ALY 6110:
DATA MANAGEMENT AND BIG DATA

Assignment 1: Cloudera Installation Using Docker on Mac

Submitted To:
Prof. Andy Chen, Faculty Lecturer
Mr. James Kong, Teaching Assistant

Submitted By:
Abhilash Dikshit

Academic Term: Spring 2023
Graduate Students at Northeastern University, Vancouver, BC, Canada
Master of Professional Studies in Analytics

June 03, 2023

Installing Docker on macOS:

- Open a web browser and go to the Docker website (https://www.docker.com/) to download the Docker Desktop installer for macOS.
- Locate the Docker Desktop installer for macOS and click on the download link.
- Once the download is complete, double-click on the installer package to start the installation process.
- Follow the on-screen instructions to complete the installation. This may involve accepting license agreements, authorizing the installation, and entering your user password.
- After the installation is finished, Docker Desktop will be launched automatically.
- Docker should now be installed on your macOS system. You can verify this by opening a terminal and running the following command:
- docker --version
- If Docker is installed correctly, it will display the version information.

I used the command line to install docker on my MacBook with the OS version Ventura 13.3.1.

Terminal command to check Docker Path:

```
(base) abidikshit@abis-air ~ % which docker
/usr/local/bin/docker
(base) abidikshit@abis-air ~ %
```

Terminal command to check Docker Version:

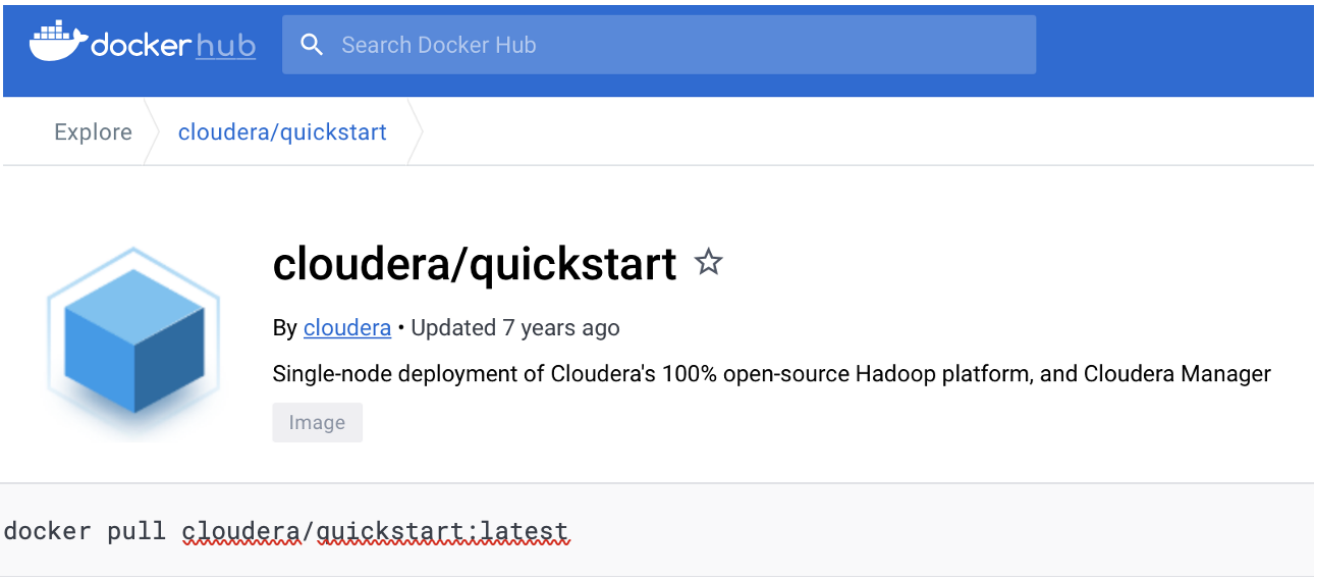
```
(base) abidikshit@abis-air ~ % docker version
Client:
 Cloud integration: v1.0.31
 Version:           23.0.5
 API version:       1.42
 Go version:        go1.19.8
 Git commit:        bc4487a
 Built:             Wed Apr 26 16:12:52 2023
 OS/Arch:           darwin/arm64
 Context:           desktop-linux

Server: Docker Desktop 4.19.0 (106363)
 Engine:
  Version:          23.0.5
  API version:      1.42 (minimum version 1.12)
  Go version:       go1.19.8
  Git commit:       94d3ad6
  Built:            Wed Apr 26 16:17:14 2023
  OS/Arch:          linux/arm64
  Experimental:     false
 containerd:
  Version:          1.6.20
  GitCommit:        2806fc1057397dbaeefbea0e4e17bddfbd388f38
 runc:
  Version:          1.1.5
  GitCommit:        v1.1.5-0-gf19387a
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
(base) abidikshit@abis-air ~ %
```

```
(base) abidikshit@abis-air ~ % docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
(base) abidikshit@abis-air ~ %
```

Importing the Cloudera QuickStart Image

You can import the Cloudera QuickStart image from Docker Hub:

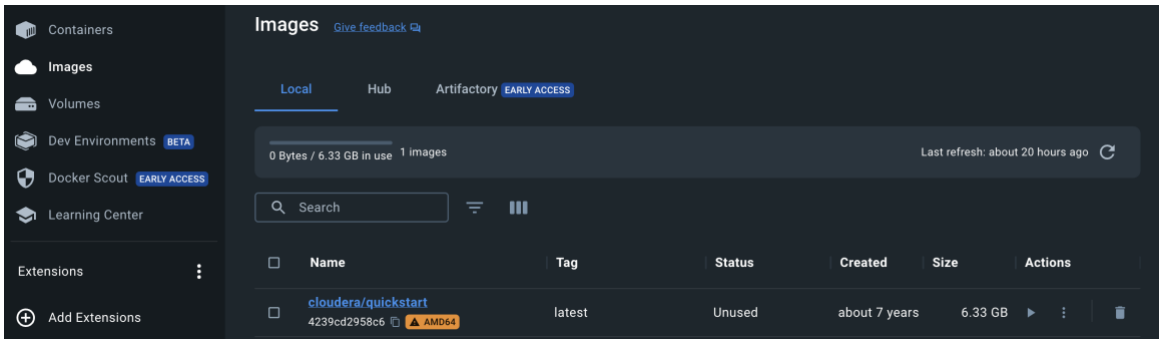
https://hub.docker.com/r/cloudera/quickstart/



Running Cloudera in a Docker Container:

- Open a terminal on your macOS system.
- Pull the Cloudera QuickStart Docker image by running the following command:

  docker pull cloudera/quickstart




```
(base) abidikshit@abis-air ~ % docker images
REPOSITORY            TAG       IMAGE ID       CREATED        SIZE
cloudera/quickstart   latest    4239cd2958c6   7 years ago    6.34GB
(base) abidikshit@abis-air ~ %
```

- Once the image is downloaded, you can start a Docker container running Cloudera by running the following command:

  docker run --hostname=quickstart.cloudera --privileged=true -t -i -v ~/Documents/dockersrc:/Src -p 8888:8888 -p 7180:7180 -p 80:80 cloudera/quickstart /usr/bin/docker-quickstart

  This command starts a new Docker container based on the Cloudera QuickStart image, assigns the hostname "quickstart.cloudera," enables privileged mode, and maps ports 8888, 7180, and 80 from the container to the corresponding ports on the host machine.

- Wait for the container to start. Once it is running, you will see the command prompt inside the container.

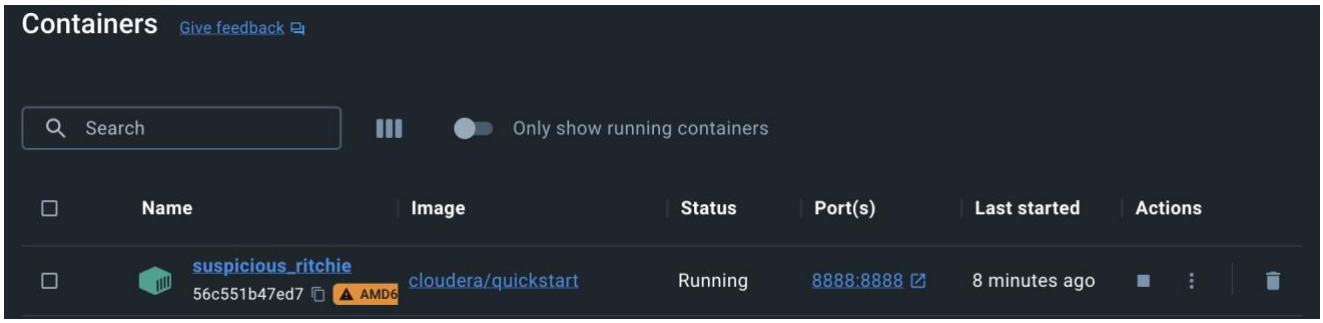
```
(base) abidikshit@abis-air ~ %
(base) abidikshit@abis-air ~ % docker run --hostname=quickstart.cloudera --privileged=true -t -i -v ~/Documents/dockersrc:/Src -p 8888:8888 cloudera/quickstart /usr/bin/docker-quickstart

WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested
Starting mysqld:                                           [  OK  ]

if [ "$1" == "start" ] ; then
    if [ "${EC2}" == 'true' ]; then
        FIRST_BOOT_FLAG=/var/lib/cloudera-quickstart/.ec2-key-installed
        if [ ! -f "${FIRST_BOOT_FLAG}" ]; then
            METADATA_API=http://169.254.169.254/latest/meta-data
            KEY_URL=${METADATA_API}/public-keys/0/openssh-key
            SSH_DIR=/home/cloudera/.ssh
            mkdir -p ${SSH_DIR}
            chown cloudera:cloudera ${SSH_DIR}
            curl ${KEY_URL} >> ${SSH_DIR}/authorized_keys
            touch ${FIRST_BOOT_FLAG}
        fi
    fi
    if [ "${DOCKER}" != 'true' ]; then
        if [ -f /sys/kernel/mm/redhat_transparent_hugepage/defrag ]; then
            echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
        fi

        cloudera-quickstart-ip
        HOSTNAME=quickstart.cloudera
        hostname ${HOSTNAME}
        sed -i -e "s/HOSTNAME=.*/HOSTNAME=${HOSTNAME}/" /etc/sysconfig/network
    fi

    (
        cd /var/lib/cloudera-quickstart/tutorial;
        nohup python -m SimpleHTTPServer 80 &
    )

    # TODO: check for expired CM license and update config.js accordingly
fi
+ '[' start == start ']'
+ '[' '' == true ']'
+ '[' true '!=' true ']'
+ cd /var/lib/cloudera-quickstart/tutorial
+ nohup python -m SimpleHTTPServer 80

nohup: appending output to `nohup.out'
JMX enabled by default
Using config: /etc/zookeeper/conf/zoo.cfg
Starting zookeeper ... STARTED
starting datanode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-datanode-quickstart.cloudera.out
Started Hadoop datanode (hadoop-hdfs-datanode):            [  OK  ]
```

```
Starting zookeeper ... STARTED
starting datanode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-datanode-quickstart.cloudera.out
Started Hadoop datanode (hadoop-hdfs-datanode):            [  OK  ]
starting journalnode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-journalnode-quickstart.cloudera.out
Started Hadoop journalnode:                                [  OK  ]
starting namenode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-namenode-quickstart.cloudera.out
Started Hadoop namenode:                                   [  OK  ]
starting secondarynamenode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-secondarynamenode-quickstart.cloudera.out
Started Hadoop secondarynamenode:                          [  OK  ]
```

```
Using CATALINA_PID:    /var/run/hadoop-httpfs/hadoop-httpfs-httpfs.pid
Started Hadoop httpfs (hadoop-httpfs):                     [  OK  ]
starting historyserver, logging to /var/log/hadoop-mapreduce/mapred-mapred-historyserver-quickstart.cloudera.out
Started Hadoop historyserver:                              [  OK  ]
starting nodemanager, logging to /var/log/hadoop-yarn/yarn-yarn-nodemanager-quickstart.cloudera.out
Started Hadoop nodemanager:                                [  OK  ]
starting resourcemanager, logging to /var/log/hadoop-yarn/yarn-yarn-resourcemanager-quickstart.cloudera.out
Started Hadoop resourcemanager:                            [  OK  ]
starting master, logging to /var/log/hbase/hbase-hbase-master-quickstart.cloudera.out
Started HBase master daemon (hbase-master):                [  OK  ]
starting rest, logging to /var/log/hbase/hbase-hbase-rest-quickstart.cloudera.out
Started HBase rest daemon (hbase-rest):                    [  OK  ]
starting thrift, logging to /var/log/hbase/hbase-hbase-thrift-quickstart.cloudera.out
Started HBase thrift daemon (hbase-thrift):                [  OK  ]
Starting Hive Metastore (hive-metastore):                  [  OK  ]
Started Hive Server2 (hive-server2):                       [  OK  ]
Starting Sqoop Server:                                     [  OK  ]
Sqoop home directory: /usr/lib/sqoop2
Setting SQOOP_HTTP_PORT:     12000
Setting SQOOP_ADMIN_PORT:     12001
Using   CATALINA_OPTS:       -Xmx1024m
Adding to CATALINA_OPTS:    -Dsqoop.http.port=12000 -Dsqoop.admin.port=12001
Using CATALINA_BASE:   /var/lib/sqoop2/tomcat-deployment
Using CATALINA_HOME:   /usr/lib/bigtop-tomcat
Using CATALINA_TMPDIR: /var/tmp/sqoop2
Using JRE_HOME:        /usr/java/jdk1.7.0_67-cloudera
Using CLASSPATH:       /usr/lib/bigtop-tomcat/bin/bootstrap.jar
Using CATALINA_PID:    /var/run/sqoop2/sqoop-server-sqoop2.pid
Starting Spark history-server (spark-history-server):      [  OK  ]
Starting Hadoop HBase regionserver daemon: starting regionserver, logging to /var/log/hbase/hbase-hbase-regionserver-quickstart.cloudera.out
hbase-regionserver.
Starting hue:                                              [FAILED]
Started Impala State Store Server (statestored):           [  OK  ]
```

Ignore the Failed HUE as it is installed and will show the workaround in later steps if HUE fails to open after installation.

```
Starting Solr server daemon:                               [  OK  ]
Using CATALINA_BASE:   /var/lib/solr/tomcat-deployment
Using CATALINA_HOME:   /usr/lib/solr/../bigtop-tomcat
Using CATALINA_TMPDIR: /var/lib/solr/
Using JRE_HOME:        /usr/java/jdk1.7.0_67-cloudera
Using CLASSPATH:       /usr/lib/solr/../bigtop-tomcat/bin/bootstrap.jar
Using CATALINA_PID:    /var/run/solr/solr.pid
Started Impala Catalog Server (catalogd) :                 [  OK  ]
Started Impala Server (impalad):                           [  OK  ]
```



Hadoop Version:

```
[root@quickstart /]# hadoop version
Hadoop 2.6.0-cdh5.7.0
Subversion http://github.com/cloudera/hadoop -r c00978c67b0d3fe9f3b896b5030741bd40bf541a
Compiled by jenkins on 2016-03-23T18:36Z
Compiled with protoc 2.5.0
From source with checksum b2eabfa328e763c88cb14168f9b372
This command was run using /usr/jars/hadoop-common-2.6.0-cdh5.7.0.jar
[root@quickstart /]#
```

Create a Test folder:

```
[root@quickstart /]# hadoop fs -ls
[root@quickstart /]# hadoop fs -mkdir Test
[root@quickstart /]# ls
bin  boot  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  packer-files  proc  root  sbin  selinux  Src  srv  sys  tmp  usr  var
[root@quickstart /]# hadoop fs -ls
Found 1 items
drwxr-xr-x   - root supergroup          0 2023-06-03 21:33 Test
[root@quickstart /]#
```

Open New Terminal Window now and run following cmd:

Do docker inspect followed by full container id or use the first 3 digit and it will show the same output.

```
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "b4166a27d396c4eb3e6ed54c96a76aa636795483f862129603c11a56190e9f7e",
    "HairpinMode":
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {
        "8888/tcp": [
            {
                "HostIp": "0.0.0.0",
                "HostPort": "8888"
            }
        ]
```

Command to Check IPv4:

```
(base) abidikshit@Abis-MacBook-Air ~ % docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 56c
172.
```

Command to Check Port:

```
(base) abidikshit@Abis-MacBook-Air ~ % docker inspect -f '{{range $key, $value := .NetworkSettings.Ports}}{{(index $value 0).HostPort}}{{end}}' 56c

8888
```

Cloudera Manager is not started by default. To see options for starting it, run

```
[root@quickstart /]# /home/cloudera/cloudera-manager --express
[QuickStart] Shutting down CDH services via init scripts...
kafka-server: unrecognized service


JMX enabled by default
Using config: /etc/zookeeper/conf/zoo.cfg
[QuickStart] Disabling CDH services on boot...
error reading information on service kafka-server: No such file or directory
[QuickStart] Starting Cloudera Manager server...
[QuickStart] Waiting for Cloudera Manager API...
[QuickStart] Starting Cloudera Manager agent...
[QuickStart] Configuring deployment...
Submitted jobs: 14
[QuickStart] Deploying client configuration...
Submitted jobs: 16
[QuickStart] Starting Cloudera Management Service...
Submitted jobs: 24
[QuickStart] Enabling Cloudera Manager daemons on boot...
--------------------------------------------------------------------------------

Success! You can now log into Cloudera Manager from the QuickStart VM's browser:

    http://quickstart.cloudera:7180

    Username: cloudera
    Password: cloudera

[root@quickstart /]#
```

- To access Hue, open a web browser on your macOS system and navigate to:

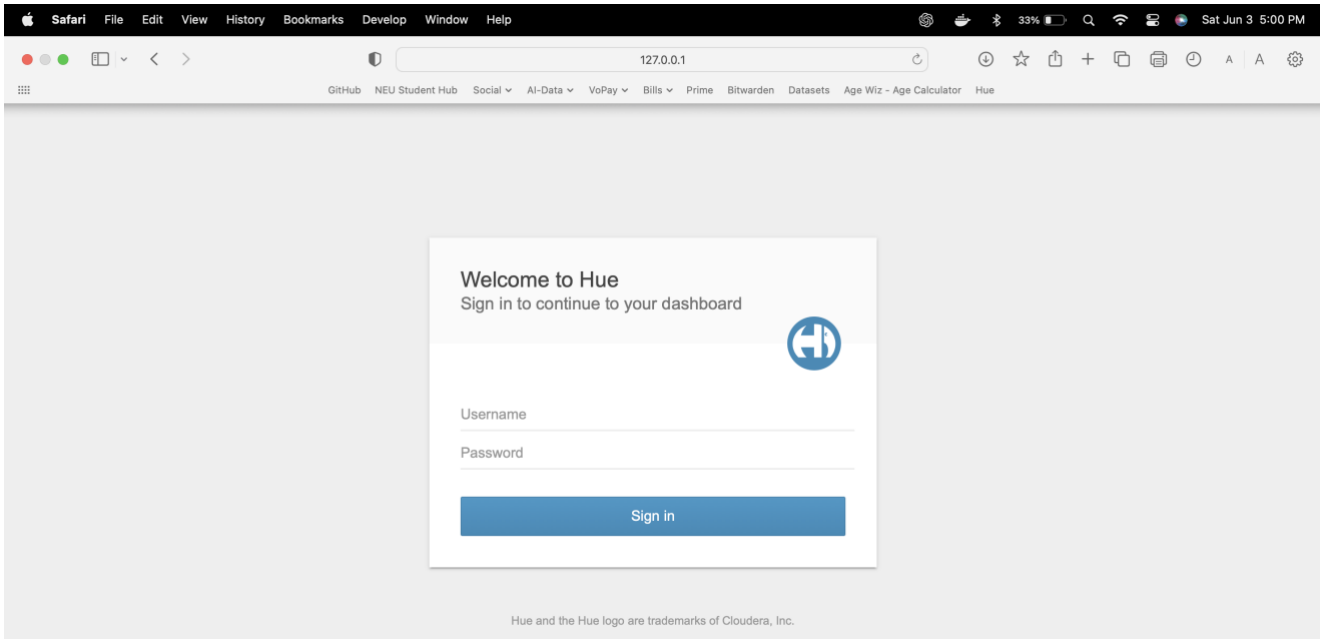
  http://localhost:8888

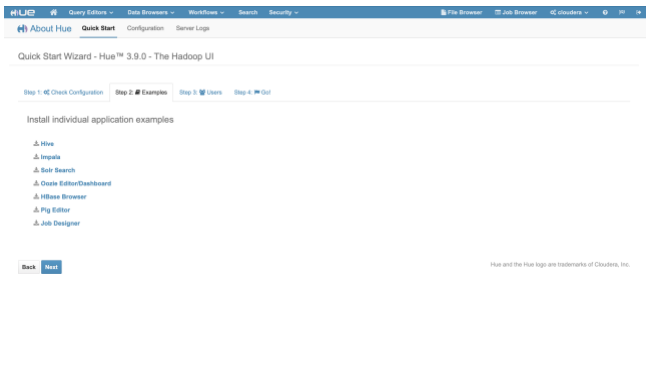
  This will open the Hue interface running inside the Docker container.

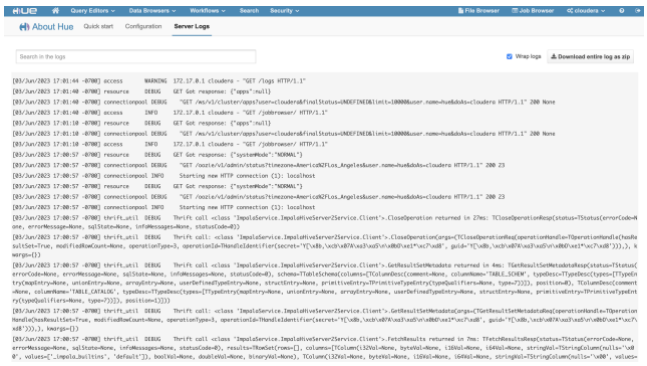
- To access the Cloudera Manager web interface, open a web browser on your macOS system and go to:

  http://localhost:7180

  This will allow you to manage your Cloudera environment using the Cloudera Manager interface.
- You can also access other Cloudera services exposed on different ports, such as Hadoop, Hive, and Impala, by referring to the Cloudera documentation or container's documentation for the specific ports and URLs.







Conclusion:

This guide provided step-by-step instructions for installing Docker on macOS and running Cloudera in a Docker container. By following these steps, you should now have Docker installed and be able to access Cloudera services through the respective URLs. Docker simplifies the deployment and management of Cloudera environments, allowing you to experiment with Big Data technologies without the need for complex installations.

References:

1. https://www.youtube.com/watch?v=b84vCNtASB4&list=TLGGYwL4l4Y8Ez4wMzA2MjAyMw&t=44s
2. https://www.youtube.com/watch?v=jILo-WnZUEY