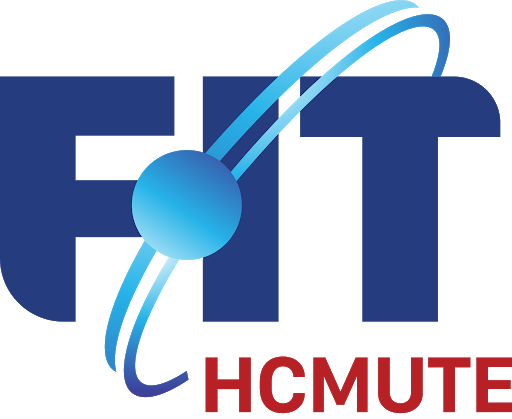
**.**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN ĐIỆN TOÁN ĐÁM MÂY**

-----🙞🙜🕮🙞🙜-----

****

**NGUYỄN THỊ NHẢ THƯ - 19133054**

**VÕ THÀNH ĐẠT - 19133019**

**Đề Tài:**

**TÌM HIỂU APACHE HIVE VÀ VIẾT ỨNG DỤNG DEMO**

**GIẢNG VIÊN HƯỚNG DẪN**

**TS. HUỲNH XUÂN PHỤNG**

**KHÓA 2019 – 2023**

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP. HCM**  **KHOA CNTT**  \*\*\*\*\*\*\* | **CỘNG HÒA XÃ 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: **Nguyễn Thị Nhả Thư** MSSV 1: **19133054**

Họ và tên Sinh viên 2: **Võ Thành Đạt** MSSV 2: **19133019**

Ngành: **Kỹ thuật dữ liệu**

Tên đề tài: **Tìm hiểu Apache Hive và viết ứng dụng demo**

Họ và tên Giáo viên hướng dẫn: **TS. Huỳnh Xuân Phụng**

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

|  |  |  |  |
| --- | --- | --- | --- |
| *Tp. Hồ Chí Minh, ngày tháng năm 2021* | | | |
|  | | Giáo viên hướng dẫn  *(Ký & ghi rõ họ tên)* |
| **ĐH SƯ PHẠM KỸ THUẬT TP. HCM**  **KHOA CNTT**  \*\*\*\*\*\*\* | | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh Phúc**  \*\*\*\*\*\*\* | | | |

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên 1: **Nguyễn Thị Nhả Thư** MSSV 1: **19133054**

Họ và tên Sinh viên 2: **Võ Thành Đạt** MSSV 2: **19133019**

Ngành: **Kỹ thuật dữ liệu**

Tên đề tài: **Tìm hiểu Apache Hive và viết ứng dụng demo**

Họ và tên Giáo viên phản biện: **TS. Huỳnh Xuân Phụng**

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

*Tp. Hồ Chí Minh, ngày tháng năm 2021*

Giáo viên phản biện

(Ký & ghi rõ họ tên)

# **LỜI CẢM ƠN**

*Lời đầu tiên nhóm xin phép được gửi lời cảm ơn chân thành và sâu sắc nhất đến với Khoa Công Nghệ Thông Tin – Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh đã tạo điều kiện cho nhóm chúng em được học tập, phát triển nền tảng kiến thức sâu sắc và thực hiện đề tài này.*

*Bên cạnh đó nhóm chúng em xin gửi đến thầy Huỳnh Xuân Phụng lời cảm ơn sâu sắc nhất. Trải qua một quá trình dài học tập và thực hiện đề tài trong thời gian qua. Thầy đã tận tâm chỉ bảo nhiệt tình nhóm chúng em trong suốt quá trình từ lúc bắt đầu cũng như kết thúc đề tài này.*

*Với sự hướng dẫn nhiệt tình, giảng dạy tận tình đầy đủ kiến thức của thầy Huỳnh Xuân Phụng, chúng em đã học tập và hiểu được những kiến thức cơ bản về Apache Hive – một dạng kho dữ liệu được sử dụng để quản lý và phân tích khối lượng dữ liệu lớn. Qua đó tụi em biết cách cài đặt và sử dụng Apache Hive.*

*Tuy nhiên lượng kiến thức là vô tận và với khả năng hạn hẹp chúng em đã rất cố gắng để hoàn thành một cách tốt nhất. Chính vì vậy việc xảy ra những thiếu sót là điều khó có thể tránh khỏi. Chúng em hi vọng nhận được sự góp ý tận tình của quý thầy (cô) qua đó chúng em có thể rút ra được bài học kinh nghiệm và hoàn thiện và cải thiện nâng cấp lại sản phẩm của mình một cách tốt nhất có thể.*

*Chúng em xin chân thành cảm ơn!*

***Nhóm thực hiện***

*Nguyễn Thị Nhả Thư - 19133054*

*Võ Thành Đạt – 19133019*

**MỤC LỤC**

[LỜI CẢM ƠN 4](#_Toc90051286)

[MỤC LỤC 5](#_Toc90051287)

[DANH MỤC CÁC HÌNH 7](#_Toc90051288)

[PHẦN MỞ ĐẦU 8](#_Toc90051289)

[**1. Tính cấp thiết của đề tài** 8](#_Toc90051290)

[**2. Đối tượng nghiên cứu** 8](#_Toc90051291)

[**3. Phạm vi nghiên cứu** 8](#_Toc90051292)

[**4. Kết quả dự kiến đạt được** 8](#_Toc90051293)

[PHẦN NỘI DUNG 9](#_Toc90051294)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VỀ APACHE HIVE 9](#_Toc90051295)

[**1.1** **Tổng quan về Hive** 9](#_Toc90051296)

[**1.2** **Kiến trúc của Hive** 10](#_Toc90051297)

[**1.3** **Hoạt động của Hive** 11](#_Toc90051298)

[**1.4** **Mô hình dữ liệu trong Hive** 12](#_Toc90051299)

[*1.4.1. Tổ chức dữ liệu* 12](#_Toc90051300)

[*1.4.2. Kiểu dữ liệu* 13](#_Toc90051301)

[**1.5** **HiveSQL (HQL)** 14](#_Toc90051302)

[CHƯƠNG 2: CÀI ĐẶT, THIẾT KẾ VÀ XÂY DỰNG DATA WAREHOUSE 15](#_Toc90051303)

[**2.1 Cài đặt** 15](#_Toc90051304)

[*2.1.1 Cài đặt môi trường Hadoop* 15](#_Toc90051305)

[*2.1.2 Cài đặt Apache Hive* 22](#_Toc90051306)

[*2.1.3 Lỗi phát sinh trong quá trình cài đặt* 25](#_Toc90051307)

[**2.2 Triển khai Hive Web Interface** 27](#_Toc90051308)

[**2.3 Thiết kế Data warehouse** 32](#_Toc90051309)

[**2.4 Xây dựng Data warehouse** 33](#_Toc90051310)

[*2.4.1 Tạo các file dữ liệu* 33](#_Toc90051311)

[*2.4.2 Tạo Database trong Hive* 34](#_Toc90051312)

[*2.4.3 Tạo Table và load dữ liệu* 34](#_Toc90051313)

[**2.5 Truy vấn và báo cáo** 41](#_Toc90051314)

[PHẦN KẾT LUẬN 45](#_Toc90051315)

[**1.** **Kết quả đạt được** 45](#_Toc90051316)

[*1.1. Kiến thức tìm hiểu được* 45](#_Toc90051317)

[*1.2. Chương trình đã làm được* 45](#_Toc90051318)

[**2.** **Ưu điểm** 45](#_Toc90051319)

[**3.** **Nhược điểm** 45](#_Toc90051320)

[**4.** **Hướng phát triển** 46](#_Toc90051321)

[TÀI LIỆU THAM KHẢO 47](#_Toc90051322)

# **DANH MỤC CÁC HÌNH**

[**Hình 1: Kiến trúc của Hive** 10](#_Toc90049634)

[**Hình 2: Sơ đồ luồng hoạt động của Hive** 10](#_Toc90049635)

[**Hình 3: Tổ chức dữ liệu trong Hive** 11](#_Toc90049636)

# **PHẦN MỞ ĐẦU**

## **1. Tính cấp thiết của đề tài**

Hiện nay, thuật ngữ Big Data được sử dụng cho các bộ tập dữ liệu khổng lồ bao gồm khối lượng lớn, tốc độ cao và nhiều loại dữ liệu đang tăng lên từng ngày. Sử dụng các hệ thống quản lý dữ liệu truyền thống, rất khó để xử lý Big Data. Do đó, Quỹ phần mềm Apache (Apache Software Foundation) đã giới thiệu một framework tên là Hadoop và trong đó là Apache Hive để giải quyết các thách thức quản lý và xử lý Big Data.

Theo.

## **2. Đối tượng nghiên cứu**

Đối với đề tài này, đối tượng nghiên cứu là Big Data. Đồng thời kèm theo đó là các công nghệ áp dụng để xây dựng warehouse đơn giản, cụ thể như:

* JDK và JRE: Bộ công cụ phát triển Java.
* Hadoop: 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.
* Apache Hive: Công cụ cơ sở hạ tầng kho dữ liệu để xử lý dữ liệu có cấu trúc trong Hadoop.
* Apache Derby: External database để cấu hình Metastore.

## **3. Phạm vi nghiên cứu**

Đề tài này chủ yếu tập trung vào việc xử lý và phân tích các câu truy vấn đối với dữ liệu lớn khi sử dụng Hive.

## **4. Kết quả dự kiến đạt được**

* Cài đặt được Hadoop và Apache Hive
* Xây dựng được một data warehouse đơn giản bằng các câu truy vấn.
* Phân tích, thống kê các dữ liệu từ các câu truy vấn tùy vào mục đích của người dùng.
* Cài đặt giao diện Hive Web Interface.

# **PHẦN NỘI DUNG**

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VỀ APACHE HIVE**

## **Tổng quan về Hive**

Apache Hive là 1 kho dữ liệu (data warehouse) hỗ trợ người sử dụng có thể dễ dàng hơn trong việc quản lý và truy vấn đối với các tập dữ liệu lớn được lưu trữ trên các hệ thống lưu trữ phân tán (distributed storage). Hive được xây dựng dựa trên cơ sở của Apache Hadoop, nó cung cấp các tính năng chính sau:

* Công cụ cho phép dễ dàng thực hiện tác vụ như trích xuất, vận chuyển và lưu trữ dữ liệu.
* Cơ chế để xử lý cho nhiều định dạng dữ liệu khác nhau.
* Truy cập tới dữ liệu dạng files được lưu trữ trực tiếp ở trong Apache HDFS hoặc đối với nhiều hệ thống lưu trữ dữ liệu khác như Apache HBase.
* Thực hiện query thông qua MapReduce.

Hive định nghĩa ra một ngôn ngữ truy vấn đơn giản có cú pháp gần giống với SQL (SQL-like query language) được gọi là HiveQL, nó cho phép người sử dụng đã quen thuộc với các truy vấn SQL thực hiện việc truy vấn dữ liệu. Ngoài ra ngôn ngữ này còn cho phép các lập trình viên người đã quen thuộc với MapReduce framework có thể nhúng các mappers và reducers cho chính họ viết ra để thực thi nhiều hơn nữa các phân tích phức tập mà không được hỗ trợ bởi các hàm đã có sẵn trong ngôn ngữ HiveQL. HiveQL cung có thể được mở rộng với các custom scalar functions (UDF’s), aggregations (UDAF’s) và các table funtions (UDTF’s)

Hive không yêu cầu dữ liệu phải được đọc và ghi dưới một định dạng của riêng Hive (Hive format). Hive hoạt động tốt trên Thrift và các định dạng dữ liệu riêng của người sử dụng.

Hive không được thiết kế để cho các giao dịch online (OLTP workloads) và không nên dùng cho các real-time queries và các cập nhật trên từng dòng trong 1 table (row-level). Hive hoạt động tốt nhất cho các batch jobs trên các tập dữ liệu lớn, mà ở đó dữ liệu được thêm vào liên tục (append-only data) ví dụ như web logs. Hive có khả năng mở rộng theo chiều ngang tốt (thực thi tốt trên 1 hadoop cluster có số tượng máy biến đổi), có khả năng tích hợp với MapReduce framework và UDF, UDAF, UDTF; có khả năng chống chịu lỗi và mềm dẻo đối với các dữ liệu đầu vào của chính nó.

Các thành phần cấu hình Hive bao gồm HCatalog và WebHCat. HCatalog là một thành phần của Hive. Đây là lớp quản lý lưu trữ cho Hadoop (table and management layer), nó cho phép người dùng với các công cụ xử lý dữ liệu khác nhau bao gồm cả Pig và MapReduce thực thi hoạt động đọc, ghi một cách dễ dàng hơn. WebHCat cung cấp một dịch vụ cho phép bạn có thể thực thi Hadoop MapReduce (hoặc YARN), Pig, Hive.

## **Kiến trúc của Hive**

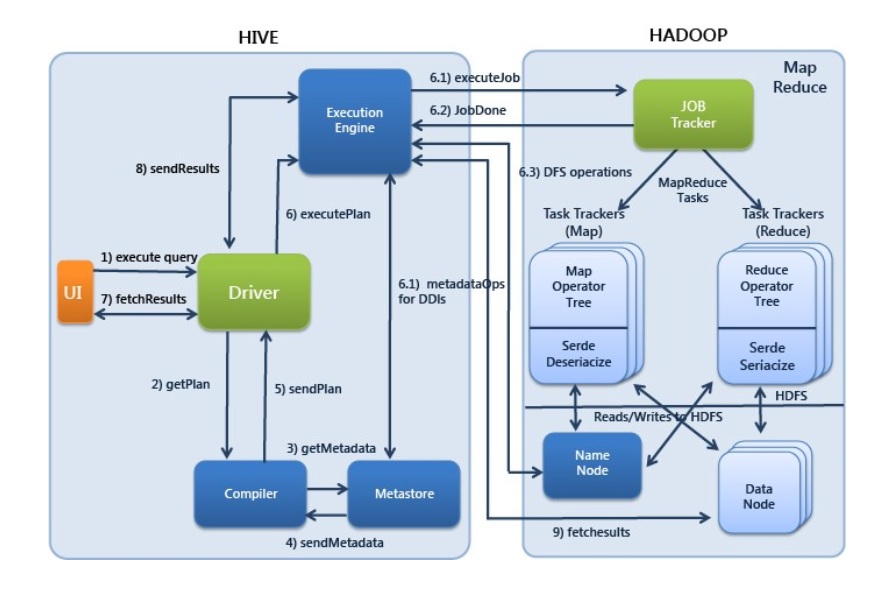
Hive có các thành phần chính là:

* Hive UI: cung cấp giao diện cho phép người sử dụng tương tác với hệ thống Hive. Hive cung cấp nhiều phương thức khác nhau cho phép người sử dụng tương tác với Hive:
* CLI: giao diện dạng shell cho phép người sử dụng tương tác trực tiếp qua command line.
* Hive Web Interface: giao diện Web cho phép người sử dụng thực hiện các truy vấn thông qua giao diện Web.
* Hive Thrift Server: cho phép các client từ nhiều ngôn ngữ lập trình khác nhau có thể thực hiện tương tác với Hive.
* Hive Driver: thành phần nhận các truy vấn và chuyển các truy vấn này thành các MapReduce Jobs để tiến hành xử lý yêu cầu của người sử dụng.
* Driver: nhận các truy vấn, thành phần này thực hiện việc quản lý các sessions và cung cấp các API để thực thi và lấy dữ liệu trên JDBC/ODBC interfaces.
* Compiler: thành phần hiện việc phân tích ngữ nghĩa đối với các query, lấy các thông tin metadata cần thiết về table và partion từ metastore để sinh ra các excution plan.
* Execute engine: thành phần thực thi các execution plan được tạo bởi compiler (submit các job tới MapReduce). Ngoài ra thành phần execution enginen này thực hiện việc quản lý các dependencies của các bước trong mỗi execution plan, thực thi từng bước này.
* Hive Metastore: thành phần lưu trữ các metadata của Hive: table, partion, buckets bao gồm cả thông tin về các column trong mỗi table, các serializers và desrializers cần thiết để thực hiện việc đọc và ghi dữ liệu. Metastore sử dụng một cơ sở dữ liệu quan hệ để lưu trữ dữ liệu của chính mình.



**Hình 1: Kiến trúc của Hive**

## **Hoạt động của Hive**



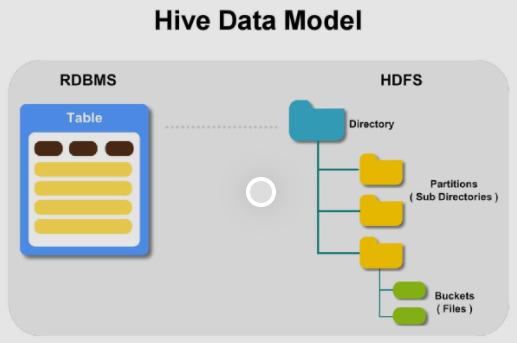
**Hình 2: Sơ đồ luồng hoạt động của Hive**

Quy trình hoạt động của Hive có thể được mô tả theo các bước sau:

1. Các truy vấn tới từ User Interface (CLI, Hive Web Interface, Thirft Server) được gửi tới thành phần Driver (Bước 1 hình 2)
2. Driver tạo ra mới 1 session cho truy vấn này và gửi query tới compiler để nhận lấy Execution Plan (Bước 2 hình 2)
3. Compilter nhận các metadata cần thiết từ Metastore (Bước 3, 4 hình 2). Các metadata này sẽ được sử dụng để kiểm tra các biểu thức bên trong query mà Compiler nhận được.
4. Plan được sinh ra bởi Compiler (thông tin về các job (map-reduce) cần thiết để thực thi query sẽ được gửi lại tới thành phần thực thi (Bước 5 hình 2)
5. Execution engine nhận yêu cầu thực thi và lấy các metadata cần thiết và yêu cầu mapreduce thực thi công việc (Bước 6.1, 6.2, 6.3 hình 2)
6. Khi output được sinh ra, nó sẽ được ghi dưới dạng 1 temporary file, temorary file này sẽ cung cấp các thông tin cần thiết cho các stages khác của plan. Nội dung của các temporary file này được execution đọc trực tiếp từ HDFS như là 1 phần của các lời gọi từ Driver (bước 7, 8, 9 hình 2)

## **Mô hình dữ liệu trong Hive**

### *1.4.1. Tổ chức dữ liệu*



**Hình 3: Tổ chức dữ liệu trong Hive**

Dữ liệu trong Hive được tổ chức thành các kiểu sau:

* Databases: là namespace cho các tables, dùng để nhóm và quản lý các nhóm tables khác nhau.
* Tables: tương tự như table trong các hệ cơ sở dữ liệu quan hệ. Trong Hive table có thể thực hiện các phép toán filter, join và union… Mặc định thì dữ liệu của Hive sẽ được lưu bên trong thư mục warehouse trên HDFS. Tuy nhiên Hive cũng cung cấp kiểu external table cho phép ta tạo ra và quản lý các table mà dữ liệu của nó đã tồn tại từ trước khi ta tạo ra table này hoặc nó được lưu trữ ở 1 thư mục khác bên trong hệ thống HDFS. Tổ chức row và column bên trong Hive có nhiều điểm tương đồng với tổ chức Row và Column trong các hệ cơ sở dữ liệu quan hệ. Hive có 2 kiểu table đó là: Managed Table và External tables.
* Partions: Mỗi table có thể có 1 hoặc nhiều các khóa mà từ đó xác định dữ liệu sẽ được lưu trữ ở đâu. Ví dụ table web\_log có thể phân chia dữ liệu của mình theo từng ngày là lưu dữ liệu của mỗi ngày trong 1 thư mục khác nhau bên dưới đường dẫn warehouse.

Ví dụ: /warehouse/web\_log/date=”01-01-2014″

* Buckets: Dữ liệu trong mỗi partion có thể được phân chia thành nhiều buckets khác nhau dựa trên 1 hash của 1 colume bên trong table. Mỗi bucket lưu trữ dữ liệu của nó bên dưới 1 thư mục riêng. Việc phân chia các partion thành các bucket giúp việc thực thi các query dễ dàng hơn.

### *1.4.2. Kiểu dữ liệu*

Kiểu dữ liệu nguyên thủy:

Mỗi columns có 1 kiểu dữ liệu cố định. Các kiểu dữ liệu nguyên thủy sau sẽ được hỗ trợ đối với Hive:

* Integers:
* TINYINT – 1 byte integer
* SMALLINT – 2 bytes integer
* INT – 4 bytes integer
* BIGINT – 8 bytes integer
* Boolean type
* BOOLEAN – TRUE/FALSE
* Floating point numbers
* FLOAT – single precision
* DOUBLE – Double precision
* String type
* STRING – sequence of characters in a specified character set

Các kiểu dữ liệu khác:

* Structs: là kiểu dữ liệu mà mỗi phần tử bên trong đó có thể được truy cập thông qua việc sử dụng ký hiệu (.)

Ví dụ, với kiểu dữ liệu STRUCT {a INT; b INT} ví dụ trường a của nó có thể truy cập thông qua c.a

* Maps (key-value tuples): là kiểu dữ liệu mà các phần tử sẽ được truy cập thông qua ký hiệu [‘element name’]. Đối với map M thực hiện việc map dữ liệu đối với khóa ‘group’ -> thì dữ liệu sẽ được sử dụng bởi trường M[‘group’]
* Arrays (indexable lists): Kiểu mảng.

## **HiveSQL (HQL)**

Ngôn ngữ truy vấn Hive cung cấp các toán tử cơ bản giống SQL. Đây là một số tác vụ mà HQL có thể làm dễ dàng.

* Tạo và quản lý tables và partitions.
* Hỗ trợ các toán tử Relational, Arithmetic và Logical khác nhau.
* Evaluate functions
* Tải về nội dung 1 table từ thư mục cục bộ hoặc kết quả của câu truy vấn đến thư mục HDFS.

Đây là ví dụ truy vấn HQL:

SELECT upper(name), salesprice

FROM sales;

SELECT category, count(1)

FROM products

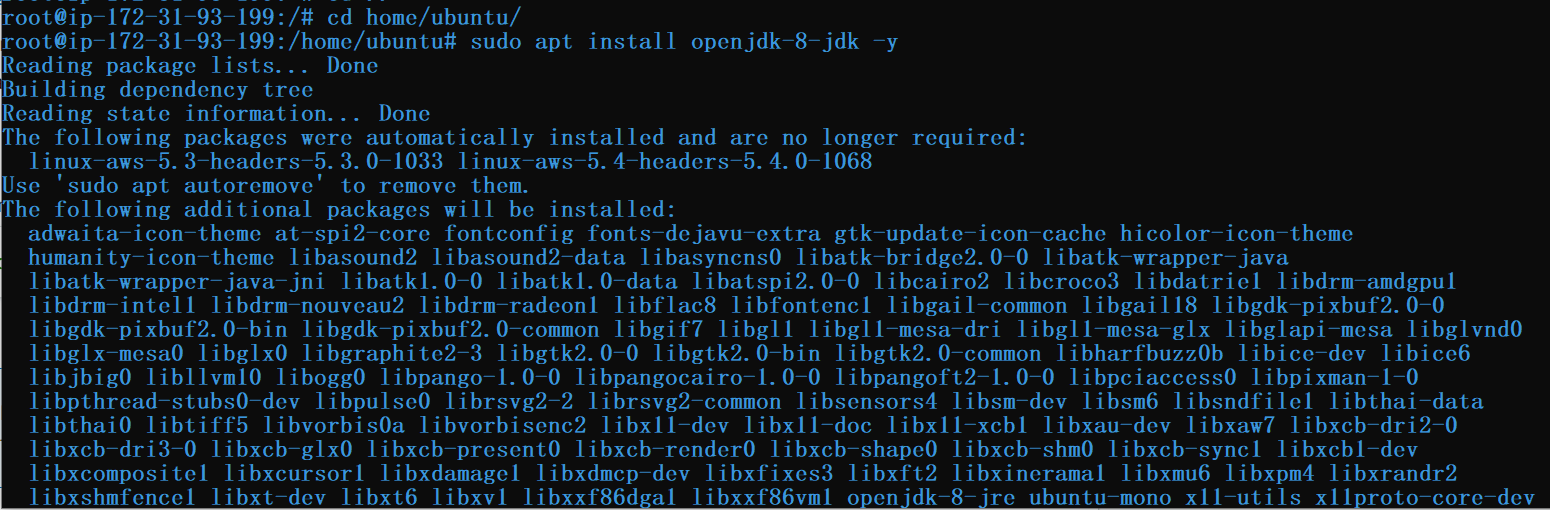
GROUP BY category;

# **CHƯƠNG 2: CÀI ĐẶT, THIẾT KẾ VÀ XÂY DỰNG DATA WAREHOUSE**

## **2.1 Cài đặt**

### *2.1.1 Cài đặt môi trường Hadoop*

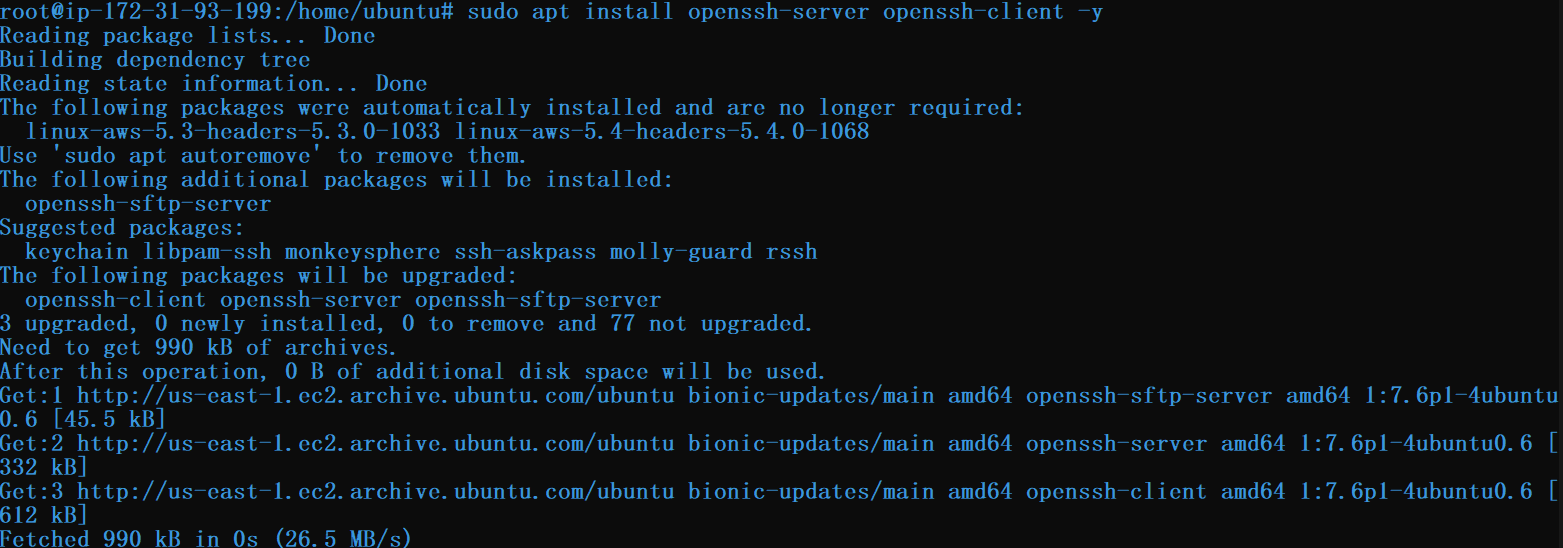
Cài đặt OpenJDK



Kiểm tra phiên bản java

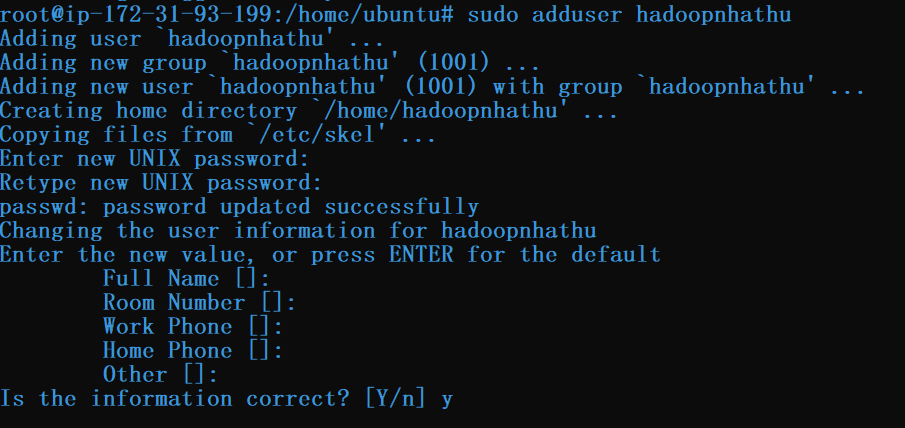


Cài đặt OpenSSH

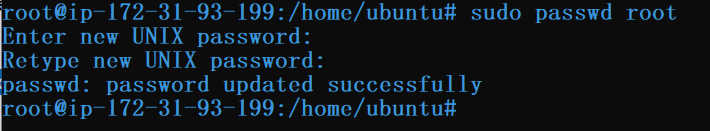


Sau khi cài đặt OpenJDK và OpenSSH, ta cần 1 người dùng để sử dụng để tiếp tục cài đặt Hadoop

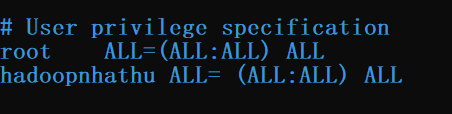
Ở đây, tụi em tạo user tên là hadoopnhathu



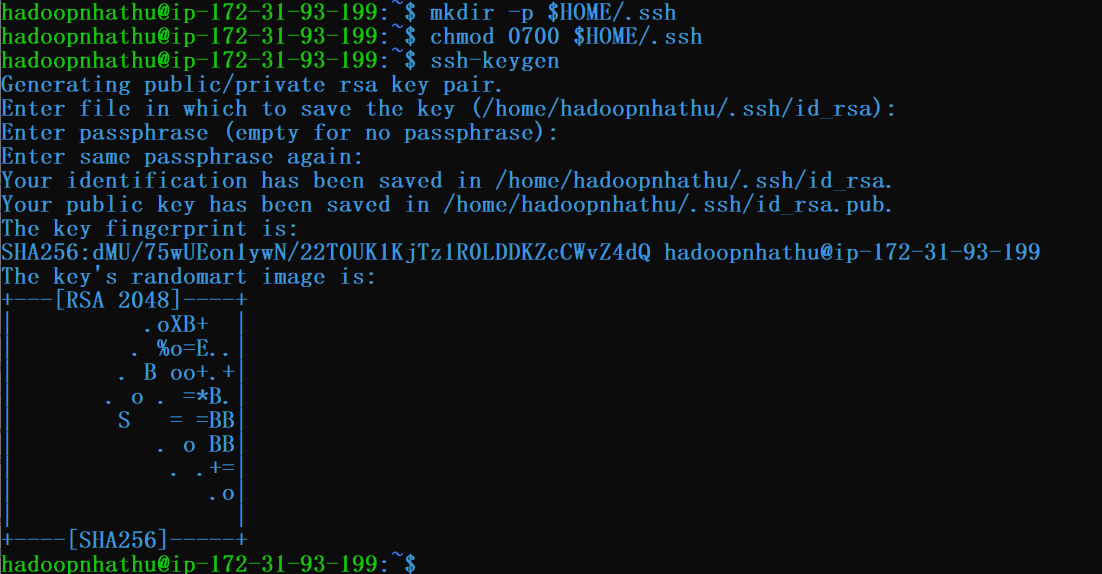
Sau khi tạo xong, ta cần thêm quyền root vào user vừa tạo



Sử dụng lệnh visudo và thêm quyền root vào dưới root ALL= (ALL:ALL) ALL



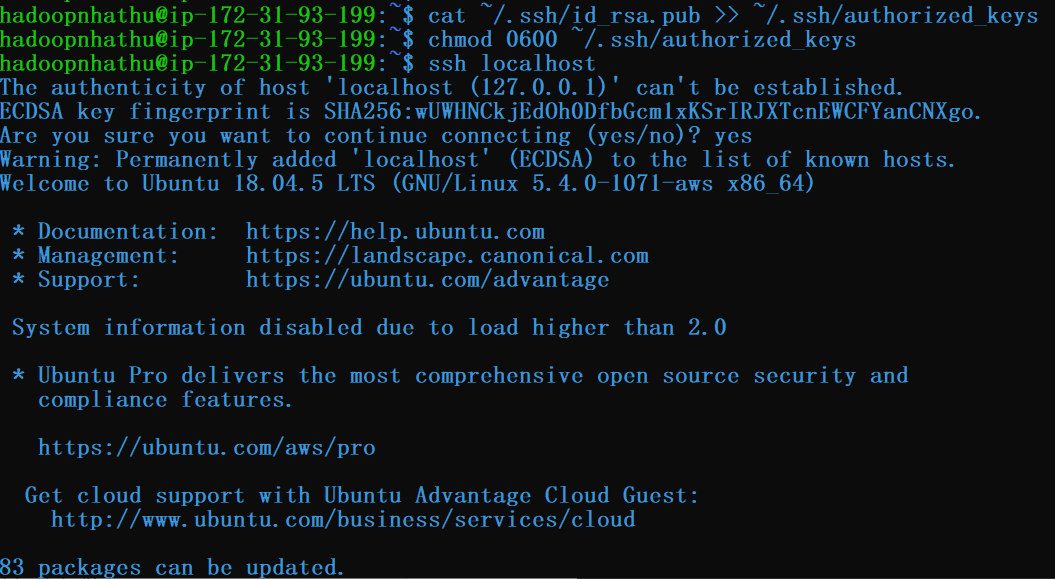
Bật không mật khẩu cho người dùng hdoop để tránh phải nhập lại mật khẩu nhiều lần.



Sử dụng lệnh cat để lưu trữ khóa công khai dưới dạng allow\_keys trong thư mục ssh.

Sau đó đặt quyền cho user bằng lệnh chmod.

Người dùng sau đó có thể ssh mà không cần dùng mật khẩu mỗi lần. Có thể kiểm tra mọi thứ đã thiết lập chính xác bằng cách sử dụng user hdoop đã tạo ở trên để ssh vào localhost.



Tiếp theo là tải gói hadoop về, ở đây chúng em sử dụng phiên bản hadoop 3.2.1

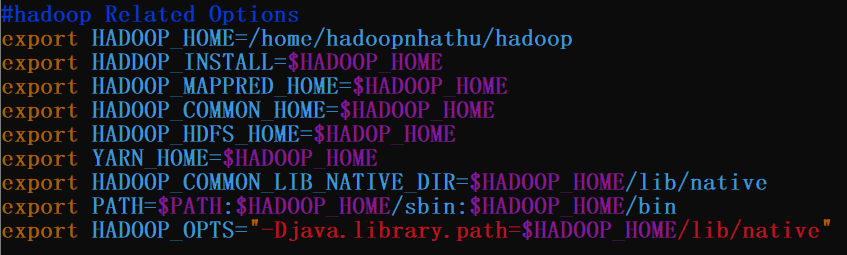
Link tải: <https://archive.apache.org/dist/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz>

Sau khi tải xong, chúng ta giải nén bằng lệnh tar xzf



Tiếp theo ta cần cấu hình các biến môi trường Hadoop(bashrc)

Xác định nội dung biến môi trường bằng các lệnh sau:



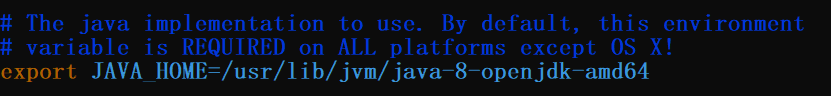
Sau khi xác định xong, ta lưu lại bằng lệnh sau



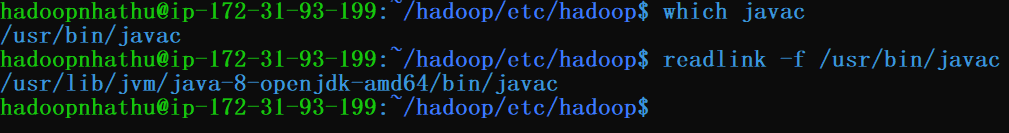
Chỉnh sửa tệp vim



Bỏ ghi chú biến $JAVA\_HOME (bỏ dấu ‘#’) và thêm đường dẫn đầy đủ đến cài đặt OpenJDK trên hệ thống



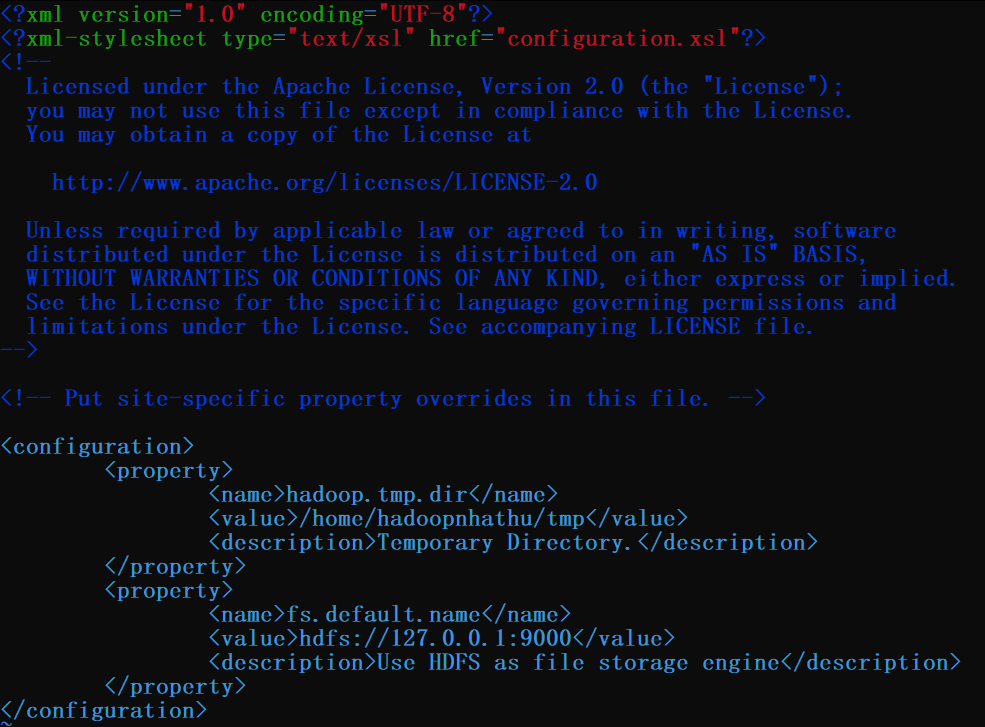
Nếu như không biết đường dẫn Java chính xác, hãy chạy lệnh sau:



Chỉnh sửa tệp core-site.xml



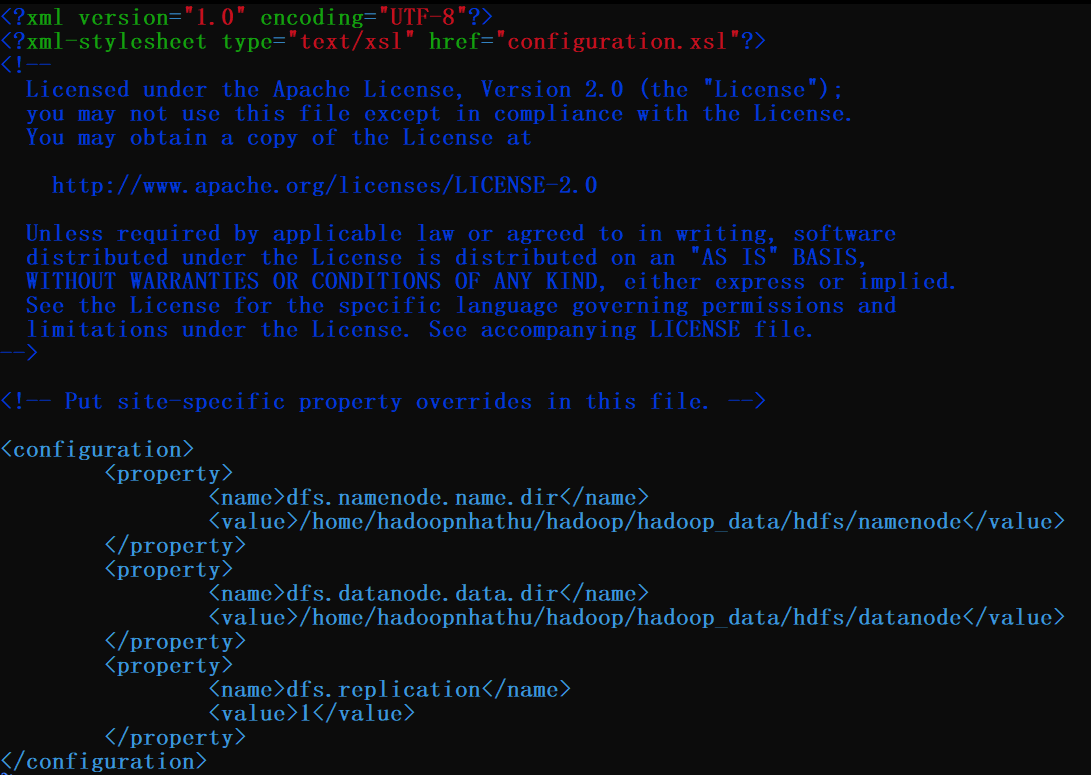
Thêm cấu hình sau để ghi đè các giá trị mặc định cho thư mục tạm thời và thêm URL HDFS để thay thế cài đặt hệ thống tệp cục bộ mặc định:



Chỉnh sửa tệp hdfs-site.xml



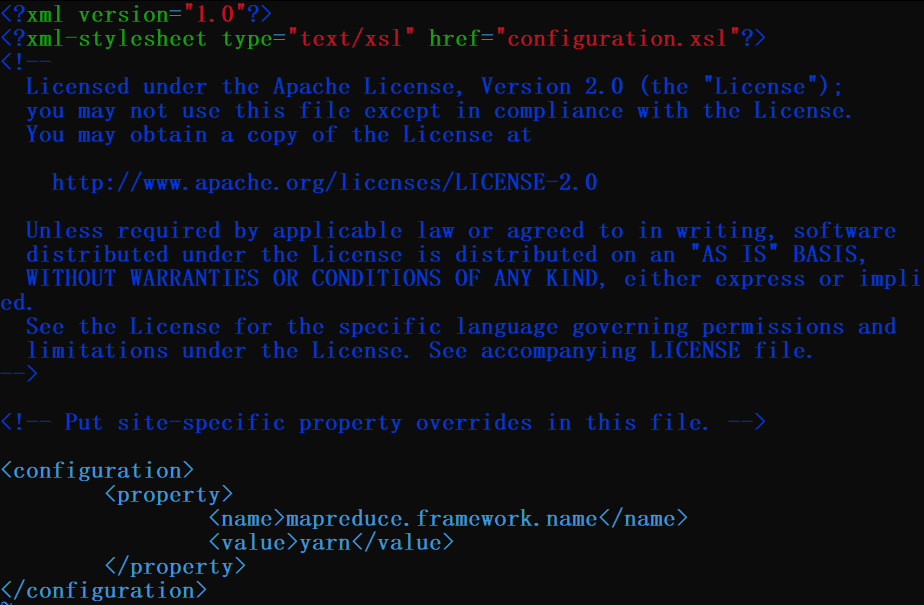
Thêm cấu hình sau vào tệp và nếu cần, hãy điều chỉnh các thư mục NameNode và DataNode cho các vị trí tùy chỉnh:



Chỉnh sửa tệp mapred-site.xml



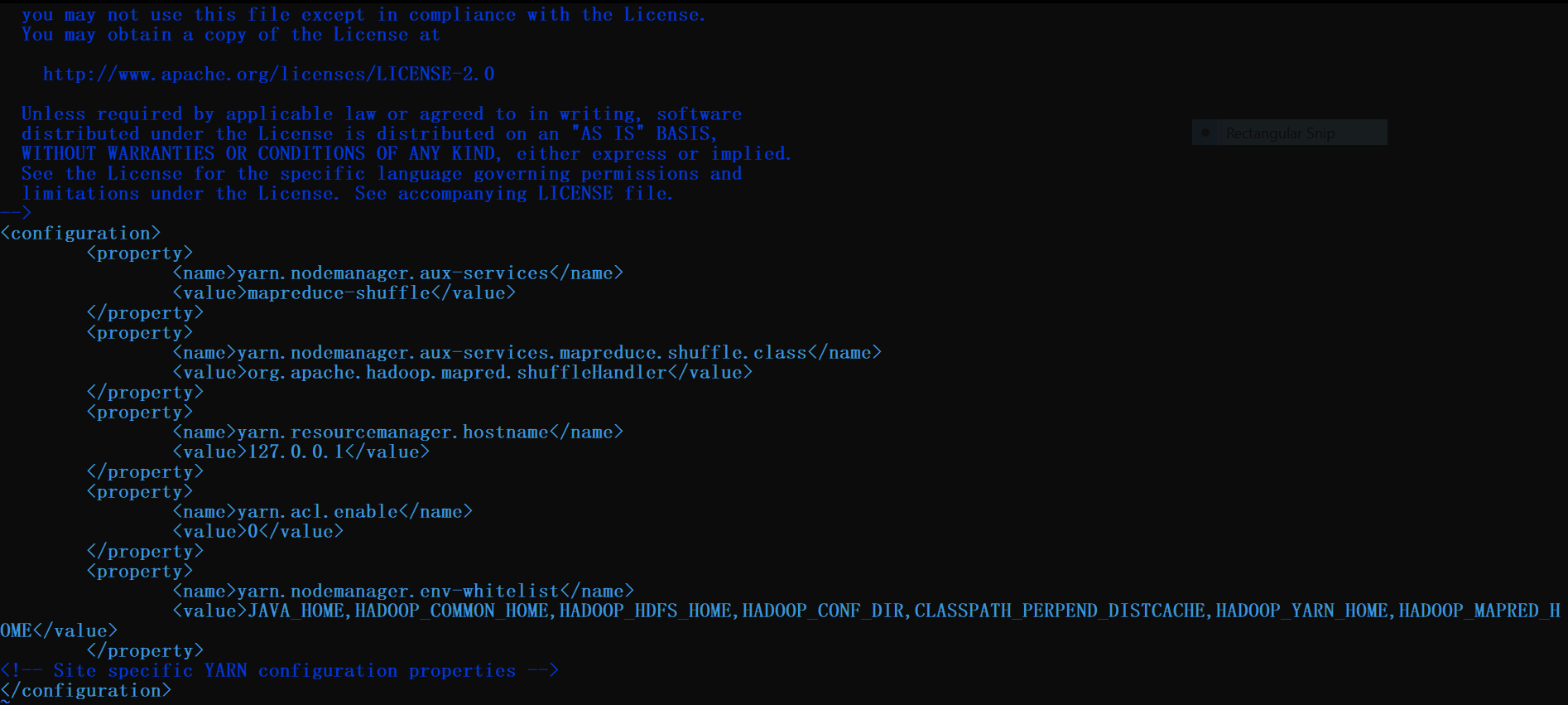
Thêm cấu hình sau để thay đổi giá trị MapReduce framework mặc định thành yarn:



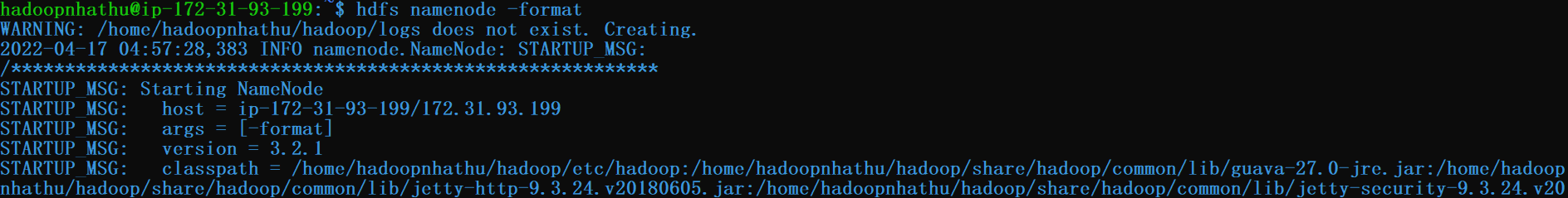
Chỉnh sửa tập tin yarn-site.xml

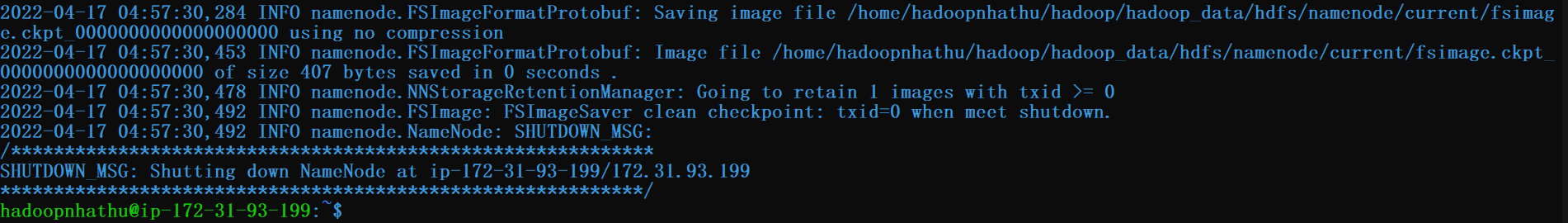


Thêm các cấu hình sau vào file

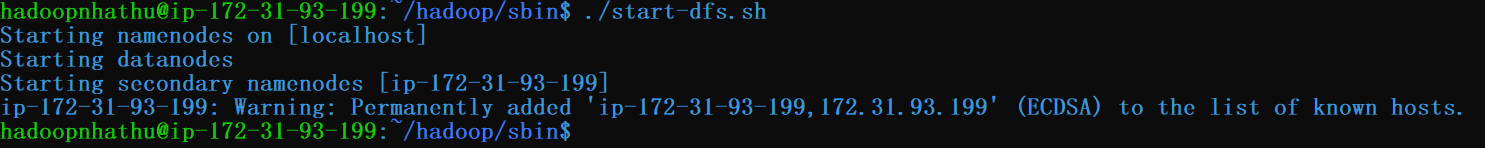


Định dạng HDFS NameNode:



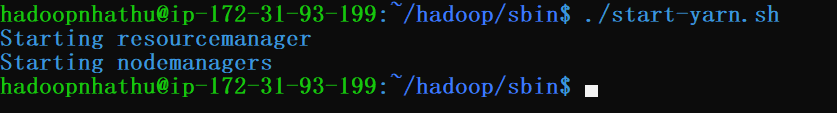


Khởi động Hadoop Cluster



Như với lệnh trước, đầu ra thông báo rằng các quá trình đang bắt đầu

Nhập lệnh đơn giản jps để kiểm tra xem tất cả các daemon có đang hoạt động và đang chạy như các quy trình Java hay không

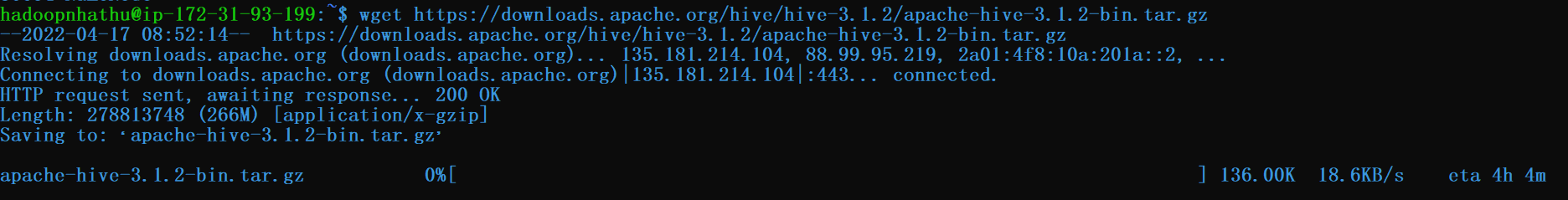


### *2.1.2 Cài đặt Apache Hive*

Tải gói apache hive về, chúng em sử dụng phiên bản hive 3.1.2

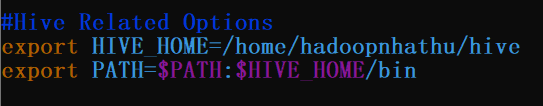
Link tải: <https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz>

Sau khi tải xong, giải nén bằng lệnh tar xzf



Cấu hình các biến môi trường Hive (bashrc)

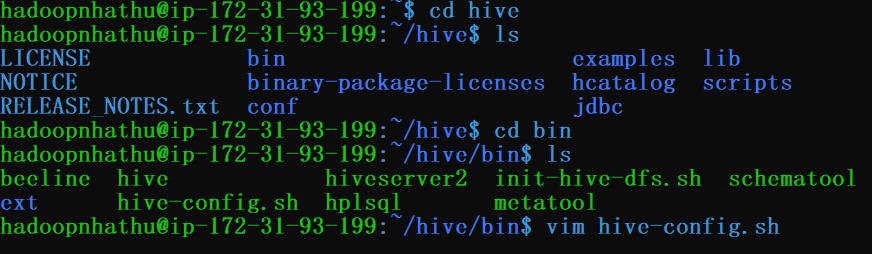
Thêm các biến môi trường hive sau vào tệp .bashrc



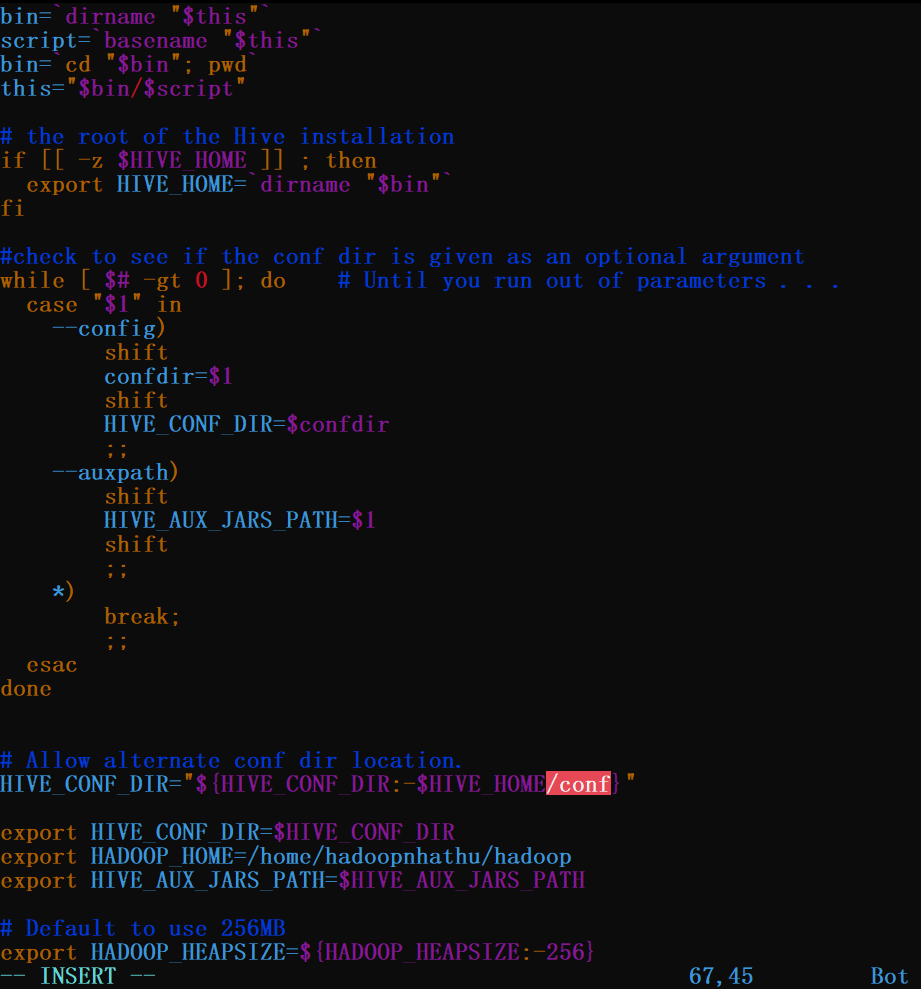
Sau khi cấu hình xong, chạy lệnh sau để lưu tệp



Chỉnh sửa tập hive-config.sh



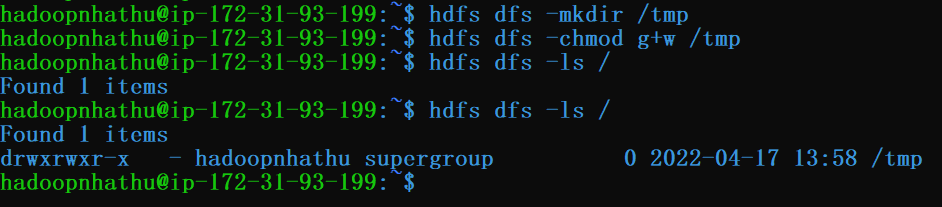
Thêm biến HADOOP\_HOME và đường dẫn đầy đủ đến thư mục Hadoop



Tạo đường dẫn thư mục Hive trong HDFS

Thêm quyền ghi và thực thi cho file tmp

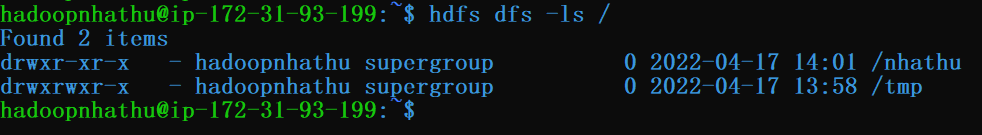
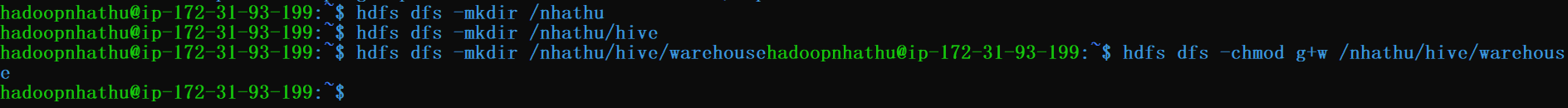
Kiểm tra quyền bằng lệnh –ls/



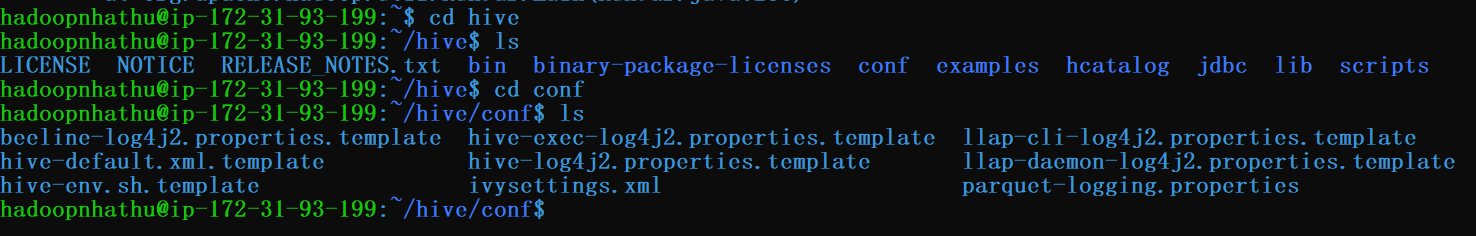
Tạo thư mục warehouse

Thêm quyền ghi và thực thi cho warehouse

Kiểm tra bằng lệnh –ls/



Cấu hình tập hive-site.xml



Sử dụng lệnh copy để tạo tệp hive-site.xml từ hive-default.xml.template

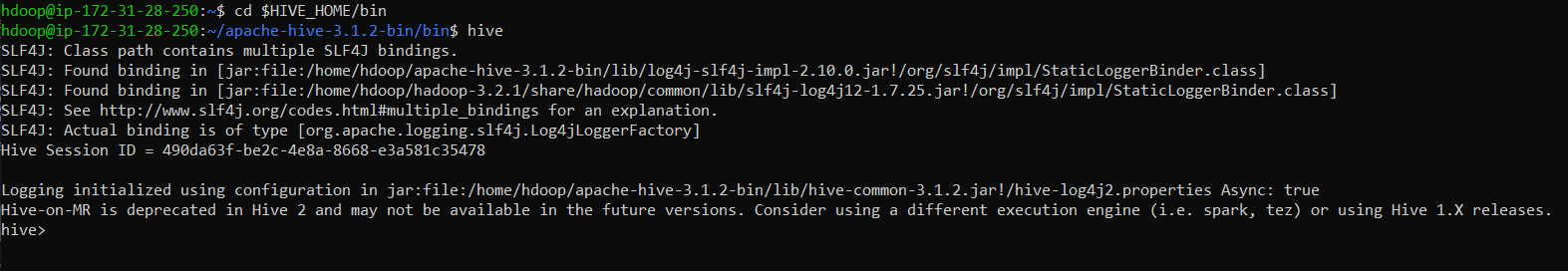


Khởi tạo Derby Database

Sau khi khởi tạo xong sẽ hiện ra schemaTool completed

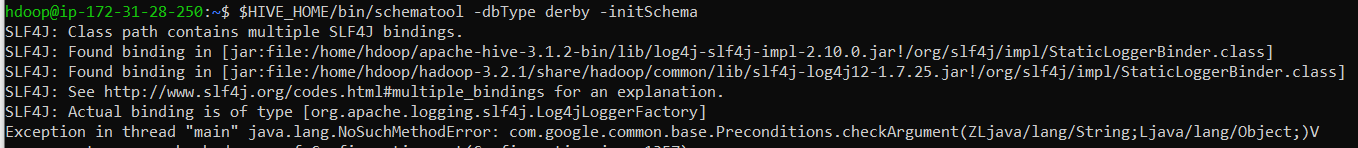


Sau đó ta có thể sử dụng Hive



### *2.1.3 Lỗi phát sinh trong quá trình cài đặt*

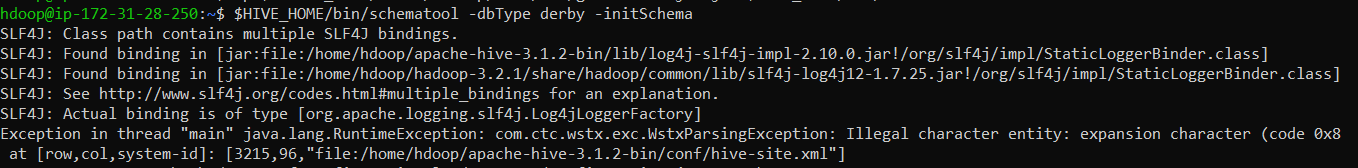
Lỗi 1:



Ta cần xóa guave của hive, rồi copy guave của hadoop qua hive để đồng bộ bằng cách sau:



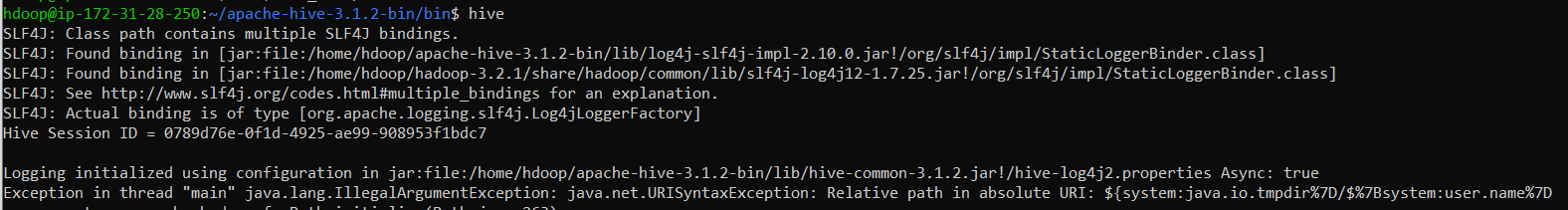
Lỗi 2:



Xóa những kí tự đặc biệt bị dư, vào hive-site.xml xóa những kí tự bị dư ở dòng 3215:



Lỗi 3:



Thêm các thẻ property sau vào đầu file hive-site.xml:

<property>

<name>system:java.io.tmpdir</name>

<value>/tmp/hive/java</value>

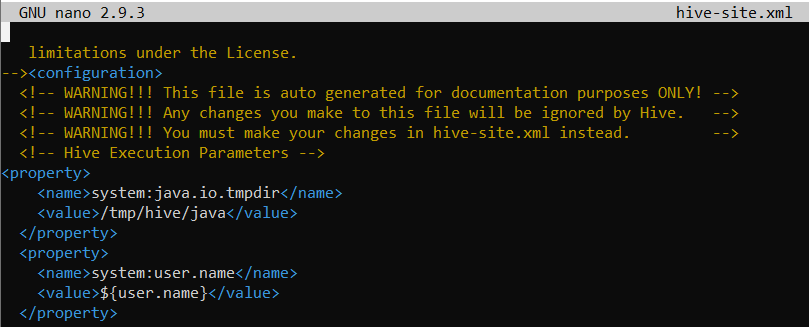
</property>

<property>

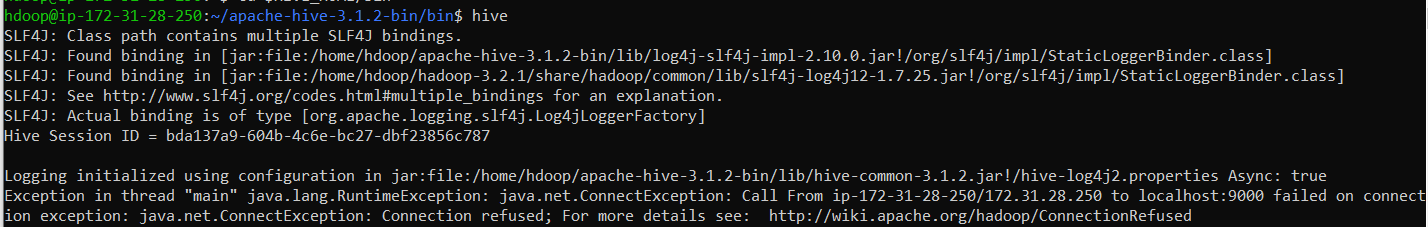
<name>system:user.name</name>

<value>${user.name}</value>

</property>



Lỗi 4:



Exception in thread "main" java.lang.RuntimeException: java.net.ConnectException: Connection refused

Sửa bằng cách thực hiện các bước sau:





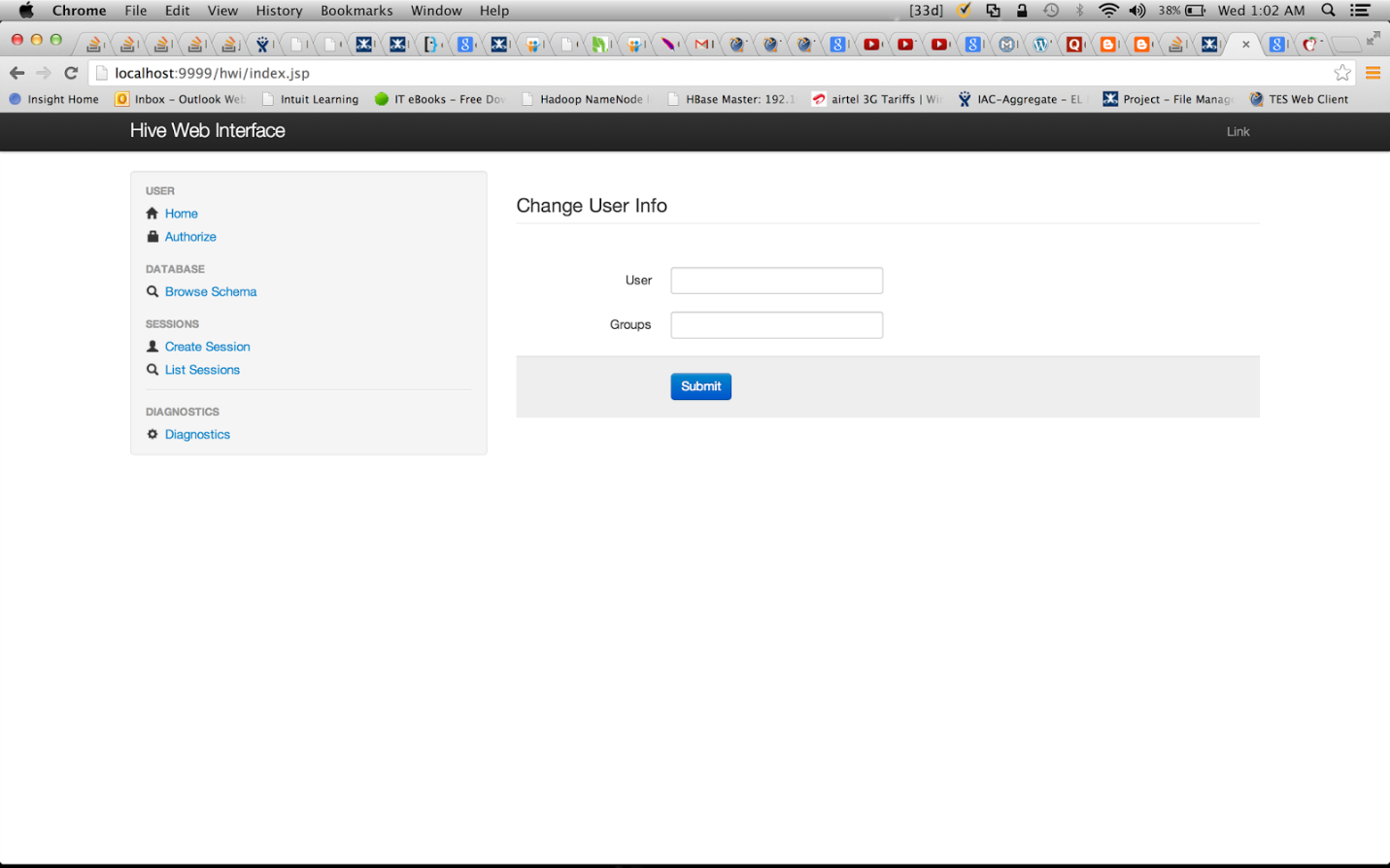




## **2.2 Triển khai Hive Web Interface**

Hive Web Interface là dao diện dùng để sử dụng và truy vấn HQL trên một giao diện người dùng thay vì trên Command Line.

Việc sử dụng HWI sẽ đơn giản và thuận tiện hơn trong việc truy vấn



Do HWI chỉ hỗ trợ cho các phiên bản Apache Hive 2.2.0 trở xuống nên để có thể sử dụng được HWI, ta cần cài đặt lại phiên bản mà HWI hỗ trợ.

Cụ thể, tụi em sẽ sử dụng các phiên bản sau:

Hadoop: <https://archive.apache.org/dist/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

Hive: <https://archive.apache.org/dist/hive/hive-2.1.0/apache-hive-2.1.0-bin.tar.gz>

Hive-source: <https://archive.apache.org/dist/hive/hive-2.1.0/apache-hive-2.1.0-src.tar.gz>

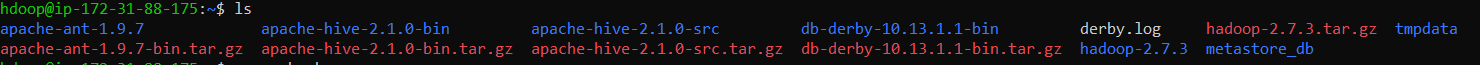
Vì bản Apache Hive này không hỗ trợ Derby nên ta cần phải cài thêm

Derby: <https://archive.apache.org/dist/db/derby/db-derby-10.13.1.1/db-derby-10.13.1.1-bin.tar.gz>

Cuối cùng là sử dụng Apache Ant để có thể kết nối với HWI

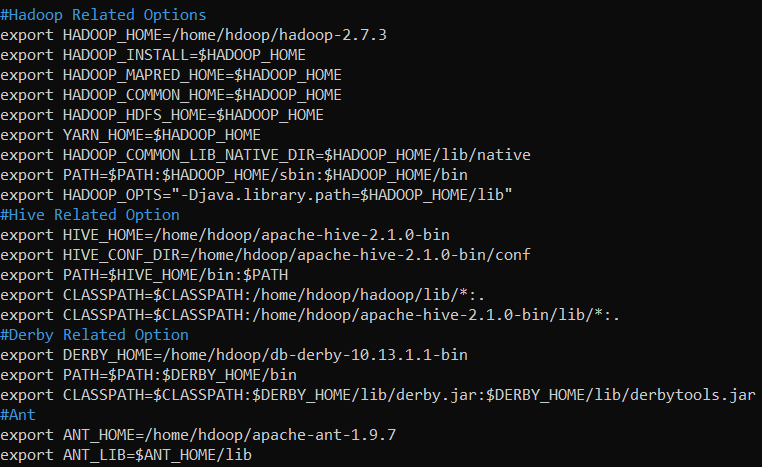
Ant: <https://archive.apache.org/dist/ant/source/apache-ant-1.9.7-src.tar.gz>

Sau khi tải xong, ta cần giải nén bằng lệnh tar xzf (đã hướng dẫn trong bước cài đặt)



Tương tự theo các bước cài đặt ở trên, sau khi giải nén xong, ta vào file bashrc để cấu hình các đường dẫn





Tiếp theo thực hiện cấu hình các file (theo hướng dẫn ở trên)

hadoop.env.sh

core-site.xml

hdfs-site.xml

mapred-site.xml

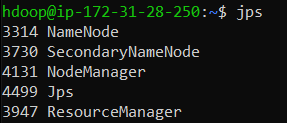
yarn-site.xml

HDFS NameNode

Hadoop Cluster

Sau khi cấu hình xong, nhập lệnh jps để kiểm tra xem tất cả các daemon có đang hoạt động và đang chạy như các quy trình Java hay không

Nếu trong trường hợp thiếu Datanode hoặc Namenode:

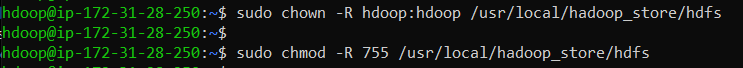


Ta thực hiện các bước sau để cấp quyền và khởi động lại hệ thống:

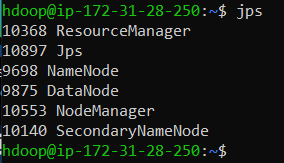
Tiến hành sửa đường dẫn của Namenode và Datanode:



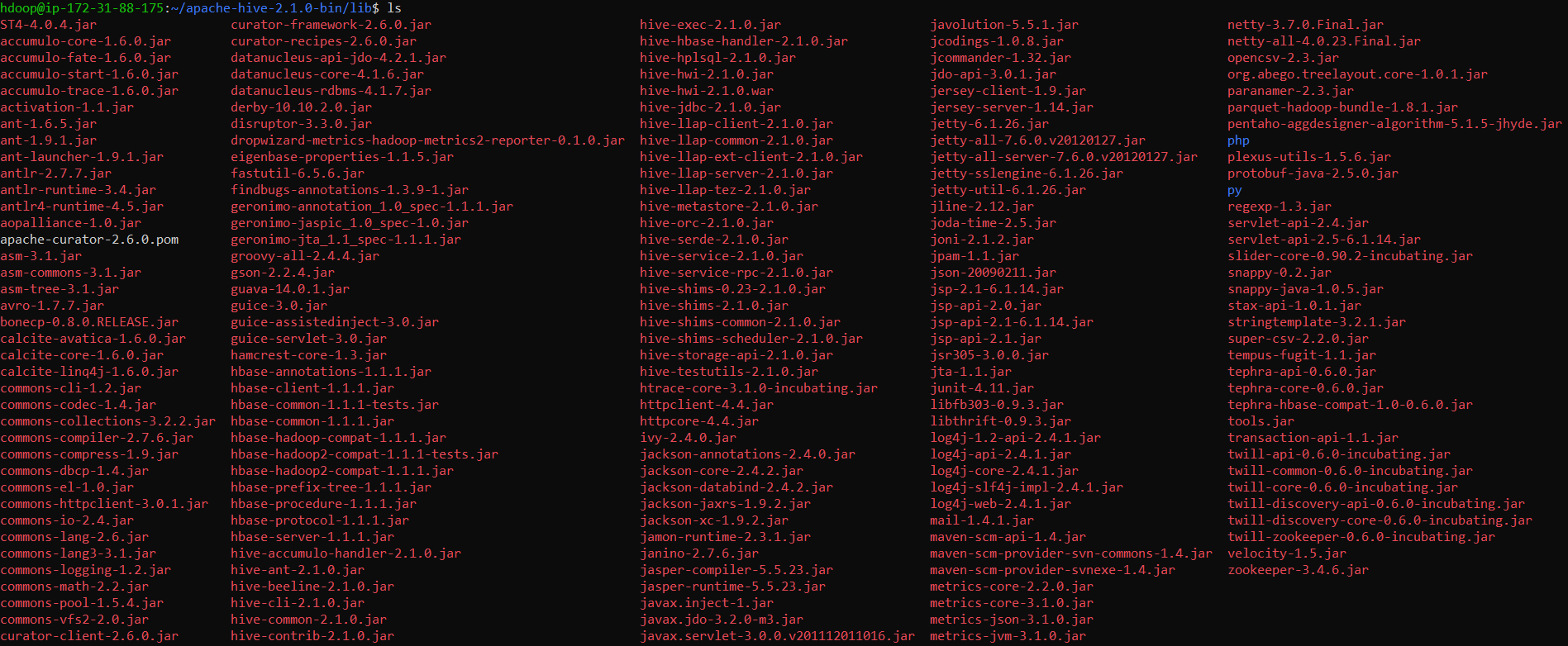
Cấp quyền cho user hdoop trên các tệp NameNode và DataNode (usr/local/hadoop\_store/hdfs):



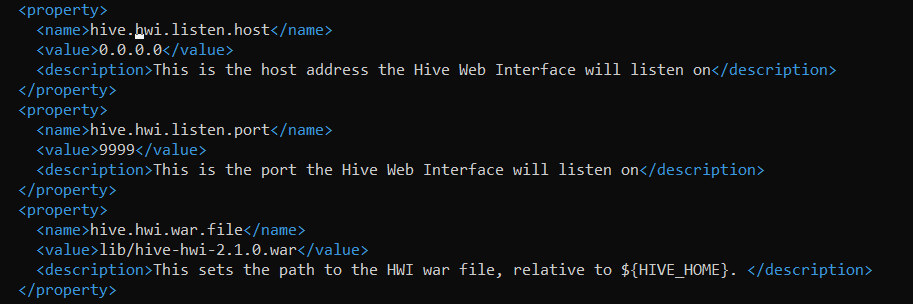
Stop .sh, chỉnh lại format NameNode và start .sh lại. Sau đó nhập jps để kiểm tra



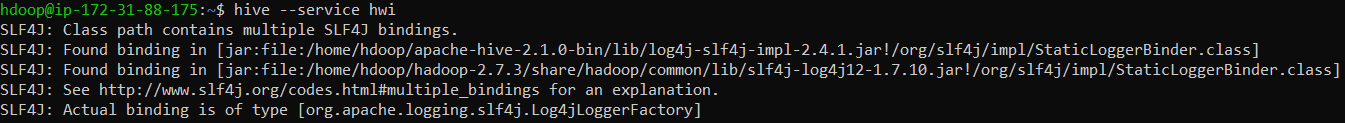
Tiến hành copy file hive-hwi-2.1.0.war từ apache-hive-2.1.0-src và copy tools.jar từ thư viện của java vào thư viện lib của Apache Hive

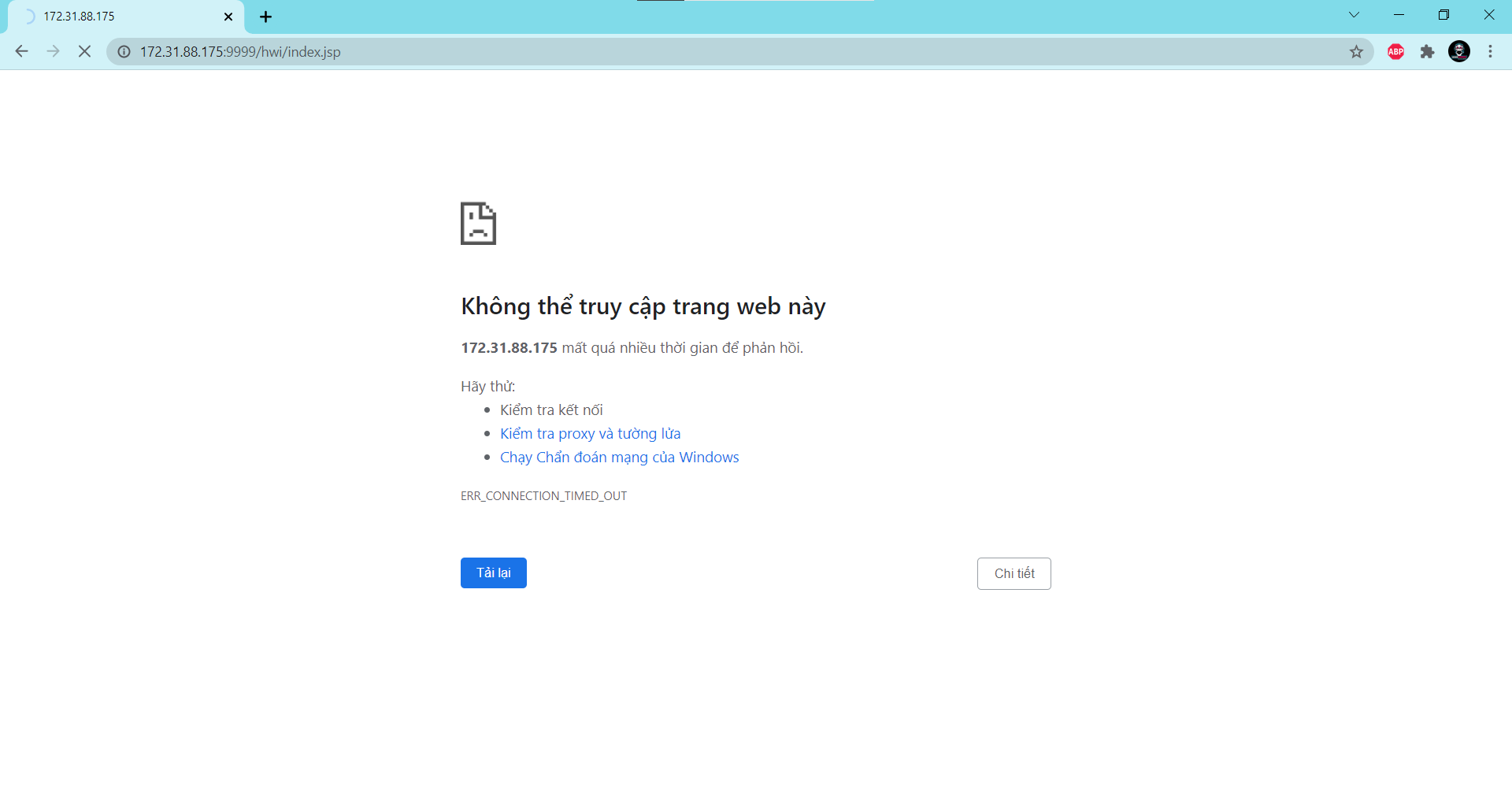


Chỉnh sửa file hive-site.xml, truy cập bằng lệnh nano hive-site.xml



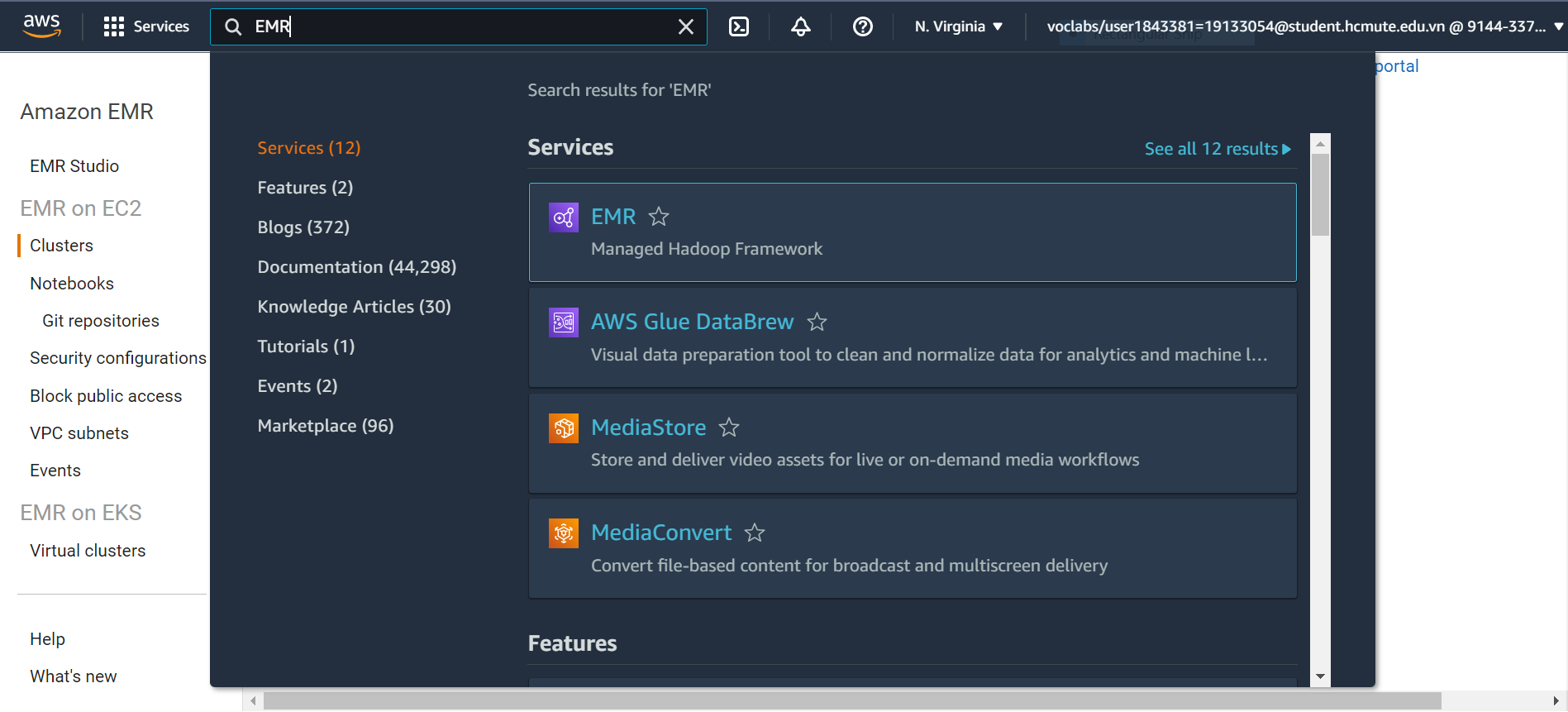
Chạy thử Hive Web Interface bằng lệnh: hive --service hwi

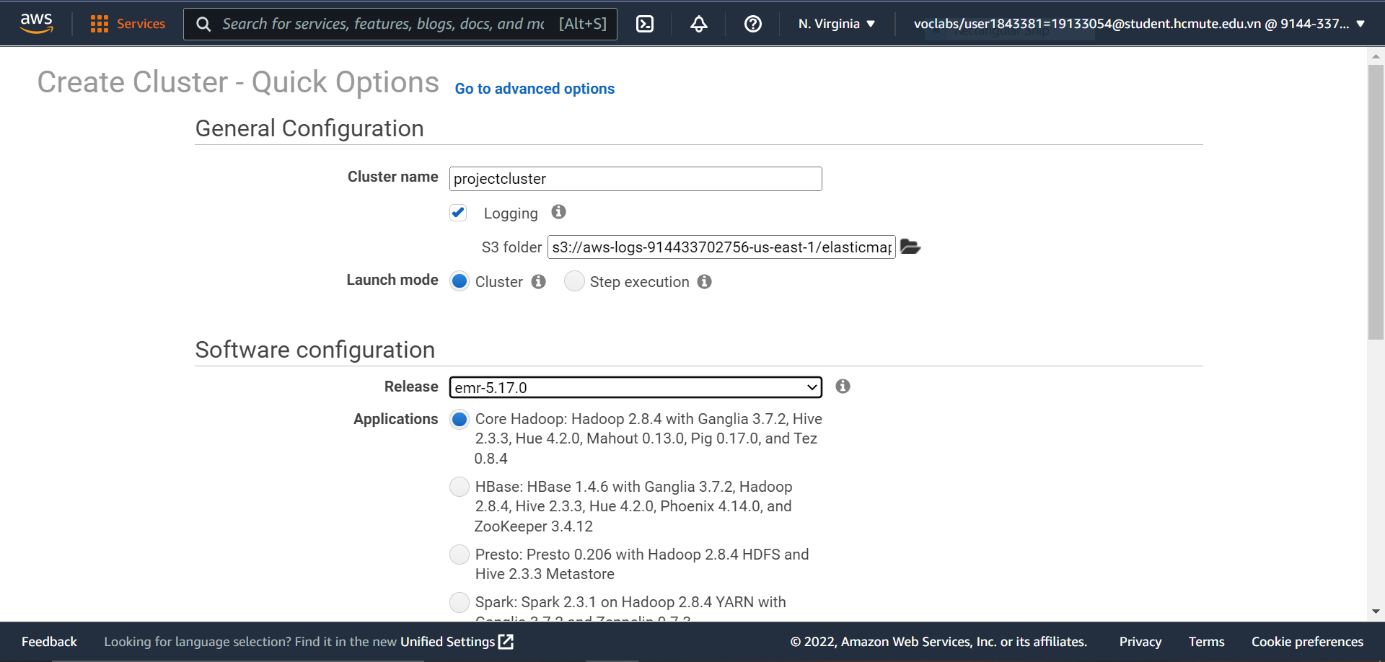




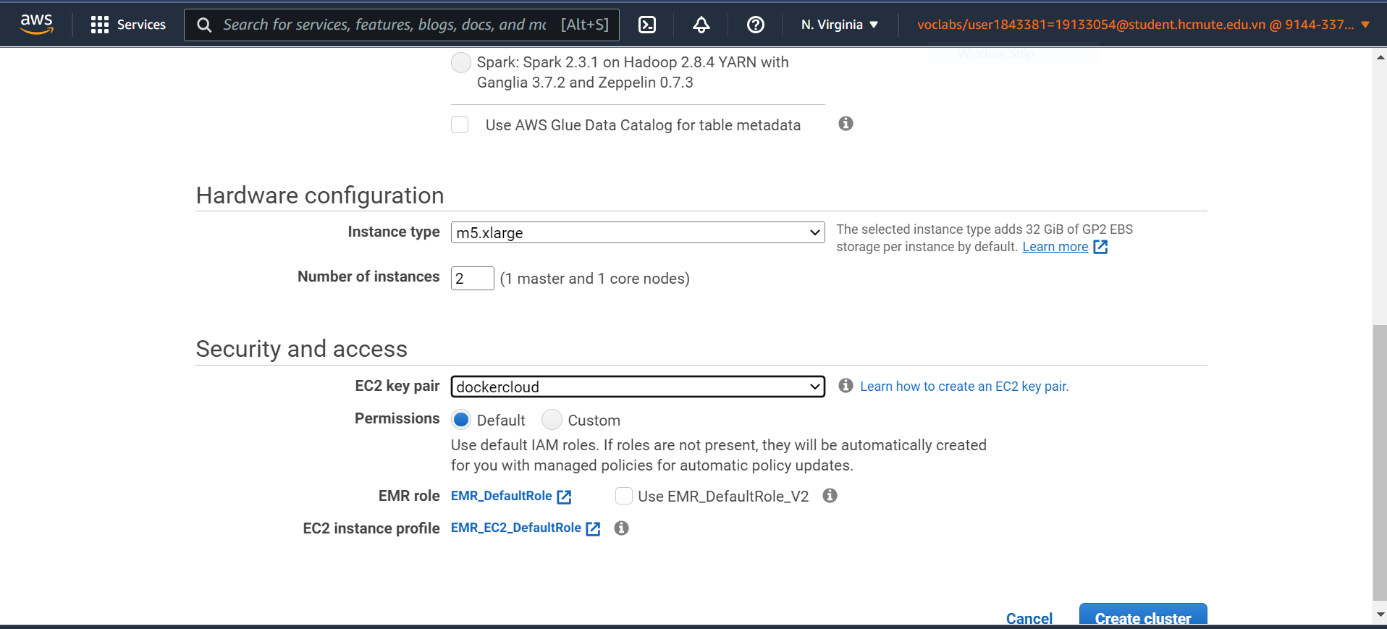
## **2.3 Cài Hive trên AWS sử dụng EMR**

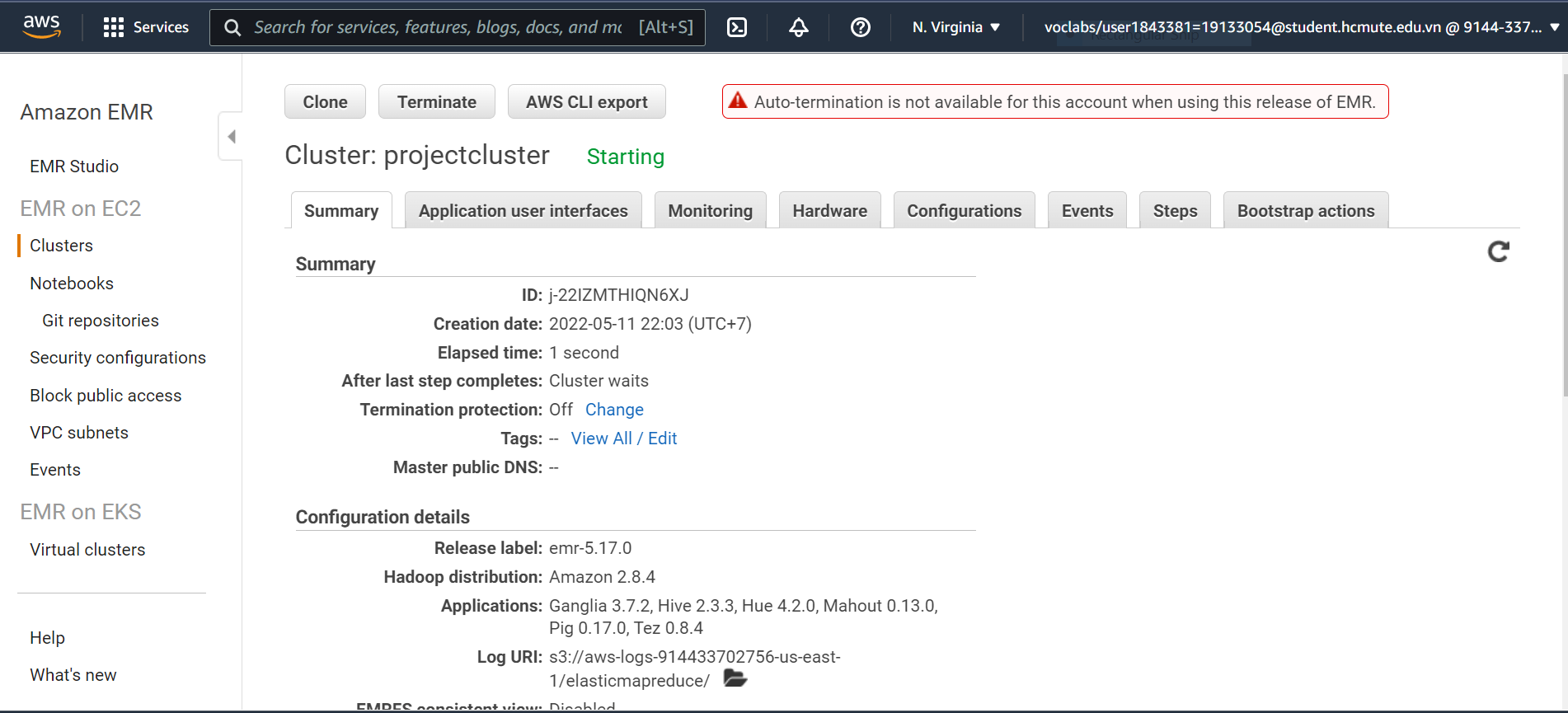
Tìm kiếm service có tên là EMR sau đó chọn Create Cluster



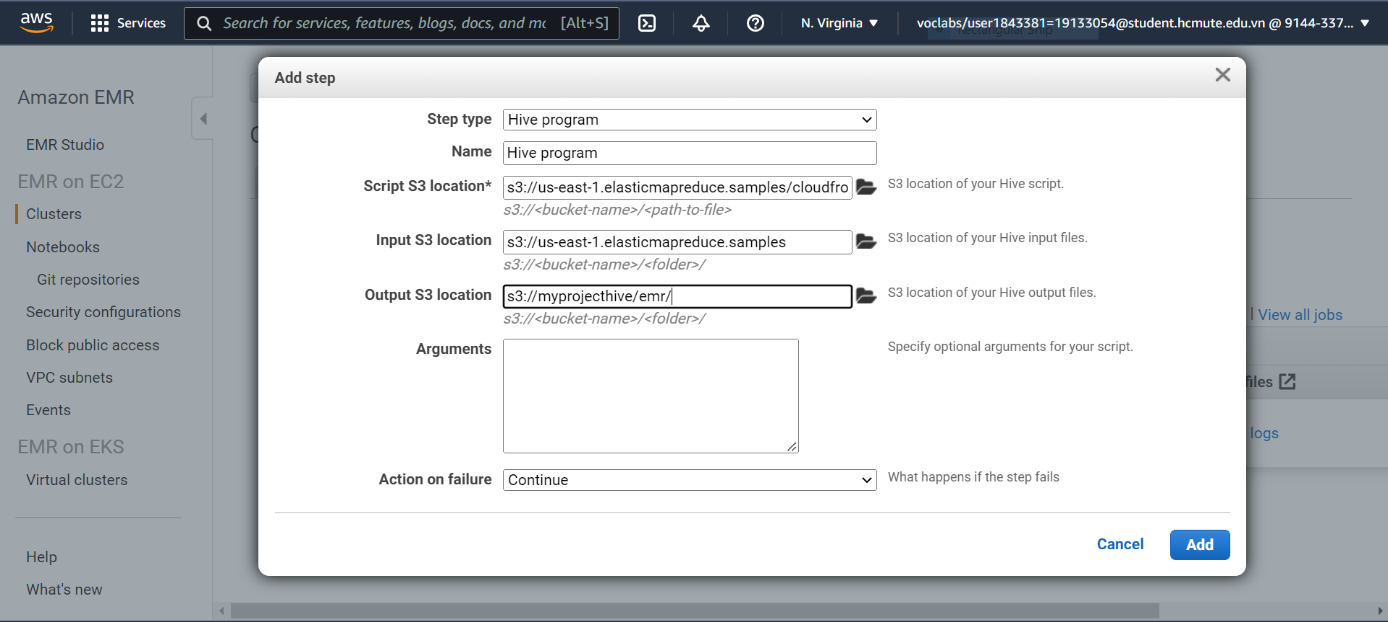


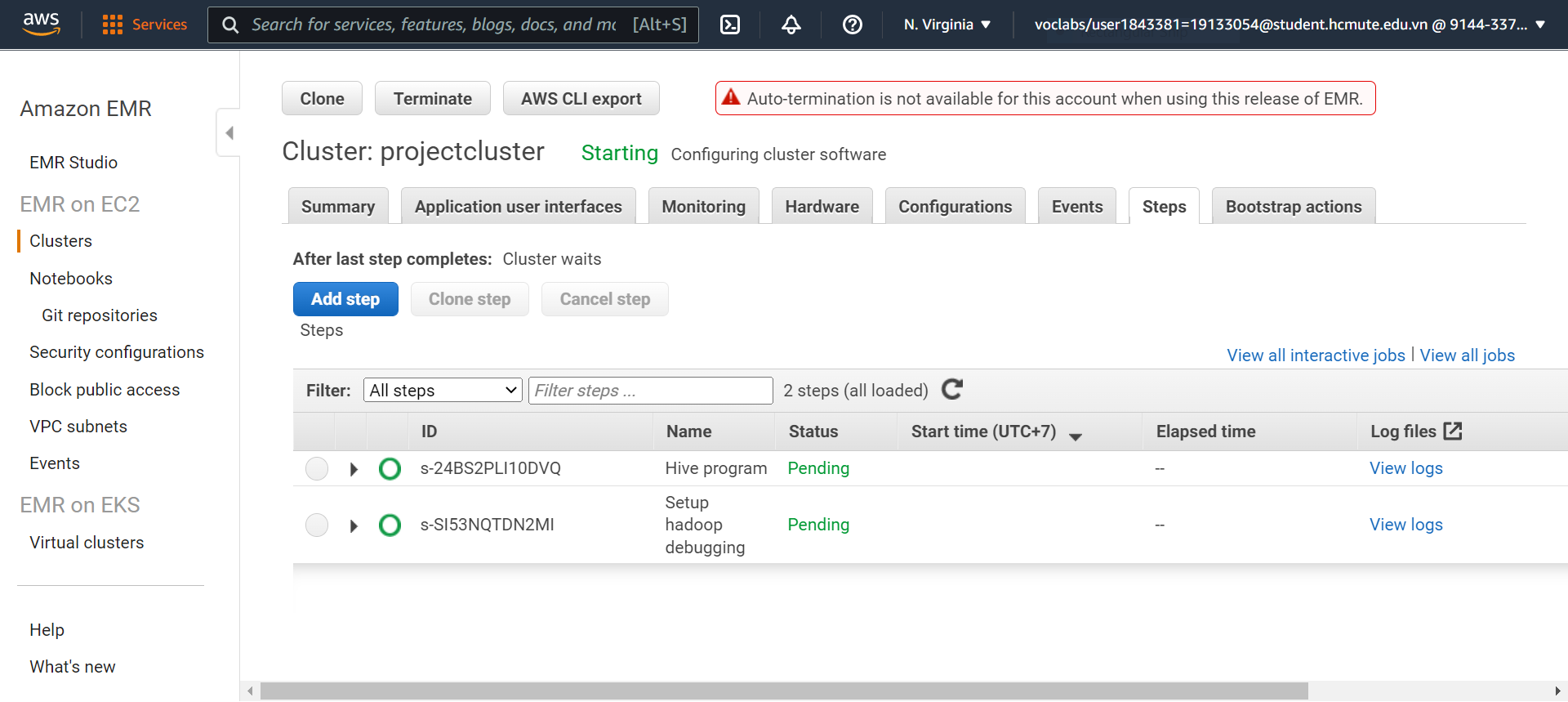
Chọn key pair có sẵn -> Create cluster



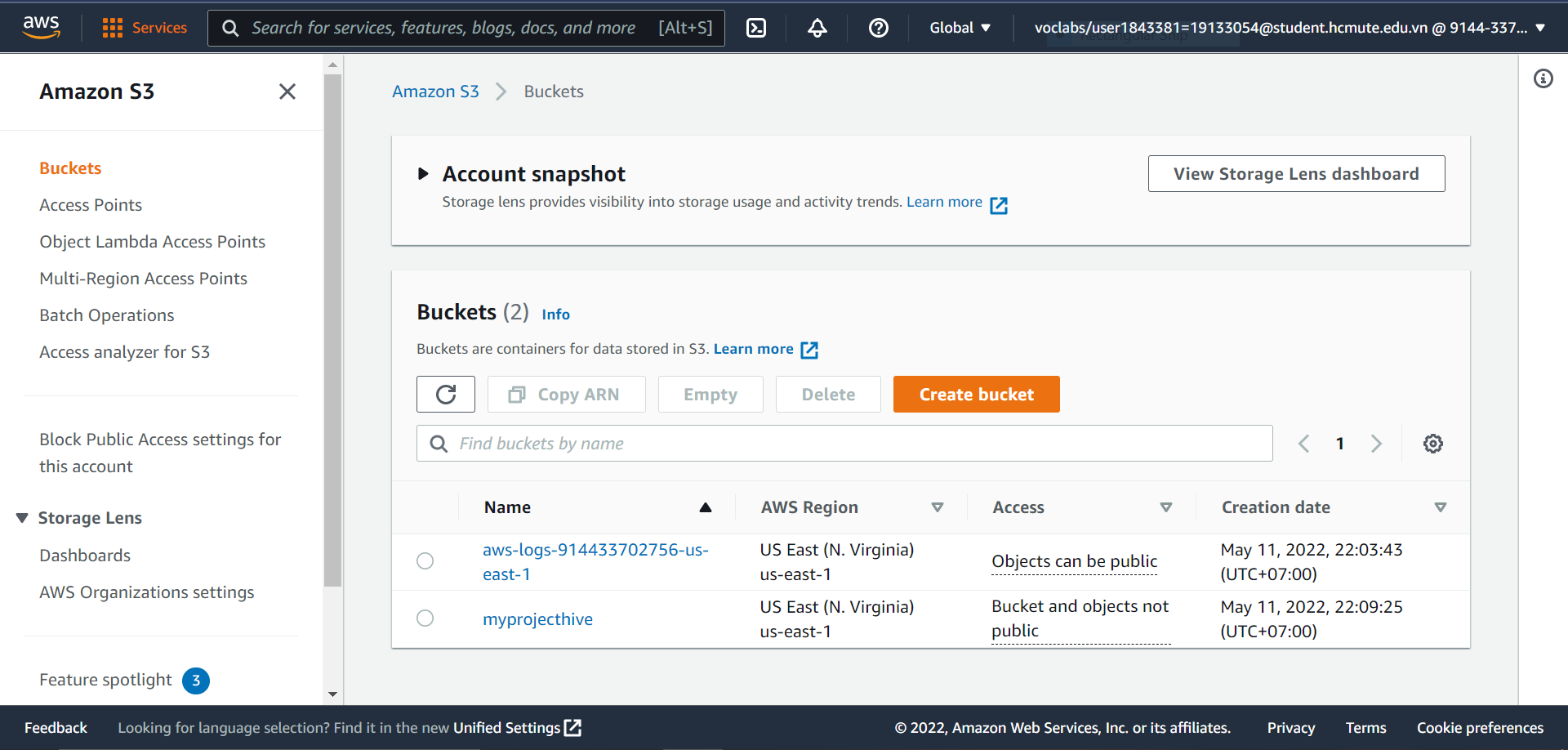


Ở tab step chọn Add step

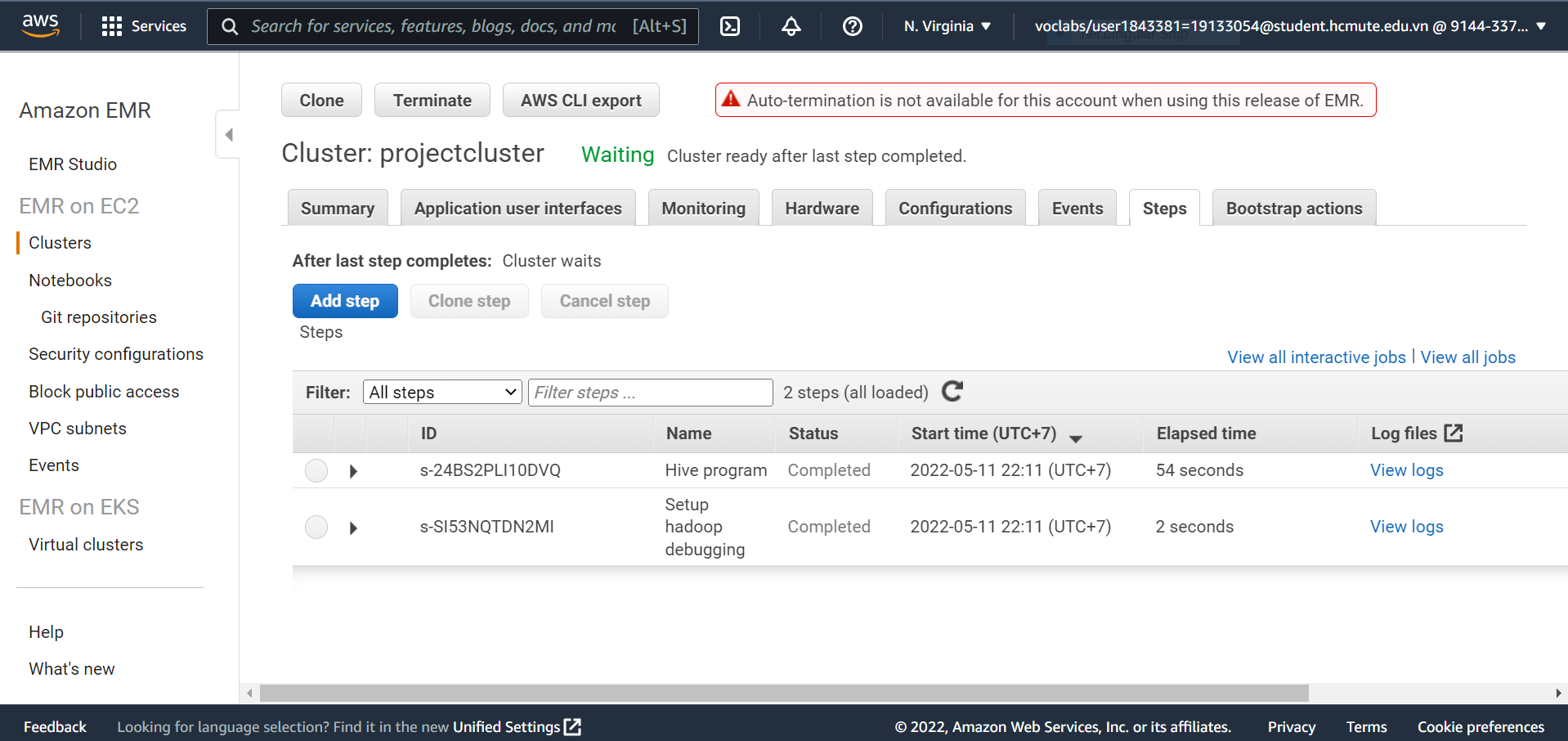


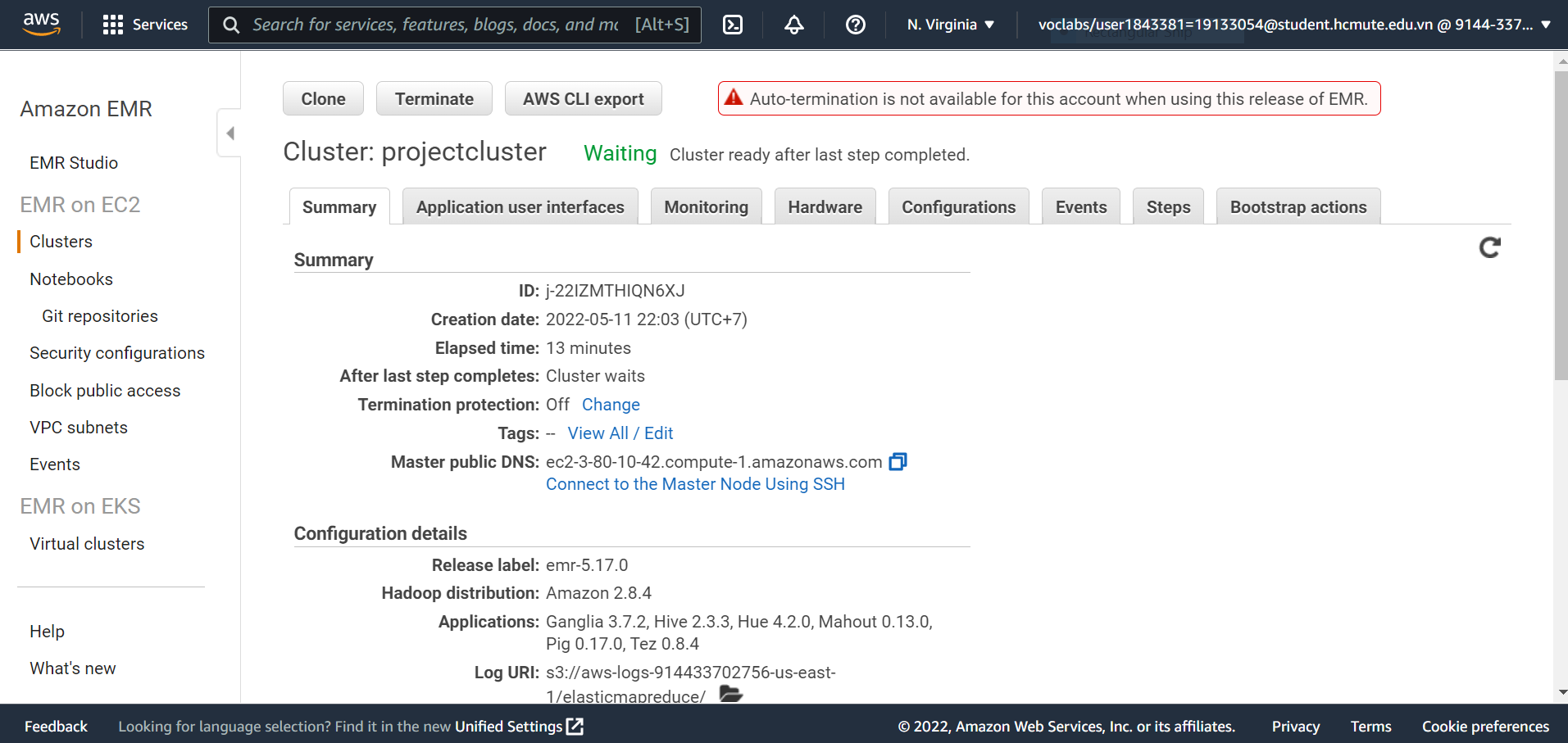


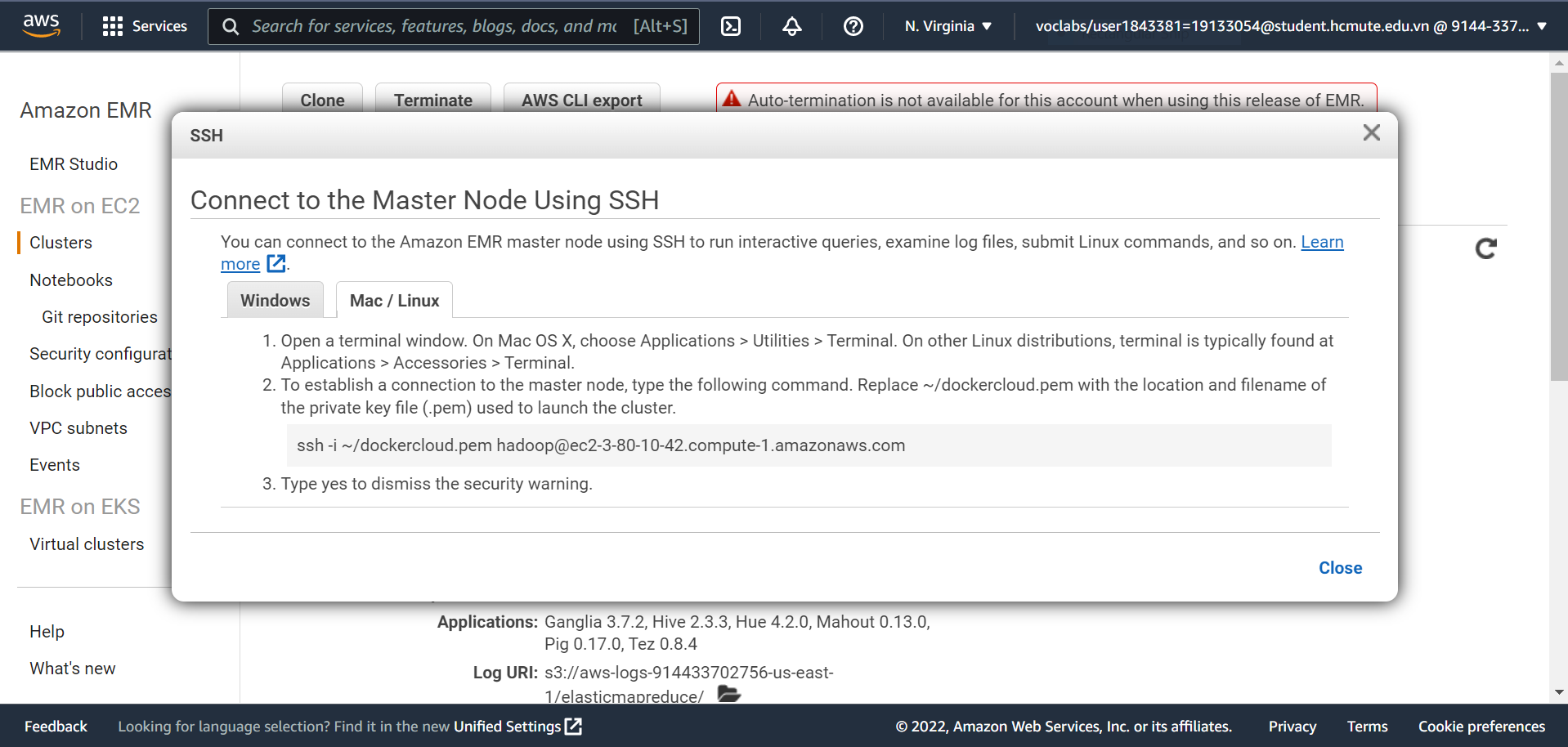
Tạo 1 bucket để lưu trữ input dữ liệu vào để xử lý



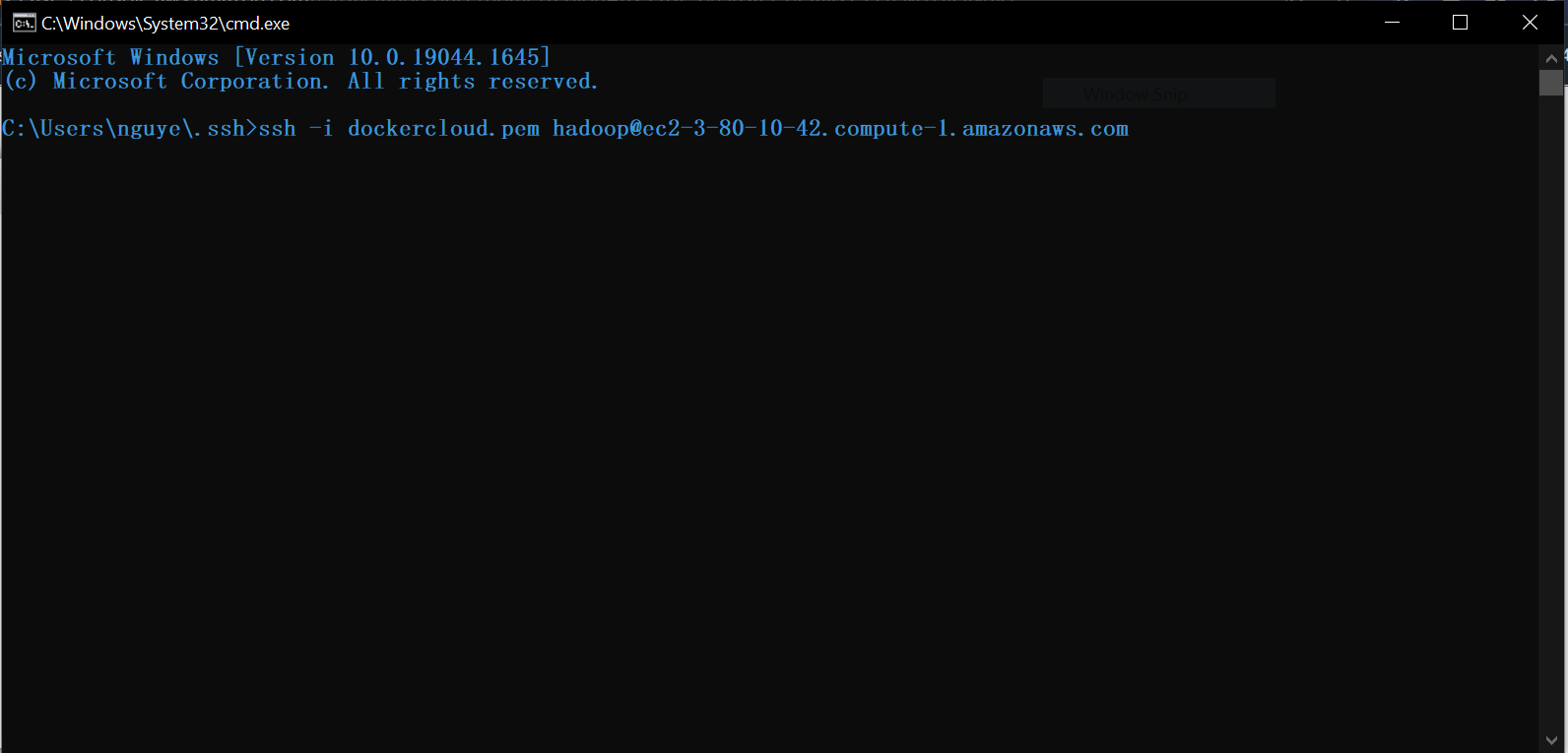
Sau khi tiến trình chuyển sang Completed thì ta chuyển sang tab Summary để lấy chuỗi kết nối





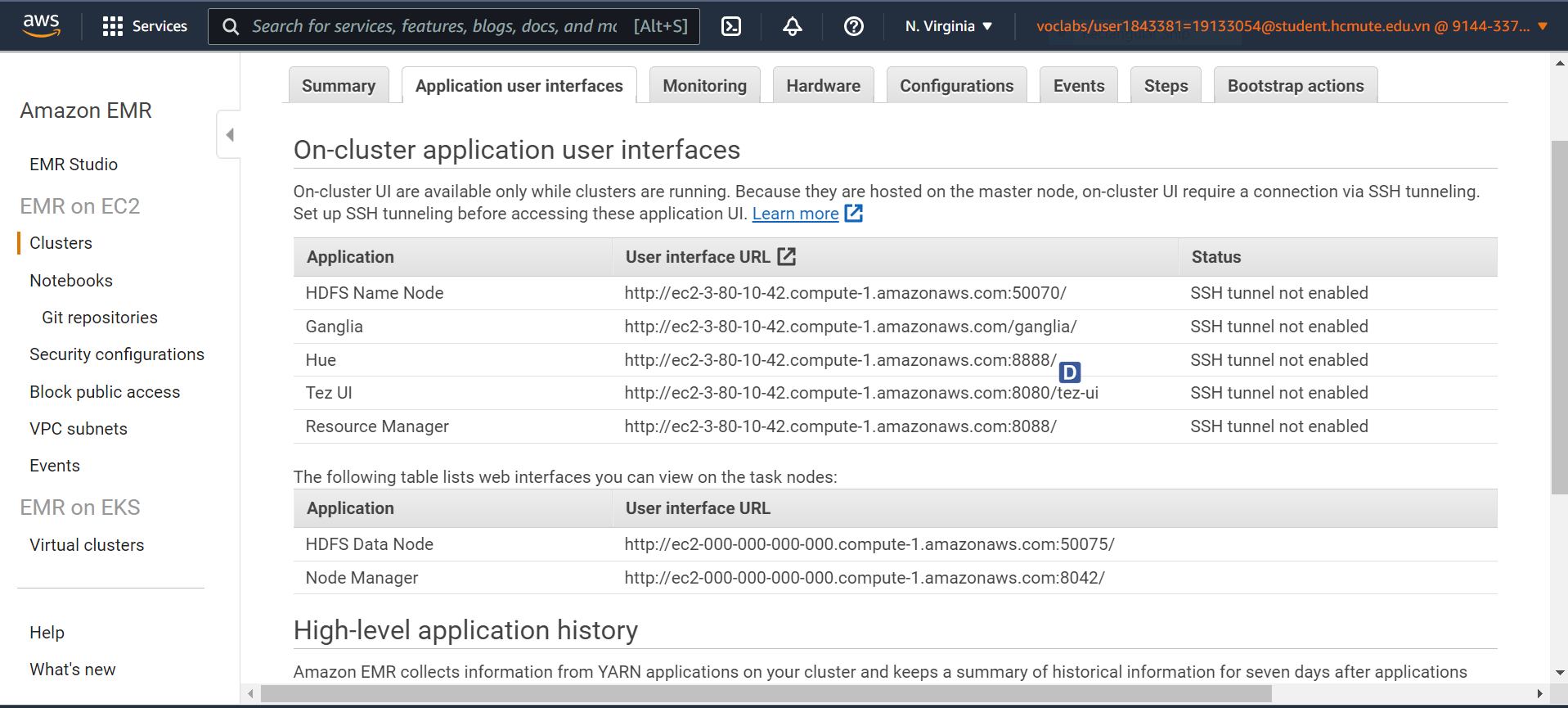


Mở command line và kết nối đến server

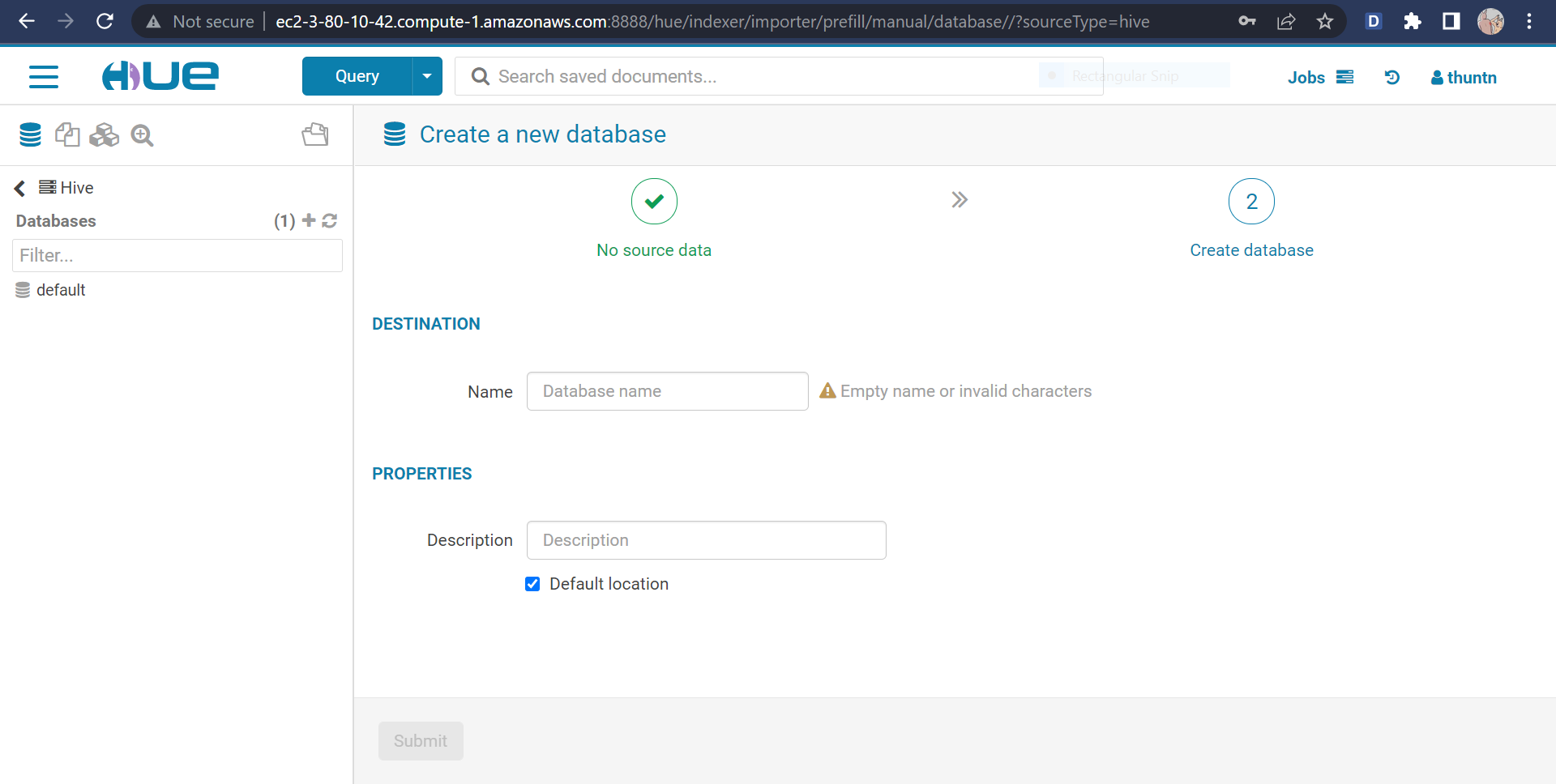




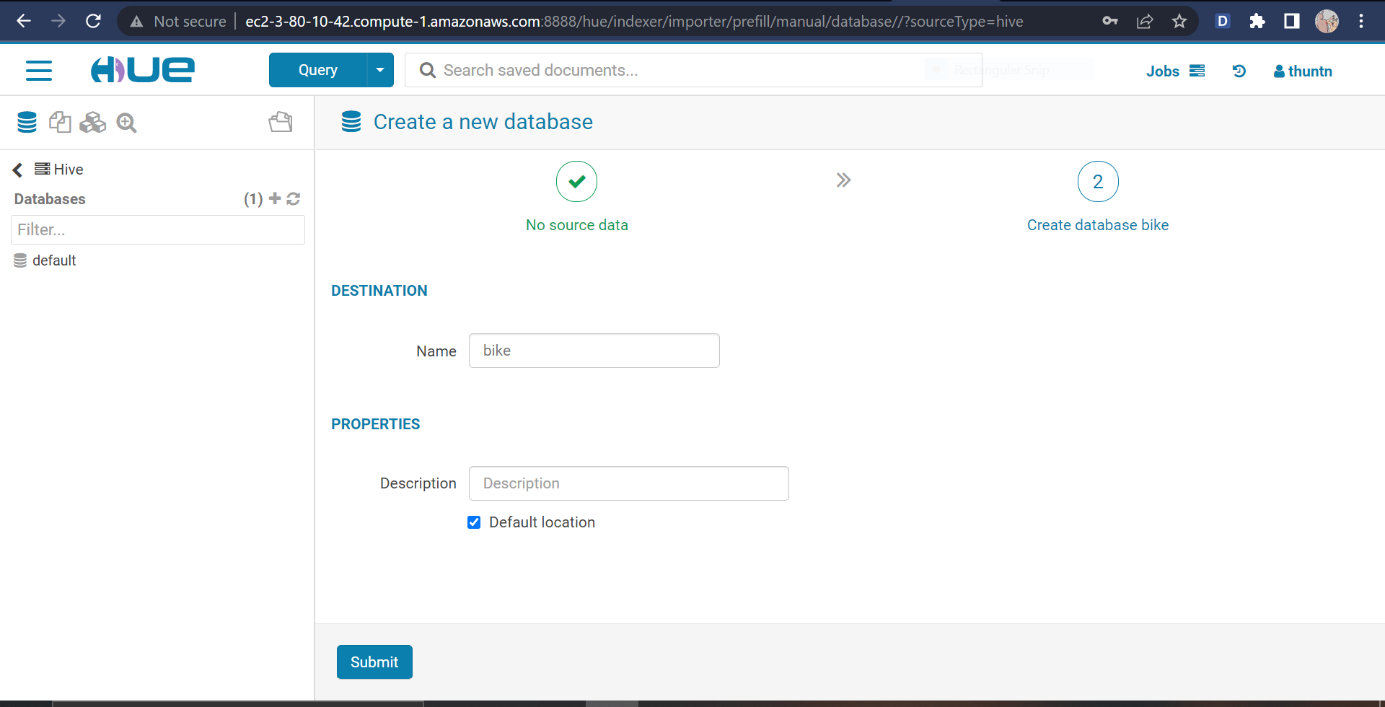
Để chạy trên server ta cần mở các port tương ứng với từng application

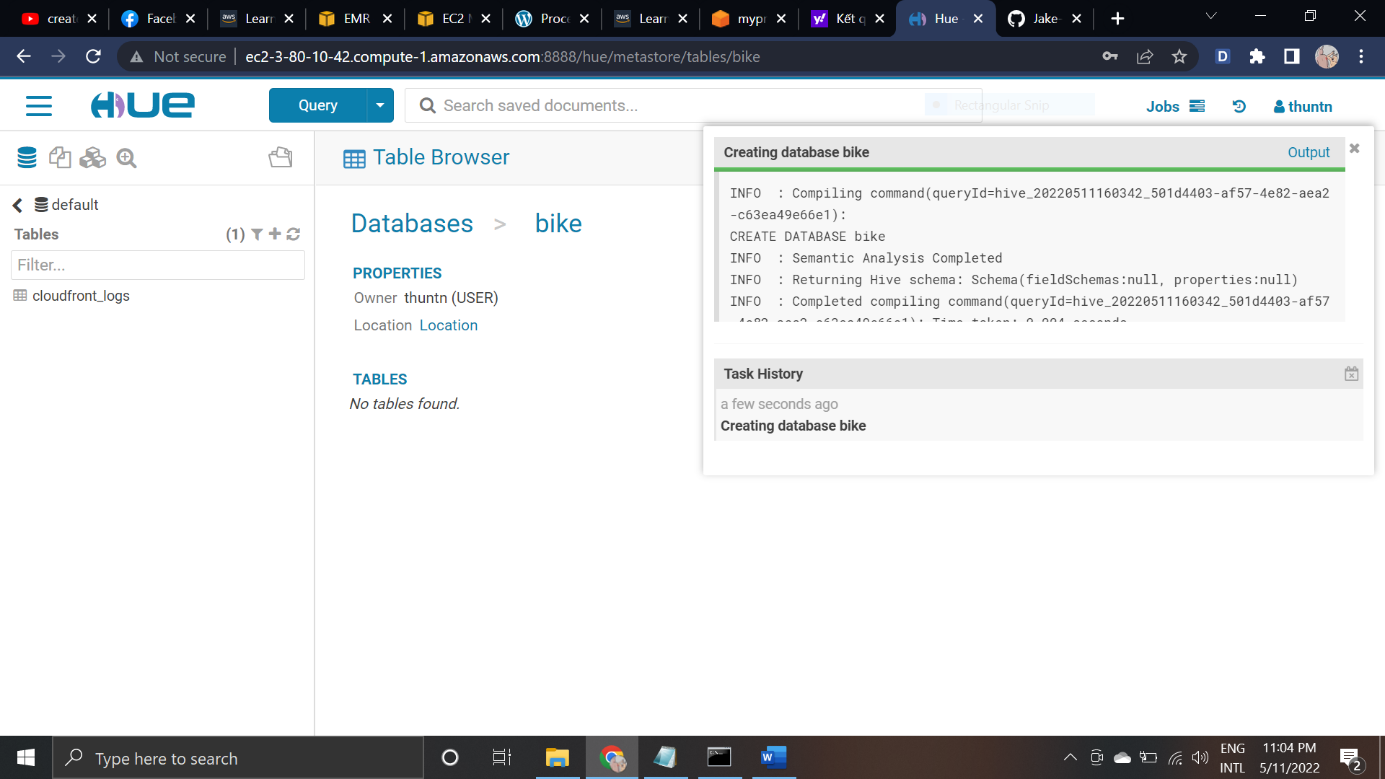


Ta mở port để chạy application HUE thực hiện tạo database và truy vấn trên hive

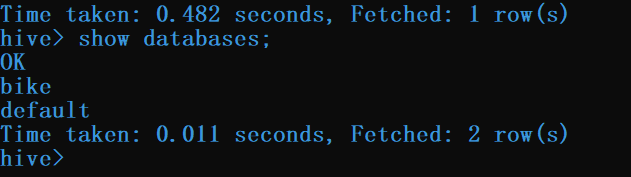


Tạo database có tên bike

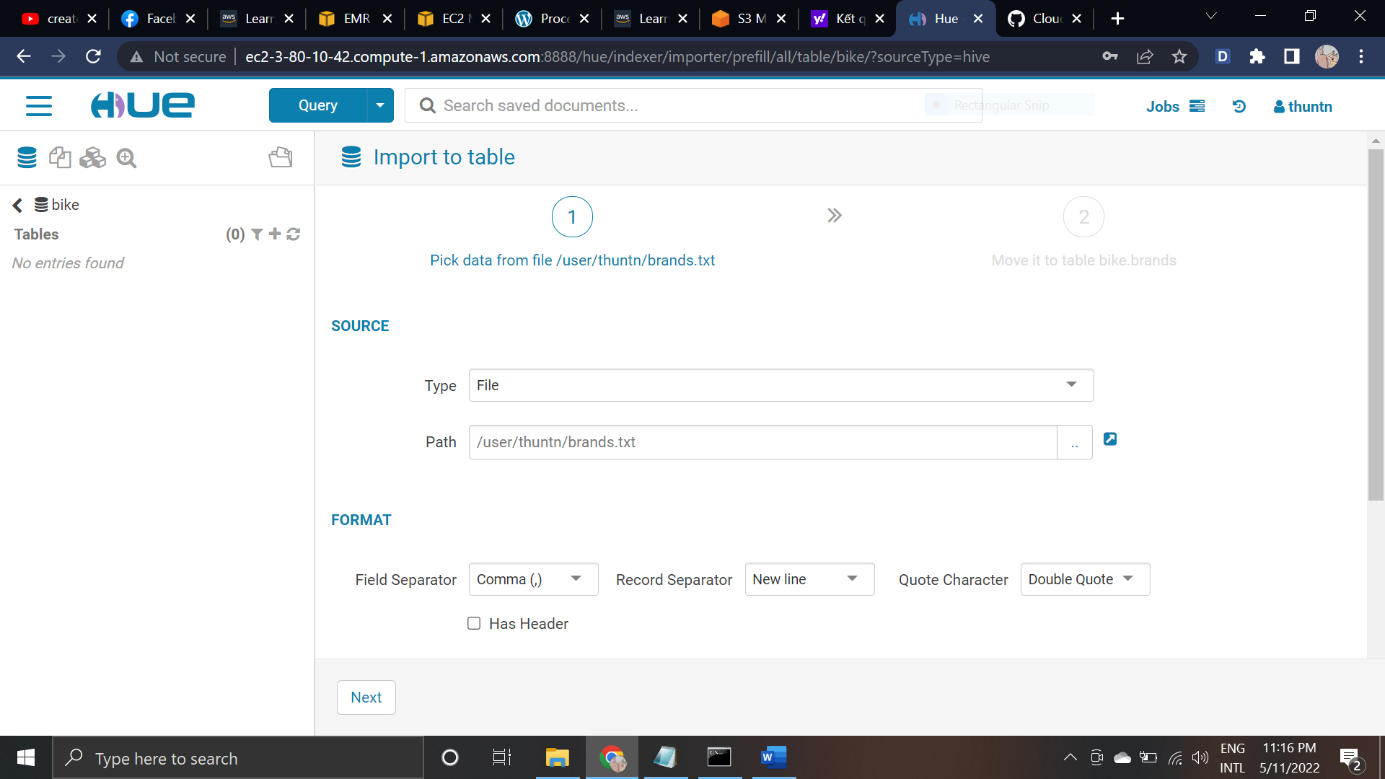


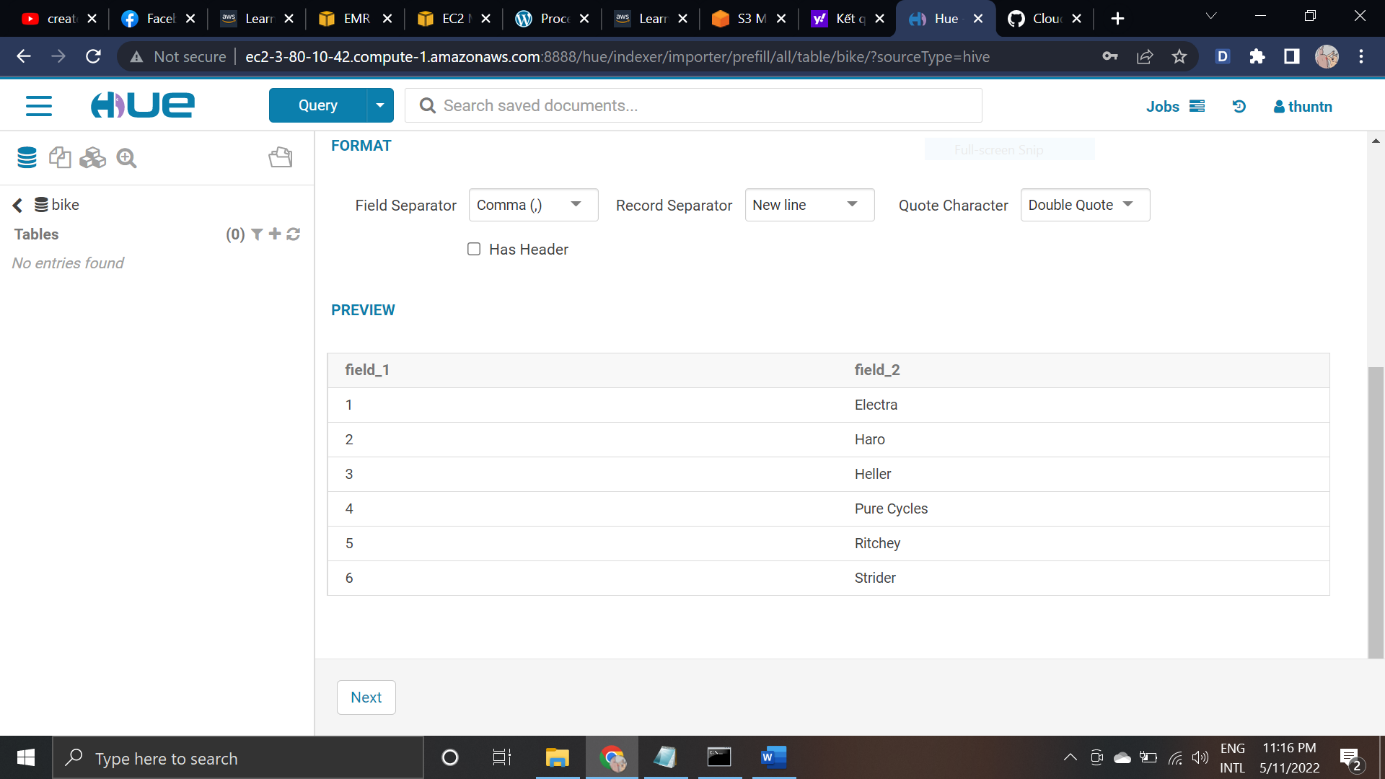


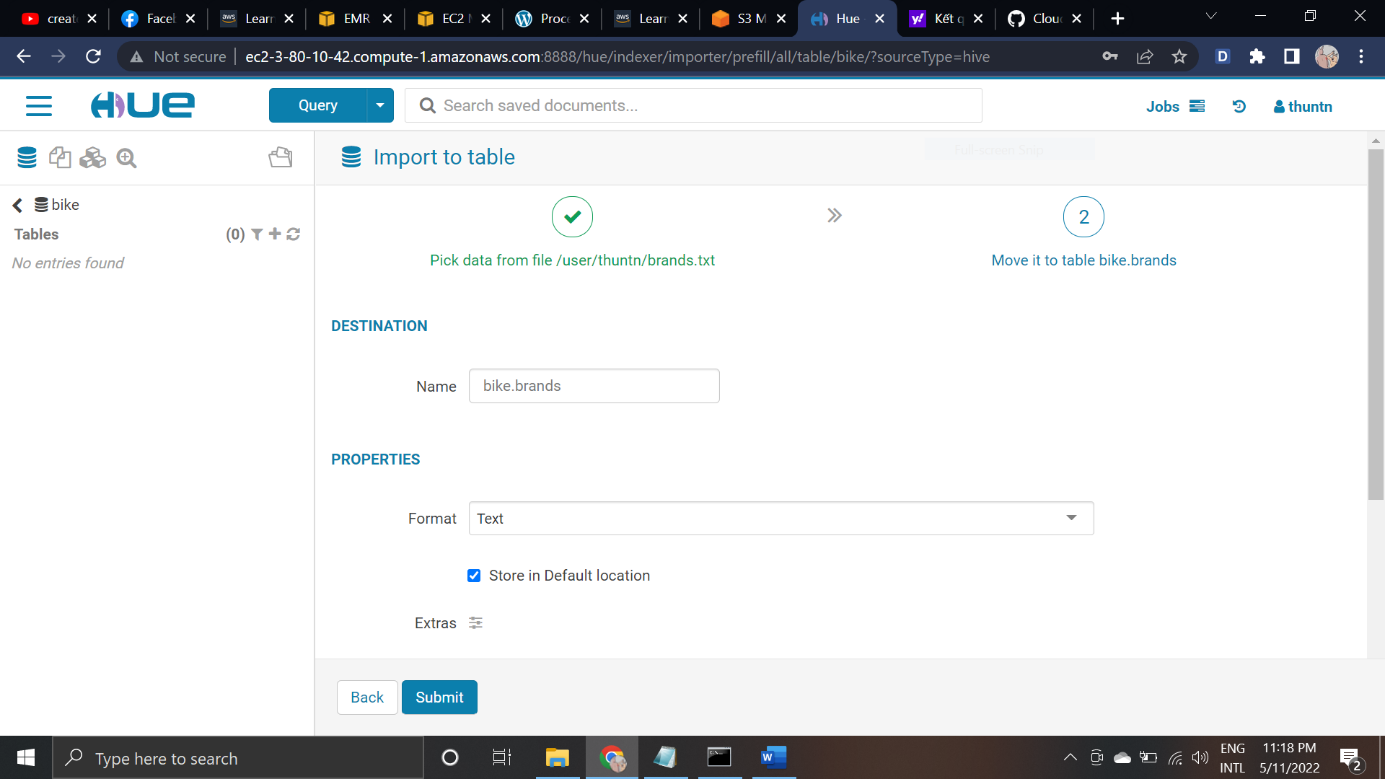
Kiểm tra dưới command line

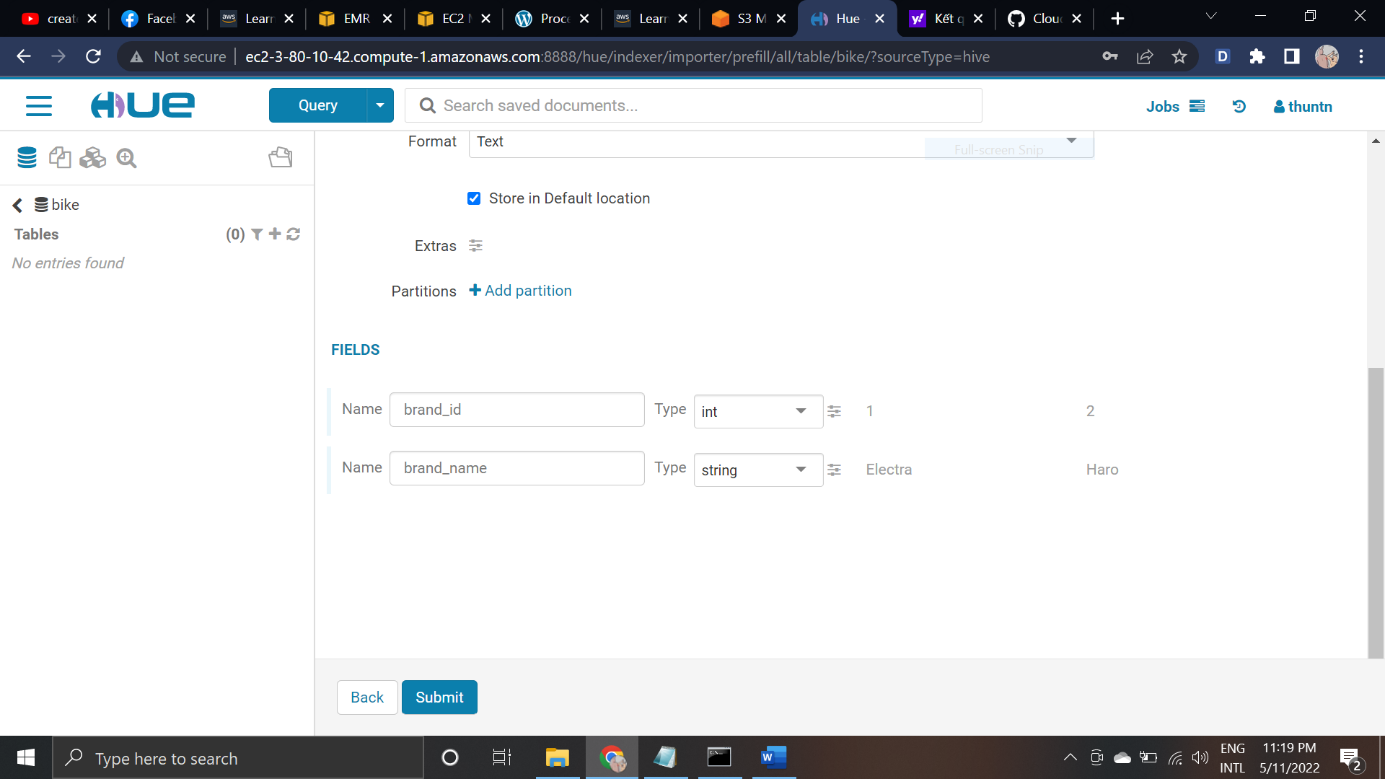


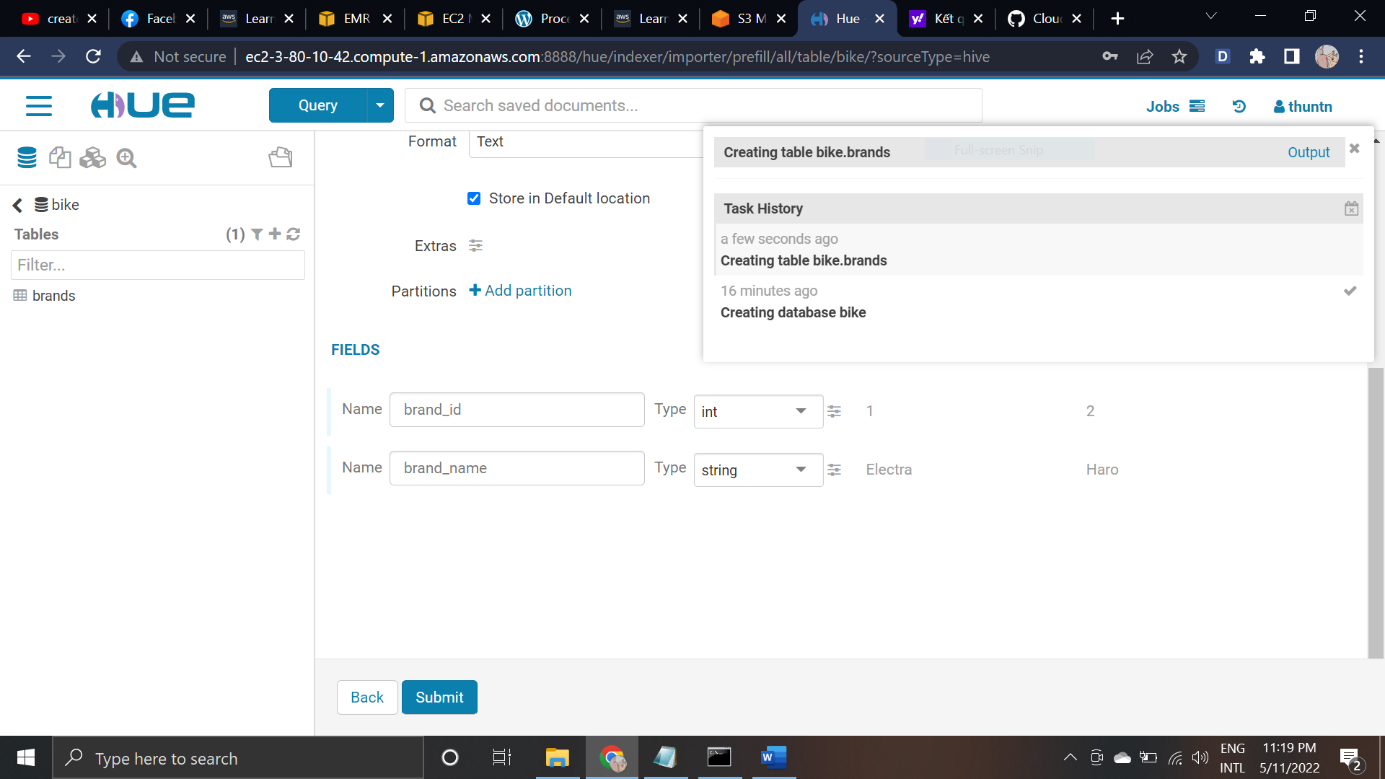
Tạo 1 table và import dữ liệu vào table bằng file txt











Kiểm tra trong command line

