Sau khi thu thập dữ liệu:

Bước 1: tải mysql

Bước 2: cài đặt các bảng:

Bảng 1: nhóm các kĩ năng lại – gọi là category cũng được

|  |
| --- |
| CREATE TABLE skill\_groups (  id INT AUTO\_INCREMENT PRIMARY KEY,  name VARCHAR(100) NOT NULL UNIQUE  ) ENGINE=InnoDB CHARSET=utf8mb4; |

Bảng 2: Lưu từng kĩ năng cụ thể:

|  |
| --- |
| CREATE TABLE skills (  id INT AUTO\_INCREMENT PRIMARY KEY,  name VARCHAR(100) NOT NULL UNIQUE  ) ENGINE=InnoDB CHARSET=utf8mb4; |

Bảng 3: gán kĩ năng vào từng nhóm

|  |
| --- |
| CREATE TABLE skill\_details (  skill\_id INT NOT NULL,  group\_id INT NOT NULL,  PRIMARY KEY (skill\_id, group\_id),  FOREIGN KEY (skill\_id) REFERENCES skills(id) ON DELETE CASCADE,  FOREIGN KEY (group\_id) REFERENCES skill\_groups(id) ON DELETE CASCADE  ) ENGINE=InnoDB CHARSET=utf8mb4; |

Bảng 4: bảng công việc chính

|  |
| --- |
| CREATE TABLE jobs (  id INT AUTO\_INCREMENT PRIMARY KEY,  group\_id INT NOT NULL, -- category của job  title VARCHAR(255) NOT NULL,  link TEXT,  location VARCHAR(255),  experience VARCHAR(100),  work\_location\_detail TEXT,  working\_time VARCHAR(255),  deadline DATE,  salary\_raw VARCHAR(100),  salary\_normalized BIGINT,  currency\_unit VARCHAR(10),  FOREIGN KEY (group\_id) REFERENCES skill\_groups(id)  ON DELETE RESTRICT  ) ENGINE=InnoDB CHARSET=utf8mb4; |

Bảng 5: Thông tin mô tả chi tiết: mô tả, yêu cầu, phúc lợi.

|  |
| --- |
| CREATE TABLE job\_details (  id INT AUTO\_INCREMENT PRIMARY KEY,  job\_id INT NOT NULL,  description TEXT,  requirements TEXT,  benefits TEXT,  FOREIGN KEY (job\_id) REFERENCES jobs(id) ON DELETE CASCADE  ) ENGINE=InnoDB CHARSET=utf8mb4; |

Bảng 6: Gán kỹ năng vào công việc (Job Skills)

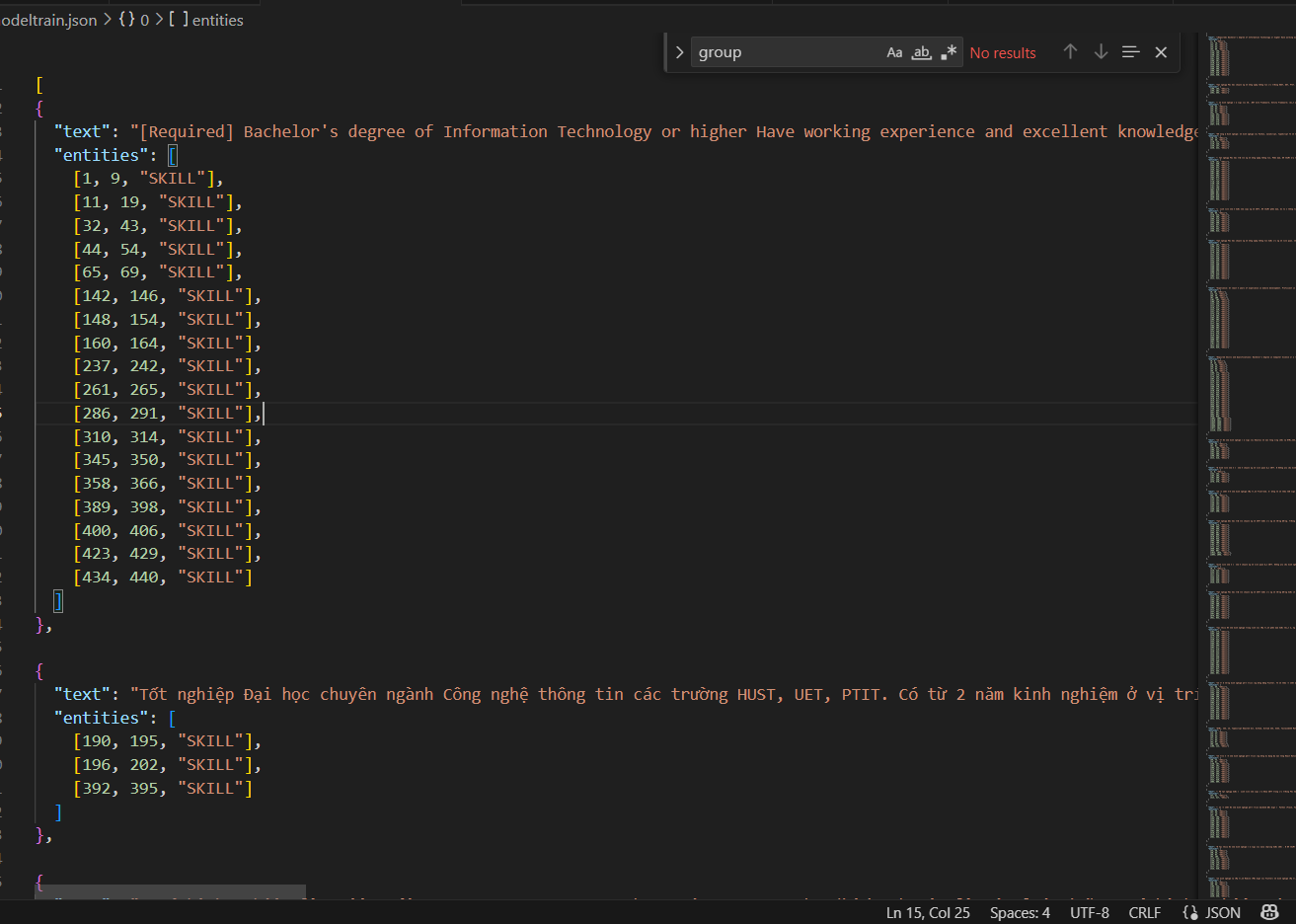
|  |
| --- |
| CREATE TABLE job\_skills (  job\_id INT NOT NULL,  skill\_id INT NOT NULL,  PRIMARY KEY (job\_id, skill\_id),  FOREIGN KEY (job\_id) REFERENCES jobs(id) ON DELETE CASCADE,  FOREIGN KEY (skill\_id) REFERENCES skills(id) ON DELETE CASCADE  ) ENGINE=InnoDB CHARSET=utf8mb4; |

**Bước 3: làm sạch dữ liệu:**

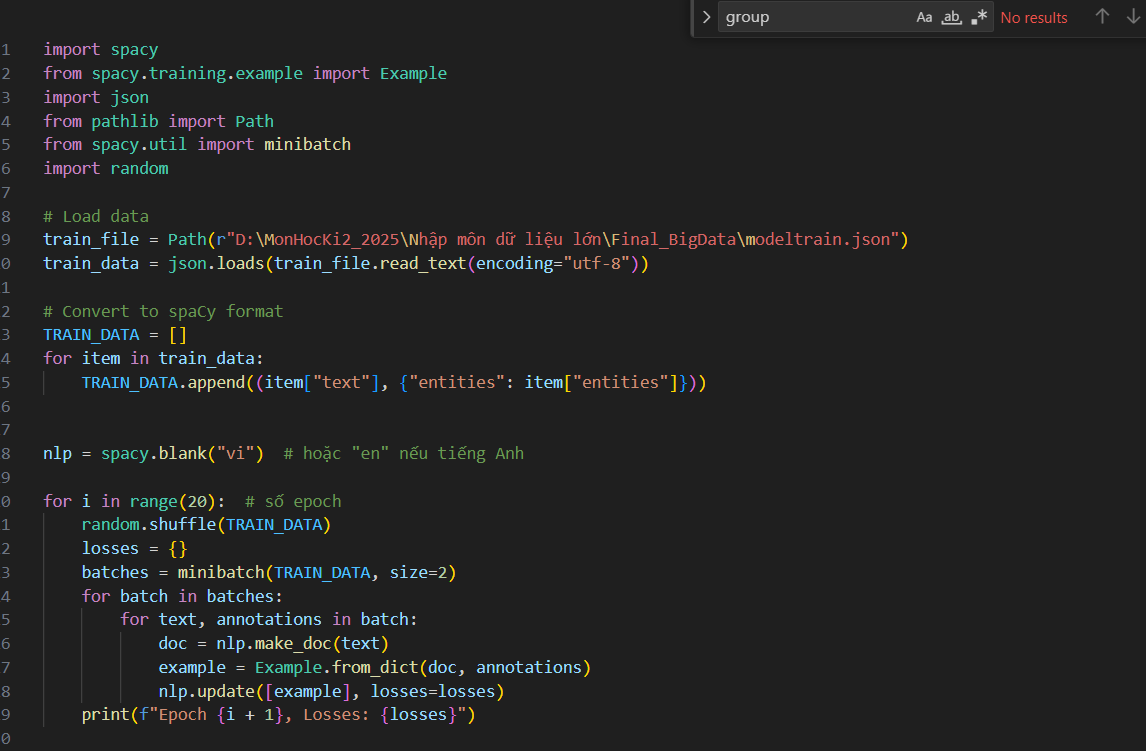
* train dữ liệu để lấy trường dữ liệu name cho bảng Skill. Bằng huấn luyện mô hình nhận diện thực thể (NER) tùy chỉnh với thư viện spaCy. Thì model NER là Nhận diện thực thể có tên là một kỹ thuật trong Xử lý ngôn ngữ tự nhiên (NLP), dùng để tìm và phân loại các cụm từ quan trọng trong văn bản.

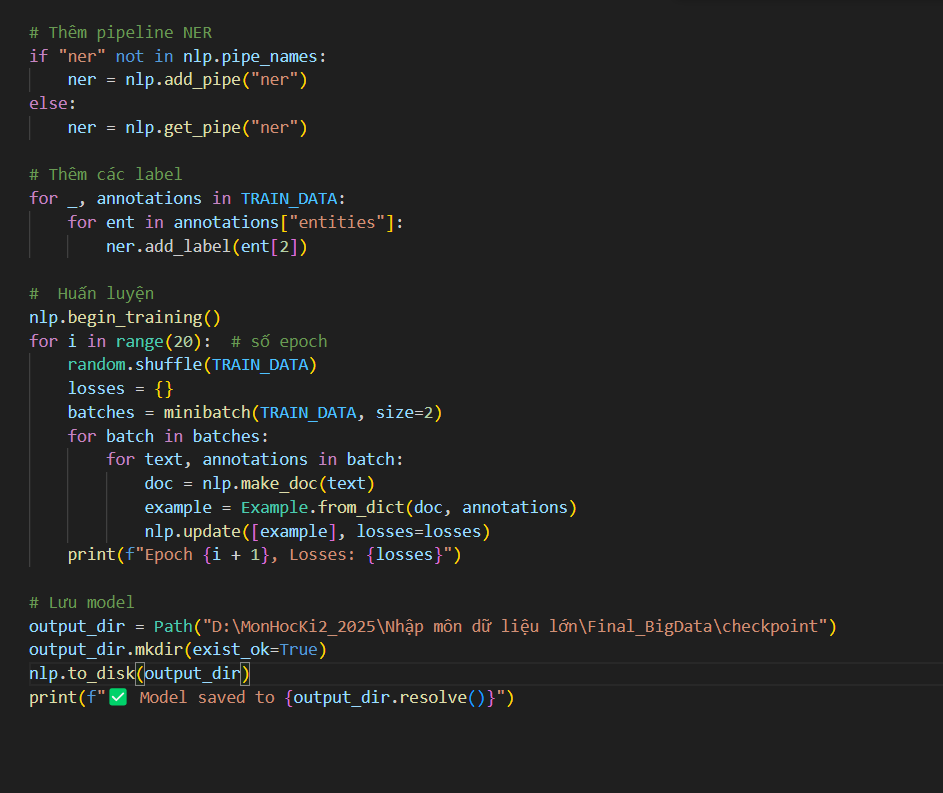
Đầu tiên: tạo file train gồm text, entities.

Có thể áp dụng trang web này để có thể chuyển label: <https://arunmozhi.in/ner-annotator/>



Bước 2: tạo thư mục python train\_model\_ner để có thể lấy dữ liệu cho bảng skill



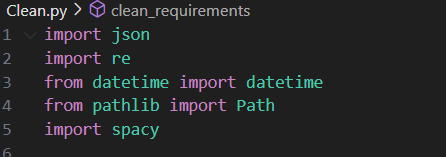


* cài đặt các thư viện cần thiết:

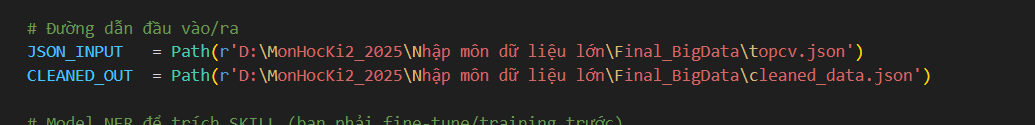
|  |
| --- |
| $ pip install spacy  $ pip install pyvi ==> giúp hỗ trợ tiếng việt |

Bước 3: clean dữ liệu:

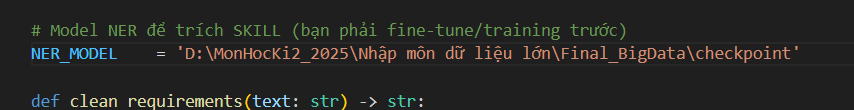
* import thư viện:



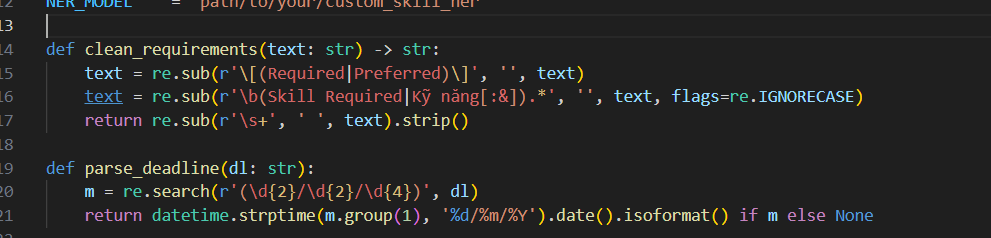
* đưa đường dẫn đầu vào:



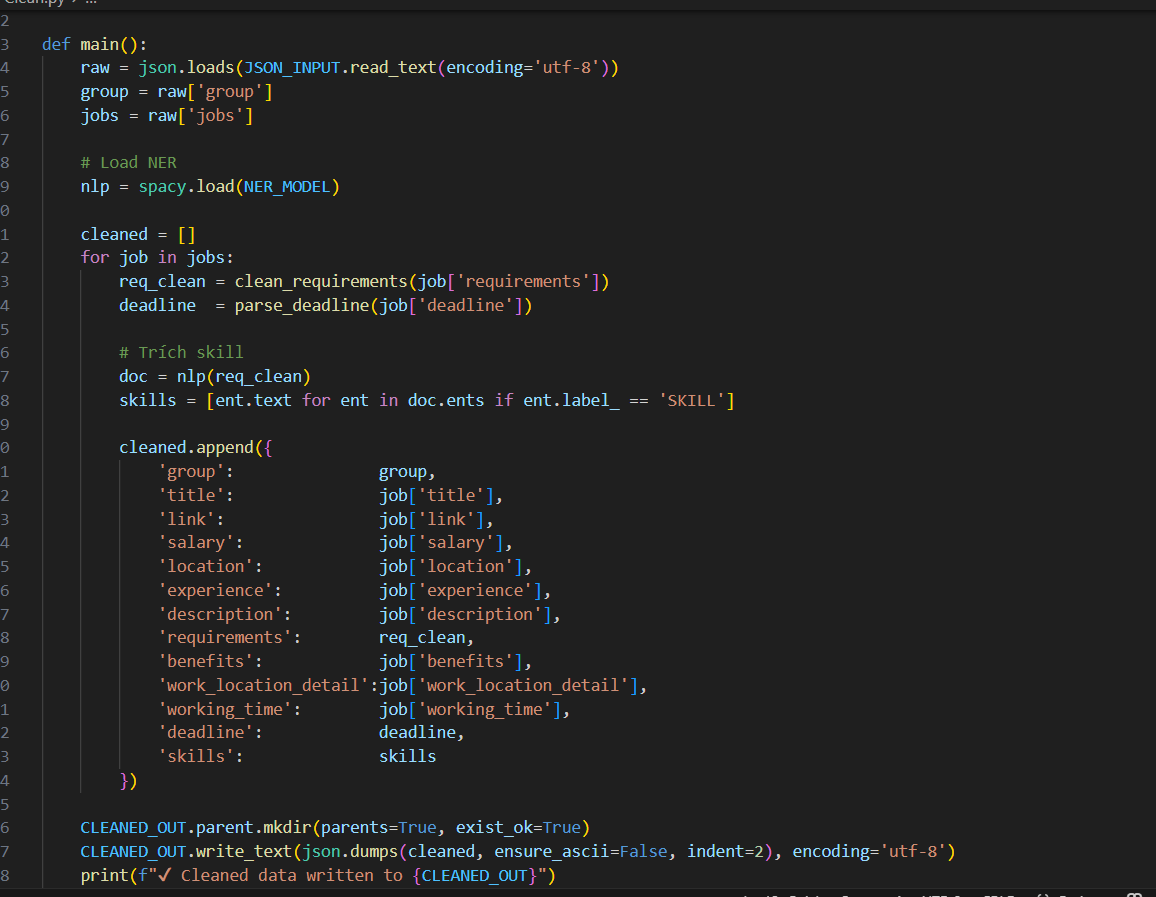
* đưa model đã huấn luyện vào mô hình:

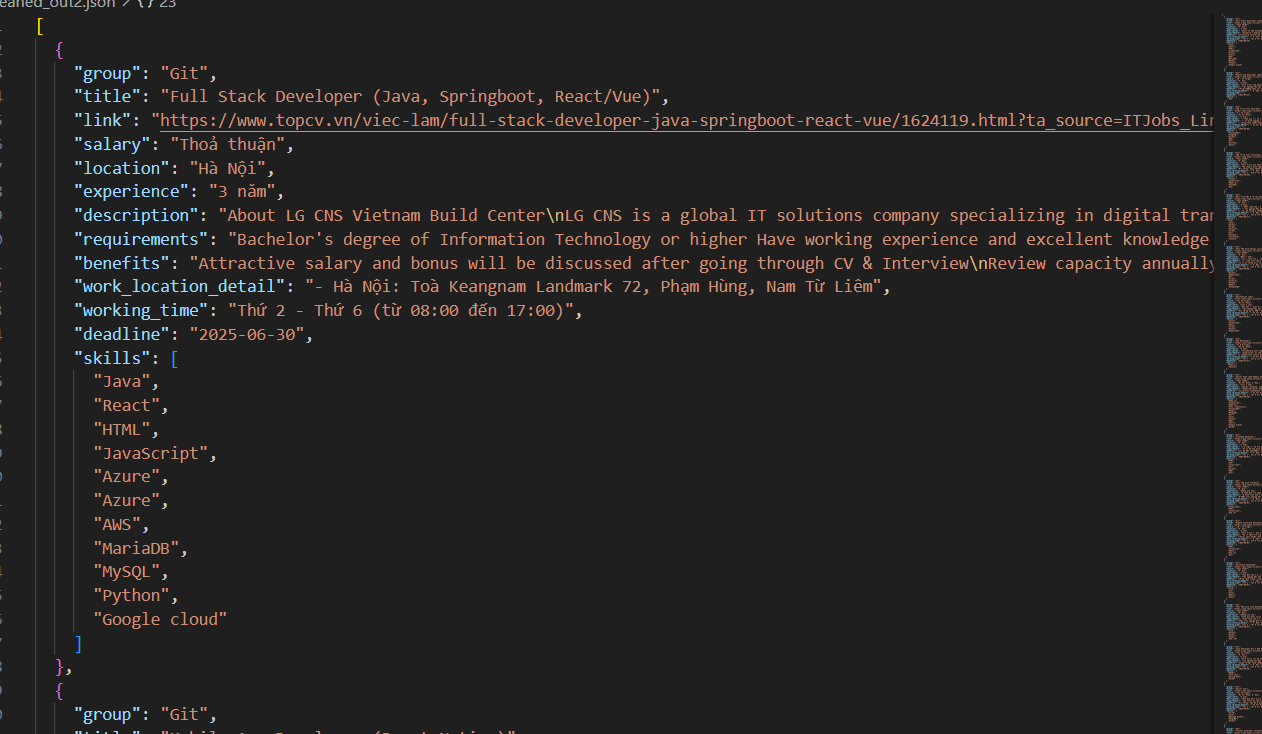


* làm sạch trường dữ liệu requirements, và ngày tháng:



* load model và đưa dữ liệu vào:



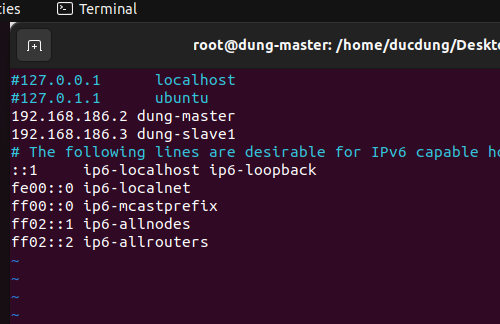
* dữ liệu mẫu làm sạch:  
  

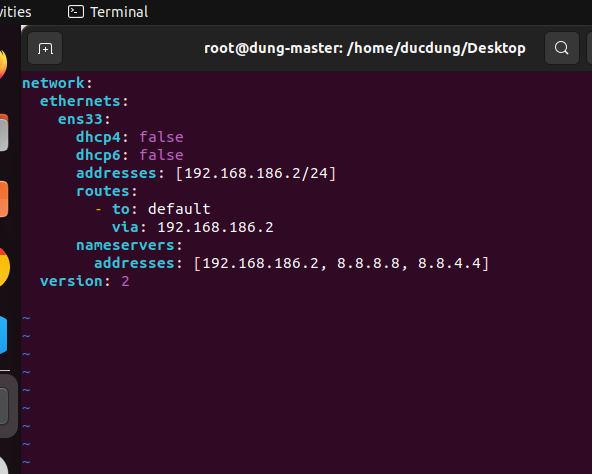
Cấu hình kết nối Hadoop giữa máy Windows và Linux để trao đổi dữ liệu

|  |
| --- |
| USE Big\_Data;  GO  -- 1. Bảng nhóm kỹ năng (skill\_groups)  CREATE TABLE dbo.skill\_groups (  id INT IDENTITY(1,1) PRIMARY KEY,  name NVARCHAR(100) NOT NULL UNIQUE  );  GO  -- 2. Bảng kỹ năng (skills)  CREATE TABLE dbo.skills (  id INT IDENTITY(1,1) PRIMARY KEY,  name NVARCHAR(100) NOT NULL UNIQUE  );  GO  -- 3. Bảng chi tiết gán kỹ năng vào nhóm (skill\_details)  CREATE TABLE dbo.skill\_details (  skill\_id INT NOT NULL,  group\_id INT NOT NULL,  CONSTRAINT PK\_skill\_details PRIMARY KEY CLUSTERED (skill\_id, group\_id),  CONSTRAINT FK\_skill\_details\_skill FOREIGN KEY (skill\_id)  REFERENCES dbo.skills(id)  ON DELETE CASCADE,  CONSTRAINT FK\_skill\_details\_group FOREIGN KEY (group\_id)  REFERENCES dbo.skill\_groups(id)  ON DELETE CASCADE  );  GO  -- 4. Bảng công việc chính (jobs)  CREATE TABLE dbo.jobs (  id INT IDENTITY(1,1) PRIMARY KEY,  group\_id INT NOT NULL, -- category của job  title NVARCHAR(255) NOT NULL,  link NVARCHAR(MAX),  location NVARCHAR(255),  experience NVARCHAR(100),  work\_location\_detail NVARCHAR(MAX),  working\_time NVARCHAR(255),  deadline DATE,  salary\_raw NVARCHAR(100),  salary\_normalized BIGINT,  currency\_unit NVARCHAR(10),  CONSTRAINT FK\_jobs\_group FOREIGN KEY (group\_id)  REFERENCES dbo.skill\_groups(id)  ON DELETE NO ACTION  );  GO  -- 5. Bảng mô tả chi tiết công việc (job\_details)  CREATE TABLE dbo.job\_details (  id INT IDENTITY(1,1) PRIMARY KEY,  job\_id INT NOT NULL,  description NVARCHAR(MAX),  requirements NVARCHAR(MAX),  benefits NVARCHAR(MAX),  CONSTRAINT FK\_job\_details\_job FOREIGN KEY (job\_id)  REFERENCES dbo.jobs(id)  ON DELETE CASCADE  );  GO  -- 6. Bảng gán kỹ năng vào công việc (job\_skills)  CREATE TABLE dbo.job\_skills (  job\_id INT NOT NULL,  skill\_id INT NOT NULL,  CONSTRAINT PK\_job\_skills PRIMARY KEY CLUSTERED (job\_id, skill\_id),  CONSTRAINT FK\_job\_skills\_job FOREIGN KEY (job\_id)  REFERENCES dbo.jobs(id)  ON DELETE CASCADE,  CONSTRAINT FK\_job\_skills\_skill FOREIGN KEY (skill\_id)  REFERENCES dbo.skills(id)  ON DELETE CASCADE  );  GO |

Mục tiêu: Cần kết nối máy Windows với một cụm Hadoop trên Linux để trao đổi dữ liệu. Máy Windows dùng Java 1.8.0\_441 , máy Linux (Ubuntu) chạy Java 1.8.0\_442 và Hadoop 3.x. Mục tiêu là crawl dữ liệu trên Windows, lưu vào CSDL MySQL hoặc MongoDB, rồi đẩy lên sau đó từ hadoop ở ubuntu cũng có thể lấy dữ liệu từ hadoop của Window. Và ngược lại.

1. Cấu hình lại mạng thành :





1. kết nối mạng từ windows sang linux:

* nếu k ping được mạng!

 Vào **Control Panel** → **Windows Defender Firewall** → **Advanced settings**.

 Chọn **Inbound Rules** → tìm **File and Printer Sharing (Echo Request – ICMPv4-In)**.d

 Nháy đúp, trong tab **Advanced** chọn bật (Enable) cho profile **Private** (và Public nếu cần).

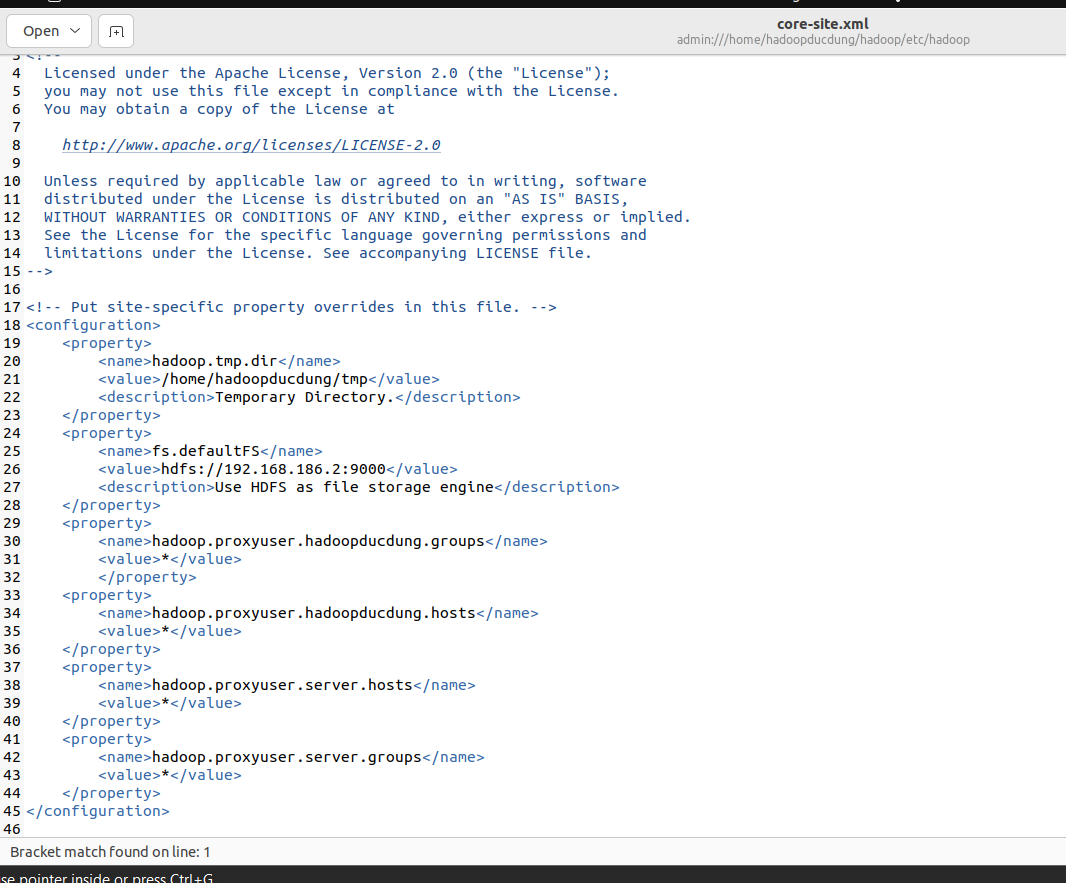
* sau bước đó mà vẫn k được !

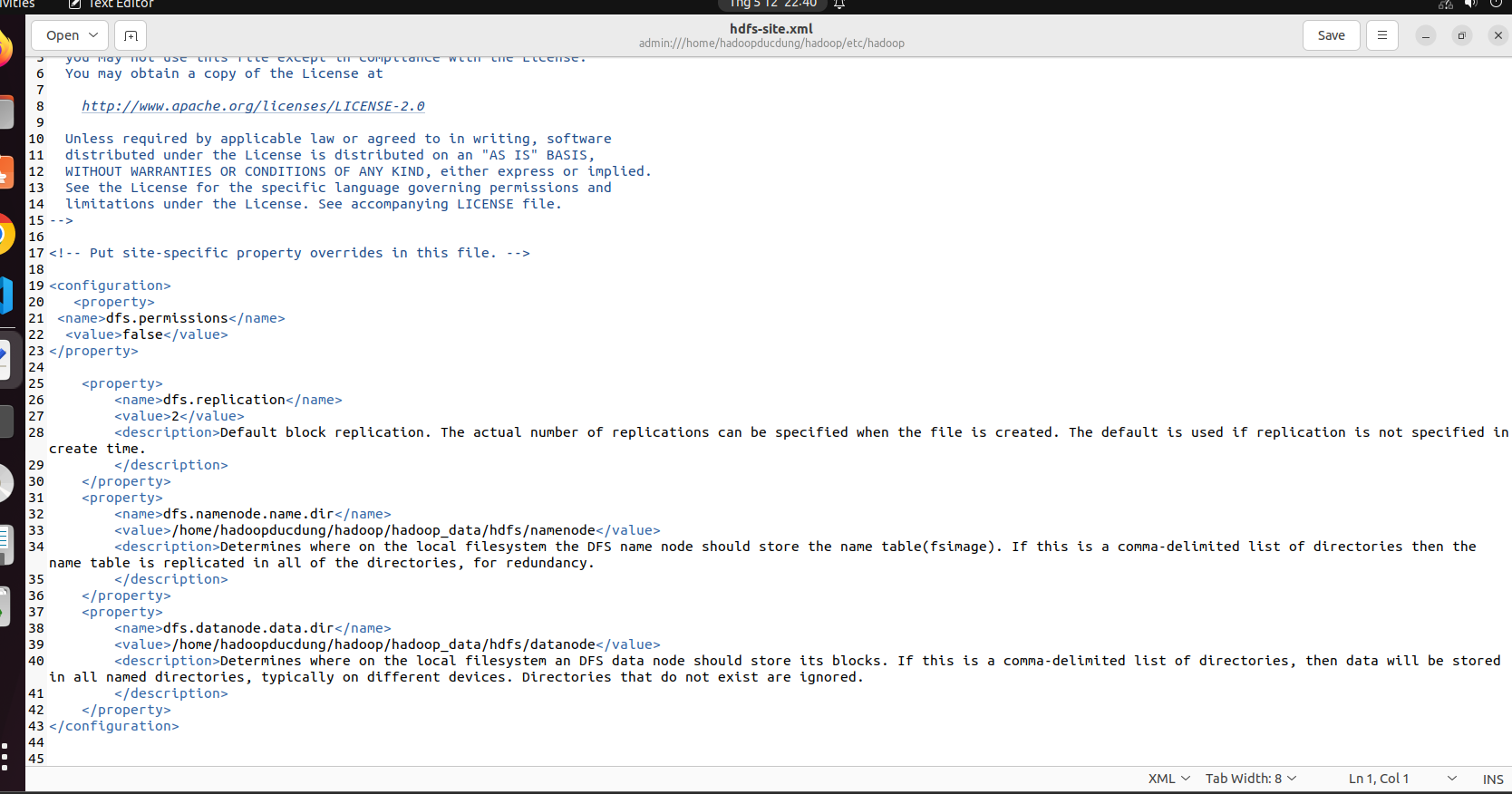
tắt tưởng lửa tại win:

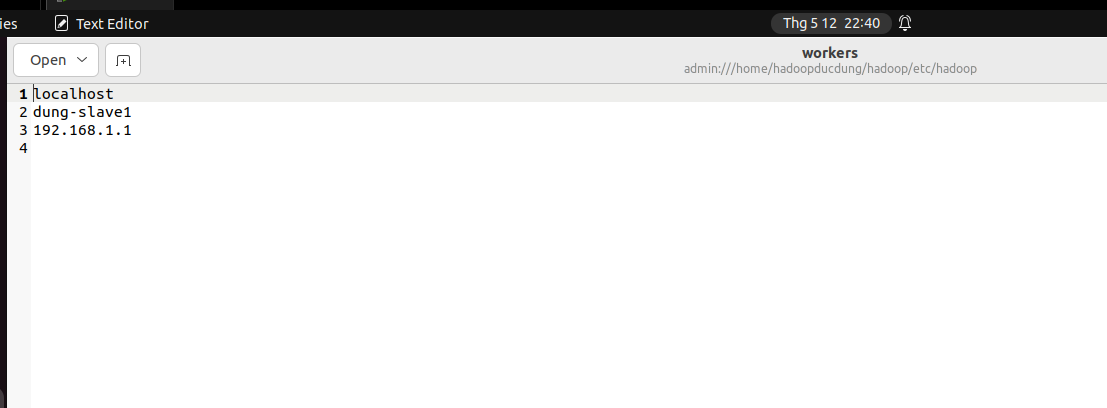
netsh advfirewall set allprofiles state off

* nếu ping thành công

B3: sửa ở file linux:







b4: thiết lập file core-site.xml trong C:\HADOOP\hadoop-3.3.6\etc\hadoop

|  |
| --- |
| <configuration>    <property>      <name>fs.defaultFS</name>      <value>hdfs://dung-master:9000</value>    </property>  </configuration> |

Chỉnh sửa lại file yarn.xml

|  |
| --- |
| <property>  <name>yarn.resourcemanager.address</name>  <value>dung-master:8032</value>  </property> |

<http://dung-master:9870/> để chạy lên giao diện UI!

Bước 5: Nếu muốn ghi lên file hadoop ở windows

1. **Trên Ubuntu**, mở file $HADOOP\_HOME/etc/hadoop/hdfs-site.xml, thêm vào trong <configuration>:

|  |
| --- |
| <property>  <name>dfs.permissions</name>  <value>false</value>  </property> |

1. **Restart HDFS**:

bash

Sao chépChỉnh sửa

$HADOOP\_HOME/sbin/stop-dfs.sh

$HADOOP\_HOME/sbin/start-dfs.sh

1. **Trên Windows**, bạn chỉ có thể trực tiếp đọc:

Nếu muốn ghi phải tạo file ở máy chính là master rồi cấp quyền :

hdfs dfs -chown Admin /win\_only ==> cấp quyền cho admin

hdfs dfs -chmod 700 /win\_only ==> cấp quyền cho user

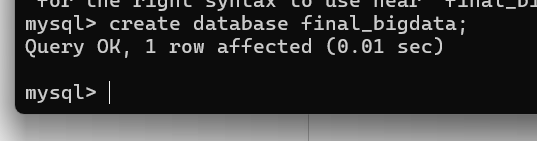
hdfs dfs -chmod 755 /win\_only ==> cho phép mọi người đọc trên trang UI

Bước 6: Cài đặt mySQL:

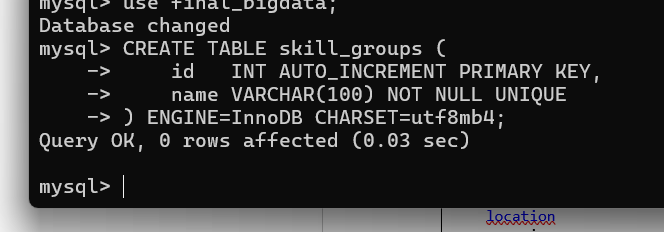
* 1. Setup mysql
  2. Tạo các bảng trong mysql

B1: tạo database:

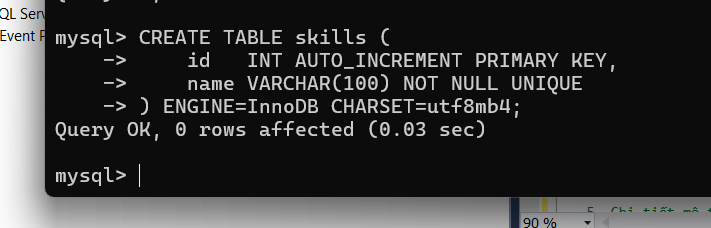
$ create database final\_bigdata;



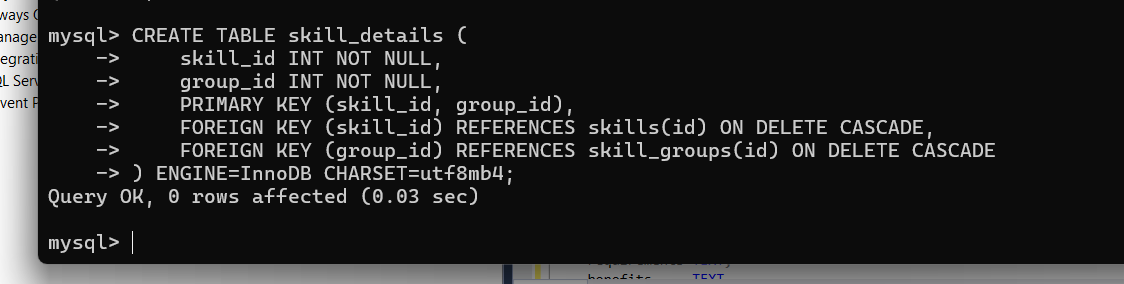
B2: Insert các bảng vào table:

--skill group 

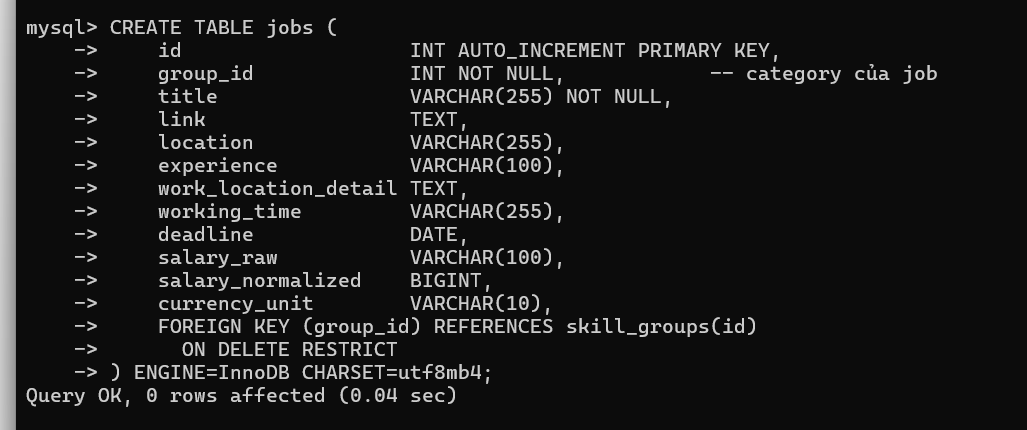
-- skill:



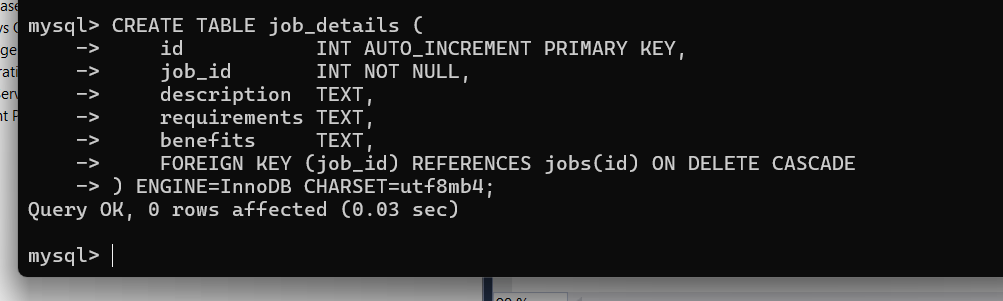
* Skill detail:



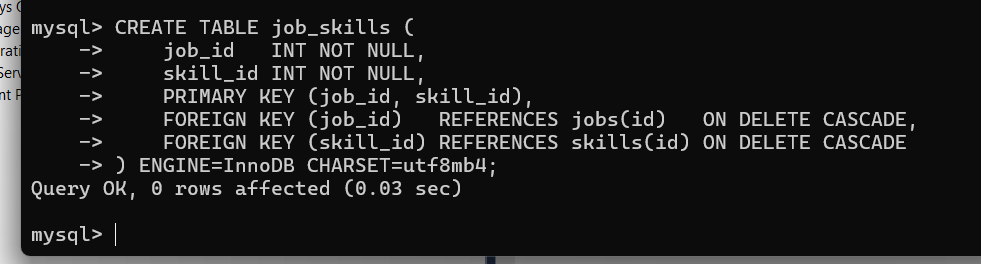
* Job:



* Job detail:



* Skill job:



Bước 7: Thêm dữ liệu vào database ở mysql để có thể lưu dữ liệu:

|  |
| --- |
| import json  import mysql.connector  from pathlib import Path  from datetime import datetime  import re  DB\_CONFIG = {      "host": "localhost",      "user": "root",      "password": "123456789",      "database": "final\_bigdata",      "charset": "utf8mb4"  }  # Load file cleaned\_data.json sau khi đã làm sạch dữ liệu  with open(r"D:\MonHocKi2\_2025\Nhập môn dữ liệu lớn\Final\_BigData\cleaned\_data.json", "r", encoding="utf-8") as f:      jobs = json.load(f)  conn = mysql.connector.connect(\*\*DB\_CONFIG)  cur = conn.cursor()  # 1. Thêm bảng group (skill\_groups)  groups = set(j['group'] for j in jobs)  category\_ids = {}  for g in groups:      cur.execute("INSERT IGNORE INTO skill\_groups(name) VALUES (%s)", (g,))  conn.commit()  cur.execute("SELECT id, name FROM skill\_groups")  for cid, cname in cur.fetchall():      category\_ids[cname] = cid  # 2. Insert skills  skill\_set = {skill for job in jobs for skill in job['skills']}  skill\_ids = {}  for s in skill\_set:      cur.execute("INSERT IGNORE INTO skills(name) VALUES (%s)", (s,))  conn.commit()  cur.execute("SELECT id, name FROM skills")  for sid, sname in cur.fetchall():      skill\_ids[sname] = sid  # 3. Thêm 2 trường dữ liệu group\_id và skill\_id vào bảng skill\_details  for job in jobs:      gid = category\_ids[job['group']]      for s in job['skills']:          sid = skill\_ids.get(s)          if sid:              cur.execute(                  "INSERT IGNORE INTO skill\_details(skill\_id, group\_id) VALUES (%s, %s)",                  (sid, gid)              )  conn.commit()  # 4. thêm các bảng jobs, job\_details, job\_skills  for job in jobs:      # parse deadline      dl = job.get('deadline')      deadline = datetime.strptime(dl, "%Y-%m-%d").date() if dl else None      # insert into jobs with group\_id      group\_id = category\_ids[job['group']]      cur.execute(          """          INSERT INTO jobs(              group\_id, title, link, location,              experience, work\_location\_detail, working\_time,              deadline, salary\_raw, salary\_normalized, currency\_unit          ) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)          """, (              group\_id,              job['title'], job['link'], job['location'],              job['experience'], job['work\_location\_detail'], job['working\_time'],              deadline,              job['salary\_raw'], job['salary\_normalized'], job['currency\_unit']          )      )      job\_id = cur.lastrowid      # thêm dữ liệu cho Job\_detals      cur.execute(          "INSERT INTO job\_details(job\_id, description, requirements, benefits) VALUES (%s,%s,%s,%s)",          (job\_id, job['description'], job['requirements'], job['benefits'])      )      # Thêm dữ liệu cho bảng job\_skills      for s in job['skills']:          sid = skill\_ids.get(s)          if sid:              cur.execute(                  "INSERT IGNORE INTO job\_skills(job\_id, skill\_id) VALUES (%s, %s)",                  (job\_id, sid)              )  conn.commit()  cur.close()  conn.close()  print("✔ Đã lưu vào MySQL thành công!") |

Bước 7: cài đặt sqoop trên Windows:

**1. Cài đặt Sqoop**

1. **Cài Java 8+** (Java HOME phải trỏ đến thư mục JDK).

* **Đưa vào path**

1. **Cài Hadoop (Standalone)**
2. **Cài Sqoop**
   * Tải Sqoop binary tương ứng bản Hadoop của bạn (vd. sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz).

ở đường link <https://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz>

* + Giải nén vào thư mục (ví dụ C:\SQOOP\sqoop-1.4.7)
  + Set SQOOP\_HOME=C:\SQOOP\sqoop-1.4.7 và thêm %SQOOP\_HOME%\bin vào PATH.

1. **Connector MySQL**
   * Copy mysql-connector-java-\*.jar vào %SQOOP\_HOME%\lib\

Có thể downloads : <https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-j-9.3.0.zip>

Sau đó mở CMD, gõ sqoop version sẽ thấy phiên bản Sqoop nếu cài đúng.

Test – đưa mysql lên hadoop:

|  |
| --- |
| sqoop import-all-tables ^  --connect jdbc:mysql://localhost:3306/final\_bigdata ^  --username root ^  --password 123456789 ^  --warehouse-dir /win\_only ^  --as-textfile ^  --m 1 |

* Cài đặt docker:

Nếu trong quá trình cài xong mà hiện lên màn hình và câu lệnh:

|  |
| --- |
| Windows Subsystem for Linux must be updated to the latest version to proceed. You can update by running 'wsl.exe --update'.  For more information please visit https://aka.ms/wslinstall  Press any key to install Windows Subsystem for Linux.  Press CTRL-C or close this window to cancel.  This prompt will time out in 60 seconds. |

Thì bật comman prompt để cài đặt : $wsl --update

|  |
| --- |
|  |

Bước 1: Tạo thư mục để làm việc với Docker

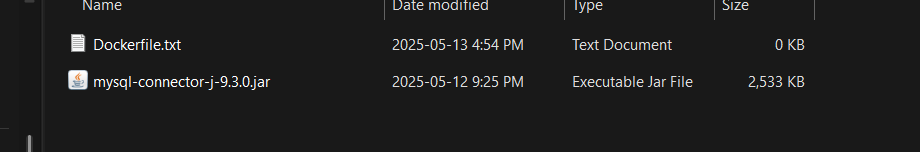
Mở CMD hoặc Explorer, tạo thư mục:

|  |
| --- |
| mkdir D:\ Docker\_Sqoop  cd D:\ Docker\_Sqoop |

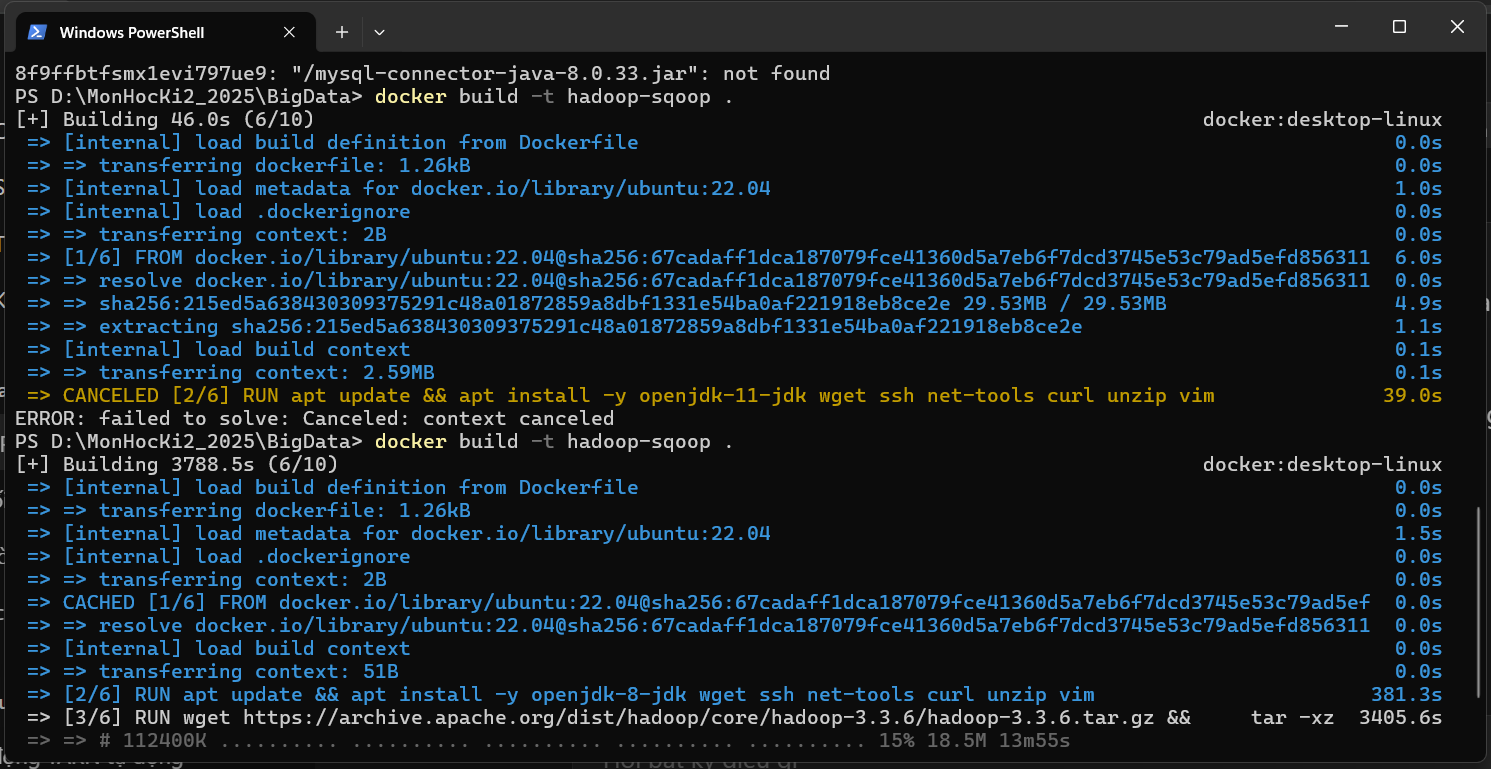
Bước 2: Tạo Dockerfile chứa Ubuntu + Java + Hadoop + Sqoop

|  |
| --- |
| FROM ubuntu:22.04  # bản unbuntu của mình  # Tắt các prompt tự động trong cài đặt Ubuntu  ARG DEBIAN\_FRONTEND=noninteractive  # Cài Java, SSH, wget, net-tools  RUN apt update && apt install -y openjdk-8-jdk wget ssh net-tools curl unzip vim  # Cài Hadoop  RUN wget https://archive.apache.org/dist/hadoop/core/hadoop-3.3.6/hadoop-3.3.6.tar.gz && \  tar -xzf hadoop-3.3.6.tar.gz -C /opt && \  mv /opt/hadoop-3.3.6 /opt/hadoop && \  rm hadoop-3.3.6.tar.gz  # Cài Sqoop  RUN wget https://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz && \  tar -xzf sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz -C /opt && \  mv /opt/sqoop-1.4.7.bin\_\_hadoop-2.6.0 /opt/sqoop && \  rm sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz  # Thiết lập biến môi trường  ENV JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64  ENV HADOOP\_HOME=/opt/hadoop  ENV SQOOP\_HOME=/opt/sqoop  ENV PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$SQOOP\_HOME/bin  # Copy file MySQL JDBC vào container nếu có  COPY mysql-connector-j-9.3.0.jar /opt/sqoop/lib/  # Tạo thư mục HDFS giả để tránh lỗi nếu bạn chạy Sqoop Local Mode  RUN mkdir -p /user/root  CMD ["/bin/bash"] |

Cài đặt và để trong thư mục!

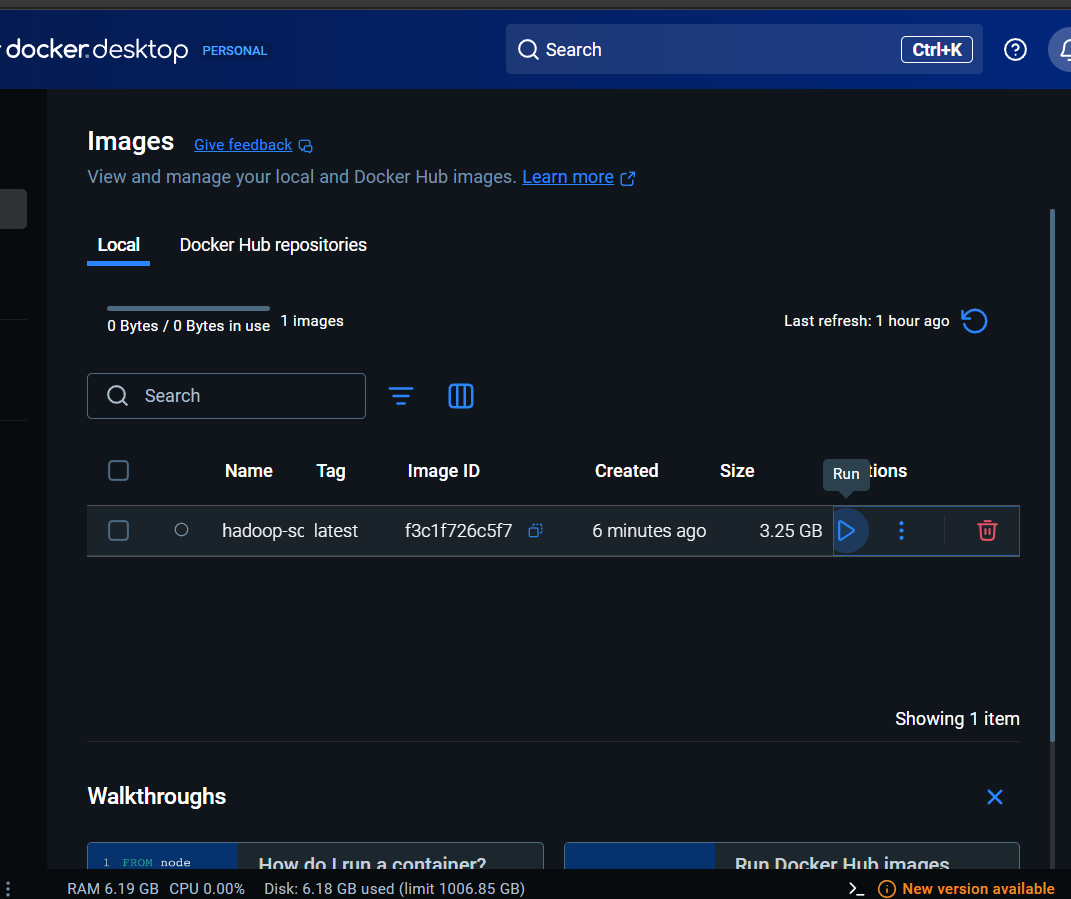


docker build -t hadoop-sqoop .

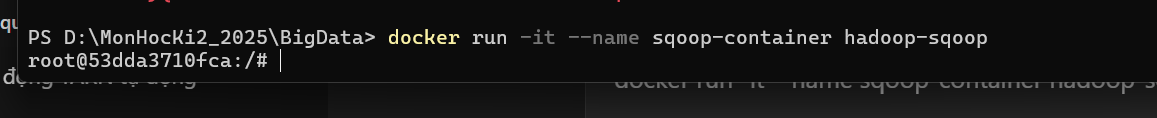


Chạy container từ image hadoop-sqoop:

Truy cập vào docker :



Khi chạy xong, bạn sẽ vào bên trong container với prompt Linux như:



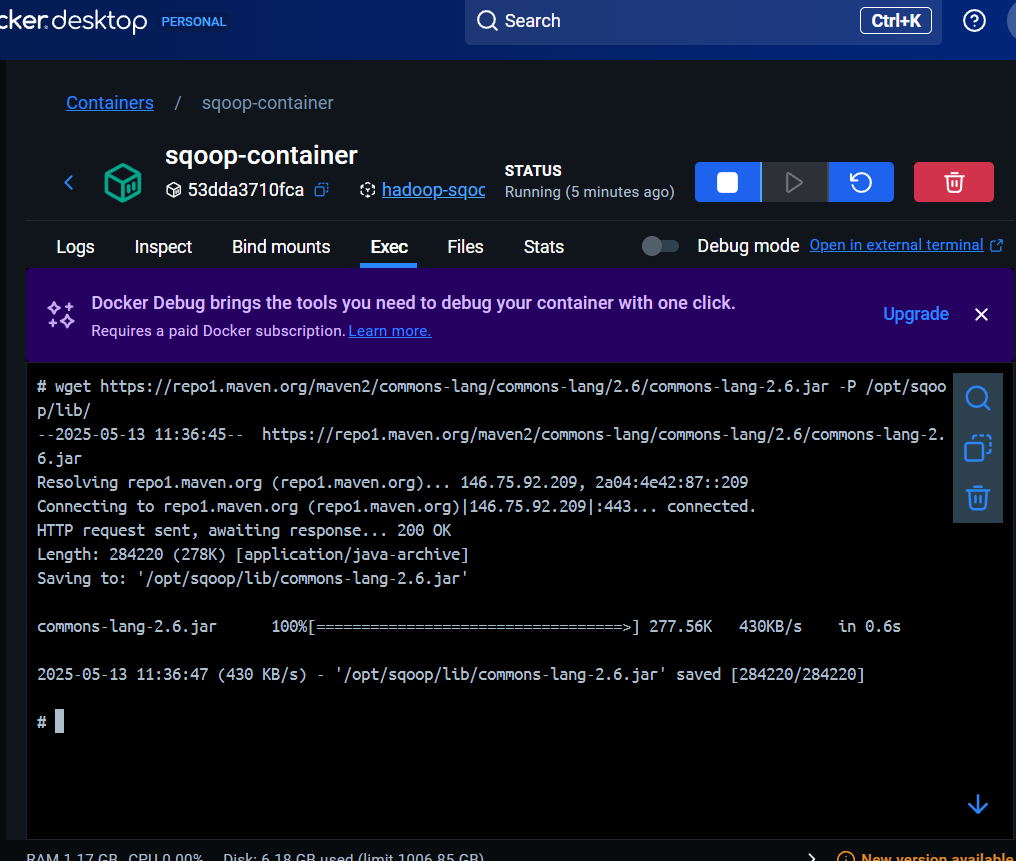
ERROR: JAVA\_HOME /usr/lib/jvm/java-11-openjdk-amd64 does not exist

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

ERRORR: java.lang.NoClassDefFoundError: org/apache/commons/lang/StringUtils

* Tải thư viện commons-lang-2.6.jar (phiên bản tương thích với Sqoop 1.4.7)

Bạn vào container hoặc Dockerfile, chạy:



Nếu hdfs đã tồn tại:

hdfs dfs -rm -r /test

chạy câu lệnh container:

docker run [OPTIONS] IMAGE[:TAG] [COMMAND] ==>

docker run -it hadoop-sqoop /bin/bash

một số lệnh container trong docker:

|  |  |
| --- | --- |
| **Mục đích** | **Lệnh** |
| Liệt kê container đang chạy | docker ps |
| Liệt kê tất cả container | docker ps -a |
| Dừng container | docker stop <container\_id> |
| Xóa container | docker rm <container\_id> |
| Xóa image | docker rmi <image\_id> |
| Vào terminal container đang chạy | docker exec -it <container\_id> /bin/bash |
| Xây dựng image từ Dockerfile | docker build -t hadoop-sqoop . |

* Kiểm tra tình trạng container:

docker ps -a

đưa dữ liệu từ sqoop lên linux:

Windows: chỉ start DataNode, bỏ qua NodeManager

|  |
| --- |
| cd C:\HADOOP\hadoop-3.3.6\sbin  hadoop-daemon.cmd start datanode |

Xử lý lỗi “fg: no job control”

trong file cấu hình Hadoop (mapred-site.xml) thêm property sau:  
  
để chạy từ docker\_container dùng câu lệnh

|  |
| --- |
| sqoop import \  --direct \  --connect jdbc:mysql://host.docker.internal:3306/final\_bigdata \  --username root --password 123456789 \  --table jobs \  --target-dir hdfs://dung-master:9000/test/skills \  --split-by id \  --num-mappers 1 \  --as-textfile |

Câu lệnh tạo bảng ở SQL server:

|  |
| --- |
| USE Big\_Data;  GO  -- 1. Bảng nhóm kỹ năng (skill\_groups)  CREATE TABLE dbo.skill\_groups (  id INT IDENTITY(1,1) PRIMARY KEY,  name NVARCHAR(100) NOT NULL UNIQUE  );  GO  -- 2. Bảng kỹ năng (skills)  CREATE TABLE dbo.skills (  id INT IDENTITY(1,1) PRIMARY KEY,  name NVARCHAR(100) NOT NULL UNIQUE  );  GO  -- 3. Bảng chi tiết gán kỹ năng vào nhóm (skill\_details)  CREATE TABLE dbo.skill\_details (  skill\_id INT NOT NULL,  group\_id INT NOT NULL,  CONSTRAINT PK\_skill\_details PRIMARY KEY CLUSTERED (skill\_id, group\_id),  CONSTRAINT FK\_skill\_details\_skill FOREIGN KEY (skill\_id)  REFERENCES dbo.skills(id)  ON DELETE CASCADE,  CONSTRAINT FK\_skill\_details\_group FOREIGN KEY (group\_id)  REFERENCES dbo.skill\_groups(id)  ON DELETE CASCADE  );  GO  -- 4. Bảng công việc chính (jobs)  CREATE TABLE dbo.jobs (  id INT IDENTITY(1,1) PRIMARY KEY,  group\_id INT NOT NULL, -- category của job  title NVARCHAR(255) NOT NULL,  link NVARCHAR(MAX),  location NVARCHAR(255),  experience NVARCHAR(100),  work\_location\_detail NVARCHAR(MAX),  working\_time NVARCHAR(255),  deadline DATE,  salary\_raw NVARCHAR(100),  salary\_normalized BIGINT,  currency\_unit NVARCHAR(10),  CONSTRAINT FK\_jobs\_group FOREIGN KEY (group\_id)  REFERENCES dbo.skill\_groups(id)  ON DELETE NO ACTION  );  GO  -- 5. Bảng mô tả chi tiết công việc (job\_details)  CREATE TABLE dbo.job\_details (  id INT IDENTITY(1,1) PRIMARY KEY,  job\_id INT NOT NULL,  description NVARCHAR(MAX),  requirements NVARCHAR(MAX),  benefits NVARCHAR(MAX),  CONSTRAINT FK\_job\_details\_job FOREIGN KEY (job\_id)  REFERENCES dbo.jobs(id)  ON DELETE CASCADE  );  GO  -- 6. Bảng gán kỹ năng vào công việc (job\_skills)  CREATE TABLE dbo.job\_skills (  job\_id INT NOT NULL,  skill\_id INT NOT NULL,  CONSTRAINT PK\_job\_skills PRIMARY KEY CLUSTERED (job\_id, skill\_id),  CONSTRAINT FK\_job\_skills\_job FOREIGN KEY (job\_id)  REFERENCES dbo.jobs(id)  ON DELETE CASCADE,  CONSTRAINT FK\_job\_skills\_skill FOREIGN KEY (skill\_id)  REFERENCES dbo.skills(id)  ON DELETE CASCADE  );  GO |

|  |
| --- |
| FROM ubuntu:22.04  bản unbuntu của mình  Tắt các prompt tự động trong cài đặt Ubuntu  ARG DEBIAN\_FRONTEND=noninteractive  Cài Java, SSH, wget, net-tools  RUN apt update && apt install -y openjdk-8-jdk wget ssh net-tools curl unzip vim  Cài Hadoop  RUN wget https://archive.apache.org/dist/hadoop/core/hadoop-3.3.6/hadoop-3.3.6.tar.gz &&  tar -xzf hadoop-3.3.6.tar.gz -C /opt &&  mv /opt/hadoop-3.3.6 /opt/hadoop &&  rm hadoop-3.3.6.tar.gz  Cài Sqoop  RUN wget https://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz &&  tar -xzf sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz -C /opt &&  mv /opt/sqoop-1.4.7.bin\_\_hadoop-2.6.0 /opt/sqoop &&  rm sqoop-1.4.7.bin\_\_hadoop-2.6.0.tar.gz  Thiết lập biến môi trường  ENV JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64 ENV HADOOP\_HOME=/opt/hadoop ENV SQOOP\_HOME=/opt/sqoop ENV PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$SQOOP\_HOME/bin  Copy file MySQL JDBC vào container nếu có  COPY mysql-connector-j-9.3.0.jar /opt/sqoop/lib/  Tạo thư mục HDFS giả để tránh lỗi nếu bạn chạy Sqoop Local Mode  RUN mkdir -p /user/root  CMD ["/bin/bash"] |