TRƯỜNG ĐẠI HỌC SƯ PHẠM VÀ KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH

BÁO CÁO ĐỒ ÁN

TÌM HIỂU APACHE HIVE VÀ DEMO DATAWAREHOUSE

Giảng viên:
T.S Huỳnh Xuân Phụng

Sinh viên thực hiện:
Nguyễn Xuân Sang
Trần Trung Hiếu

Năm học: 2021-2022

ĐH SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CNTT
****

Ã HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh Phúc
****

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1 :..... .....**MSSV 1**: .....
Họ và tên Sinh viên 2 :..... .....**MSSV 2**: .....
Ngành: Công nghệ Thông tin Tên đề tài:
.....
.....
Họ và tên Giáo viên hướng dẫn: .....
NHẬN XÉT
Về nội dung đề tài khối lượng thực hiện:
.....
.....
.....
.....
.....
.....
.....
.....
Ưu điểm:
.....
.....
.....
.....
.....
.....
.....
.....
Khuyết điểm
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

ĐH SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CNTT
****

Ã HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh Phúc
****

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1 :..... .....**MSSV 1**: .....
Họ và tên Sinh viên 2 :..... .....**MSSV 2**: .....
Ngành: Công nghệ Thông tin Tên đề tài:
.....
.....
Họ và tên Giáo viên hướng dẫn: .....
NHẬN XÉT
Về nội dung đề tài khối lượng thực hiện:
.....
.....
.....
.....
.....
.....
.....
.....
Ưu điểm:
.....
.....
.....
.....
.....
.....
.....
.....
Khuyết điểm
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

MỤC LỤC

BÁO CÁO ĐỒ ÁN 1
I. GIỚI THIỆU 7
1. Tóm tắt đề tài 7
2. Giới thiệu đề tài 7
3. Lý do chọn đề tài 7
II. NỘI DUNG 7
1. Hadoop 7
a. Giới thiệu 7
b. Nhiệm vụ 8
c. Kiến trúc 8
- *Hadoop Distributed File System (HDFS)* 9
- *Hadoop MapReduce* 10
- *Hadoop Common* 10
- *Hadoop YARN* 11

d. Cách thức hoạt động 11
e. Lý do sử dụng 12
2. Apache Hive 12
a. Giới thiệu 12
b. Đặc trưng 13
c. Kiến trúc 13
d. Ưu điểm nổi trội 14
e. Cách làm việc 15
3. DataWareHouse 16
a. Giới thiệu 16
b. Lợi ích 19
c. Cách thức hoạt động 19
d. Phân loại 20
e. Thành phần 21
f. Đối tượng sử dụng 21

g. Kiến trúc 22
h. Sự phát triển 23
i. Cloud Data Warehouse 24
4. Demo DataWareHouse 25
a. Cài đặt máy chủ EC2 25
b. Cài đặt Hadoop 28
c. Cài đặt Hive 32
d. Sử dụng HiveQL 34
III. TỔNG KẾT 37

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh đã đưa môn học điện toán đám mây vào chương trình giảng dạy. Đặc biệt, em xin gửi lời cảm ơn chân thành đến giảng viên bộ môn – Thầy Huỳnh Xuân Phụng đã dạy dỗ và tâm huyết truyền đạt những kiến thức quý giá cho em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học của thầy, em đã trau dồi cho bản thân nhiều kiến thức bổ ích, tinh thần học tập nghiêm túc và hiệu quả. Đây chắc chắn sẽ là những kiến thức có giá trị sâu sắc, là hành trang để em vững bước sau này.

Bộ môn Điện toán đám mây là môn học thú vị, bổ ích và có tính thực tế cao. Đảm bảo cung cấp đầy đủ kiến thức, kỹ năng, giúp sinh viên có thể ứng dụng vào thực tế. Tuy nhiên, do khả năng tiếp thu thực tế còn nhiều hạn hẹp, kiến thức chưa sâu rộng. Mặc dù bản thân đã cố gắng hết sức nhưng chắc chắn bài tiểu luận khó tránh khỏi những thiếu sót, kính mong thầy xem xét và góp ý để bài tiểu luận của em được hoàn thiện và tốt hơn.

Em xin chân thành cảm ơn!”

I. GIỚI THIỆU

1. *Tóm tắt đề tài*

Tìm hiểu về Hadoop và xây dựng một DataWarehouse đơn giản

2. *Giới thiệu đề tài*

Trong thời đại công nghệ 4.0 ngày nay, có lẽ các bạn được nghe rất nhiều về AI, big data Machine Learning hay điện toán đám mây… Nhưng tất cả những công nghệ đó đều phải dựa vào tài nguyên của người dùng đó là Big data.

Vậy Big Data là gì? Big Data là một tập hợp dữ liệu rất lớn và rất phức tạp đến nỗi những công cụ, kỹ thuật xử lý dữ liệu truyền thống không thể nào đảm đương được. Hiện nay BIg data đang là một trong những ưu tiên hàng đầu của các công ty công nghệ trên toàn thế giới, vậy những kỹ thuật hiện đại nào sẽ giúp các công ty giải quyết được vấn đề của Big Data? Và hôm nay mình xin giới thiệu về Hadoop, một framework được dùng phổ biến nhất để giải quyết các bài toán về Big Data.

3. *Lý do chọn đề tài*

Muốn được nghiên cứu và tìm hiểu về Bigdata củng như là các công cụ để thực thi và làm việc với bigdata.

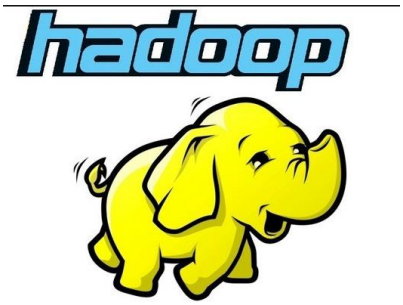
II. NỘI DUNG

1. *Hadoop*

a. Giới thiệu

- Hadoop là gì ? **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.
- Hadoop hiện thực mô hình MapReduce, mô hình mà ứng dụng sẽ được chia nhỏ ra thành nhiều phân đoạn khác nhau được chạy song song trên nhiều node khác nhau.

- Hadoop được viết bằng Java tuy nhiên vẫn hỗ trợ C++, Python, Perl bằng cơ chế streaming.



b. Nhiệm vụ

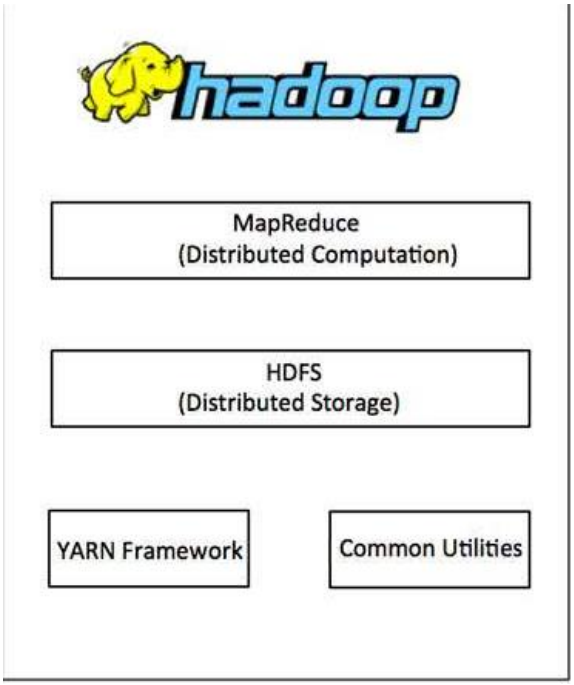
- Xử lý và làm việc khối lượng dữ liệu khổng lồ tính bằng Petabyte.
- Xử lý trong môi trường phân tán, dữ liệu lưu trữ ở nhiều phần cứng khác nhau, yêu cầu xử lý đồng bộ
- Các lỗi xuất hiện thường xuyên.
- Băng thông giữa các phần cứng vật lý chứa dữ liệu phân tán có giới hạn.

c. Kiến trúc

- Một cụm Hadoop nhỏ gồm 1Master node và nhiều Worker/slave node. Toàn bộ cụm chứa 2 lớp, một lớp MapReduce Layer và lớp kia là HDFS Layer.
- Mỗi lớp có các thành phần liên quan riêng:

✓ Master node gồm JobTracker, TaskTracker, NameNode, và DataNode. Worker/Slave node gồm DataNode, và TaskTracker.

✓ Cũng có thể Worker/Slave node chỉ là dữ liệu hoặc node để tính toán.



- *Hadoop Distributed File System (HDFS)*
    - Đây là hệ thống file phân tán cung cấp truy cập thông lượng cao cho ứng dụng khai thác dữ liệu. **Hadoop Distributed File System (HDFS)** là hệ thống tập tin ảo.
    - Khi chúng ta di chuyển 1 tập tin trên HDFS, nó tự động chia thành nhiều mảnh nhỏ. Các đoạn nhỏ của tập tin sẽ được nhân rộng và lưu trữ trên nhiều máy chủ khác để tăng sức chịu lỗi và tính sẵn sàng cao.
    - HDFS sử dụng kiến trúc master/slave, trong đó master gồm một NameNode để quản lý hệ thống file metadata và một hay nhiều slave DataNodes để lưu trữ dữ liệu thực tại.
    - Một tập tin với định dạng HDFS được chia thành nhiều khối và những khối này được lưu trữ trong một tập các DataNodes. NameNode định nghĩa ánh xạ từ các khối đến các DataNode. Các DataNode điều hành các tác vụ đọc và ghi dữ liệu lên hệ

thống file. Chúng cũng quản lý việc tạo, huỷ, và nhân rộng các khối thông qua các chỉ thị từ NameNode.

- *Hadoop MapReduce*

    - Đây là hệ thống dựa trên YARN dùng để xử lý song song các tập dữ liệu lớn. Là cách chia một vấn đề dữ liệu lớn hơn thành các đoạn nhỏ hơn và phân tán nó trên nhiều máy chủ. Mỗi máy chủ có 1 tập tài nguyên riêng và máy chủ xử lý dữ liệu trên cục bộ. Khi máy chủ xử lý xong dữ liệu, chúng sẽ gởi trở về máy chủ chính.
    - **MapReduce** gồm một single master (máy chủ) JobTracker và các slave (máy trạm) TaskTracker trên mỗi cluster-node. Master có nhiệm vụ quản lý tài nguyên, theo dõi quá trình tiêu thụ tài nguyên và lập lịch quản lý các tác vụ trên các máy trạm, theo dõi chúng và thực thi lại các tác vụ bị lỗi. Những máy slave TaskTracker thực thi các tác vụ được master chỉ định và cung cấp thông tin trạng thái tác vụ (task-status) để master theo dõi.
    - JobTracker là một điểm yếu của Hadoop Mapreduce. Nếu JobTracker bị lỗi thì mọi công việc liên quan sẽ bị ngắt quãng.

- *Hadoop Common*

    - Đây là các thư viện và tiện ích cần thiết của Java để các module khác sử dụng. Những thư viện này cung cấp hệ thống file và lớp OS trừu tượng, đồng thời chứa các mã lệnh Java để khởi động Hadoop.

- *Hadoop YARN*
    - Quản lý tài nguyên của các hệ thống lưu trữ dữ liệu và chạy phân tích.

d. Cách thức hoạt động

Hadoop hoạt động gồm có 4 giai đoạn:

❖ **Giai đoạn 1:**

Một user hay một ứng dụng có thể submit một job lên Hadoop (hadoop job client) với yêu cầu xử lý cùng các thông tin cơ bản:

1. Nơi lưu (location) dữ liệu input, output trên hệ thống dữ liệu phân tán.
2. Các java class ở định dạng jar chứa các dòng lệnh thực thi các hàm map và reduce.
3. Các thiết lập cụ thể liên quan đến job thông qua các thông số truyền vào.

❖ **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.
- Khi "chạy Hadoop" có nghĩa là chạy một tập các trình nền – daemon, hoặc các chương trình thường trú, trên các máy chủ khác nhau trên

mạng của bạn. Những trình nền có vai trò cụ thể, một số chỉ tồn tại trên một máy chủ, một số có thể tồn tại trên nhiều máy chủ.

Các daemon bao gồm:

- NameNode
- DataNode
- SecondaryNameNode
- JobTracker
- TaskTracker

❖ **Giai đoạn 4:**

e. Lý do sử dụng

Điểm thuật lợi:

- Robus and Scalable – Có thể thêm node mới và thay đổi chúng khi cần.
- Affordable and Cost Effective – Không cần phần cứng đặc biệt để chạy Hadoop.
- Adaptive and Flexible – Hadoop được xây dựng với tiêu chí xử lý dữ liệu có cấu trúc và không cấu trúc.
- Highly Available and Fault Tolerant – Khi 1 node lỗi, nền tảng Hadoop tự động chuyển sang node khác.

2. Apache Hive

a. Giới thiệu

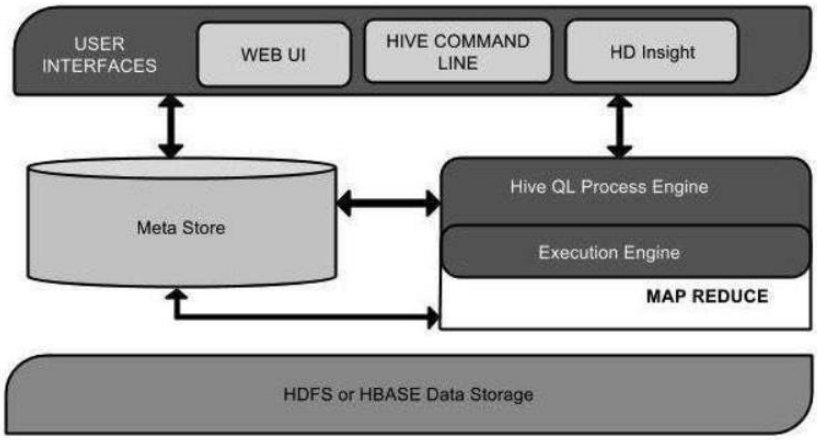
Cũng giống như SQL, ngôn ngữ truy vấn Hive cũng cung cấp các toán tử cơ bản để xử lý cơ sở dữ liệu, HiveSQL có thể tạo và quản lý các tables và partitions dễ dàng, bên cạnh đó, nó cũng hỗ trợ các toán tử Relational,

Logical, Arithmetic, Evaluate functions, và nhiều các loại toán tử khác nữa. Phương thức hoạt động của HiveSQL là tải về nội dung của một table từ thư mục cục bộ hoặc kết quả của các câu truy vấn đến thư mục HDFS.

b. Đặc trưng

- Thứ 1 nó được thiết kế dành cho OLAP
- Thứ 2: nó lưu trữ các lược đồ trong cơ sở dữ liệu và xử lý các dữ liệu này bên trong HDFS
- Thứ 3: nó cung cấp ngôn ngữ kiểu SQL để truy vấn cơ sở dữ liệu được thuận lợi và dễ dàng, và được gọi là HiveSQL (hay HQL)
- Thứ 4: chính vì sử dụng ngôn ngữ kiểu SQL, nên trông Hive rất quen thuộc, dễ dàng sử dụng nhanh chóng đối với các lập trình viên mới bắt đầu và đặc biệt có khả năng mở rộng.

c. Kiến trúc



Kiến trúc của Hive có rất nhiều thành phần khác nhau, tuy nhiên, có 5 thành phần chính được sử dụng nhiều nhất dưới đây:

- Thành phần quan trọng đầu tiên với tên gọi *User Interface:* Đây chính là giao diện người dùng mà Hive hỗ trợ, bao gồm: Hive Web UI, Hive command line và Hive HD Insight, nó giúp tạo ra sự tương tác giữa người dùng với HDFS.

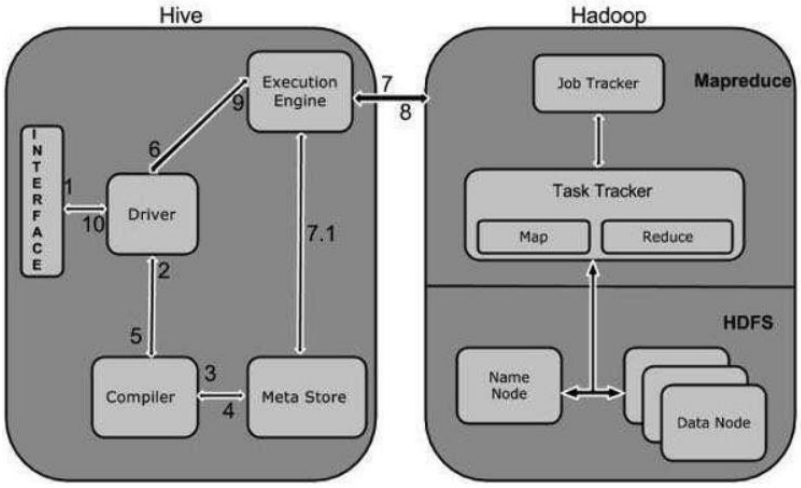
- Thành phần thứ 2. *Meta Store:* đây chính là nơi mà Hive chọn các máy chủ cơ sở dữ liệu để lưu trữ như: các loại lược đồ, các metadata, các cột, các bảng, các loại dữ liệu trong một bảng, một cột và dữ liệu ánh xạ của HDFS.
- Thành phần 3. *HiveQL Process Engine*: HiveQL làm việc tương tự nhe SQL để truy vấn các thông tin về lược đồ trên hệ thống. Ngoài ra, đây còn là một phương pháp nhằm thay thế cho chương trình MapReduce. Vì thế, các lập trình viên thay vì phải viết chương trình MapReduce bằng Java tương đối phức tạp và mất khá nhiều thời gian, thì họ có thể viết những câu truy vấn bằng HiveQL để xử lý công việc được dễ dàng hơn.
- Thành phần thứ 4. *Execution Engine*: đây là phần kết hợp giữa 2 công cụ xử lý: HiveQL + MapReduce, và nó chính là công cụ thực thi Hive Execution Engine. Công cụ này giúp thực thi và xử lý các câu truy vấn dữ liệu.
- Và cuối cùng, là thành phần thứ 5. *HDFS hoặc HBASE:* đây chính là hệ thống các tệp phân tán của Hadoop. Và HBASE chính là các kỹ thuật dùng để lưu trữ dữ liệu vào hệ thống các tệp phân tán đó.

d. Ưu điểm nổi trội

- Nó là một cơ sở dữ liệu SQL thực, với bộ dữ liệu rất lớn.
- Nó được tích hợp công cụ BI, các trường sử dụng EDW, bảng ACID, ngoài ra nó còn tích hợp cả Hbase giúp xử lý thông tin, dữ liệu chính xác và nhanh chóng hơn.
- Nó hỗ trợ Spark mạnh mẽ, tương tác tốt với Druid, ngoài ra với cơ chế bảo mật dữ liệu mạnh mẽ, Apache Hive sẽ giúp ích rất nhiều cho các lập trình viên trong vấn đề bảo mật thông tin người dùng.
- Apache Hive hỗ trợ lưu trữ các loại tệp dữ liệu khác nhau trên HDFS bao gồm: Apache ORC, Apache Parquet, CSV, JSON, ACID

- Kết hợp SQL trên Hadoop (HPL & SQL)

e. Cách làm việc



Quy trình làm việc của Hive và Hadoop

Quy trình làm việc:

- Bước 1: Thực thi các dòng lệnh query: giao diện sử dụng của Hive giống như Command line, hoặc các giao diện người dùng web, gửi truy vấn đến trình điều khiển để thực thi các dòng lệnh
- Bước 2: Nhận kế hoạch: trình điều khiển với sự trợ giúp của trình biên dịch, sau đó phân tích các cú pháp truy cấp để kiểm tra các cú pháp, các kế hoạch và yêu cầu truy vấn.
- Bước 3: Nhận metadata: các trình biên dịch sẽ gửi yêu cầu nhận metadata đến Metastore.
- Bước 4: gửi kế hoạch: các trình biên dịch sau khi kiểm tra thật kỹ các yêu cầu sẽ gửi lại kế hoạch cho trình điều khiển xử lý tiếp. Và đến đây, thì việc phân tích cú pháp và biên dịch một truy vấn đã được hoàn tất.
- Bước 5: Thực hiện kế hoạch: trình điều khiển sẽ gửi kế hoạch ở phía trên đến các công cụ thực thi.

- Bước 6: Thực thi công việc: MapReduce sẽ có nhiệm vụ thực thi các công việc trên. Công cụ này sẽ gửi công việc đến các JobTracker ở bên trong node Name, sau đó nó gán công việc này cho các TaskTracker.
- Bước 7: các hoạt động của metadata: trong quá trình thực hiện, các công cụ thực thi sẽ triển khai các hoạt động của metadata với Metastore.
- Bước 8: Lấy kết quả: các công cụ thực thi sẽ lấy kết quả từ các node Data
- Bước 9: Gửi kết quả: sau khi thực thi xong, các công cụ sẽ gửi kết quả đến trình điều khiển, cuối cùng, các trình điều khiển sẽ gửi toàn bộ kết quả xử lý được đến giao diện Hive.
- Bước 10: các lập trình viên có thể sử dụng các kết quả được gửi đến Hive để phục vụ cho công việc của mình và hoàn thành các bước xử lý dữ liệu tiếp theo.

3. *DataWareHouse*

a. Giới thiệu

Data Warehouse có nghĩa là kho dữ liệu là một loại quản lý dữ liệu hệ thống được thiết kế để cho phép và hỗ trợ kinh doanh thông minh hoạt động BI, đặc biệt là phân tích. Data Warehouse chỉ nhằm mục đích thực hiện các truy vấn và phân tích và thường chứa một lượng lớn dữ liệu. Dữ liệu trong Data Warehouse thường được lấy từ nhiều nguồn như tệp nhật ký ứng dụng và ứng dụng giao dịch.

Data Warehouse tập trung và tổng hợp một lượng lớn dữ liệu từ nhiều nguồn. Khả năng phân tích Data Warehouse cho phép các tổ chức thu được những hiểu biết kinh doanh có giá trị từ dữ liệu của họ để cải thiện việc ra quyết định. Theo thời gian, nó xây dựng một hồ sơ lịch sử có thể là vô giá đối với các nhà Data Science và nhà phân tích kinh doanh.

Một Data Warehouse điển hình thường bao gồm các yếu tố sau:

- Một cơ sở dữ liệu quan hệ để lưu trữ và quản lý dữ liệu.

- Giải pháp trích xuất, tải và biến đổi ELT để chuẩn bị dữ liệu cho phân tích.
- Khả năng phân tích thống kê, báo cáo và khai thác dữ liệu.
- Các công cụ phân tích khách hàng để trực quan hóa và trình bày dữ liệu cho người dùng doanh nghiệp.
- Các ứng dụng phân tích khác, phức tạp hơn tạo ra thông tin có thể hành động bằng cách áp dụng khoa học dữ liệu và thuật toán trí tuệ nhân tạo AI hoặc các tính năng đồ thị và không gian cho phép nhiều loại phân tích dữ liệu hơn trên quy mô lớn.

Cơ sở dữ liệu hỗ trợ quyết định Data Warehouse được duy trì tách biệt với cơ sở dữ liệu hoạt động của tổ chức. Tuy nhiên, Data Warehouse không phải là một sản phẩm mà là một môi trường. Đây là một cấu trúc của một hệ thống thông tin cung cấp cho người dùng thông tin hỗ trợ quyết định hiện tại và quá khứ, cái mà khó truy cập hoặc hiện diện trong kho dữ liệu vận hành truyền thống.

Data Warehouse là cốt lõi của hệ thống BI được xây dựng để phân tích và báo cáo dữ liệu. Bạn có biết rằng một cơ sở dữ liệu được thiết kế 3NF cho một hệ thống kiểm kê, nhiều cơ sở có các bảng liên quan với nhau.

Ví dụ: Một báo cáo về thông tin hàng tồn kho hiện tại có thể bao gồm hơn 12 điều kiện tham gia. Điều này có thể nhanh chóng làm chậm thời gian phản hồi của truy vấn và báo cáo. Nhiệm vụ Data Warehouse cung cấp một thiết kế mới có thể giúp giảm thời gian phản hồi và giúp tăng cường hiệu suất của các truy vấn cho báo cáo và phân tích.

Hệ thống Data Warehouse còn được gọi bằng tên sau:

- Hệ thống hỗ trợ quyết định (DSS)
- Hệ thống điều hành thông tin
- Hệ thống thông tin quản lý
- Giải pháp kinh doanh thông minh
- Ứng dụng phân tích
- Kho dữ liệu

b. Lợi ích

Data Warehouse mang lại lợi ích bao trùm và duy nhất là cho phép các tổ chức phân tích một lượng lớn dữ liệu biến thể và trích xuất giá trị đáng kể từ nó, cũng như lưu giữ hồ sơ lịch sử.

Bốn đặc điểm độc đáo (được mô tả bởi nhà khoa học máy tính William Inmon, người được coi là cha đẻ của kho dữ liệu) cho phép các kho dữ liệu mang lại lợi ích bao trùm này là:

- Theo định hướng chủ đề: Họ có thể phân tích dữ liệu về một chủ đề hoặc lĩnh vực chức năng cụ thể chẳng hạn như bán hàng.
- Tích hợp: Kho dữ liệu tạo ra sự nhất quán giữa các kiểu dữ liệu khác nhau từ các nguồn khác nhau.
- Cố định dữ liệu: Khi dữ liệu nằm trong kho dữ liệu, nó ổn định và không thay đổi.
- Biến thể thời gian: Phân tích kho dữ liệu xem xét sự thay đổi theo thời gian.

Một Data Warehouse được thiết kế tốt sẽ thực hiện các truy vấn rất nhanh chóng, cung cấp thông tin lượng dữ liệu cao và cung cấp đủ tính linh hoạt cho người dùng cuối hoặc giảm khối lượng dữ liệu để kiểm tra kỹ hơn nhằm đáp ứng nhiều nhu cầu khác nhau cho dù ở mức độ rất tốt, chi tiết. Kho dữ liệu đóng vai trò là nền tảng chức năng cho môi trường phần mềm BI trung gian cung cấp cho người dùng cuối các báo cáo, trang tổng quan và các giao diện khác.

c. Cách thức hoạt động

Data Warehouse hoạt động như một kho lưu trữ trung tâm nơi thông tin đến từ một hoặc nhiều nguồn dữ liệu. Dữ liệu chảy vào kho dữ liệu từ hệ thống giao dịch và các cơ sở dữ liệu liên quan khác.

Dữ liệu có thể là/được:

1. Cấu trúc
2. Bán cấu trúc
3. Dữ liệu phi cấu trúc

Dữ liệu được xử lý, chuyển đổi và nhập để người dùng có thể truy cập dữ liệu đã xử lý trong Data Warehouse thông qua các công cụ Business Intelligence, SQL client và bảng tính. Data Warehouse hợp nhất thông tin đến từ các nguồn khác nhau vào một cơ sở dữ liệu toàn diện.

Bằng cách hợp nhất tất cả các thông tin này ở một nơi, một tổ chức có thể phân tích khách hàng của mình một cách toàn diện hơn. Điều này giúp đảm bảo rằng nó đã xem xét tất cả các thông tin có sẵn. Data Warehouse làm cho khai thác dữ liệu là có thể làm được. Khai thác dữ liệu đang tìm kiếm các mẫu trong dữ liệu để có được doanh thu và lợi nhuận cao hơn.

d. Phân loại

- *Enterprise Data Warehouse (Data Warehouse doanh nghiệp)*

  Data Warehouse doanh nghiệp hay còn gọi kho dữ liệu doanh nghiệp là một kho tập trung. Chức năng cung cấp dịch vụ hỗ trợ quyết định trên toàn doanh nghiệp. Ngoài ra cung cấp một cách tiếp cận thống nhất để tổ chức và đại diện dữ liệu. Và thêm nữa là cung cấp khả năng phân loại dữ liệu theo chủ đề và cấp quyền truy cập theo các bộ phận đó.

- *Operational Data Store (Kho lưu trữ dữ liệu hoạt động)*

  Kho lưu trữ dữ liệu hoạt động, còn được gọi là ODS, không có gì ngoài kho lưu trữ dữ liệu cần thiết khi cả Data Warehouse và hệ thống OLTP không hỗ trợ các tổ chức báo cáo nhu cầu. Trong ODS, kho dữ liệu được làm mới theo thời gian. Do đó, nó được ưa thích rộng rãi cho các hoạt động thường ngày như lưu trữ hồ sơ của nhân viên.

- *Data Mart*

  Một data mart là một tập hợp con của Data Warehouse, được thiết kế đặc biệt cho một ngành kinh doanh cụ thể, chẳng hạn như bán hàng, tài chính,

bán hàng hoặc tài chính. Trong một data mart độc lập, dữ liệu có thể thu thập trực tiếp từ các nguồn.

e. Thành phần

Quản lý phụ tải: Quản lý phụ tải còn được gọi là quản lý phía cầu. Nó thực hiện với tất cả các hoạt động liên quan đến việc trích xuất và tải dữ liệu vào kho. Các hoạt động này bao gồm các phép biến đổi để chuẩn bị dữ liệu để nhập vào kho dữ liệu.

Quản lý warehouse: Quản lý warehouse thực hiện các hoạt động liên quan đến việc quản lý dữ liệu trong kho, được thực hiện các hoạt động như phân tích dữ liệu để đảm bảo tính nhất quán, tạo các chỉ mục và khung nhìn, tạo ra sự không chuẩn hóa và tổng hợp, chuyển đổi và hợp nhất dữ liệu nguồn và lưu trữ và dữ liệu.

Trình quản lý truy vấn: Trình quản lý truy vấn còn được gọi là thành phần phụ trợ. Nó thực hiện tất cả các hoạt động liên quan đến việc quản lý các truy vấn của người dùng. Các hoạt động của các thành phần Data Warehouse này là các truy vấn trực tiếp đến các bảng thích hợp để lên lịch thực hiện các truy vấn.

Công cụ truy cập của người dùng cuối:

Công cụ này được phân loại thành năm nhóm khác nhau như:

- Báo cáo dữ liệu;
- Công cụ truy vấn;
- Công cụ phát triển ứng dụng;
- Công cụ EIS;
- Công cụ OLAP và công cụ khai thác dữ liệu.

f. Đối tượng sử dụng

Data Warehouse là cần thiết cho tất cả các loại người dùng như:

- Những người ra quyết định dựa vào khối lượng dữ liệu.
- Người dùng sử dụng các quy trình phức tạp, tùy chỉnh để lấy thông tin từ nhiều nguồn dữ liệu.

- Nó cũng được sử dụng bởi những người muốn công nghệ đơn giản để truy cập dữ liệu
- Nó cũng cần thiết cho những người muốn có một cách tiếp cận có hệ thống để đưa ra quyết định.
- Nếu người dùng muốn hiệu suất nhanh trên một lượng dữ liệu khổng lồ cần thiết cho các báo cáo, lưới hoặc biểu đồ, thì Data Warehouse sẽ trở nên hữu ích.
- Data Warehouse là bước đầu tiên nếu bạn muốn khám phá 'các mẫu ẩn' của luồng dữ liệu và nhóm.

g. Kiến trúc

Kiến trúc của một Data Warehouse được xác định bởi các nhu cầu cụ thể của tổ chức. Các kiến trúc phổ biến bao gồm:

Simple: Tất cả các Data Warehouse đều có chung một thiết kế cơ bản, trong đó siêu dữ liệu, dữ liệu tóm tắt và dữ liệu thô được lưu trữ trong kho lưu trữ trung tâm của kho. Kho lưu trữ được cung cấp bởi các nguồn dữ liệu ở một đầu và được người dùng cuối truy cập để phân tích, báo cáo và khai thác ở đầu kia.

Simple with a staging area: Dữ liệu hoạt động phải được làm sạch và xử lý trước khi đưa vào kho. Mặc dù điều này có thể được thực hiện theo chương trình, nhiều kho dữ liệu bổ sung thêm một vùng phân bố cho dữ liệu trước khi dữ liệu vào kho, để đơn giản hóa việc chuẩn bị dữ liệu.

Hub and spoke: Việc thêm các kho dữ liệu giữa kho lưu trữ trung tâm và người dùng cuối cho phép một tổ chức tùy chỉnh kho dữ liệu của mình để phục vụ các ngành kinh doanh khác nhau. Khi dữ liệu đã sẵn sàng để sử dụng, nó sẽ được chuyển đến data mart thích hợp.

Sandboxes: Sandboxes là các khu vực riêng tư, bảo mật, an toàn cho phép các công ty khám phá nhanh chóng và không chính thức các bộ dữ liệu mới hoặc các cách phân tích dữ liệu mà không cần phải tuân thủ hoặc tuân thủ các quy tắc và giao thức chính thức của kho dữ liệu.

h. Sự phát triển

Khi các kho dữ liệu lần đầu tiên xuất hiện vào cuối những năm 1980, mục đích của chúng là giúp dữ liệu chuyển từ các hệ thống vận hành sang các hệ thống hỗ trợ quyết định DSS. Những kho dữ liệu ban đầu này đòi hỏi một lượng lớn dự phòng. Hầu hết các tổ chức có nhiều môi trường DSS phục vụ những người dùng khác nhau. Mặc dù các môi trường DSS sử dụng nhiều dữ liệu giống nhau, việc thu thập, làm sạch và tích hợp dữ liệu thường được sao chép cho từng môi trường.

Khi các kho dữ liệu trở nên hiệu quả hơn, chúng đã phát triển từ các kho thông tin hỗ trợ nền tảng BI truyền thống thành các cơ sở hạ tầng phân tích rộng rãi hỗ trợ nhiều loại ứng dụng, chẳng hạn như phân tích hoạt động và quản lý hiệu suất. Việc lặp lại kho dữ liệu đã tiến triển theo thời gian để mang lại giá trị gia tăng gia tăng cho doanh nghiệp.

Ngày nay, AI và máy học đang biến đổi hầu hết mọi ngành, dịch vụ và tài sản doanh nghiệp và Data Warehouse cũng không ngoại lệ. Việc mở rộng dữ liệu lớn và ứng dụng các công nghệ kỹ thuật số mới đang thúc đẩy sự thay đổi về các yêu cầu và khả năng của kho dữ liệu.

Các kho dữ liệu độc lập là bước đi mới nhất trong quá trình nâng cấp này, cung cấp các doanh nghiệp khả năng trích xuất giá trị lớn hơn từ dữ liệu trong khi giảm chi phí và cải thiện độ tin cậy và hiệu suất kho dữ liệu.

i. Cloud Data Warehouse

Cloud Data Warehouse sử dụng đám mây để nhập và lưu trữ dữ liệu từ các nguồn dữ liệu khác nhau.

Các kho dữ liệu ban đầu được xây dựng với các máy chủ tại chỗ. Các kho dữ liệu tại chỗ này tiếp tục có nhiều lợi thế ngày nay. Trong nhiều trường hợp, chúng có thể cải thiện khả năng quản trị, bảo mật, chủ quyền dữ liệu và độ trễ tốt hơn. Tuy nhiên, kho dữ liệu tại chỗ không co giãn bằng và chúng yêu cầu dự báo phức tạp để xác định cách mở rộng kho dữ liệu cho các nhu cầu trong tương lai. Việc quản lý các kho dữ liệu này cũng có thể rất phức tạp.

Mặt khác, một số ưu điểm của Cloud Data Warehouse bao gồm:

- Hỗ trợ co giãn, mở rộng quy mô cho các yêu cầu lưu trữ hoặc tính toán lớn hoặc thay đổ.
- Dễ sử dụng.
- Dễ quản lý.
- Tiết kiệm chi phí.

Các kho dữ liệu đám mây tốt nhất được quản lý hoàn toàn, đảm bảo rằng ngay cả những người mới bắt đầu cũng có thể tạo và sử dụng kho dữ liệu chỉ với một vài cú nhấp chuột. Một cách dễ dàng để bắt đầu di chuyển sang Cloud Data Warehouse là chạy kho dữ liệu đám mây của bạn tại chỗ, đằng sau tường lửa trung tâm dữ liệu tuân thủ các yêu cầu về chủ quyền và bảo mật dữ liệu. Ngoài ra, hầu hết các kho dữ liệu đám mây đều tuân theo mô hình và trả tiền khi sử dụng, giúp tiết kiệm thêm chi phí cho khách hàng.

j. Modern Data Ware House

Cho dù họ là thành viên của nhóm CNTT, kỹ thuật dữ liệu, phân tích kinh doanh hay khoa học dữ liệu, những người dùng khác nhau trong tổ chức có nhu cầu khác nhau về kho dữ liệu.

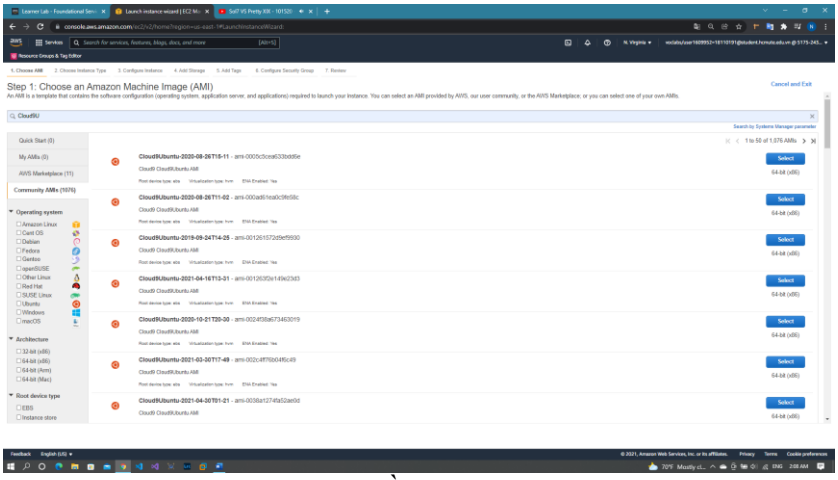
Một kiến trúc dữ liệu hiện đại giải quyết những nhu cầu khác nhau bằng cách cung cấp một cách để quản lý tất cả các loại dữ liệu, khối lượng công việc, và phân tích. Bao gồm các mẫu kiến trúc với các thành phần cần thiết được tích hợp để làm việc cùng nhau theo các phương pháp hay nhất trong ngành. Modern Data Warehouse bao gồm:

- Cơ sở dữ liệu hội tụ giúp đơn giản hóa việc quản lý tất cả các loại dữ liệu và cung cấp các cách khác nhau để sử dụng dữ liệu.
- Dịch vụ nhập và chuyển đổi dữ liệu tự phục vụ.
- Hỗ trợ xử lý SQL, máy học, đồ thị và không gian.
- Nhiều tùy chọn phân tích giúp bạn dễ dàng sử dụng dữ liệu mà không cần di chuyển dữ liệu.
- Quản lý tự động để cung cấp, mở rộng quy mô và quản trị đơn giản.

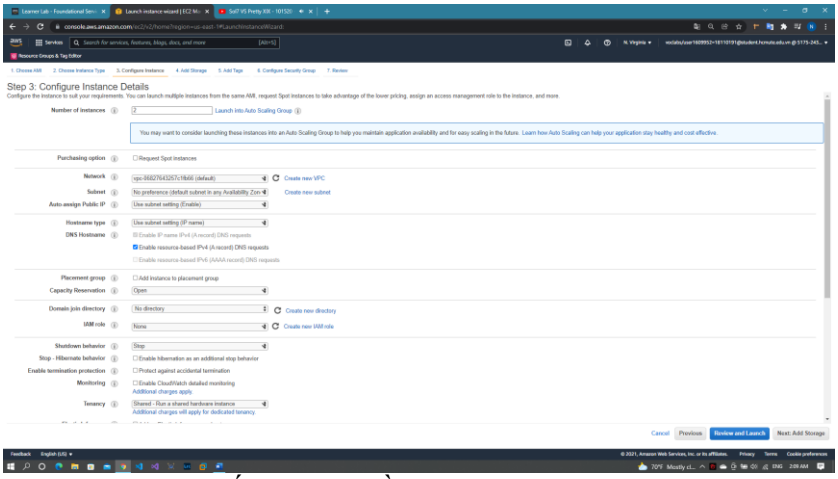
Một kho dữ liệu hiện đại có thể hợp lý hóa quy trình công việc dữ liệu một cách hiệu quả theo cách mà các kho khác không làm được. Điều này có nghĩa là tất cả mọi người, từ các nhà phân tích và kỹ sư dữ liệu đến các nhà khoa học dữ liệu và nhóm CNTT, có thể thực hiện công việc hiệu quả hơn và theo đuổi công việc đổi mới đưa tổ chức tiến lên mà không có sự chậm trễ và phức tạp.

4. Demo DataWareHouse

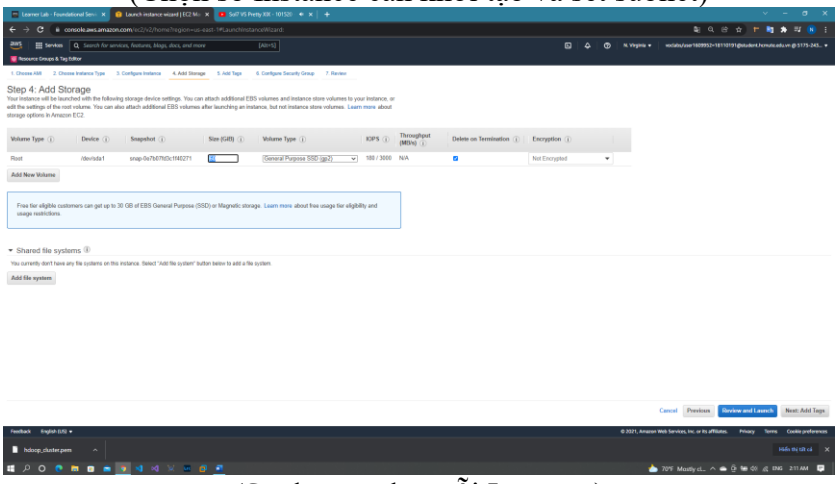
a. Cài đặt máy chủ EC2



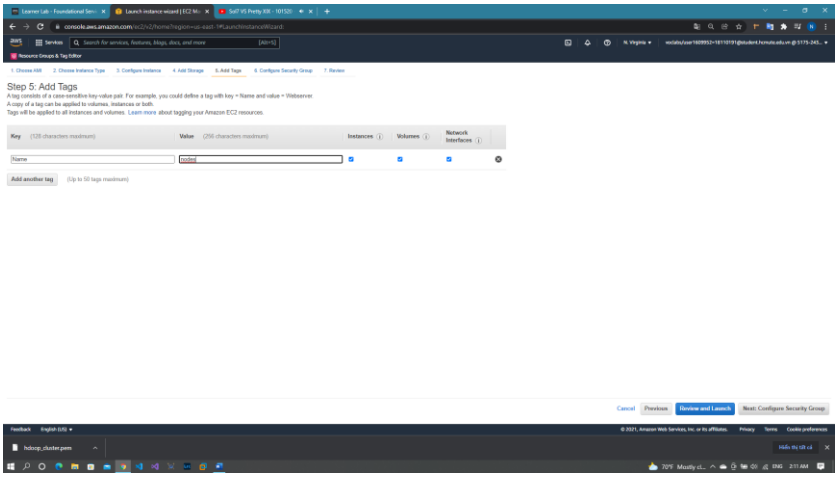
(Chọn máy chủ cần khởi tạo-Cloud9U)



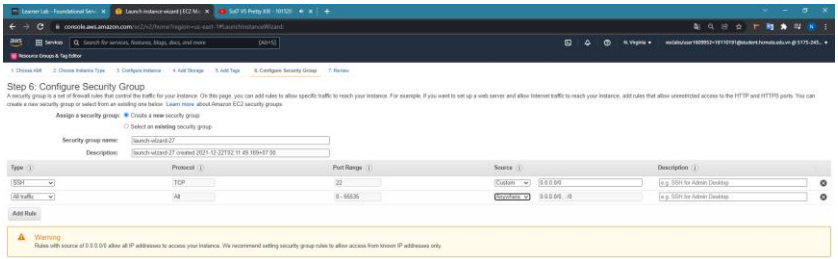
(Chọn số Instance cần khởi tạo và set subnet)



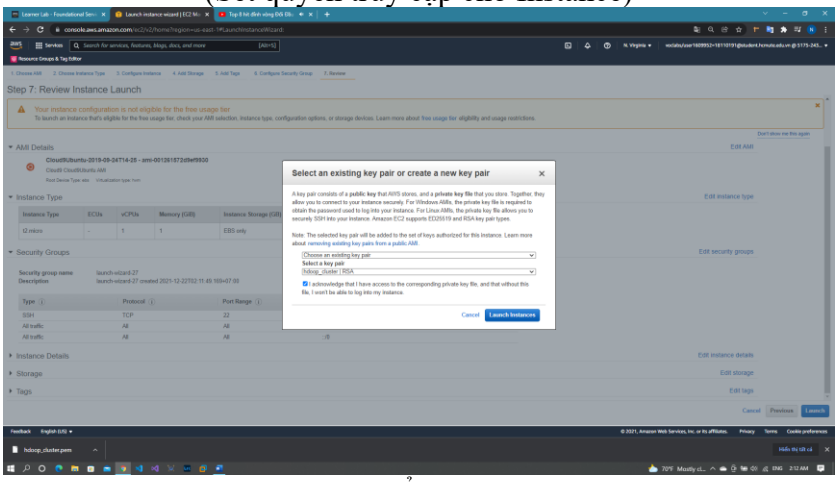
(Set lượng cho mỗi Instance)



(Set Key cho từng Instance)



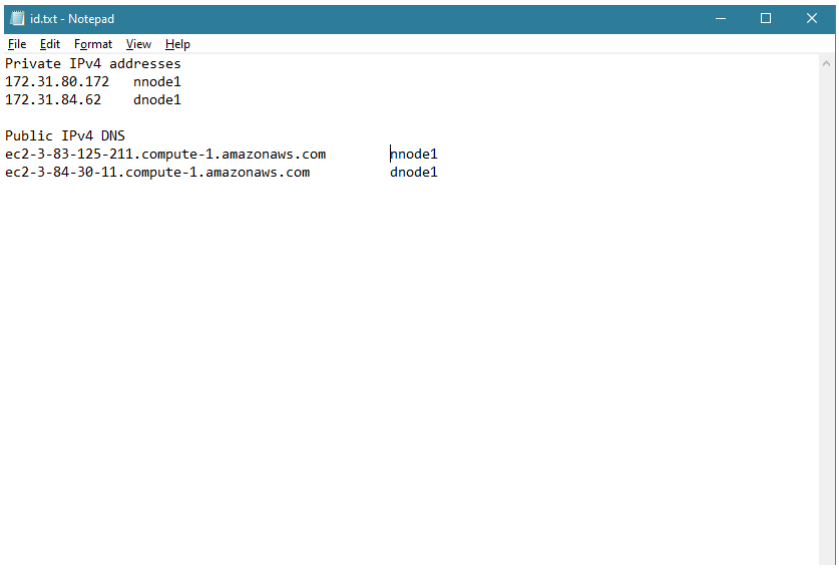
(Set quyền truy cập cho Instance)



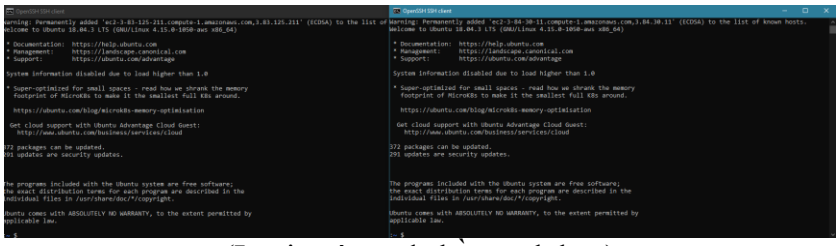
(Khởi tạo Key Pair để truy cập các Instance)

(Kiểm tra các Instance đã khởi tạo)

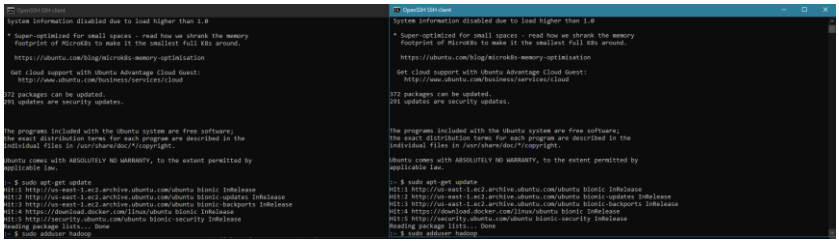
b. Cài đặt Hadoop



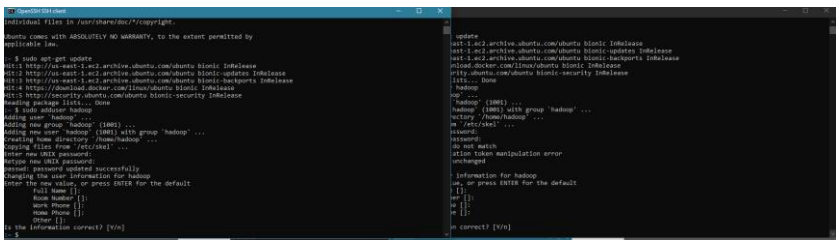
(Lưu lại Private IP và Public DNS của từng node)

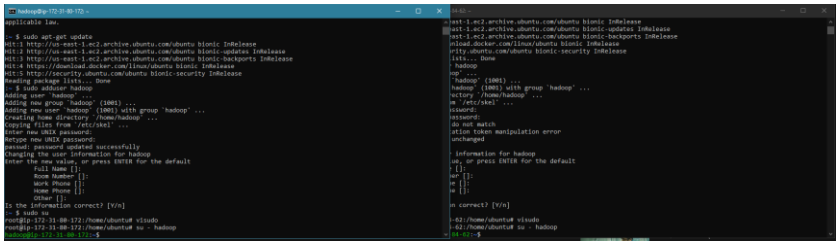


(Login các node bằng ssh key)

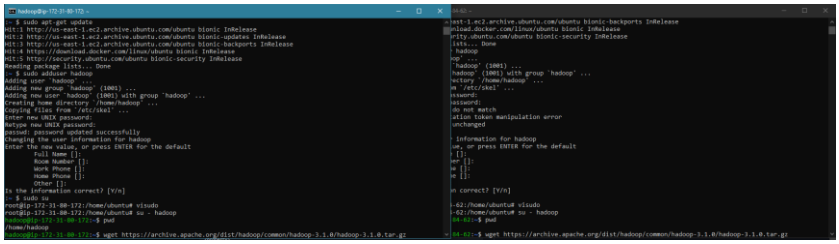


(Khởi tạo user cho các node)

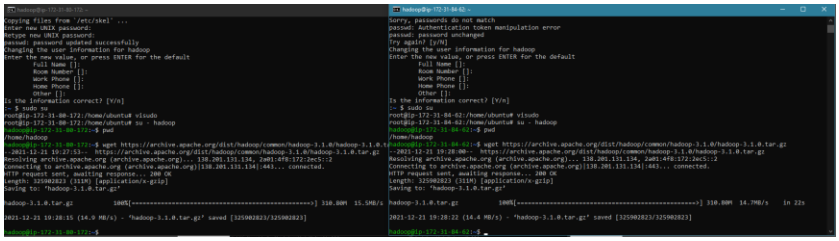


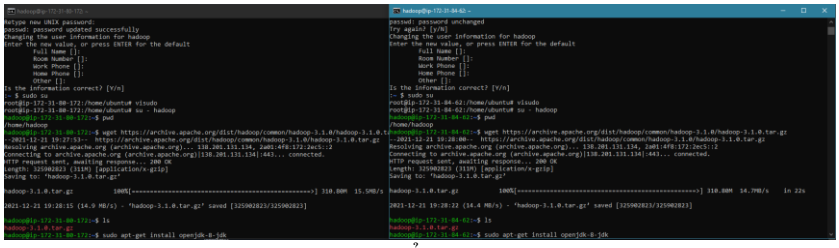


(Truy cập và làm việc với user riêng)

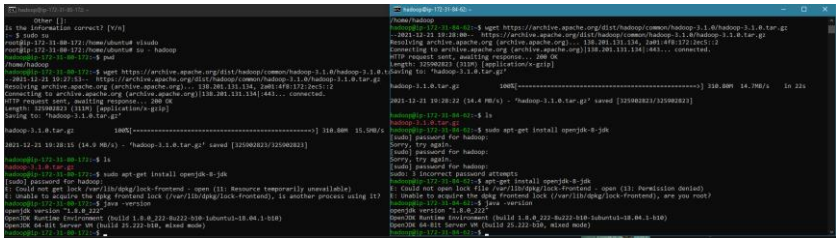


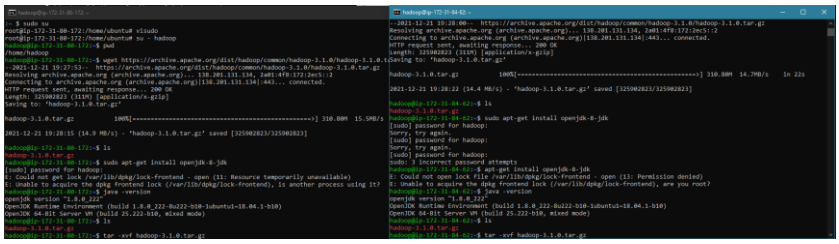
(Dowload Hadoop)



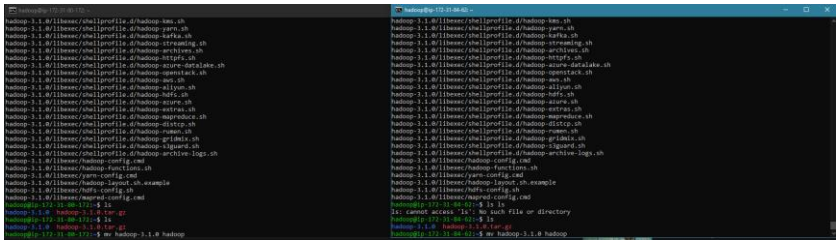


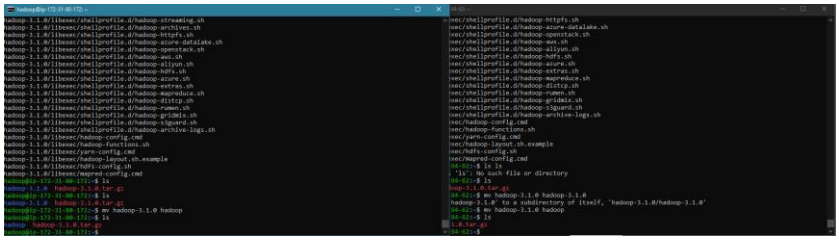
(Dowload và kiểm tra Java JDK)

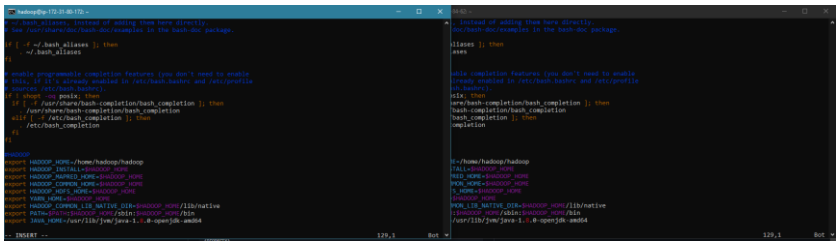




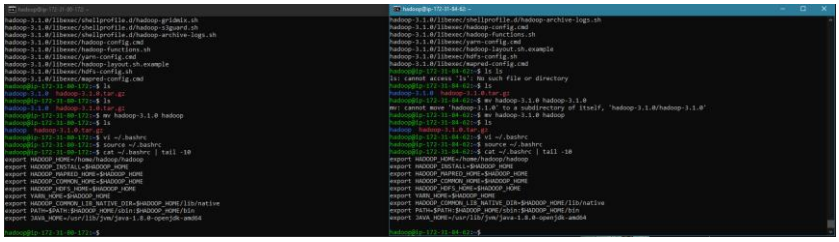
(Giải nén file Hadoop)

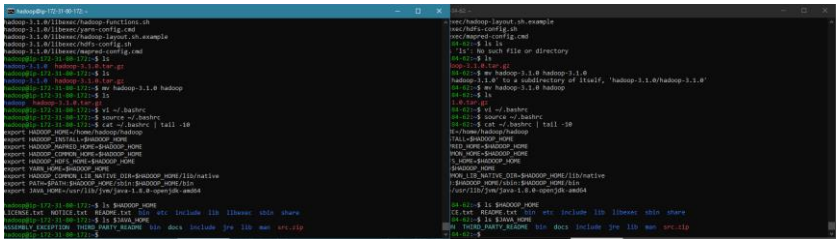


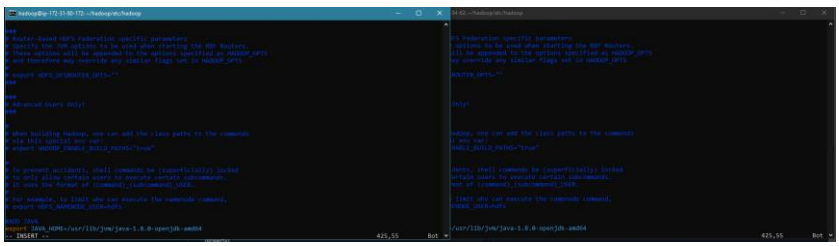




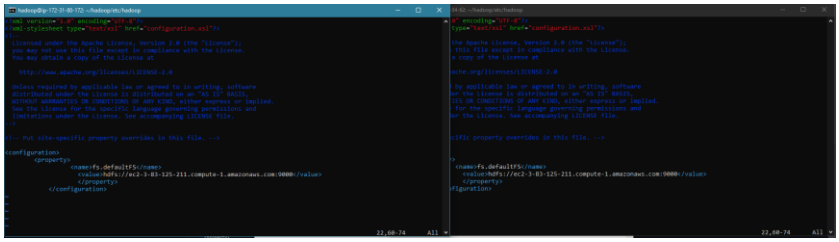
(Edit lại file .bashrc)

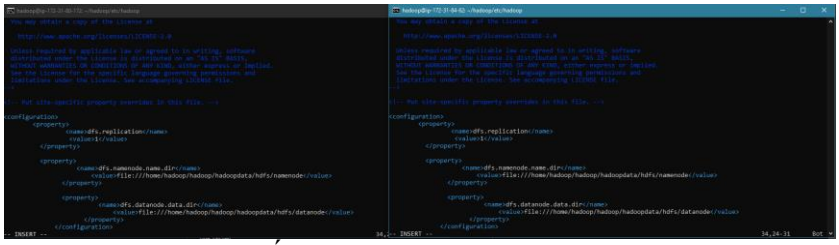




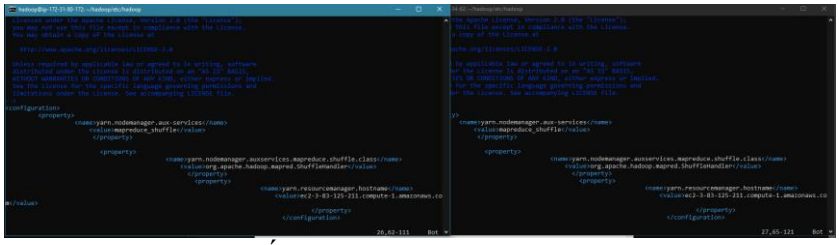


(Cấu hình file Core-site.xml)

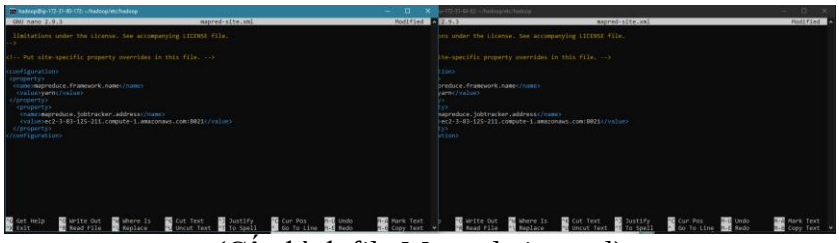




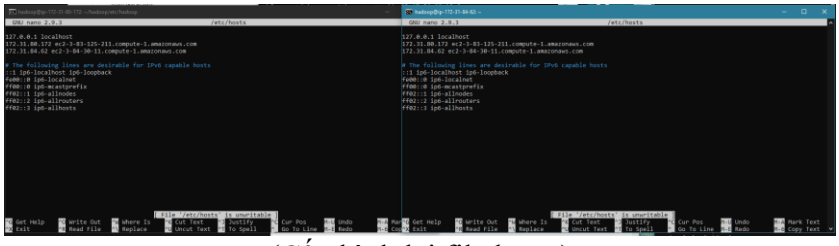
(Cấu hình file hdfs-site.xml)



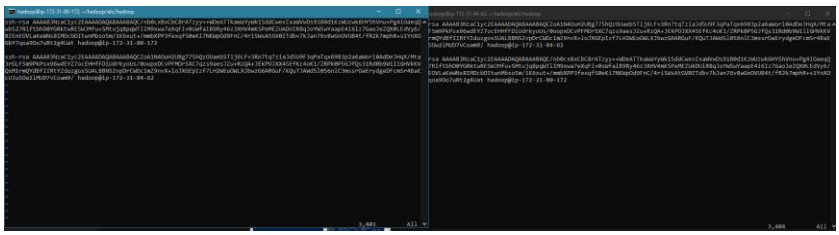
(Cấu hình file Yarn-site.xml)



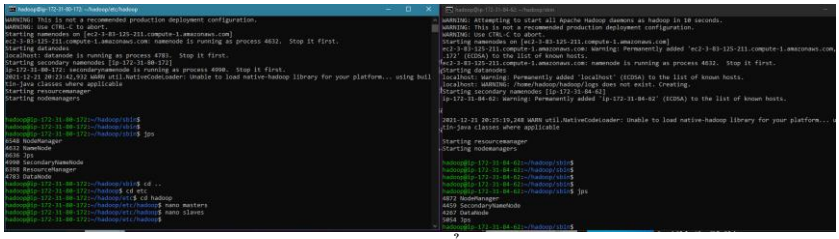
(Cấu hình file Mapred-site.xml)



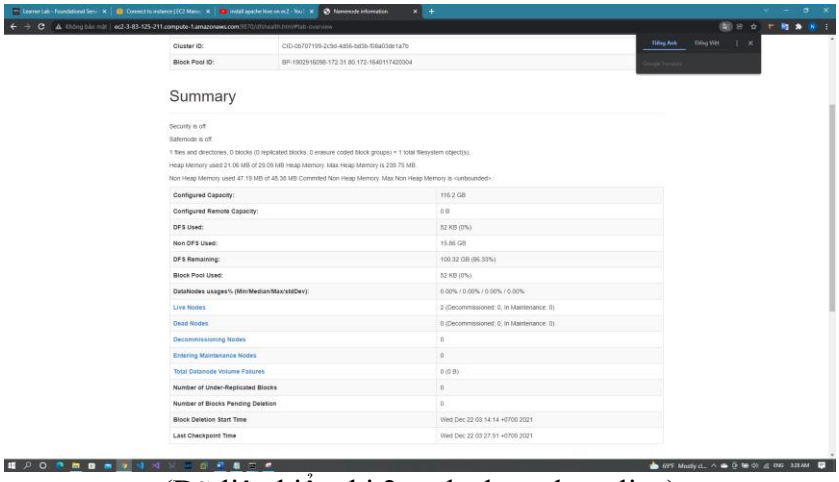
(Cấu hình lại file hosts)



(Sao chép ssh key cho 2 node)

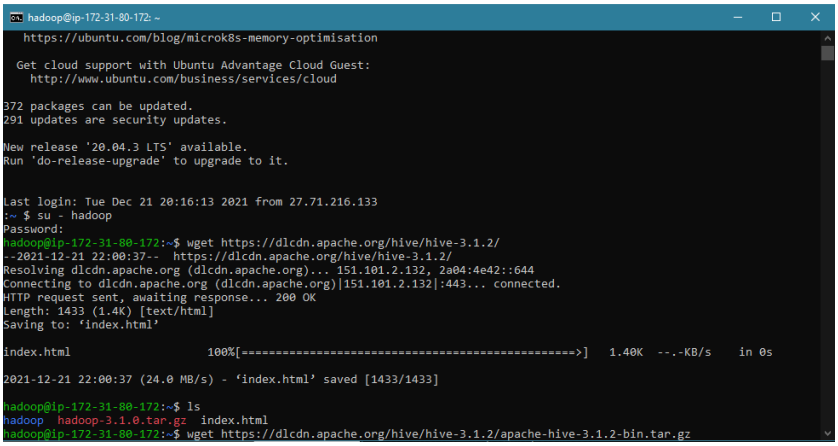


(Chạy và kiểm tra)

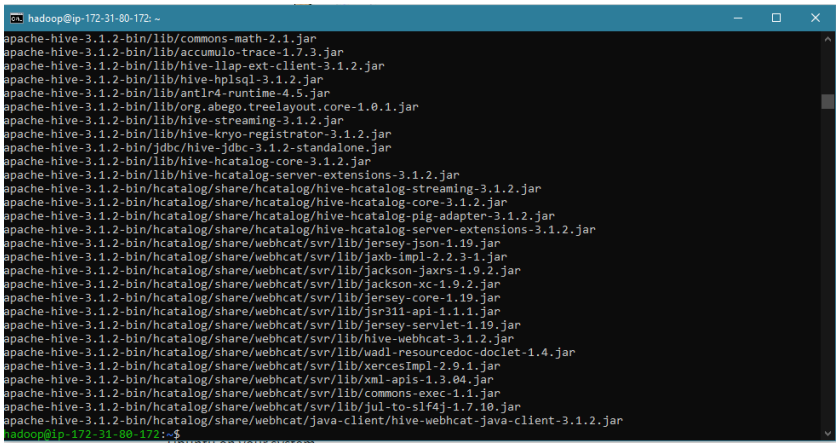


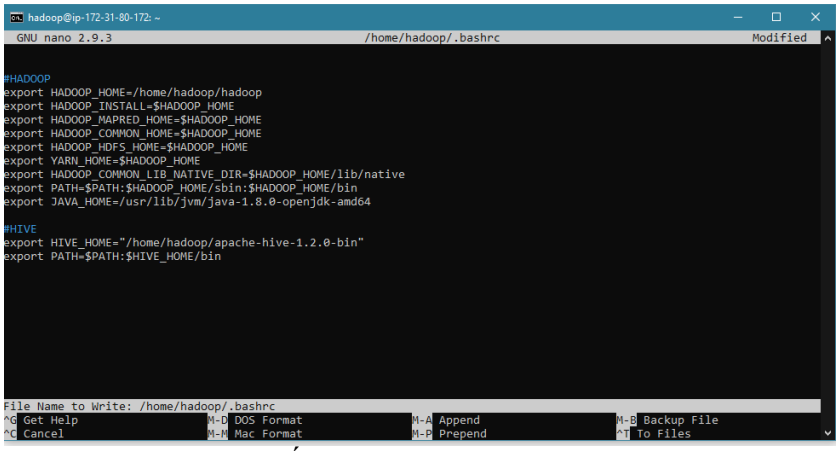
(Dữ liệu hiển thị 2 node đang được live)

c. Cài đặt Hive

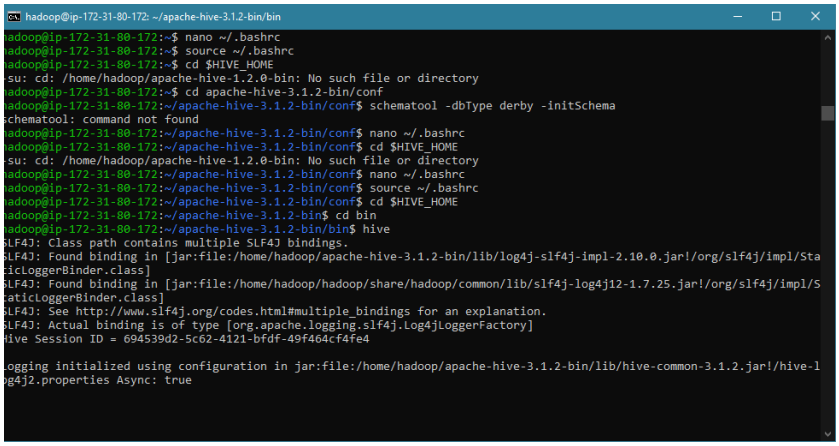


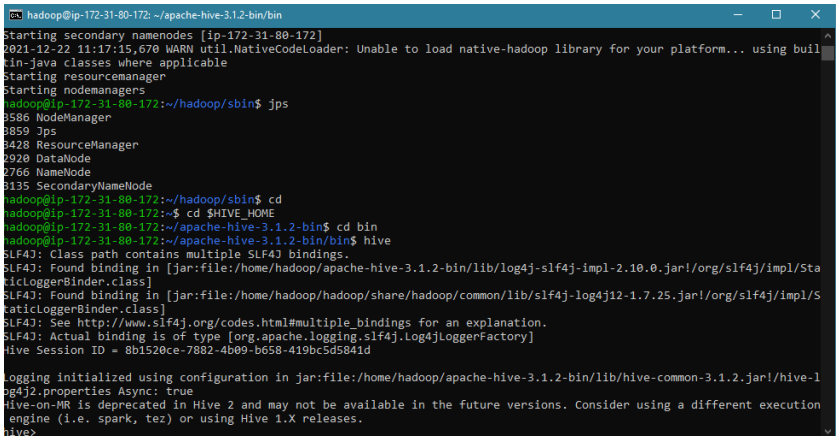
(Tải Hive)





(Cấu hình lại file .bashrc)





(Khởi động hive)

d. Sử dụng HiveQL

Demo sử dụng câu lệnh

// Khởi tạo data mark

```
--Bảng tính tổng số ca trong từng năm tong_so_ca
CREATE TABLE tong_so_ca
AS(SELECT thoi_gian,sum(tong_so)
AS tong_so_ca
FROM covid_19
GROUP BY thoi_gian);

--Số ca dương tính của Trung Quốc
CREATE TABLE so_lieu_tau
AS(SELECT Country, sum(tong_so)
AS tong_so_ca
FROM covid_19
WHERE Country='China'
GROUP BY Country);

--Tổng số ca dương tính theo tên nước
```

```
CREATE TABLE tong_so_ca_theo_quoc_gia
AS(SELECT Country, sum(tong_so)
AS tong_so_Ca FROM covid_19
GROUP BY Country);


--Tổng số ca dương tính năm 2015 của các nước
SELECT Country,Sum(so_luong)
FROM covid_19
WHERE thoi_gian
LIKE '%2015%' GROUP BY Country;

--Thống kê số ca theo năm
CREATE TABLE so_ca_theo_nam_duong_lich
AS(SELECT thoi_gian, Sum(tong_so) as tong_so
FROM covid_19
GROUP BY thoi_gian);

--Thống kê số ca theo tháng của 1 năm
CREATE TABLE so_ca_theo_thang_trong_mot_nam
AS(SELECT Count(Country), Sum(so_ca)
FROM covid_19
WHERE Date_time LIKE '%09/2015%'
GROUP BY Country);

--Thống kê số ca từng năm của từng nước
CREATE TABLE tong_so_ca_theo_tung_nam_o_cac_nuoc
AS(SELECT Country,thoi_gian, Sum(tong_so)
AS tong_so_ca
```

```
FROM covid_19
GROUP BY Country, thoi_gian);


///CÂU TRUY VẤN
--Nước có số ca nhiễm lớn nhất theo năm cần tìm
SELECT Country,MAX(tong_so_ca)
FROM tong_so_ca_theo_tung_nam_o_cac_nuoc
WHERE Country='Japan'
AND thoi_gian='2015'
GROUP BY Country;

SELECT Country,MAX(tong_so_ca)
FROM tong_so_ca_theo_tung_nam_o_cac_nuoc
WHERE thoi_gian='2015'
GROUP BY Country;

--Số ca trong năm 2015
 SELECT  * FROM so_ca_theo_nam_duong_lich
WHERE thoi_gian=2015;

--Năm có tổng số ca cao nhất
SELECT * FROM so_ca_theo_nam_duong_lich
WHERE so_ca=(SELECT MAX(so_ca)
FROM so_ca_theo_nam_duong_lich);

--Tìm nước có số ca nhiễm cao nhất năm 2015
SELECT * FROM tong_so_ca_theo_tung_nam_o_cac_nuoc
WHERE thoi_gian=2015
```

```
AND tong_so_ca=(SELECT MAX(tong_so_ca)
FROM tong_so_ca_theo_tung_nam_o_cac_nuoc
WHERE thoi_gian=2015);


#Exports to HDFS directory
INSERT OVERWRITE DIRECTORY '/home/hdoop/epdata/data.txt' ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT * FROM
tong_so_ca limit ;


#Exports to LOCAL directory
INSERT OVERWRITE LOCAL DIRECTORY
'/home/hdoop/epdata/export' ROW FORMAT DELIMITED FIELDS
TERMINATED BY ',' SELECT * FROM emp.tong_so_ca;

cat /home/hdoop/epdata/export/000000_0
```

III. TỔNG KẾT

MỨC ĐỘ HOÀN THÀNH

STT	TÊN THÀNH VIÊN	MỨC ĐỘ HOÀN THÀNH
1	Nguyễn Xuân Sang	100%
2	Trần Trung Hiếu	100%