

IT5384

Lưu trữ và phân tích dữ liệu lớn

IT5384

09/2020
Thanh-Chung Dao Ph.D.

1

Agenda



Zeppelin notebook

What and why we need it?
Installation using Docker
Usage



Load, inspect, and save data

Loading data from different sources
Simple inspecting commands
Saving data

2

2

Zeppelin notebook

- A web-based interface for interactive data analytics
 - Easy to write and access your code
 - Support many programming languages
 - Scala (with Apache Spark), Python (with Apache Spark), SparkSQL, Hive, Markdown, Angular, and Shell
 - Data visualization
- Monitoring Spark jobs

3

3

Installation using Docker

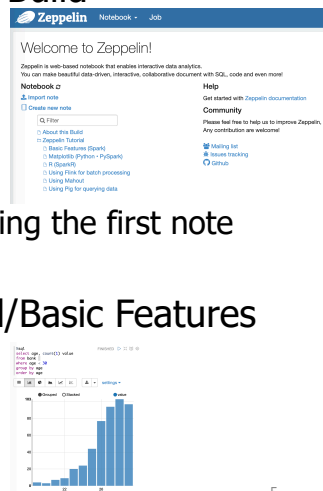
- Install Docker and login
 - <https://docs.docker.com/docker-for-windows/install/>
 - <https://docs.docker.com/docker-for-mac/install/>
- Download lecture's git repository
 - <https://github.com/bk-blockchain/big-data-class>
- Run Zeppelin using docker-composer
 - docker-compose up -d --build spark_master
 - <http://localhost>

4

4

Zeppelin usage

- Run the first node: "About this Build"
 - Check Spark version
- Check Spark running mode
 - <http://localhost:4040>
 - Need to start Spark first by running the first note
- Run the second node: "Tutorial/Basic Features (Spark)"
 - Load data into table
 - SQL example



5

5

Useful Docker commands

- Login to a container
 - `docker ps` (get any container id)
 - `docker exec -it container_id bash`
- List all containers: `docker ps -a`
- Stop a container: `docker stop container_id`
- Start a stopped container: `docker start container_id`

6

6

Load, inspect, and save data

- Data is always huge that does not fit on a single machine
 - Data is distributed on many storage nodes
- Data scientists can likely focus on the format that their data is already in
 - Engineers may wish to explore more output formats
- Spark supports a wide range of input and output sources

7

7

Data sources

- File formats and filesystems
 - Local or distributed filesystem, such as NFS, HDFS, or Amazon S3
 - File formats including text, JSON, SequenceFiles, and protocol buffers
- Structured data sources through Spark SQL
 - Apache Hive
 - Parquet
 - JSON
 - From RDDs
- Databases and key/value stores
 - Cassandra, HBase, Elasticsearch, and JDBC dbs

8

8

File Formats

- Formats range from unstructured, like text, to semistructured, like JSON, to structured, like SequenceFiles

Table 5-1. Common supported file formats

Format name	Structured	Comments
Text files	No	Plain old text files. Records are assumed to be one per line.
JSON	Semi	Common text-based format, semistructured; most libraries require one record per line.
CSV	Yes	Very common text-based format, often used with spreadsheet applications.
SequenceFiles	Yes	A common Hadoop file format used for key/value data.
Protocol buffers	Yes	A fast, space-efficient multilanguage format.
Object files	Yes	Useful for saving data from a Spark job to be consumed by shared code. Breaks if you change your classes, as it relies on Java Serialization.

From Learning Spark [1]

→

9

Lab: loading, inspecting, and saving data

- On the Zeppelin notebook
 - <http://localhost:8080/#/notebook/2EAMFFAH7>

10

10

References

- [1] Karau, Holden, et al. *Learning spark: lightning-fast big data analysis*. " O'Reilly Media, Inc.", 2015.

11