Installing docker if not exists:

1. Check if docker already exists:

```
sudo docker --version
```

2. If not installed install it by following steps:

```
sudo apt-get update
```

```
echo \
  "deb [arch=$(dpkg --print-architecture)
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Pull image and run:

```
docker run -it adityasm1238/cclab:2.2
```

Once inside the container to start all the data nodes and name nodes run:

```
start-nodes
```

Wait until hadoop **safemode** is turned **off**
Status of safe mode can be checked using following command:

```
hdfs dfsadmin -safemode get
```

To run hive:

Once the safe mode is off, run the following command to open hive shell:

```
run-hive
```

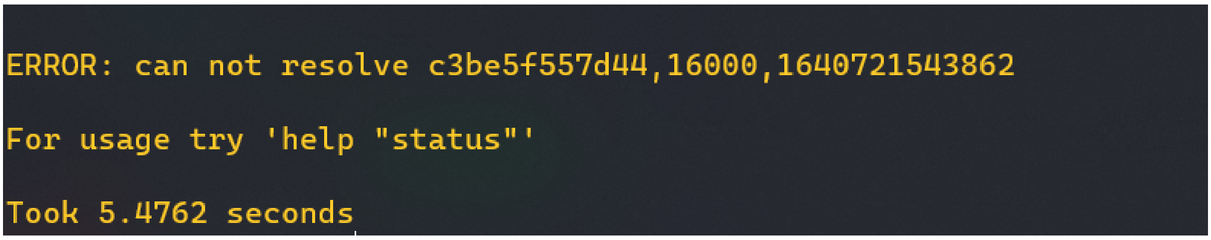
To run hbase:

Once the safe mode is off, run the following command to open hbase shell:

```
hbase shell
```

Then inside hbase shell try running :

```
status
```

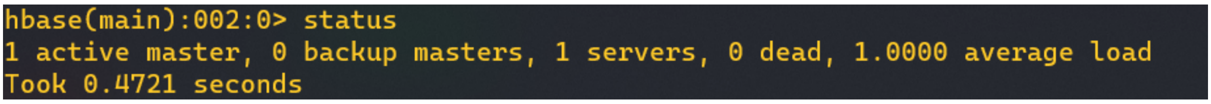


```
ERROR: can not resolve c3be5f557d44,16000,1640721543862

For usage try 'help "status"'

Took 5.4762 seconds
```

Wait until the above command stops giving error, this happens because the hbase nodes take approx 3 min to start.



```
hbase(main):002:0> status
1 active master, 0 backup masters, 1 servers, 0 dead, 1.0000 average load
Took 0.4721 seconds
```

It means hbase is up and running.

To run pig:

Once the safe mode is off, use ‘nano’ to edit and save the program along with data file, Run the program by using:

```
pig -x local program_name.pig
```

To run hadoop:

Write all programs, and data files using nano, Once the safe mode is off, put the data files into hdfs:

```
hdfs dfs -put /path/to/file/in/local /
```

To check if files are present in hdfs:

```
hdfs dfs -ls /
```

To export classpath run the following command to get classpath value

```
hadoop classpath
```

Copy paste the value in class path variable:

```
export CLASSPATH=<copied_value>
```

Compile the programs:

```
javac *.java -d .
```

Create **Manifest.txt** file using nano and make jar file:

```
jar cfm filename.jar Manifest.txt packagename/*
```

Once the jar is created, before running confirm there is no previous output folder in hdfs if present delete it using:

```
hdfs dfs -rm -r -f /output
```

To run jar:

```
hadoop jar filename.jar /datafile.csv /output
```

To check output:

```
hdfs dfs -cat /output/*
```