```
# Install Java
!apt-get update
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

# Download and extract Hadoop
!wget -q https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
!tar -xzf hadoop-3.4.0.tar.gz

# Set environment variables
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["HADOOP_HOME"] = "/content/hadoop-3.4.0"
os.environ["PATH"] += f":{os.environ['HADOOP_HOME']}/bin"

# Verify Java and Hadoop installation
!echo $JAVA_HOME
!hadoop version
```

Defining the Dataset The dataset involves lines containing either a name, email address, or both.

Example would be:

Tom Jones - tom.jones@gmail.com

Ethan Smith

Chandler Johnson

Brian Flaunders: bflaunders@yahoo.com

Samantha Lipson

The python mapper script I crafted will read each line separately, check if it contains an email address using a regex pattern, and output that.

```
%%writefile email_mapper.py
#!/usr/bin/env python3
import sys
import re

def email_mapper(line):
    pattern = re.compile(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}
    if re.search(pattern, line):
        return True
    return False

for line in sys.stdin:
    line = line.strip()
    if email_mapper(line):
        print(line)
```

```
!chmod +x email_mapper.py
```

```
!echo -e "Lebron James - LebronJamess@NBA.com\nLos Angeles Lakers\nEmpty Line
```

```
!echo -e "Lebron James - LebronJamess@NBA.com\nLos Angeles Lakers\nEmpty Lin
```

```
!hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.4.0.jar \
    -files email_mapper.py \
    -input input.txt \
    -output output \
    -mapper email_mapper.py \
    -reducer NONE
```

```
!cat output/*
```

```
Lebron James - LebronJamess@NBA.com
Lawyer: JaneJohnson@hotmail.com
```

Markdown Report

short description of the technology used:

Docker is a platform that allows developers to package applications into containers—standardized executable components combining application source code with the operating system libraries and dependencies required to run that code in any environment. Docker's container technology offers a lighter-weight form of virtualization, providing almost the same isolation as virtual machines but without the overhead of having to include a full OS in each application's files. Docker is efficient for resource use, rapid deployment, and consistent operation across systems. A con is that containers still share the host OS kernel, which can lead to security vulnerabilities if not managed correctly. The principle of "build once, run anywhere" can be applied using Docker, demonstrating modern deployment methodologies.

Redis is a memory key value data storer, used as a database, cache, and message broker. It can use strings, lists, maps, sets, and much more. Unlike most databases that store data on disk, Redis stores data in memory, which allows for much efficient and timely data retrieval. Redis is exceptionally fast and efficient, supports rich data types, and can be easily scaled in distributed environments using Redis Cluster. Unfortunatley though, its data size is limited by memory, and persistence configuration can compromise performance.

Google Colab is a cloud service that supports Python and Jupyter scripts for machine learning applications. Colab removes the necessity for complex hardware setups and software configurations by providing a fully prepared execution environment, which is particularly beneficial for students and researchers. It requires no previous setup, free access to hardware accelerators, and integration with Google Drive; but it does have imited session durations. Google Colab can be used for programming, data analysis, and machine learning easily.

Java is object oriented programming language designed to have as few implementation dependents as possible. Its enables applications to be written once and run anywhere again. Java achieves platform independence through the use of the Java Virtual Machine, which abstracts the application from hardware-specific details. Benefits of Java include platform independence, strong memory management, extensive standard libraries. But, compared to other languages, Java may require more memory and has a slower runtime.

Java's concept of write once, run anywhere is its most significant feature, illustrating all platform compatibility.

Hadoop is a open source framework that processes large data sets using simple programming algorithms or models. Hadoop is designed to scale up from a single server to several, efficiently. Hadoop has high scalability, is cost efficient, flexible data processing, fault tolerance. There is complexity in setup and management, and slow in small data operations. Hadoop and MapReduce are often discussed in courses on big data technologies, showcasing how large data sets can be managed and processed efficiently.

Sources:

docs.docker.com/

redis.io

research.google.com/colaboratory/faq.html

docs.oracle.com/en/java/

hadoop.apache.org/

svn.apache.org/repos/asf/hadoop/common/site/main/publish/index.html

...

describe and explain the logic of the Docker system that you have built:

Explain the decision you have made What are the containers involved How they communicate

The choice of openjdk:8-jdk as a base image is because of Hadoop's requirement for Java. This image provides a Java environment, which simplifies setting up. Installing tools like wget for downloading necessary files, vim for editing configurations, and ssh and pdsh for cluster management; emphasizes versaitlity, a manageable environment, when dealing with distributed systems like Hadoop. Hadoop is downloaded directly from its offical website. Copying custom configuration files into the Hadoop directory in the container is crucial for tailoring Hadoop's behavior to the needs of your project, such as setting up correct networking and storage options. Exposing ports like 9870 (NameNode web UI) and 8088 (ResourceManager web UI) is essential for accessing Hadoop's management interfaces from outside the container, facilitating monitoring and management. If all services, or Hadoop nodes, run within a single container, the communication happens internally, which simplifies the network complexity but deviates from production-like environments where each node would typically be in its own container. In a more scalable setup, each Hadoop component would be housed in separate containers. These containers would communicate over a Docker defined network, which isolates traffic and secures communication channels.

explain how to run the system by starting the container system:

Open a terminal and then run:

bash

docker build -t my-hadoop-system .

This command tells Docker to build a new image, -t my-hadoop-system: This tags the created image with the name "my-hadoop-system," making it easier to reference. .: This

Mazen Hamza
7:15 PM Today

FROM openjdk:8-jdk

RUN apt-get update && apt-get install -y wget vim ssh pdsh

RUN wget -q https://downloads.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz && \
tar -xzf hadoop-3.4.0.tar.gz -C /opt/ && \
rm hadoop-3.4.0.tar.gz
ENV HADOOP_HOME=/opt/hadoop-3.4.0
ENV PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

COPY config/* $HADOOP_HOME/etc/hadoop/

EXPOSE 9870 9864 8088 8042
RUN $HADOOP_HOME/bin/hdfs namenode -format
CMD ["start-all.sh"]

Mazen Hamza
7:28 PM Today

To start the Docker system:

bash

docker build -t hadoop-setup .
docker run -p 9870:9870 -p 8088:8088 -d hadoop-setup

Mazen Hamza
7:41 PM Today

Once the image is built, you can run it as a container using:

bash

docker run -d -p 9870:9870 -p 8088:8088 --name hadoop-instance my-hadoop-system specifies the build context as the current directory. Docker will look for the Dockerfile here. Then the dockerfile begins pulling the base image, running commands to install software, and copying files. When the process is completed without error, you will see a message saying so. "docker run" creates a container from the image, "-d" runs the container in the background. "-p 9870:9870 -p 8088:8088" reports the container's ports to the host, which is needed for accessing the NameNode and ResourceManager web UIs from the browser. "--name hadoop-instance": assigns a name to the container. "my-hadoop-system" is the name chosen of the image to run.

describe exactly what you have done:

Java Installation updates the package lists and installs the headless version of OpenJDK 8. Hadoop Setup downloads and extracts Hadoop 3.4.0 silently using wget with the quiet option -q. Then I set JAVA_HOME to the directory where Java is installed. Also set HADOOP_HOME to the directory where Hadoop is extracted. Appends the Hadoop binary directory to the PATH environment variable, enabling execution of Hadoop commands from the shell. Print the Java installation path to verify JAVA_HOME is set correctly. Checks the installed Hadoop version to ensure it's correctly installed and ready to work. The email_mapper script extracts lines containing email addresses from the input and prints them out. The regex pattern utilized identifies normal email formats within the text. The code reads from standard input (sys.stdin), which is typical in Hadoop streaming tasks where data is piped into the script. I use echo-e to test the code locally,it does this using sample text directly into the Python script to test email extraction functionality and regex efficiency.Then it configures and runs a Hadoop streaming job using email_mapper.py as the mapper. The outputs lines from the input that contain email addresses. The output after is: Lebron James - LebronJames@NBA.com
Lawyer: JaneJohnson@hotmail.com so I know the code works correctly and efficiently.

```
Start coding or generate with AI.
```