**TÀI LIỆU HƯỚNG DẪN THỰC HÀNH**

**Tên bài thực hành:**

**Cài đặt và triển khai Hadoop Single Node**

**10/2023**

**MỤC LỤC**

[1. CƠ SỞ LÝ THUYẾT 2](#_Toc147489038)

[1.1. Giới thiệu chung 2](#_Toc147489039)

[1.2. Nguyên lý hoạt động 8](#_Toc147489040)

[1.3. Nguyên lý dịch & gỡ rối 9](#_Toc147489041)

[2. BÀI THÍ NGHIỆM THỰC HÀNH 10](#_Toc147489042)

[2.1. Mục đích - Yêu cầu 10](#_Toc147489043)

[**2.1.1. Mục đích 10**](#_Toc147489044)

[**2.1.2. Yêu cầu: 10**](#_Toc147489045)

[**2.1.3. Thời gian thực hiện 11**](#_Toc147489046)

[2.2. Chuẩn bị 11](#_Toc147489047)

[**2.2.1. Danh mục thiết bị thực hành 11**](#_Toc147489048)

[2.3. Nội dung 12](#_Toc147489049)

[**2.3.1. Các bước thực hiện 12**](#_Toc147489050)

[**2.3.2. Kết quả thực hiện 18**](#_Toc147489051)

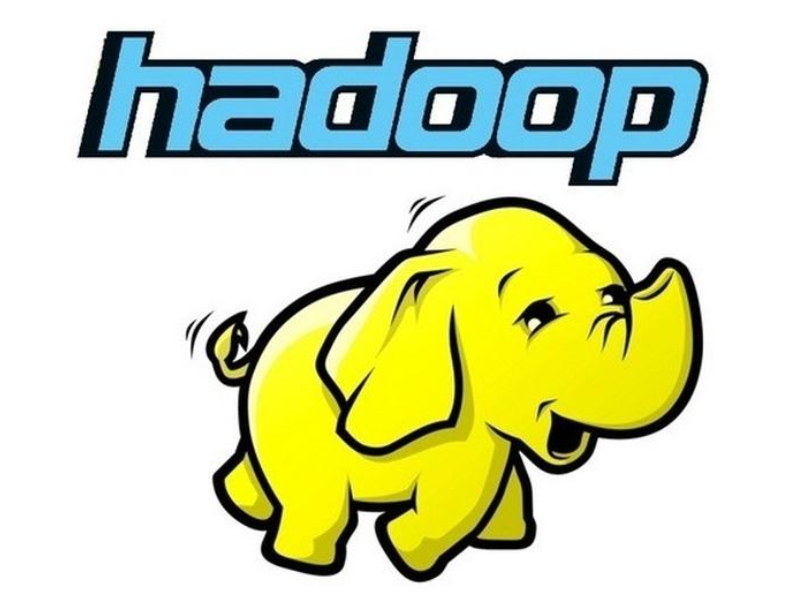
[TÀI LIỆU HƯỚNG DẪN THỰC HÀNH 20](#_Toc147489052)

[PHỤ LỤC 20](#_Toc147489053)

# 1. CƠ SỞ LÝ THUYẾT

## 1.1. Giới thiệu chung

***Làm quen với kiến trúc***



Hình 1 Biểu tượng của Hadoop

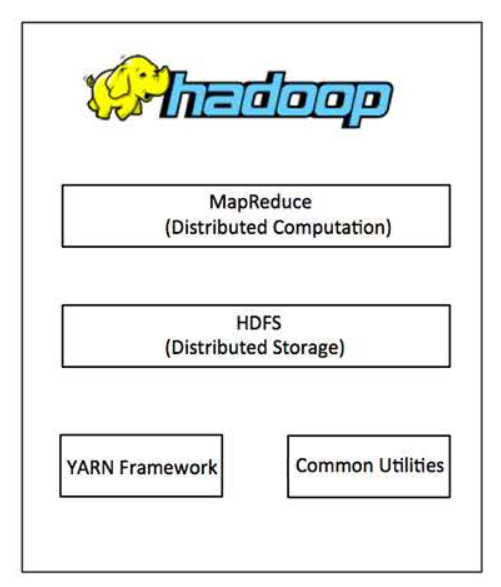
**+** Khái niệm***:***

* Hadoop là một Apache framework nguồn mở viết bằng Java cho phép phát triển các ứng dụng phân tán có cường độ dữ liệu lớn một cách miễn phí. Nó được thiết kế để mở rộng quy mô từ một máy chủ đơn sang hàng ngàn máy tính khác có tính toán và lưu trữ cục bộ (local computation and storage). Hadoop được phát triển dựa trên ý tưởng từ các công bố của Google về mô hình Map-Reduce và hệ thống file phân tán Google File System (GFS). Và có cung cấp cho chúng ta một môi trường song song để thực thi các tác vụ Map-Reduce.
* Nhờ có cơ chế streaming mà Hadoop có thể phát triển trên các ứng dụng phân tán bằng cả java lẫn một số ngôn ngữ lập trình khác như C++, Pyhthon, Pearl,....

**+** Đặc điểm

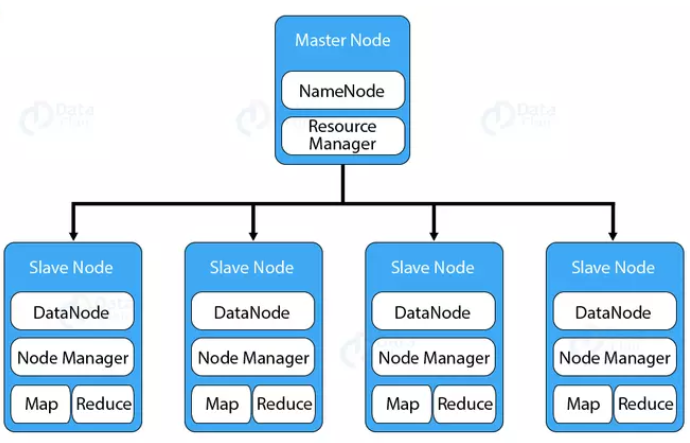
* Hadoop:
* Phân tán (Distributed): Hadoop cho phép xử lý dữ liệu trên nhiều máy tính cùng một lúc, chia công việc thành các phần nhỏ gọi là "blocks" và phân phối chúng trên các nút trong cụm máy tính.
* Tolerant to Failures (Chống lỗi): Hadoop có khả năng chống lỗi, tức là nếu một hoặc vài máy tính trong cụm gặp sự cố, nó có thể tự động chuyển công việc sang các máy tính khác để đảm bảo tính liên tục của quá trình xử lý.
* Scalability (Khả năng mở rộng): Hadoop có thể mở rộng dễ dàng bằng cách thêm máy tính vào cụm để xử lý lượng dữ liệu lớn hơn. Khả năng mở rộng này giúp Hadoop phù hợp với các tình huống tăng cường nhu cầu về lưu trữ và xử lý dữ liệu.
* Hệ thống tệp dựa trên HDFS (Hadoop Distributed File System): Hadoop sử dụng HDFS để lưu trữ dữ liệu phân tán. HDFS chia dữ liệu thành các khối nhỏ (blocks) và sao lưu chúng trên nhiều máy tính trong cụm. Điều này giúp cải thiện hiệu suất và độ tin cậy của lưu trữ dữ liệu.
* Sử dụng MapReduce: Hadoop sử dụng mô hình lập trình MapReduce để xử lý và phân tích dữ liệu. Nó cho phép người dùng định nghĩa các công việc xử lý dữ liệu dưới dạng các bộ từ (map) và hợp nhất (reduce), giúp tận dụng sức mạnh tính toán của nhiều máy tính.
* Hỗ trợ đa ngôn ngữ và framework: Hadoop hỗ trợ nhiều ngôn ngữ lập trình và framework khác nhau như Java, Python và R, cho phép các nhà phân tích dữ liệu sử dụng ngôn ngữ và công cụ phù hợp với nhu cầu của họ.
* Cộng đồng mạnh mẽ: Hadoop được phát triển và duy trì bởi một cộng đồng rộng lớn, có nhiều đóng góp từ các công ty công nghệ hàng đầu, điều này đảm bảo sự phát triển và hỗ trợ liên tục cho framework này.
* Hadoop Ecosystem (Hệ sinh thái Hadoop): Ngoài framework cơ bản, Hadoop có một hệ sinh thái phong phú với nhiều công cụ và dự án bổ sung như Hive, Pig, Spark, HBase, và nhiều công cụ khác để hỗ trợ việc xử lý và phân tích dữ liệu lớn.

+ Cấu trúc



Hình 2 Cấu trúc của Hadoop

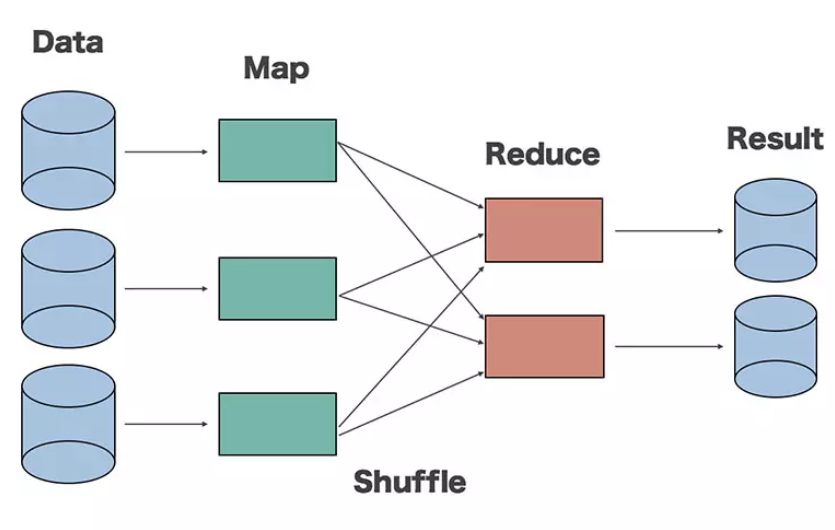
* Hadoop có một cấu trúc liên kết master-slave. Trong cấu trúc này, chúng ta có một node master và nhiều node slave . Chức năng của node master là gán một tác vụ cho các node slave khác nhau và quản lý tài nguyên. Các node slave là máy tính thực tế có thể không mạnh lắm. Các node slave lưu trữ dữ liệu thực trong khi trên master chúng ta có metadata.
* Kiến trúc Hadoop gồm có ba lớp chính đó là:
* HDFS (Hadoop Distributed File System)
* Map-Reduce
* Yarn
* HDFS (Hadoop Distributed File System)



Hình 3 kiến trúc HDFS (Hadoop Distributed File System)

* Là hệ thống file phân tán, cung cấp khả năng lưu trữ dữ liệu khổng lồ và tính năng tối ưu hoá việc sử dụng băng thông giữa các node. HDFS có thể được sử dụng để chạy trên một cluster lớn với hàng chục ngàn node.
* Cho phép truy xuất nhiều ổ đĩa như là 1 ổ đĩa. Nói cách khác, chúng ta có thể sử dụng một ổ đĩa mà gần như không bị giới hạn về dung lượng. Muốn tăng dung lượng chỉ cần thêm node (máy tính) vào hệ thống.
* Có kiến trúc Master-Slave
* NameNode chạy trên máy chủ Master, có tác vụ quản lý Namespace và điều chỉnh truy cập tệp của client
* DataNode chạy trên các nút Slave. có tác vụ lưu trữ business data thực tế
* Một tập tin với định dạng HDFS được chia thành nhiều block và những block này được lưu trữ trong một tập các DataNodes
* Kích thước 1 block thông thường là 64MB, kích thước này có thể thay đổi được bằng việc cấu hình.

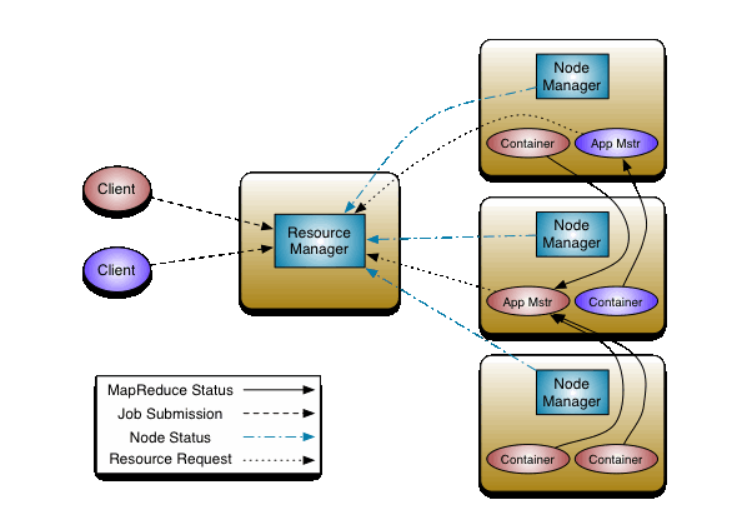
• Map-Reduce



Hình 4 Cấu trúc của Map-Reduce

* Map-Reduce là một framework dùng để viết các ứng dụng xử lý song song một lượng lớn dữ liệu có khả năng chịu lỗi cao xuyên suốt hàng ngàn cluster(cụm) máy tính
* Map-Reduce thực hiện 2 chức năng chính đó là Map và Reduce
* Map: Sẽ thực hiện đầu tiên, có chức năng tải, phân tích dữ liệu đầu vào và được chuyển đổi thành tập dữ liệu theo cặp key/value
* Reduce: Sẽ nhận kết quả đầu ra từ tác vụ Map, kết hợp dữ liệu lại với nhau thành tập dữ liệu nhỏ hơn.

• Yarn



Hình 5 Kiến trúc của yarn

* YARN (Yet-Another-Resource-Negotiator) là một framework hỗ trợ phát triển ứng dụng phân tán YARN cung cấp daemons và APIs cần thiết cho việc phát triển ứng dụng phân tán, đồng thời xử lý và lập lịch sử dụng tài nguyên tính toán (CPU hay memory) cũng như giám sát quá trình thực thi các ứng dụng đó.
* Bên trong YARN, chúng ta có hai trình quản lý ResourceManager và NodeManage
* ResourceManager: Quản lý toàn bộ tài nguyên tính toán của cluster.
* NodeManger: Giám sát việc sử dụng tài nguyên của container và báo cáo với ResourceManger. Các tài nguyên ở đây là CPU, memory, disk, network,...
* Quá trình 1 ứng dụng chạy trên YARN được mô tả bằng sơ đồ trên qua các bước sau:
* Client giao 1 task cho Resource Manager
* Resource Manager tính toán tài nguyên cần thiết theo yêu cầu của ứng dụng và tạo 1 App Master (App Mstr). Application Master được chuyển đến chạy 1 một node tính toán. Application Master sẽ liên lạc với các NodeManager ở các node khác để ra yêu cầu công việc cho node này.
* Node Manager nhận yêu cầu và chạy các task trên container
* Các thông tin trạng thái thay vì được gửi đến JobTracker sẽ được gửi đến App Master.
* ResourceManger có hai thành phần quan trọng đó là Scheduler và ApplicationManager: Scheduler có trách nhiệm phân bổ tài nguyên cho các ứng dụng khác nhau. Đây là Scheduler thuần túy vì nó không thực hiện theo dõi trạng thái cho ứng dụng. Nó cũng không sắp xếp lại các tác vụ bị lỗi do lỗi phần cứng hoặc phần mềm. Bộ lập lịch phân bổ các tài nguyên dựa trên các yêu cầu của ứng dụng
* ApplicationManager có chức năng sau:
* Chấp nhận nộp công việc.
* Đàm phán container đầu tiên để thực thi ApplicationMaster. Một nơi chứa kết hợp các yếu tố như CPU, bộ nhớ, đĩa và mạng.
* Khởi động lại container ApplicationMaster khi không thành công.
* Chúng ta có thể mở rộng YARN ngoài một vài nghìn node thông qua tính năng YARN Federation. Tính năng này cho phép chúng ta buộc nhiều cụm YARN thành một cụm lớn. Điều này cho phép sử dụng các cụm độc lập, ghép lại với nhau cho một job rất lớn

## *1.2. Nguyên lý hoạt động*

***Nguyên lý hoạt động của quy trình này có thể được mô tả như sau:***

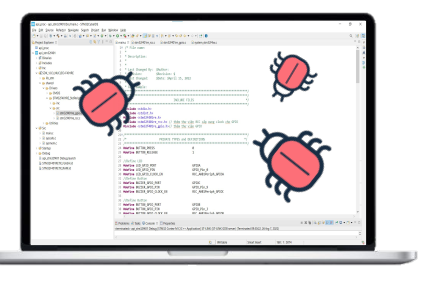
Nguyên lý hoạt động của Hadoop dựa trên hai thành phần chính: Hadoop Distributed File System (HDFS) và MapReduce. Dưới đây là phần giải thích chi tiết về cách Hadoop hoạt động:

* Hadoop Distributed File System (HDFS):
  + Lưu trữ phân tán: HDFS chia dữ liệu thành các khối nhỏ (blocks), thường có kích thước 128 MB hoặc 256 MB. Các khối dữ liệu này được lưu trữ trên nhiều máy tính trong cụm Hadoop. Việc chia nhỏ dữ liệu và lưu trữ nó trên nhiều nút giúp tối ưu hóa tính toàn vẹn và hiệu suất của hệ thống.
  + Sao lưu và kiểm tra lỗi: HDFS sử dụng cơ chế sao lưu (replication) để sao chép dữ liệu trên nhiều máy tính khác nhau trong cụm. Mặc định, mỗi khối dữ liệu có ba bản sao. Điều này đảm bảo tính chịu lỗi của hệ thống, vì nếu một máy tính gặp sự cố, dữ liệu vẫn có sẵn trên các bản sao khác.
  + Duyệt dữ liệu hiệu quả: HDFS cho phép duyệt dữ liệu một cách hiệu quả bằng cách phân tách và phân phối nó trên các nút. Mỗi khối dữ liệu được lưu trữ tại vị trí cụ thể và có thể được truy cập song song, giúp tăng tốc quá trình đọc và ghi dữ liệu.
* MapReduce:
* Phân tán xử lý dữ liệu: MapReduce là mô hình lập trình và xử lý dữ liệu phân tán. Quá trình xử lý được chia thành hai pha chính: pha Map và pha Reduce. Trong pha Map, dữ liệu được chia thành các cặp key-value và được xử lý độc lập trên các nút. Sau đó, kết quả từ các pha Map được tổng hợp và nhóm lại theo key trong pha Reduce.
  + Parallel Processing (Xử lý song song): Các công việc Map và Reduce có thể được thực hiện song song trên nhiều máy tính trong cụm Hadoop. Điều này giúp tận dụng sức mạnh tính toán của nhiều máy tính để xử lý dữ liệu lớn một cách nhanh chóng và hiệu quả.
  + Tính chịu lỗi và khả năng mở rộng: MapReduce có khả năng tự động phân phối công việc và xử lý lại công việc nếu một máy tính gặp sự cố. Điều này đảm bảo tính chịu lỗi và khả năng mở rộng của hệ thống.

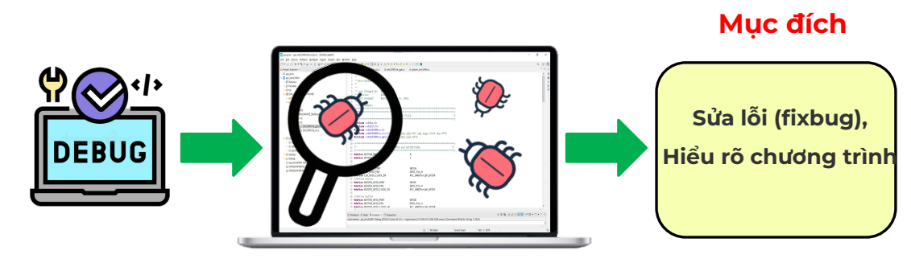
## *1.3. Nguyên lý dịch & gỡ rối*

***+*** Bug được hiểu là thuật ngữ dùng để chỉ các lỗi kỹ thuật có thể xảy ra trong quá trình thiết kế và vận hành các chương trình lập trình.

+ Bug khiến cho phần mềm, ứng dụng không thực thi được hoặc thực thi sai.

**

+ Debug là quá trình tìm kiếm và phát hiện nguyên nhân gây ra lỗi

****

***+ Các tùy chọn với chế độ debug***

* Thu thập thông tin và logs cẩn thận:
* Thu thập thông tin chi tiết về lỗi hoặc vấn đề mà bạn gặp phải, bao gồm thông báo lỗi, stack trace, và mô tả về các bước bạn đã thực hiện trước khi gặp lỗi.
* Kiểm tra các logs của Hadoop, bao gồm logs của Hadoop HDFS và Hadoop MapReduce. Logs này thường nằm trong thư mục /logs của các nút trong cụm.
* Kiểm tra cấu hình (Configuration):
* Đảm bảo rằng các tệp cấu hình của Hadoop (core-site.xml, hdfs-site.xml, mapred-site.xml,...) được cấu hình chính xác với các giá trị như địa chỉ IP, cổng, đường dẫn tệp dữ liệu, v.v.
* Kiểm tra xem các biến môi trường (environment variables) liên quan đến Hadoop như HADOOP\_HOME, HADOOP\_CONF\_DIR, và JAVA\_HOME có được đặt đúng.
* Kiểm tra quyền truy cập (Permissions):
* Đảm bảo rằng bạn có quyền truy cập và ghi vào các thư mục và tệp liên quan đến Hadoop. Điều này bao gồm quyền truy cập vào thư mục dữ liệu HDFS và quyền thực thi các tệp lệnh Hadoop.
* Sử dụng các công cụ gỡ rối (Debugging Tools):
* Sử dụng các công cụ gỡ rối như hadoop fsck để kiểm tra trạng thái của HDFS, yarn logs để xem logs của ứng dụng YARN, và jps để xem các tiến trình Java đang chạy.
* Sử dụng log viewer của Hadoop để theo dõi logs một cách tiện lợi.

# 2. BÀI THÍ NGHIỆM THỰC HÀNH

## 2.1. Mục đích - Yêu cầu

### 2.1.1. Mục đích

+ Giúp sinh viên làm quen với các công nghệ hiện đại, nắm rõ cách thức cài đặt hadoop

+ Làm việc với các tập dữ liệu lớn, hiểu cách phân tán xử lý dữ liệu, và làm quen với các khái niệm quan trọng như Hadoop Distributed File System (HDFS) và MapReduce.

+ sử dụng các công cụ và thư viện Hadoop để trích xuất thông tin từ dữ liệu lớn và tạo ra các báo cáo và đối tượng dữ liệu

### 2.1.2. Yêu cầu:

* Nắm vững kiến thức được giới thiệu, tự tìm hiểu thêm nhiều chức năng khác
* Hoàn thành bài thí nghiệm thực hành theo yêu cầu của giảng viên

### 2.1.3. Thời gian thực hiện

+ Thời gian mỗi buổi thực hành là từ 3-4 giờ, chia làm 10-12 nhóm nhỏ, mỗi nhóm 2 sinh viên/1 máy tính. Sinh viên tìm hiểu cơ sở lý thuyết ngắn gọn và các bước thực hiện có thể thao tác dễ dàng.

## 2.2. Chuẩn bị

### 2.2.1. Danh mục thiết bị thực hành

+ Phần cứng: 01 máy tính. Tất cả được được đồng bộ theo số thứ tự từ 1-20

+ Phần mềm:

* Hadoop Distribution: Bạn cần phải tải và cài đặt một bản phân phối Hadoop, chẳng hạn như Apache Hadoop hoặc một phân phối Hadoop thương mại như Cloudera, Hortonworks, hay MapR.
* Java Development Kit (JDK): Hadoop được viết bằng Java, vì vậy bạn cần cài đặt JDK trên tất cả các máy tính trong cụm.
* Cấu hình hệ thống: Bạn cần cấu hình các tệp cấu hình Hadoop, chẳng hạn như core-site.xml, hdfs-site.xml, và mapred-site.xml. Điều này bao gồm định cấu hình các đường dẫn, cổng và các thiết lập khác.
* Cài đặt và triển khai phần mềm quản lý cụm (như Apache ZooKeeper hoặc Apache Ambari): Các phần mềm này giúp quản lý và theo dõi các nút trong cụm Hadoop.
* Hệ điều hành: Các máy tính trong cụm cần chạy một hệ điều hành tương thích với Hadoop, chẳng hạn như Linux.
* Cài đặt và cấu hình tường lửa (firewall): Điều này quan trọng để đảm bảo rằng các cổng mạng cần thiết cho Hadoop được mở và tường lửa không gây cản trở.

**Danh mục thiết bị thực hành phòng lab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stt | Tên thiết bị | Số lượng | Thông số  kỹ thuật | Vai trò |
| 1 | Máy tính để bàn | 01 | - Intel Core i3-2100 3.1 Ghz/3M Cache  - 4GB DDR3 (nâng cấp lên ram 8GB thêm 300K)  -SSD 120G Gb chuẩn SATA 3 - 6Gb/s Ổ SSD | -Tính toán, viết chương trình, chạy chương trình |

## 2.3. Nội dung

### 2.3.1. Các bước thực hiện

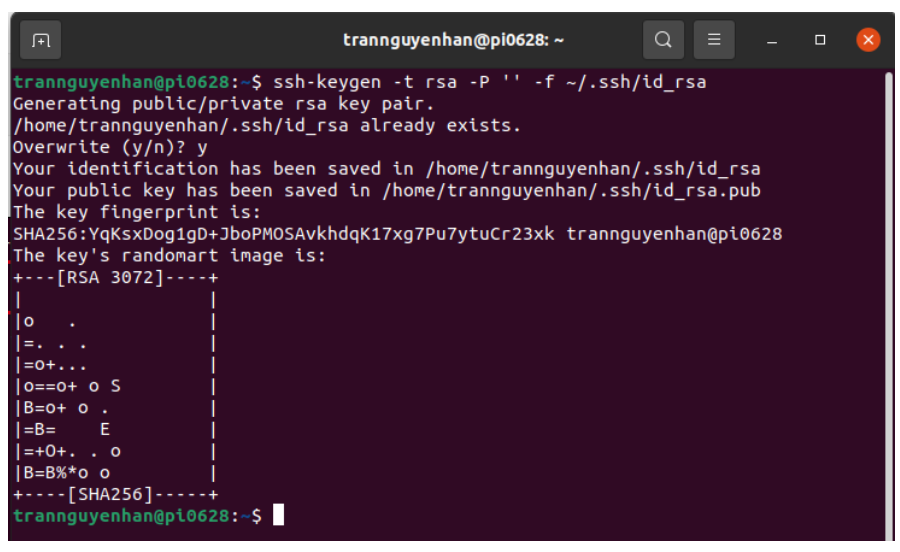
***Nội dung 1:*** Cài đặt và triển khai Hadoop Single Node

**Bước 1: Kiểm tra điều kiện trước khi cài đặt**

|  |
| --- |
| /\**Máy bạn phải có Openjdk ( bản 8, 11 hay 15 đều được), nếu chưa có thì bạn có thể cài theo câu lệnh sau :* \*/  ***sudo apt-get install openjdk-11-jdk -y***  */\*Máy có SSH client và SSH server, nếu chưa có thì bạn có thể cài theo câu lệnh sau :\*/*  ***sudo apt-get install -y openssh-server openssh-client*** |

**Bước 2: Thiết lập Private key cho Hadoop**

|  |
| --- |
| */\* Tạo cặp khóa SSH và xác định vị trí sẽ được lưu trữ :\*/*  ***ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa*** |

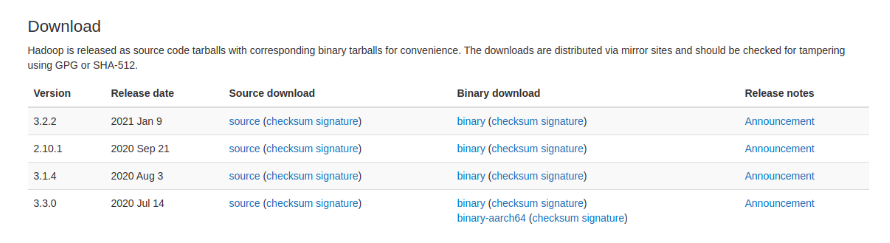
****

Hình 6 Hệ thống tiến hành tạo và lưu cặp khóa

|  |
| --- |
| */\* Sử dụng lệnh cat để lưu public key vào authorized\_keys trong thư mục của SSH:\*/*  ***cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys***  ***/\*****Phân quyền cho người dùng với lệnh****chmod****:\*/*  ***chmod 0777 ~/.ssh/authorized\_keys***  */\*Xác minh mọi thứ được thiết lập chính xác bằng cách ssh đến localhost: \*/*  ***ssh localhost***  ***/\**** *Nếu như bước dưới cùng không đăng nhập được ssh key trên ubuntu thì thay thế chmod ở trên bằng lệnh sau* ***\*/***  ***chmod og-wx ~/.ssh/authorized\_keys*** |

**Bước 3: Download và Cài đặt Hadoop trên Ubuntu**

|  |
| --- |
| */\*Tải về một phiên bản Hadoop trên trang phân phối chính thức của Hadoop tại :*[*https://hadoop.apache.org/releases.html*](https://hadoop.apache.org/releases.html)*\*/* |

****

Hình 7 Giao diện tải hadoop

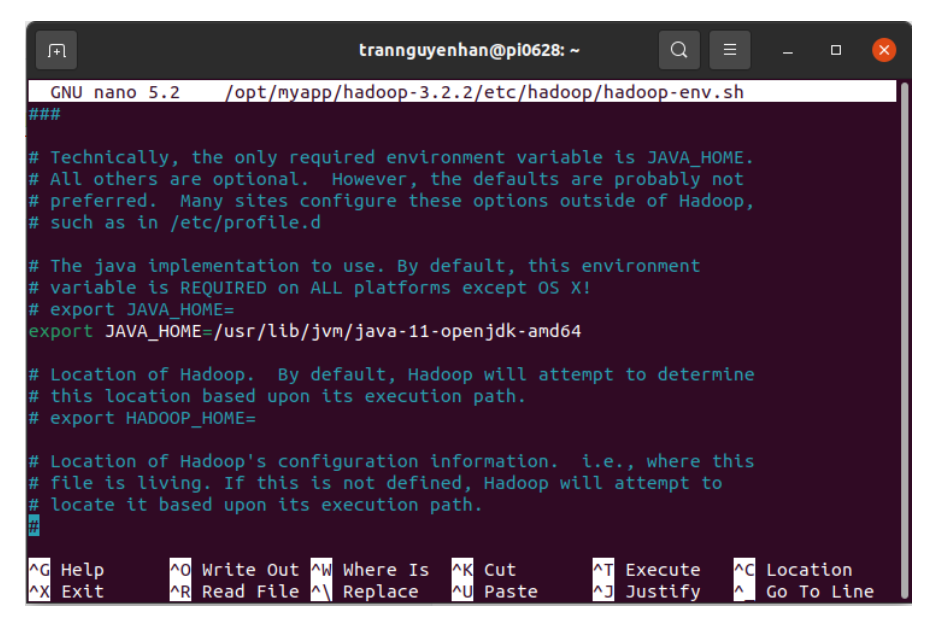
|  |
| --- |
| */\*Ấn vào phần binary trong Binary download. Bây giờ để file nén mà bạn vừa tải về vào bất kì chỗ nào và giải nén nó ra bằng lệnh : \*/*  ***tar xvzf hadoop-3.2.2.tar.gz*** |

**Bước 4: Cấu hình và triển khai Hadoop Single Node (Pseudo-Distributed Mode)**

|  |
| --- |
| */\*Để cấu hình Hadoop cho chế độ phân phối giả chúng ta sẽ chỉnh sửa các tập file cấu hình của Hadoop trong đường dẫn****etc/hadoop****và trong file cấu hình môi trường gồm các file sau :*  *.bashrc*  *hadoop-env.sh*  *core-site.xml*  *hdfs-site.xml*  *mapred-site.xml*  *yarn-site.xml*  ***Cấu hình biến môi trường Hadoop ( file .bashrc)***  *Mở file của .bashrc của bạn bằng trình soạn thảo nano : \*/*  ***sudo nano ~/.bashrc***  ***/\*****Xác định biến mỗi trường****Hadoop****bằng cách thêm các biến sau vào cuối file ( nhớ chỉnh sửa đường dẫn****$HADOOP\_HOME****cho đúng với đường dẫn mà bạn đã đặt hadoop)\*/*  ***#Hadoop Related Options***  ***JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64***  ***export HADOOP\_HOME=/opt/myapp/hadoop-3.2.2***  ***/\* (sửa lại đường dẫn thư mục HADOOP trên máy của bạn) \*/***  ***export HADOOP\_INSTALL=$HADOOP\_HOME***  ***export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME***  ***export HADOOP\_COMMON\_HOME=$HADOOP\_HOME***  ***export HADOOP\_HDFS\_HOME=$HADOOP\_HOME***  ***export HADOOP\_YARN\_HOME=$HADOOP\_HOME***  ***export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native***  ***export PATH=PATH:HADOOP\_HOME/sbin:$HADOOP\_HOME/bin***  ***export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/native"***  ***/\****Áp dụng các thay đổi trên bằng cách thực hiện lệnh sau : \*/  ***source ~/.bashrc*** |

**Bước 5: Chỉnh sửa file hadoop-env.sh**

|  |
| --- |
| */\*Mở file hadoop-env.sh bằng trình soạn thảo nano : \*/*  ***sudo nano $HADOOP\_HOME/etc/hadoop/hadoop-env.sh***  */\*Tìm tới vị trí như hình bên dưới, bỏ comment ( bỏ dấu #) phần JAVA\_HOME và thêm vào đầy đủ đường dẫn Openjdk trên máy của bạn :\*/* |



Hình 8 Giao diện chỉnh sửa file hadoop-env.sh

**Bước 6: Chỉnh sửa file hadoop-env.sh**

|  |
| --- |
| */\*Mở file <hadoop\_directory>/etc/hadoop/core-site.xml bằng trình soạn thảo nano : \*/*  */\* Sửa: <user>: Tên user của ubuntu, hadoop-3.2.2: thư mục cài đặt của Hadoop \*/*  ***sudo nano $HADOOP\_HOME/etc/hadoop/core-site.xml***  */\*Thêm vào giữa 2 thẻ configuration để được nội dung đầy đủ như sau :\*/*  ***<configuration>***  ***<property>***  ***<name>hadoop.tmp.dir</name>***  ***<value>/home/<user>/hadoop-3.2.2/tmpdata</value>***  ***</property>***  ***<property>***  ***<name>fs.default.name</name>***  ***<value>hdfs://localhost:9000</value>***  ***</property>***  ***</configuration>*** |

**Bước 7: Chỉnh sửa file hdfs-site.xml**

|  |
| --- |
| */\** *Mở file core-site.xml bằng trình soạn thảo nano : \*/*  ***sudo nano $HADOOP\_HOME/etc/hadoop/core-site.xml***  */\*Thêm vào giữa 2 thẻ configuration để được nội dung đầy đủ như sau : \*/*  ***<configuration>***  ***<property>***  ***<name>dfs.data.dir</name>***  ***<value>/opt/myapp/hadoop-3.2.2/dfsdata/namenode</value>***  ***</property>***  ***<property>***  ***<name>dfs.data.dir</name>***  ***<value>/opt/myapp/hadoop-3.2.2/dfsdata/datanode</value>***  ***</property>***  ***<property>***  ***<name>dfs.replication</name>***  ***<value>2</value>***  ***</property>***  ***</configuration>*** |

**Bước 8: Chỉnh sửa file mapred-site.xml**

|  |
| --- |
| ***/\*Mở file core-site.xml bằng trình soạn thảo nano : \*/***  *sudo nano $HADOOP\_HOME/etc/hadoop/mapred-site.xml*  ***/\*Thêm vào giữa 2 thẻ configuration để được nội dung đầy đủ như sau : \*/***  ***<configuration>***  ***<property>***  ***<name>mapreduce.framework.name</name>***  ***<value>yarn</value>***  ***</property>***  ***</configuration>*** |

**Bước 9: Chỉnh sửa file yarn-site.xml**

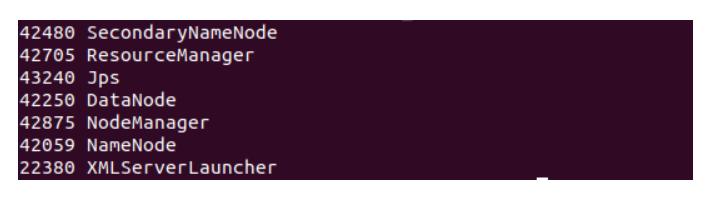
|  |
| --- |
| */\** *Mở file core-site.xml bằng trình soạn thảo nano :\*/*  ***sudo nano $HADOOP\_HOME/etc/hadoop/yarn-site.xml***  */\*Thêm vào giữa 2 thẻ configuration để được nội dung đầy đủ như sau : \*/*  ***<configuration>***  ***<property>***  ***<name>yarn.nodemanager.aux-services</name>***  ***<value>mapreduce\_shuffle</value>***  ***</property>***  ***<property>***  ***<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>***  ***<value>org.apache.hadoop.mapred.ShuffleHandler</value>***  ***</property>***  ***<property>***  ***<name>yarn.resourcemanager.hostname</name>***  ***<value>127.0.0.1</value>***  ***</property>***  ***<property>***  ***<name>yarn.acl.enable</name>***  ***<value>0</value>***  ***</property>***  ***<property>***  ***<name>yarn.nodemanager.env-whitelist</name>***  ***<value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PERPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAPRED\_HOME</value>***  ***</property>***  ***</configuration>*** |

**Bước 10: Format HDFS namenode**

|  |
| --- |
| */\** *Các bạn phải định dạng lại namenode trước khi bắt đầu các dịch vụ đầu tiên : \*/*  ***hdfs namenode –format***  */\** *Trong trường hợp gặp lỗi Cannot create directory /home/hadoop/tmpdata... thì xem lại user ở bước 6 đã đúng với user hiện tại hay chưa \*/* |

**Bước 11: Start Hadoop Cluster**

|  |
| --- |
| */\* Tại thư mục sbin, thực hiện các lệnh sau để chạy và khởi động Hadoop : \*/*  ***./start-all.sh***  */\*Chạy lệnh jsp để kiểm tra các trình daemon đang chạy : \*/*  ***Jps*** |

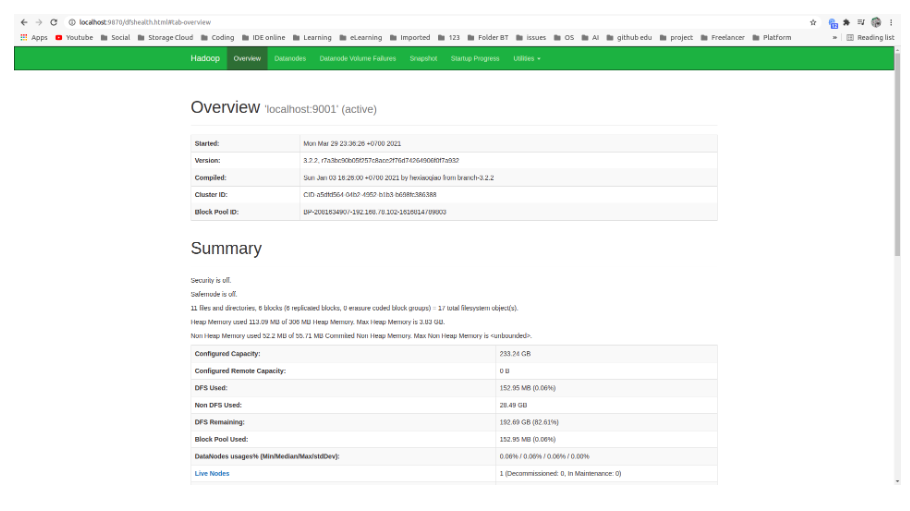
****

Hình 9 Số trình daemon như sau thì bạn đã cấu hình đúng

### 2.3.2. Kết quả thực hiện

***Truy cập Hadoop UI từ trình duyệt***

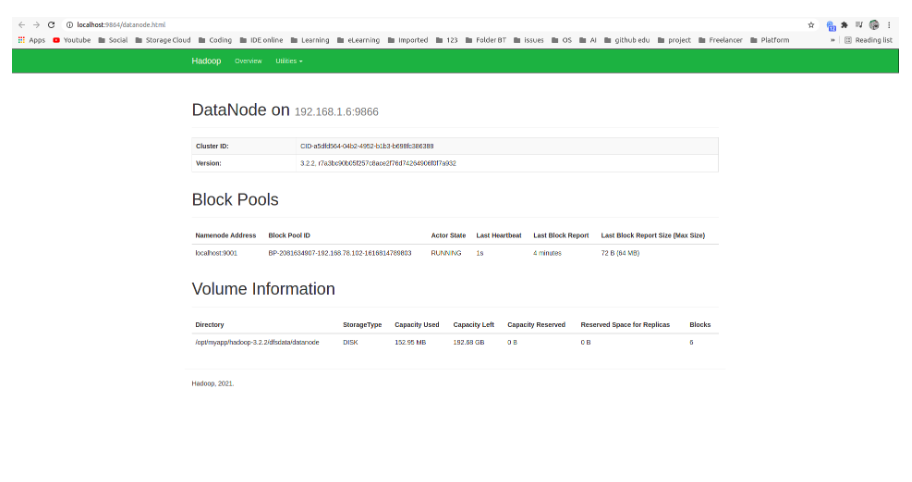
|  |
| --- |
| */\*Các bạn có thể kiểm tra Hadoop đã được cài đặt thành công hay chưa tại cổng mặc định của namenode là 9870 : \*/*  ***localhost:9870*** |



Hình 10 Truy cập Hadoop UI từ trình duyệt

***Kiểm tra datanode tại cổng mặc định 9864***

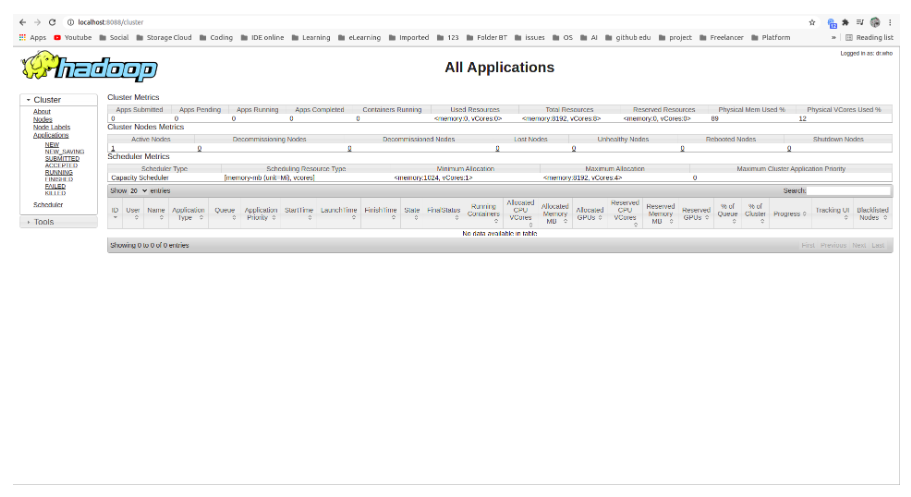
|  |
| --- |
| */\*Kiểm tra datanode tại cổng mặc định 9864 : \*/*  ***localhost:9864*** |



Hình 11 datanode tại cổng mặc định 9864

***Kiểm tra Yarn resource manager tại cổng 8088:***

|  |
| --- |
| */\*Kiểm tra Yarn resource manager tại cổng 8088 : \*/*  ***locahost:8088*** |



Hình 12 Yarn resource manager tại cổng 8088

# TÀI LIỆU HƯỚNG DẪN THỰC HÀNH

[1] Tài liệu: https://www.tailieubkhn.com/2021/03/install-hadoop-single-node.html

# PHỤ LỤC

**Phiếu báo cáo kết quả thực hành (Sinh viên)**

Tên bài: ……………………………………………………………………………...

Họ và tên sinh viên……………………………Mã sinh viên……………………...

………………………………………………………………..……………………...

Nhóm…………………..Lớp……………… ..Ngày…..tháng…..năm…………..

Giáo viên hướng dẫn…………………………Ca thực tập……………………….

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Nội dung  thực hành | Mức độ  hoàn thành (%) | Thời gian  hoàn thành | Đánh giá kết quả (100) |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| Thảo luận sinh viên: | | | | |

**Phiếu đánh giá kết quả thực hành (Giảng viên)**

Tên bài: ……………………………………………………………………………...

Họ và tên sinh viên……………………………Mã sinh viên……………………...

Nhóm…………………..Lớp…………………Ngày…..tháng…..năm…………..

Giáo viên hướng dẫn………………………….Ca thực tập……………………….

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thứ tự** | **Nội dung đánh giá** | **Điểm chuẩn** | **Yêu cầu** | **Ghi chú** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Tổng điểm: | | | | |
| Nhận xét giảng viên: | | | | |