BUAN 6346

Big Data

Project 1

Project 1 includes 2 steps as follows. For each step follow the description and requirements of the step and provide your answers/results in the report.

- There is not a page limit for the report.
- Make sure to document every step you take and explain what you have done.

For each step you need to provide the following deliverables in your report **along with explanations**:

- **Codes**: provide all the codes you used in a clear and readable format
- **Logs**: provide all the logs (wherever applicable)
- **Screenshots**: provide screenshots (wherever applicable)

In every step it is clearly specified what type of deliverable is required in the report.

Step 0: VM Setup

In order to be able to complete this project you need to do the following steps in your VM. Please follow these steps exactly as mentioned. **[No documentation required for this step.]**

1. Create a new virtual machine (VM) from the same image file. This is to make sure that if anything happens, you will not use all the progress you have made on the other VM.
2. Follow assignment 1 to enable all the services and set up your Hadoop.
3. Change the Python default interpreter to Python2.7 in your VM by executing the following command in your Terminal:

   ```
   sudo ln -sf /usr/local/bin/python2.7 /usr/bin/python
   ```

4. Verify that you default Python interpreter has changed by running in the Terminal:

   ```
   Python --version
   ```

5. Create a directory in the user directory of your VM and call it `required_packages`:

   ```
   /home/training/required_packages
   ```

6. Copy the contents of the Required Packages on eLearning to your VM to the directory you created in the previous step (`required_packages`)
7. Move the file `pip-20.3.4-py2.py3-none-any.whl` from `required_packages` to the following directory:

   ```
   /home/training/
   ```

8. Change your working directory to `/home/training`
9. Upgrade the pip on your machine by executing the following command in Terminal:

   ```
   sudo pip install –no-index pip-20.3.4-py2.py3-none-any.whl
   ```

10. Verify that the pip has been upgraded by executing the following:

    ```
    pip --version
    ```

11. Change your directory to `/home/training/required_packages`
12. Install all the packages in the directory by executing the following in Terminal:

    ```
    sudo pip install –no-index ./*
    ```

    you should be able to see all the packages installed and get a successful message at the end.

13. Verify that all the packages have been installed by running the following:

```
pip list
```

Step 1: Data Ingestion

In this step you will use various methods and tools from the Hadoop ecosystem to ingest data from different sources and store them on HDFS.

A. Direct File Transfer

In this step you will manually transfer a dataset from your VM's local storage to HDFS. The dataset contains information from the Bitcoin blockchain and is provided by blockchain.com. you first need to pull the data from the website using their API and store it on your local storage, then you will be able to manually transfer it to HDFS.

1. Setup your Python code using the following information to pull data from the API and store it in a text file on your local storage. [Deliverable: Python code, console output]
    a. This can be done outside VM
    b. Required packages: `requests, datetime`
    c. Your code should pull the hash information of all the blocks in a day for 7 different days
    d. Your code should use the hash information to pull the entire block information for the blocks pulled
    e. Your code should store all this information in a text file on your local storage [Deliverable: screenshots of pulled data in a text file]
        i. If your doing this outside of VM, your code will store on your computer and then you should transfer it to your VM
        ii. If you are doing this inside the VM, it will store it on the VM's storage
2. Transfer the data from your VM's local storage to HDFS [Deliverable: screenshot of data in HDFS]

B. Stream Ingestion using Flume

In this step you should use Apache Flume to store the streaming data being pulled from a website's API. **Alpha Vantage** provides real-time and historical stock market data through their API. They offer both free and paid plans, and their API includes streaming capabilities for real-time data. You will use this website's API to get live data from the stock market and store it in HDFS.

1. Get you free API key from Alpha Vantage

    https://www.alphavantage.co/support/#api-key

2. Setup your Python code using the following information to pull data from the API and send it to `localhost` on a specific port (should be done inside VM) [Deliverable: Python code]
    a. Required packages (already installed in Step 0): `requests, json, socket`
    b. Your code should be able to connect to the Alpha Vantage API using the API key provided by them
    c. Your code should pull the stock market data for the following symbols, duration and frequency:
        i. Symbols: choose 5 symbols from stock market (examples: AAPL, IBM, …)
        ii. Duration: 9:30 am to 4 pm (ET) on 5 business days (check NYSE)
        iii. Frequency: every 1 minute
    d. Your code should print this data in the output [Deliverable: screenshot of console output]

    e. The code should also send the data to a socket connected to the `localhost` on a specific `port` (example: `localhost, port=12345`) [Deliverable: screenshot of both Python code and Flume agent running side by side while the data streams]
3. Setup and configure you Flume agent so that it captures the data from the host `localhost` and stores it on HDFS [Deliverable: Flume console output/log]
    a. Configure Flume by creating a Flume configuration file with the characteristics mentioned below. [Deliverable: Flume configuration]
    b. Your Flume agent should be able to capture the streaming data from the host and port
    c. Your Flume agent should be able to store the captured data in HDFS
    d. Configure your Flume sink so that it creates files with a maximum size of 512KB when storing the data
    e. The data should be directly stored on HDFS [Deliverable: screenshot of data in HDFS]

C. Data Ingestion using Sqoop

In this step you should use Sqoop to import a dataset from MySQL database directly into Hive (technically into HDFS while creating the corresponding table in the Hive Metastore).

The dataset is provided on eLearning. It contains 4 csv files each representing a table containing information on Bitcoin blockchain.

1. Download the file from eLearning and transfer it to your VM
2. Upload each file from your VM's local storage to MySQL database on your VM [Deliverable: SQL code used to upload data to MySQL, logs]
3. Verify the upload by using both MySQL and Sqoop to explore the database [Deliverable: screenshots of SQL code and result and Sqoop command and result confirming the data upload]
4. Use Sqoop to import each table from MySQL directly into Hive [Deliverable: Sqoop command, logs]
5. Verify the import by browsing Hive [Deliverable: HiveQL code and result]
6. Show where the actual data is stored on HDFS [Deliverable: screenshot of location of data stored on HDFS]