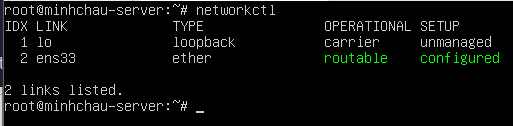
Cài đặt Hadoop Eco System

* Ubuntu Server 18.04
* Hadoop 2.7.7
* Login với vai trò root (pass: root) để thực hiện những công việc sau

# **Thiết lập IP tĩnh cho master**

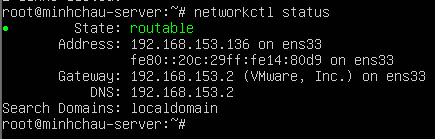
* Kiểm tra các thiết bị mạng

# networkctl



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

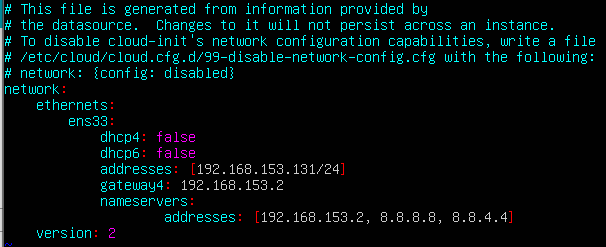
# networkctl status



* Cấu hình IP tĩnh

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

* Thêm vào các nội dung sau



* 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

# **Cài đặt Oracle Java 14**

# add-apt-repository ppa:linuxuprising/java

# apt update

* Khởi động lại máy

# reboot

# apt install -y oracle-java14-installer

* Chấp nhận liscence
* Quản lý phiên bản Java (chọn phiên bản Oracle Manual Mode)

# update-alternatives –config java

# **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

# **Cấu hình host/hostname**

## **Kiểm tra ip của các máy master, slave**

# ifconfig

Ví dụ:

* Master: 192.168.153.131
* Slave: 192.168.153.132

## **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:

192.168.153.131 master

192.168.153.132 slave-1

…

## **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 master

## **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 slave-1
* Restartmáy

# reboot

# **Tạo user hadoop**

* Tạo user hadoopuser để quản lý các permission cho đơn giản

# addgroup hadoopgroup

# adduser hadoopuser

# usermod -g hadoopgroup hadoopuser

# groupdel hadoopuser

# **Cài đặt Hadoop 2.7.7**

* Chuyển qua hadoopuser

# su hadoopuser

* Chuyển qua thư mục /home/hadoopuser 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

# **Cấu hình các thông số cho Hadoop**

## **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-14-oracle # 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

## **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-14-orale/

## **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/hadoopuser/tmp</value>

<description>Temporary Directory.</description>

</property>

<property>

<name>fs.defaultFS</name>

<value>hdfs://master:54310</value>

<description>Use HDFS as file storage engine</description>

</property>

</configuration>

## 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>master: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>

## **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/hadoopuser/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/hadoopuser/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>

## **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>master:8030</value>

</property>

<property>

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

<value>master:8032</value>

</property>

<property>

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

<value>master:8088</value>

</property>

<property>

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

<value>master:8031</value>

</property>

<property>

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

<value>master:8033</value>

</property>

</configuration>

# **Chỉ ra các máy slaves (chỉ cấu hình ở master)**

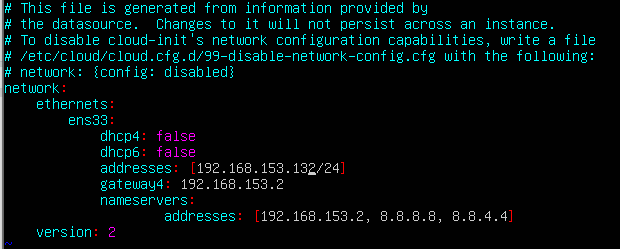
# vim ~/hadoop/etc/hadoop/slaves

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

# slave-1

# **Tạo máy slave-1**

* 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…



* **Lưu ý:**
  + Một số lệnh cần phải có quyền root mới thực hiện được.

# **Cài đặt ssh key giữa các node**

Thao tác này chỉ thực hiện trên master

* Đăng nhập với hadoopuser

# sudo su - hadoopuser

* Tạo ssh key

# ssh-keygen -t rsa -P ""

* Nhấn Enter để chấp nhận giá trị mặc định

# cat /home/hadoopuser/.ssh/id\_rsa.pub >> /home/hadoopuser/.ssh/authorized\_keys

# chmod 600 /home/hadoopuser/.ssh/authorized\_keys

* Share ssh key giữa master - master

# ssh-copy-id -i ~/.ssh/id\_rsa.pub master

* Share ssh key giữa master - slave

# ssh-copy-id -i ~/.ssh/id\_rsa.pub slave-1

## **Test kết nối ssh**

* Test kết nối tới master

# ssh hadoopuser@master

* Đăng xuất

# logout

* Test kết nối tới slave

# ssh hadoopuser@slave

* Đăng xuất

# logout

# **Format namenode**

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

# hadoop namenode -format

# **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

Last contact: Tue Feb 18 12:28:56 UTC 2020

# **Chạy thử**

## **Tạo test**

* Ra thư mục gốc, tạo file test

# cd ~

# vim test.sh

* Tạo nội dung như sau

!/bin/bash

# test the hadoop cluster by running wordcount

# create input files

mkdir input

echo "Hello World" >input/file2.txt

echo "Hello Hadoop" >input/file1.txt

# create input directory on HDFS

hadoop fs -mkdir -p input

put input files to HDFS

hdfs dfs -put ./input/\* input

# run wordcount

hadoop jar $HADOOP\_HOME/share/hadoop/mapreduce/sources/hadoop-mapreduce-examples-2.7.7-sources.jar org.apache.hadoop.examples.WordCount input output

# print the input files

echo -e "\ninput file1.txt:"

hdfs dfs -cat input/file1.txt

echo -e "\ninput file2.txt:"

hdfs dfs -cat input/file2.txt

# print the output of wordcount

echo -e "\nwordcount output:"

hdfs dfs -cat output/part-r-00000

**Giải thích:** Test trên sẽ tạo ra 2 file file2.txt và file1.txt có nội dung lần lượt là Hello World và Hello Hadoop. 2 file này lần lượt được đưa vào trong HDFS, sau đó sẽ chạy một job có nhiệm vụ đếm số lần xuất của hiện của mỗi từ có trong file1.txt và file2.txt.

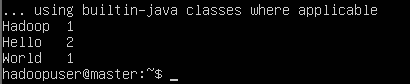
* Chạy test

# ./test.sh

* Nếu báo lỗi Permission denied thì thực hiện lệnh

# chmod +x test.sh

* Chạy xong mà nhận được kết quả dạng như bên dưới là mọi thứ OK.



* Trong trường hợp bạn muốn test lại thì phải xóa kết quả cũ bằng lệnh sau

# rm -rf input

# hadoop fs -rm -r input

# hadoop fs -rm -r output

rồi chạy lại lệnh

# ./test.sh