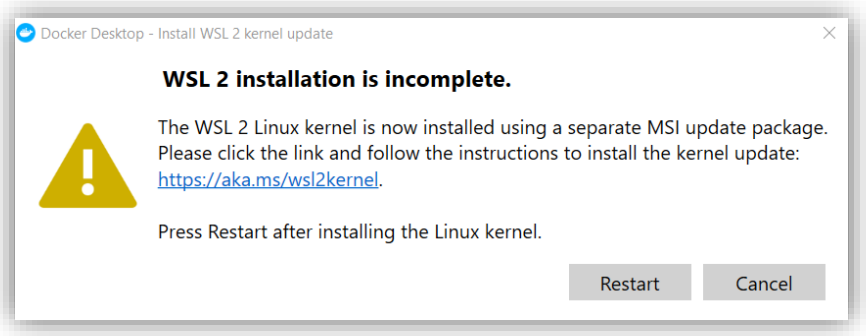
1) HDFS

a) Docker - download & install from this URL:

Docker Desktop for Mac and Windows | Docker

Windows: In case the following exception occurs, follow the link in the message and then download and install WSL2:



b) Clone Hadoop+Hive repository: GitHub - FreeUniDataEngineering/hadoop_hive

c) Commands **`docker-compose up`** or **`docker-compose up -d`** to build, create and launch containers

d) Hadoop components - short overview

https://aws.amazon.com/emr/details/hadoop/what-is-hadoop/

https://www.geeksforgeeks.org/hadoop-history-or-evolution/

e) Docker commands: **`docker ps, docker container list, docker container stats`**

f) **`docker exec -ti CONTAINER bash`**

`ls, pwd, cd, rm, cat, echo 'hello, world!' > hello.txt` commands

g) Copy files from container to local fs:

`docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH`

`docker cp 059af7d2f810:./etc/hosts C:/Users/Gigi/Desktop/hd/hosts`

Copy files from local fs to container:

`docker cp [OPTIONS] SRC_PATH CONTAINER:DEST_PATH`

`docker cp C:/Users/Gigi/Desktop/hd/hosts 059af7d2f810:./etc/hosts`

d) HDFS from command-line:

- **help** :)))
- **version**
- **env**
- dfs:
  - List all available commands - **help**
  - **hdfs dfs -ls /**
  - **hdfs dfs -mkdir /new_dir**
  - no **cd** here :)))
  - **hdfs dfs -put /hello.txt /new_dir/hello_again.txt**
  - **hdfs dfs -count -h -v /new_dir**
  - **hdfs dfs -mv /new_dir/hello_again.txt /new_dir/new_subdir/hello_again_moved.txt**
  - **hdfs dfs -rm -r /new_dir**
  - **head/tail: hdfs dfs -head /new_dir/hello.txt & hdfs dfs -tail /new_dir/hello.txt**

e) HDFS Python Client:

- run **pip install hdfs** inside the container named 'edge' (running on port 7777)
- **from hdfs import InsecureClient**
- **InsecureClient('http://namenode:50070', root='/')**
- **list(hdfs_path, status=False)**
- **walk(hdfs_path, depth=0, status=False)**
- **content(hdfs_path)**
- **read(hdfs_path, offset=0, encoding=None)**
- **makedirs(hdfs_path)**
- **rename(hdfs_src_path, hdfs_dst_path)**
- **delete(hdfs_path, recursive=False)**
- **write(hdfs_path, data=None, overwrite=False, append=False, blocksize=None, replication=None, encoding=None)**
- **download(hdfs_path, local_path, overwrite=False)**
- **upload(hdfs_path, local_path)**

More about HDFS Client: API reference — HdfsCLI 2.5.8 documentation

2) Google Cloud Storage:

- UI:
  - Create Buckets
  - Upload, Make Public, and Delete Objects in Your Bucket
  - Enable Version Control - display versions & restore
- Command Line: Gsutil:
  - Needs Google Cloud SDK installed: Installing Cloud SDK | Cloud SDK Documentation | Google Cloud
  - **gsutil ls**
  - **gsutil ls gs://buckety-bucket/**
  - **gsutil cat gs://buckety-bucket/DataLake/testv2.txt**
  - **gsutil cp D:\Workspace\test.txt gs://buckety-bucket/DataLake/test.txt**
  - **gsutil cp gs://buckety-bucket/DataLake/test.txt D:\Workspace\hello_from_GCP.txt**
  - **gsutil compose gs://buckety-bucket/DataLake/test1.txt gs://buckety-bucket/DataLake/test2.txt gs://buckety-bucket/DataLake/test3.txt**

3) Python Client:

- Python 3.6+ with Pip
- **pip install --upgrade google-cloud-storage**
- Set up env var GOOGLE_APPLICATION_CREDENTIALS. Value: location of Private Key
- **from google.cloud import storage**
- **storage.Client()**
- Bucket constructor: **storage_client.bucket(bucket_name)**
- **storage_client.create_bucket(bucket, location="us")**
- **list_buckets()**
- **get_bucket(bucket_name)**
- **list_blobs(bucket_name)**
- **list_blobs(bucket_name, prefix=prefix, delimiter=delimiter)**
- **bucket.blob(source_blob_name)**

- **blob.download_to_filename(destination_file_name)**
- **blob.upload_from_filename(source_file_name)**
- **bucket.labels**
- **bucket.patch()** = update
- **blob.delete()**
- **bucket.delete()**
- **blob_to_be_created.compose(list of source blobs [blob1, blob2…])**

Doc: Storage Client — google-cloud-storage documentation (googleapis.dev)