

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN



MÔN HỌC: ĐIỆN TOÁN ĐÁM MÂY

ĐỀ TÀI:

TÌM HIỂU PARQUET VÀ VIẾT ỨNG DỤNG DEMO

Giảng viên hướng dẫn: TS. Huỳnh Xuân Phụng

Sinh viên thực hiện :

Nguyễn Chí Trường 17133069

Đinh Quang Huy 17133026

Mai Bình Nam 17133039

Tp.hcm, tháng 1 năm 2021

Mục lục

I.	TÌM HIỂU VỀ PARQUET	1
1.	Định nghĩa apache parquet	1
2.	Kiểu dữ liệu trong apache parquet	1
3.	Kiểu dữ liệu logic trong apache parquet	2
4.	Định dạng của một tệp Apache Parquet trong hadoop	3
5.	Ưu điểm và nhược điểm của Apache Parquet	4
6.	Lợi ích của việc sử dụng định dạng tệp Parquet so với CSV	5
II.	Tìm hiểu về Hadoop và Mapreduce	7
1.	Tìm hiểu về Hadoop	7
2.	Nguyên tắc hoạt động Hadoop	7
3.	Tìm hiểu về MapReduce	8
III.	Cài đặt Hadoop	9
1.	Thiết lập IP tĩnh cho master	9
2.	Cài đặt Java 8	10
3.	Cài đặt SSH	10
4.	Cấu hình host/hostname	11
4.1.	Kiểm tra ip của các máy master, slave	11
4.2.	Cấu hình host	11
4.3.	Cài đặt hostname cho master (thực hiện trên máy master)	11
4.4.	Cài đặt hostname cho slave (thực hiện trên máy slave)	11
5.	Tạo user hadoop	11
6.	Cài đặt Hadoop 2.7.7	11
7.	Cấu hình các thông số cho Hadoop	12
7.1.	File .bashrc	12
7.2.	File hadoop-env.sh	12
7.3.	File core-site.xml	12
7.4.	File mapred-site	13
7.5.	File hdfs-site.xml	14
7.6.	File yarn-site.xml	15
8.	Chỉ ra các máy slaves (chỉ cấu hình ở master)	16
9.	Tạo máy quanghuy2-slave	16
10.	Cài đặt ssh key giữa các node	16

11.	Format namenode	17
12.	Kiểm tra xem mọi thứ đã ổn	17
IV.	Cài đặt Spark và samba.....	19
1.	Cài đặt spark	19
2.	Cài đặt samba (để chia sẻ file giữa ubuntu với windows).....	20
	Tài liệu tham khảo	22

I. TÌM HIỂU VỀ PARQUET

1. Định nghĩa apache parquet

Apache Parquet là một định dạng tệp nhị phân lưu trữ dữ liệu theo kiểu cột. Dữ liệu trong tệp Parquet tương tự như bảng kiểu RDBMS nơi bạn có các cột và hàng. Nhưng thay vì truy cập dữ liệu một hàng tại một thời điểm, bạn thường truy cập vào một cột tại một thời điểm.

Apache Parquet là một trong những định dạng lưu trữ dữ liệu lớn hiện đại. Nó có một số lợi thế, một số trong đó là:

- Lưu trữ cột: truy xuất dữ liệu hiệu quả, nén hiệu quả, v.v ...
- Siêu dữ liệu nằm ở cuối tệp: cho phép các tệp Parquet được tạo từ luồng dữ liệu. (phổ biến trong các kịch bản dữ liệu lớn)
- Được hỗ trợ bởi tất cả các sản phẩm dữ liệu lớn của Apache

2. Kiểu dữ liệu trong apache parquet

Tương tự các dạng lưu trữ khác, kiểu dữ liệu được lưu trữ trên apache parquet bao gồm:

- BOOLEAN: 1 bit boolean
- INT32: Các int có dấu 32 bit
- INT64: Các int có dấu 64 bit
- INT96: Số nguyên có dấu 96 bit
- FLOAT: Giá trị số thực 32-bit IEEE
- DOUBLE: Giá trị số thực 64-bit IEEE
- BYTE_ARRAY: mảng byte dài tùy ý.

So sánh kiểu dữ liệu trên SQL so với Parquet:

SQL Type	Parquet Type
BIGINT	INT64
BOOLEAN	BOOLEAN
N/A	BYTE_ARRAY
FLOAT	FLOAT
DOUBLE	DOUBLE
INTEGER	INT32
VARBINARY(12)*	INT96

3. Kiểu dữ liệu logic trong apache parquet

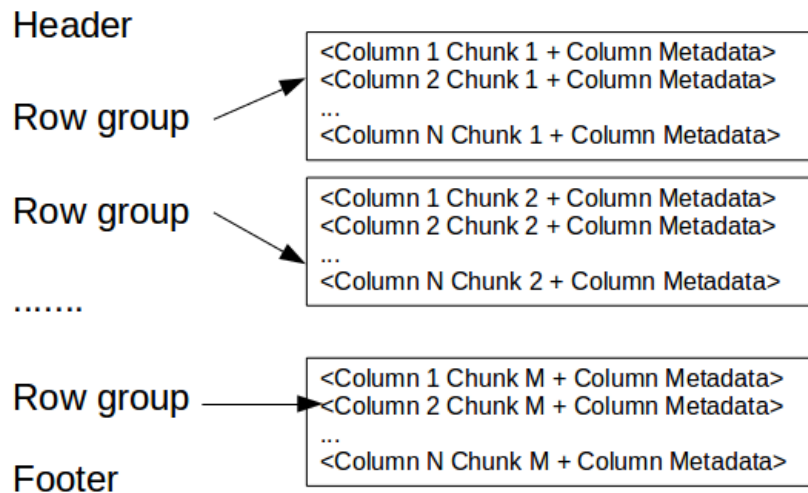
SQL Type	SQL Description	Parquet Logical Type	Parquet Description
DATE	Thời gian dạng YYYY-MM-DD	DATE	Ngày nhưng không bao gồm thời gian trong ngày
VARCHAR	Chuỗi kí tự	UTF8 (Strings)	Chuỗi kí tự được mã hóa UTF8
None		INT_8	Số nguyên 8 bit, có dấu
None		INT_16	Số nguyên 16 bit, có dấu
INT		INT_32	Số nguyên 32 bit, có dấu
None		UINT_8	Số nguyên 8 bit, không dấu

None		UINT_16	Số nguyên 16 bit, không dấu
None		UINT_32	Số nguyên 32 bit, không dấu
None		UINT_64	Số nguyên 64 bit, không dấu
DECIMAL*	Số thập phân với độ chính xác 38 chữ số	DECIMAL	
TIME	Giờ, phút, giây, mili giây	TIME_MILLIS	
TIMESTAMP	Năm, tháng, ngày và giây	TIMESTAMP_MILLIS	
INTERVAL	Khoản thời gian	INTERVAL	

4. Định dạng của một tệp Apache Parquet trong hadoop

Để hiểu định dạng tệp Apache Parquet trong Hadoop:

- Nhóm hàng: Phân vùng dữ liệu theo chiều ngang hợp lý thành các hàng. Một nhóm hàng bao gồm một đoạn cột cho mỗi cột trong tập dữ liệu.
- Nhóm cột: Một đoạn dữ liệu cho một cột cụ thể. Các phần cột này nằm trong một nhóm hàng cụ thể và được đảm bảo là liên tiếp trong tệp.
- Trang: Các đoạn cột được chia thành các trang được viết ngược nhau. Các trang có chung một tiêu đề và người đọc có thể bỏ qua trang mà họ không quan tâm.



Hình 4.1: Minh họa định dạng tệp Parquet.

Ở hình 4.1, phần header chỉ chứa một chữ số “PAR1” (4 bytes) để xác định tệp này có định dạng là một tệp parquet.

Phần footer chứa:

- Tệp metadata: chứa vị trí của tất cả các vị trí bắt đầu cột.
- Độ dài của tệp metadata (4 bytes)
- Một chữ số “PAR1” (4 bytes)

5. Ưu điểm và nhược điểm của Apache Parquet

Ưu điểm:

- Lưu trữ dạng cột như Apache Parquet được thiết kế để mang lại hiệu quả so với các tệp dựa trên hàng như CSV. Khi truy vấn, lưu trữ dạng cột, bạn có thể bỏ qua dữ liệu không liên quan rất nhanh chóng. Kết quả là, các truy vấn tổng hợp ít tốn thời gian hơn so với cơ sở dữ liệu hướng

hàng. Cách lưu trữ này đã giúp tiết kiệm phần cứng và giảm thiểu độ trễ khi truy cập dữ liệu.

- Apache Parquet được xây dựng hoàn thiện. Do đó, nó có thể hỗ trợ các cấu trúc dữ liệu lồng nhau nâng cao. Bố cục của tệp dữ liệu Parquet được tối ưu hóa cho các truy vấn xử lý khối lượng lớn dữ liệu, trong phạm vi gigabyte cho từng tệp riêng lẻ.
- Parquet được xây dựng để hỗ trợ các tùy chọn nén linh hoạt và các chương trình mã hóa hiệu quả. Vì kiểu dữ liệu cho mỗi cột là khá giống nhau nên việc nén từng cột là đơn giản (điều này làm cho các truy vấn nhanh hơn). Dữ liệu có thể được nén bằng cách sử dụng một trong một số codec có sẵn; do đó, các tệp dữ liệu khác nhau có thể được nén khác nhau.

6. Lợi ích của việc sử dụng định dạng tệp Parquet so với CSV

CSV là một định dạng đơn giản và phổ biến rộng rãi được sử dụng bởi nhiều công cụ như Excel, Google Trang tính và nhiều công cụ khác có thể tạo tệp CSV. Mặc dù các tệp CSV là định dạng mặc định cho các đường ống xử lý dữ liệu nhưng nó có một số nhược điểm:

- Amazon Athena và Spectrum sẽ tính phí dựa trên lượng dữ liệu được quét trên mỗi truy vấn.
- Google và Amazon sẽ tính phí bạn theo lượng dữ liệu được lưu trữ trên GS / S3.
- Các khoản phí Dataproc của Google dựa trên thời gian.

Parquet đã giúp người dùng giảm yêu cầu lưu trữ ít nhất một phần ba trên các bộ dữ liệu lớn, ngoài ra, nó còn cải thiện đáng kể thời gian quét và giải mã hóa, do đó chi phí tổng thể.

Bảng sau đây so sánh mức tiết kiệm cũng như tốc độ tăng tốc thu được khi chuyển đổi dữ liệu thành Parquet từ CSV.

Dataset	Kích thước trên Amazon S3	Thời gian chạy truy vấn	Dữ liệu được quét	Giá cả
Dữ liệu được lưu trữ dưới dạng tệp CSV	1 TB	236 giây	1,15 TB	\$ 5,75
Dữ liệu được lưu trữ ở định dạng Apache Parquet	130 GB	6,78 giây	2,51 GB	\$0,01
Tiết kiệm	Giảm 87% khi sử dụng Parquet	Nhanh hơn 34 lần	Dữ liệu được quét ít hơn 99%	Tiết kiệm 99,7%

II. TÌM HIỂU VỀ HADOOP VÀ MAPREDUCE

1. Tìm hiểu về Hadoop

Hadoop là một Apache framework mã nguồn mở cho phép phát triển các ứng dụng phân tán (distributed processing) để lưu trữ và quản lý các tập dữ liệu lớn. Những năm 2000, Google công bố tài liệu nghiên cứu cách tiếp cận và nguyên tắc thiết kế để xử lý khối lượng lớn dữ liệu đã được đánh chỉ mục trên web. Những nguyên tắc cơ bản thiết kế là:

Thứ nhất, thực tế nếu ta có đến hàng trăm hay thậm chí hàng ngàn cỗ máy lưu trữ thì lỗi xảy ra là điều hiển nhiên chứ không phải ngoại lệ, do đó giám sát liên tục, phát hiện lỗi, kháng lỗi và tự động phục hồi phải được tích hợp với hệ thống.

Thứ hai, các tập tin rất lớn so với tiêu chuẩn truyền thống. Tập tin có dung lượng hàng GB và hàng tỷ đối tượng là rất phổ biến. Do đó, những giả định về thiết kế và các thông số như vận hành I/O hay kích thước khối phải xem xét lại.

Thứ ba, hầu hết các tập tin được cập nhật bằng cách thêm dữ liệu mới hơn là ghi đè lên dữ liệu hiện có. Việc ghi dữ liệu ngẫu nhiên trong một tập tin trên thực tế là không xảy ra. Khi ghi, các tập tin chỉ đọc và thường đọc theo thứ tự. Vì đây kiểu truy cập vào các tập tin lớn, nên sự bổ sung thêm trở thành tiêu điểm của việc tối ưu hóa hiệu suất và bảo đảm hoàn tất giao dịch.

2. Nguyên tắc hoạt động Hadoop

Giai đoạn 1: Một người dùng hay một ứng dụng có thể đưa (submit) một công việc (Job) lên Hadoop (Hadoop Job Client) với yêu cầu xử lý cùng các thông tin cơ bản.

Giai đoạn 2: Hadoop Job Client submit job (file jar, file thực thi) và các thiết lập cho JobTracker. Sau đó, master sẽ phân phối tác vụ đến các máy slave để theo dõi và quản lý tiến trình các máy này, đồng thời cung cấp thông tin về tình trạng và chẩn đoán liên quan đến job-client.

Giai đoạn 3: TaskTrackers trên các node khác nhau thực thi tác vụ MapReduce và trả về kết quả output được lưu trong hệ thống file.

3. Tìm hiểu về MapReduce

Là thành phần quan trọng của góp phần làm nên sức mạnh của Hadoop. MapReduce được chia thành hàm là Map và Reduce. Những hàm này được định nghĩa bởi người dùng là hai quá trình liên tiếp khi xử lý dữ liệu.

Map nhận input là tập các cặp khóa/giá trị và output là tập các cặp khóa/giá trị trung gian và ghi xuống đĩa cứng và thông báo cho Reduce nhận dữ liệu đọc.

Reduce sẽ nhận khóa trung gian I và tập các giá trị ứng với khóa đó, ghép nối chúng lại để tạo thành một tập khóa nhỏ hơn. Các cặp khóa/giá trị trung gian sẽ được đưa vào cho hàm reduce thông qua một con trỏ vị trí (iterator). Điều này cho phép ta có thể quản lý một lượng lớn danh sách các giá trị để phù hợp với bộ nhớ.

Ở giữa Map và Reduce thì còn 1 bước trung gian đó chính là Shuffle. Sau khi Map hoàn thành xong công việc của mình thì Shuffle sẽ làm nhiệm vụ chính là thu thập cũng như tổng hợp từ khóa/giá trị trung gian đã được map sinh ra trước đó rồi chuyển qua cho Reduce tiếp tục xử lý.

III. CÀI ĐẶT HADOOP

1. Thiết lập IP tĩnh cho master

- Ubuntu Server 18.04
- Hadoop 2.7.7
- Login với vai trò root (pass: root) để thực hiện những công việc sau
- Kiểm tra các thiết bị mạng

```
# networkctl
```

```
root@quanghuy1-server:/# networkctl
IDX LINK                TYPE          OPERATIONAL SETUP
  1 lo                    loopback      carrier    unmanaged
  2 ens33                 ether         routable   configured

2 links listed.
root@quanghuy1-server:/#
```

- In trạng thái của từng địa chỉ IP trên hệ thống

```
# networkctl status
```

```
2 links listed.
root@quanghuy1-server:/# networkctl status
• State: routable
  Address: 192.168.248.131 on ens33
           fe80::20c:29ff:feef:6b82 on ens33
  Gateway: 192.168.248.2 (VMware, Inc.) on ens33
    DNS: 192.168.248.2
         8.8.8.8
         8.8.4.4
root@quanghuy1-server:/# _
```

- Cấu hình IP tĩnh

```
# vim /etc/netplan/50-cloud-init.yaml
```

- Thêm vào các nội dung sau

```
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    ens33:
      dhcp4: false
      dhcp6: false
      addresses: [192.168.248.131/24]
      gateway4: 192.168.248.2
      nameservers:
        addresses: [192.168.248.2, 8.8.8.8, 8.8.4.4]
  version: 2
~
~
~
```

- Lưu file và chạy lệnh sau để lưu cấu hình mới

```
# netplan apply
```

- Hệ thống đã được cấu hình theo IP mới, để kiểm tra chạy 1 trong 2 lệnh sau

```
# ifconfig
```

```
# ip addr show
```

2. Cài đặt Java 8

```
# add-apt-repository ppa:linuxuprising/java
```

```
# apt update
```

- Khởi động lại máy

```
# reboot
```

```
# apt install -y java-8-openjdk-amd64
```

- Chấp nhận liscence

- Quản lý phiên bản Java (chọn phiên bản Oracle Manual Mode)

```
# update-alternatives -config java
```

3. Cài đặt SSH

```
# apt-get install ssh
```

```
# apt install openssh-server
```

```
# reboot
```

Cấu hình SSH

```
# vim /etc/ssh/sshd_config
```

- Tìm đoạn `PubkeyAuthentication yes`. Bỏ dấu `#` phía trước thành

```
...
```

```
PubkeyAuthentication yes
```

```
...
```

- Tìm đoạn `PasswordAuthentication no` đổi thành

```
...
```

```
PasswordAuthentication yes
```

```
...
```

- Sau khi sửa thì nhấn phím ESC, nhập `:wq` để lưu và thoát khỏi vim.

- Khởi động lại SSH

```
# service sshd restart
```

4. Cấu hình host/hostname

4.1. Kiểm tra ip của các máy master, slave

- # ifconfig
- Ví dụ:
- quanghuy1-server: 192.168.248.131
- quanghuy2-slave: 192.168.248.132

4.2. Cấu hình host

```
# vim /etc/hosts
```

- Nhấn phím i để chuyển sang chế độ insert, bổ sung thêm 2 host master và slave như sau:
192.168.248.131 quanghuy1-server
192.168.248.132 quanghuy2-slave

4.3. Cài đặt hostname cho master (thực hiện trên máy master)

```
# vim /etc/hostname
```

- Trong file này sẽ xuất hiện hostname mặc định của máy, xóa đi và đổi thành quanghuy1-server

4.4. Cài đặt hostname cho slave (thực hiện trên máy slave)

```
# vim /etc/hostname
```

- Trong file này sẽ xuất hiện hostname mặc định của máy, xóa đi và đổi thành quanghuy2-slave
 - Restart máy
- ```
reboot
```

## 5. Tạo user hadoop

- Tạo user hadoopuser để quản lý các permission cho đơn giản
- ```
# addgroup hadoopgroup  
# adduser quanghuyhadoop  
# usermod -g hadoopgroup quanghuyhadoop  
# groupdel quanghuyhadoop
```

6. Cài đặt Hadoop 2.7.7

- Chuyển qua hadoopuser
- ```
su quanghuyhadoop
```

- Chuyển qua thư mục /home/quanghuyhadoop để download file:  
# wget <https://archive.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz>
- Giải nén file  
# tar -xzf hadoop-2.7.7.tar.gz
- Đổi tên thư mục giải nén thành hadoop cho dễ quản lý  
# mv hadoop-2.7.7 hadoop

## 7. Cấu hình các thông số cho Hadoop

### 7.1. File .bashrc

- ```
# vim ~/.bashrc
```
- Thêm vào cuối file .bashrc nội dung như sau:
export HADOOP_HOME=/home/hadoopuser/hadoop #Đường dẫn tới
hadoop home
export JAVA_HOME=/usr/lib/jvm/ java-8-openjdk-amd64
Đường dẫn tới javahome
export PATH=\$PATH:\$HADOOP_HOME/bin
export PATH=\$PATH:\$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=\$HADOOP_HOME
export HADOOP_COMMON_HOME=\$HADOOP_HOME
export HADOOP_HDFS_HOME=\$HADOOP_HOME
export YARN_HOME=\$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=\$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=\$HADOOP_HOME/lib"
 - Nhấn Esc, nhập :wq để lưu và thoát file.
 - Soucre file .bashrc
source ~/.bashrc

7.2. File hadoop-env.sh

- ```
vim ~/hadoop/etc/hadoop/hadoop-env.sh
```
- Tìm đoạn export JAVA\_HOME=... sửa thành như sau:  
# export JAVA\_HOME=/usr/lib/jvm/ java-8-openjdk-amd64/

### 7.3. File core-site.xml

- ```
# vim ~/hadoop/etc/hadoop/core-site.xml
```

- Cấu hình lại thông tin như sau:

```
<configuration>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/quanghuyhadoop/tmp</value>
    <description>Temporary Directory.</description>
  </property>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://quanghuy1-server:54310</value>
    <description>Use HDFS as file storage
engine</description>
  </property>
</configuration>
```

7.4. File mapred-site

```
# cd ~/hadoop/etc/hadoop/
# cp mapred-site.xml.template mapred-site.xml
# vim mapred-site.xml
```

- Chỉnh sửa lại thông tin như sau:

```
<configuration>
  <property>
    <name>mapreduce.jobtracker.address</name>
    <value>quanghuy1-server:54311</value>
    <description>The host and port that the MapReduce
job tracker runs at. If "local", then jobs are run in-
process as a single map and reduce task.
    </description>
  </property>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
    <description>The framework for running mapreduce
jobs</description>
```



```
    </property>
</configuration>
```

7.5. File hdfs-site.xml

```
# vim ~/hadoop/etc/hadoop/hdfs-site.xml
```

- Chỉnh sửa lại thông tin cấu hình như sau:

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>2</value>
        <description>Default block replication. The actual
number of replications can be specified when the file is
created. The default is used if replication is not specified
in create time.
        </description>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>/home/ quanghuy1-
server/hadoop/hadoop_data/hdfs/namenode</value>
        <description>Determines where on the local
filesystem the DFS name node should store the name
table(fsimage). If this is a comma-delimited list of
directories then the name table is replicated in all of the
directories, for redundancy.
        </description>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>/home/ quanghuy1-server
/hadoop/hadoop_data/hdfs/datanode</value>
        <description>Determines where on the local
filesystem an DFS data node should store its blocks. If this
is a comma-delimited list of directories, then data will be
```

stored in all named directories, typically on different devices. Directories that do not exist are ignored.

```
</description>
```

```
</property>
```

```
</configuration>
```

7.6. File yarn-site.xml

- Chuyển đến thư mục ~/hadoop/hadoop-yarn-project/hadoop-yarn/conf

```
# vim ~/hadoop/etc/hadoop/yarn-site.xml
```

Chỉnh sửa lại thông tin cấu hình như sau:

```
<configuration>
```

```
  <property>
```

```
    <name>yarn.nodemanager.aux-services</name>
```

```
    <value>mapreduce_shuffle</value>
```

```
  </property>
```

```
  <property>
```

```
    <name>yarn.resourcemanager.scheduler.address</name>
```

```
    <value> quanghuy1-server:8030</value>
```

```
  </property>
```

```
  <property>
```

```
    <name>yarn.resourcemanager.address</name>
```

```
    <value> quanghuy1-server:8032</value>
```

```
  </property>
```

```
  <property>
```

```
    <name>yarn.resourcemanager.webapp.address</name>
```

```
    <value> quanghuy1-server:8088</value>
```

```
  </property>
```

```
  <property>
```

```
    <name>yarn.resourcemanager.resource-  
tracker.address</name>
```

```
    <value> quanghuy1-server:8031</value>
```

```
  </property>
```

```
  <property>
```

```
    <name>yarn.resourcemanager.admin.address</name>
```

```

        <value> quanghuy1-server:8033</value>
    </property>
</configuration>

```

8. Chỉ ra các máy slaves (chỉ cấu hình ở master)

```
# vim ~/hadoop/etc/hadoop/slaves
```

Thêm hostname của các máy slave: mỗi máy slave đặt trên 1 dòng

```
# quanghuy2-slave
```

9. Tạo máy quanghuy2-slave

- Tắt máy master.
- Copy master ra, đổi tên thành slave
- Mở máy slave, chỉnh lại IP tĩnh và các thông số cho phù hợp: hosts, hostname...

```

# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    ens33:
      dhcp4: false
      dhcp6: false
      addresses: [192.168.153.132/24]
      gateway4: 192.168.153.2
      nameservers:
        addresses: [192.168.153.2, 8.8.8.8, 8.8.4.4]
  version: 2

```

- Lưu ý:

- o Một số lệnh cần phải có quyền root mới thực hiện được.

10. Cài đặt ssh key giữa các node

Thao tác này chỉ thực hiện trên master

- Đăng nhập với hadoopuser


```
# sudo su - quanghuyhadoop
```
- Tạo ssh key


```
# ssh-keygen -t rsa -P ""
```
- Nhấn Enter để chấp nhận giá trị mặc định


```
# cat /home/quanghuyhadoop/.ssh/id_rsa.pub >> /home/quanghuyhadoop/.ssh/authorized_keys
```

```
# chmod 600 /home/quanghuyhadoop/.ssh/authorized_keys
```
- Share ssh key giữa master - master


```
# ssh-copy-id -i ~/.ssh/id_rsa.pub quanghuy1-server
```

- Share ssh key giữa master - slave

```
# ssh-copy-id -i ~/.ssh/id_rsa.pub quanghuy2-slave
```

Test kết nối ssh

- Test kết nối tới server

```
# ssh quanghuyhadoop@quanghuy1-server
```

- Đăng xuất

```
# logout
```

- Test kết nối tới slave

```
# ssh quanghuyhadoop@quanghuy2-slave
```

- Đăng xuất

```
# logout
```

11.Format namenode

- Thao tác này chỉ thực hiện trên master.
- Cập nhật lại các thông tin cấu hình của master

```
# hadoop namenode -format
```

12.Kiểm tra xem mọi thứ đã ổn

- Trên master chúng ta chạy lệnh sau để khởi động các thành phần có trong Hadoop

```
# start-all.sh
```

- Kiểm tra các thành phần có chạy đủ bằng lệnh sau

```
# jps
```

- Nếu xuất hiện output dạng như sau thì có nghĩa là các thành phần đã chạy đủ

```
2003 NameNode
```

```
2412 ResourceManager
```

```
2669 Jps
```

```
2255 SecondaryNameNode
```

- Kiểm tra các máy slave còn hoạt động hay không

```
# hdfs dfsadmin -report
```

- Nếu thấy xuất hiện output như sau thì có nghĩa là máy slave vẫn đang hoạt động

```
20/02/18 12:28:56 WARN util.NativeCodeLoader: Unable to load
native-hadoop library for your platform... using builtin-java
classes where applicable
```

Configured Capacity: 10340794368 (9.63 GB)
Present Capacity: 8154087424 (7.59 GB)
DFS Remaining: 8154054656 (7.59 GB)
DFS Used: 32768 (32 KB)
DFS Used%: 0.00%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

Live datanodes (1): # Số datanode (slave node) đang hoạt động

Name: 192.168.33.12:50010 **(slave)**
Hostname: ubuntu-bionic
Decommission Status : Normal
Configured Capacity: 10340794368 (9.63 GB)
DFS Used: 32768 (32 KB)
Non DFS Used: 2169929728 (2.02 GB)
DFS Remaining: 8154054656 (7.59 GB)
DFS Used%: 0.00%
DFS Remaining%: 78.85%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1

IV. CÀI ĐẶT SPARK VÀ SAMBA

1. Cài đặt spark

Su quanghuyhadoop, tải file spark về

```
wget https://downloads.apache.org/spark/spark-3.0.1/spark-3.0.1-bin-hadoop2.7.tgz
```

```
sudo mv spark-3.0.1-bin-hadoop2.7 /opt/spark
```

Thiết lập biến môi trường trong file profile

```
echo "export SPARK_HOME=/opt/spark" >> ~/.profile
echo "export
PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin" >>
~/.profile
echo "export PYSPARK_PYTHON=/usr/bin/python3" >> ~/.profile
```

```
# ~/.profile: executed by Bourne-compatible login shells.

if [ "$BASH" ]; then
  if [ -f ~/.bashrc ]; then
    . ~/.bashrc
  fi
fi

mesg n || true
export SPARK_HOME=/opt/spark
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
export PYSPARK_PYTHON=/usr/bin/python3
```

Tiến hành bật hdfs

Khởi động spark-shell

```
root@quanghuy1-server:~# su quanghuyhadoop
quanghuyhadoop@quanghuy1-server:/root$ cd /opt/spark/bin
quanghuyhadoop@quanghuy1-server:/opt/spark/bin$ ./spark-shell
21/01/06 04:56:33 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://quanghuy1-server:4040
Spark context available as 'sc' (master = local[*], app id = local-1609909007164).
Spark session available as 'spark'.
Welcome to

  ____
 /_  __ \  _ __| | | |
/_ __/ \_\ \_ \ | | |
 \___)____)___)___|_|_|
version 3.0.1

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_275)
Type in expressions to have them evaluated.
Type :help for more information.

scala> _
```

2. Cài đặt samba (để chia sẻ file giữa ubuntu với windows)

```
sudo apt update
sudo apt install samba
```

Cấu hình thư mục cần share trong conf

```
sudo nano /etc/samba/smb.conf
[sambashare]
    comment = Samba on Ubuntu
    path = /home/quanghuyhadoop/sharefile
    read only = no
    browsable = yes
```

Khởi động lại dịch vụ

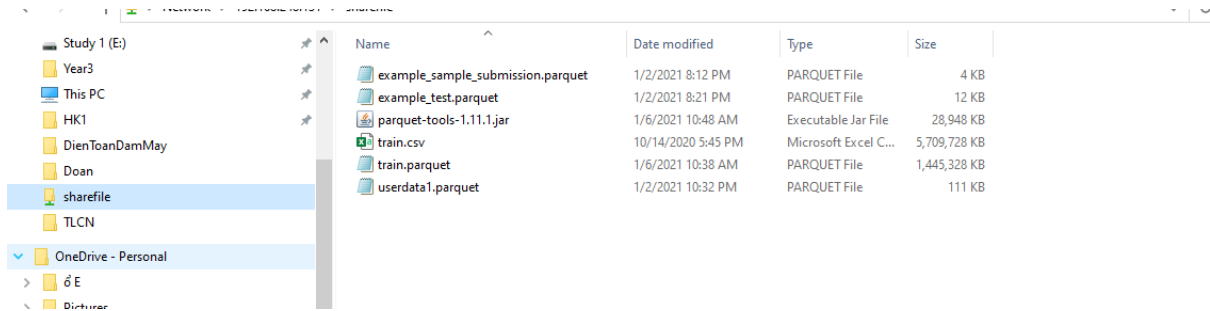
```
sudo service smb restart
```

Kho phép samba chạy

```
sudo ufw allow samba
```

Cài tài khoản và mật khẩu root -pass 123456

```
sudo smbpasswd -a username
```



Name	Date modified	Type	Size
example_sample_submission.parquet	1/2/2021 8:12 PM	PARQUET File	4 KB
example_test.parquet	1/2/2021 8:21 PM	PARQUET File	12 KB
parquet-tools-1.11.1.jar	1/6/2021 10:48 AM	Executable Jar File	28,948 KB
train.csv	10/14/2020 5:45 PM	Microsoft Excel C...	5,709,728 KB
train.parquet	1/6/2021 10:38 AM	PARQUET File	1,445,328 KB
userdata1.parquet	1/2/2021 10:32 PM	PARQUET File	111 KB

TÀI LIỆU THAM KHẢO

- [1]. <https://phoenixnap.com/kb/install-spark-on-ubuntu>
- [2]. <https://javalibs.com/artifact/org.apache.parquet/parquet-tools>
- [3]. <https://ubuntu.com/tutorials/install-and-configure-samba#3-setting-up-samba>