings icon.
    b. Change it from default to report. In this mode only results can be viewed without queries.
    c. Copy the URL and share with others (who have access to the server node and cluster).



    d. As the above image shows, three modes are available to share your notebooks:
        i. Default – In this mode, the notebook can be edited by anyone who has access to the notebook (edit queries and re-run to display different results).
        ii. Simple – This mode is similar to default, the only difference is that all the available options are invisible. Options are visible only when you hover your mouse over the cell. This mode gives a cleaner view of the results when shared.
        iii. Report – When this mode is enabled, only the final results are visible (read only). The notebook cannot be edited.

CONCLUSION

Clearly, Apache Zeppelin is in the Incubator stage, but it does show promise as a cross-platform notebook not tied to a particular platform, tool, or programming language. Our intent here was to demonstrate how you can install Apache Zeppelin on your own system and start experimenting with its many capabilities. In the future, we want to use Zeppelin for exploratory data analysis and also write more interpreters for it to improve the visualization capability, i.e., incorporate Google Charts and similar tools.

THANK YOU!

REFERENCES

1. Apache Zeppelin. Accessed from https://zeppelin.incubator.apache.org/ on June 7, 2015.
2. Cloudera Distribution of Apache Hadoop. Accessed from http://www.cloudera.com/content/cloudera/en/downloads/cdh/cdh-5-4-0.html on June 7, 2015.
3. Apache Spark. Accessed from https://spark.apache.org/ on June 7, 2015.
4. Apache Hive. Accessed from https://hive.apache.org/ on June 5, 2015.
5. Apache Maven. Accessed from http://xmodulo.com/how-to-install-maven-on-centos.html on June 5, 2015.
6. GitHub. Accessed from https://github.com/ on June 5, 2015.
7. HDFS. Accessed from https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html on June 5, 2015.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at **intel.com.**

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.