We recommend you use Linux or MacOS, rather than Windows, please use Hadoop 2.8.5 to match the Hadoop in docker.

Contents

Contents

1. How to run WordCount with HDFS using IDEA. (macOS as an example)

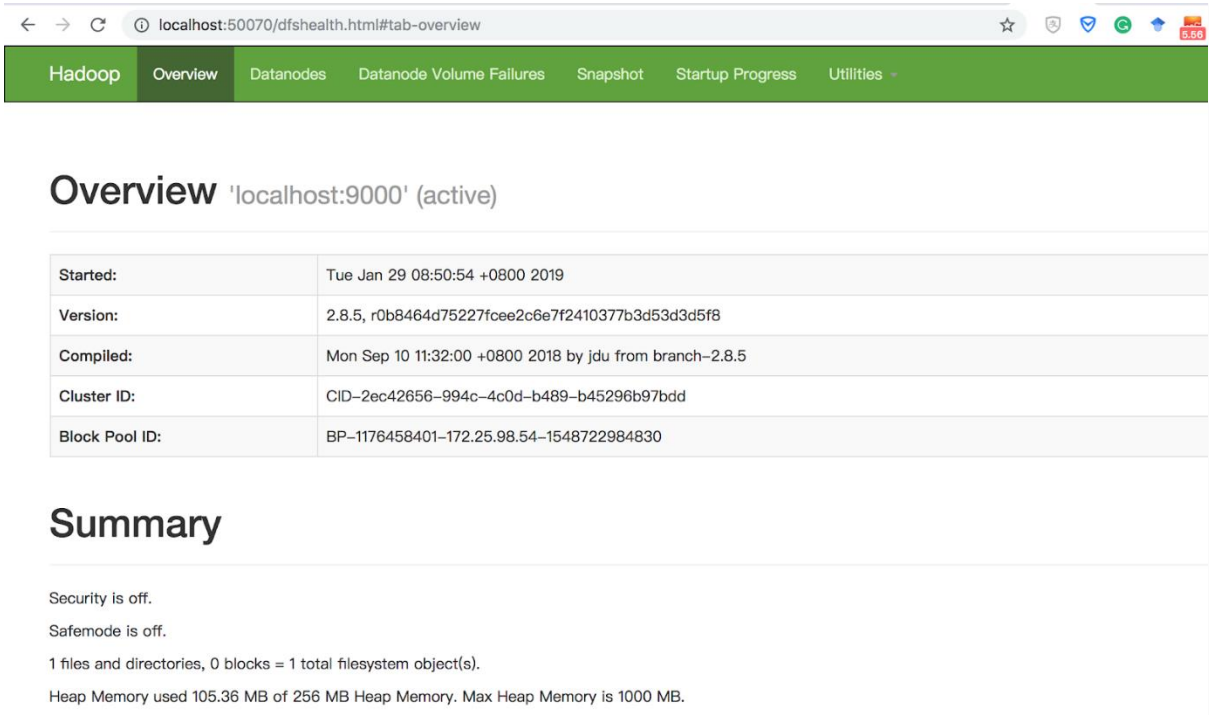
Once the Hadoop is up and running, you can check the web-ui:

Default for namenode: http://your namenode host name:50070/

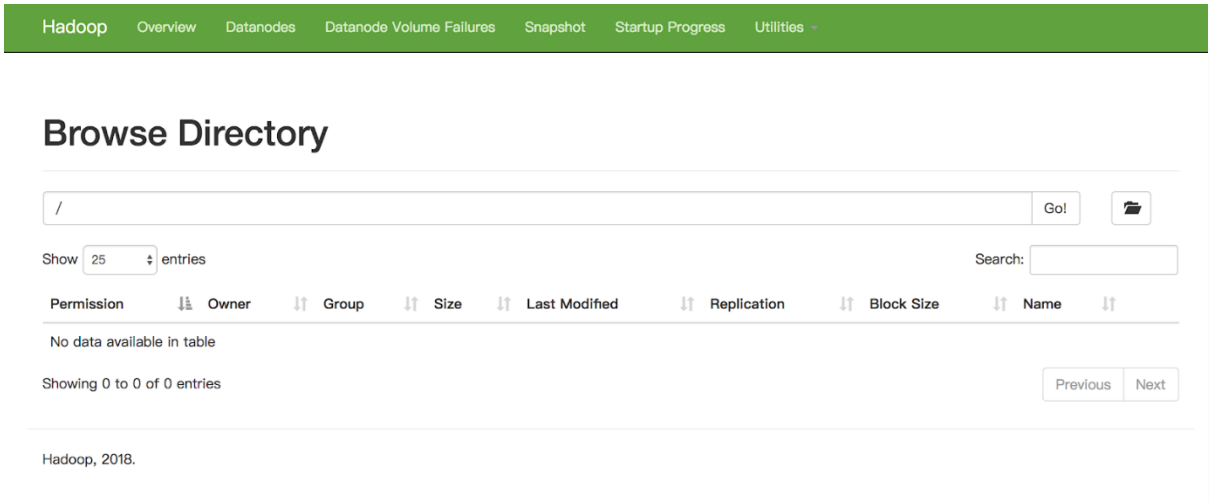
Default for ResourceManger: http://your_resourcemanger host name:8088/

For example, you run the namenode in your own machine, you can use http://localhost:50070/

You will see:



And you can click utilities to see the hdfs.

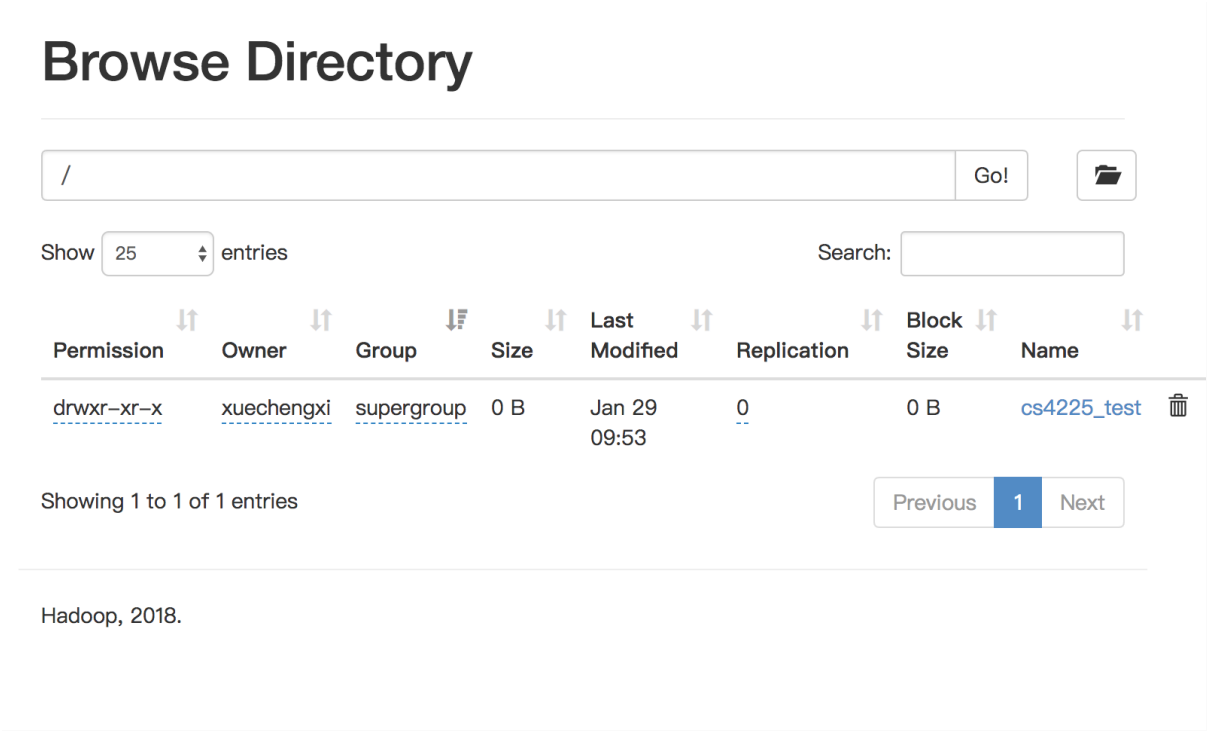


The path to access the HDFS depends on the parameter **<name>fs.defaultFS</name>** in core-site.xml. In this example, we use hdfs://localhost:9000/.

We can make a directory using **"hdfs dfs -mkdir "**

```
xuechengxi@r-54-98-25-172 [09:53:22] [/usr/local/hadoop-2.8.5/bin]
-> % ./hdfs dfs -mkdir /cs4225_test
19/01/29 09:53:34 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
platform... using builtin-java classes where applicable
xuechengxi@r-54-98-25-172 [09:53:35] [/usr/local/hadoop-2.8.5/bin]
-> % ./hdfs dfs -ls /
19/01/29 09:53:40 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
platform... using builtin-java classes where applicable
Found 1 items
drwxr-xr-x   - xuechengxi supergroup          0 2019-01-29 09:53 /cs4225_test
xuechengxi@r-54-98-25-172 [09:53:41] [/usr/local/hadoop-2.8.5/bin]
-> %
```

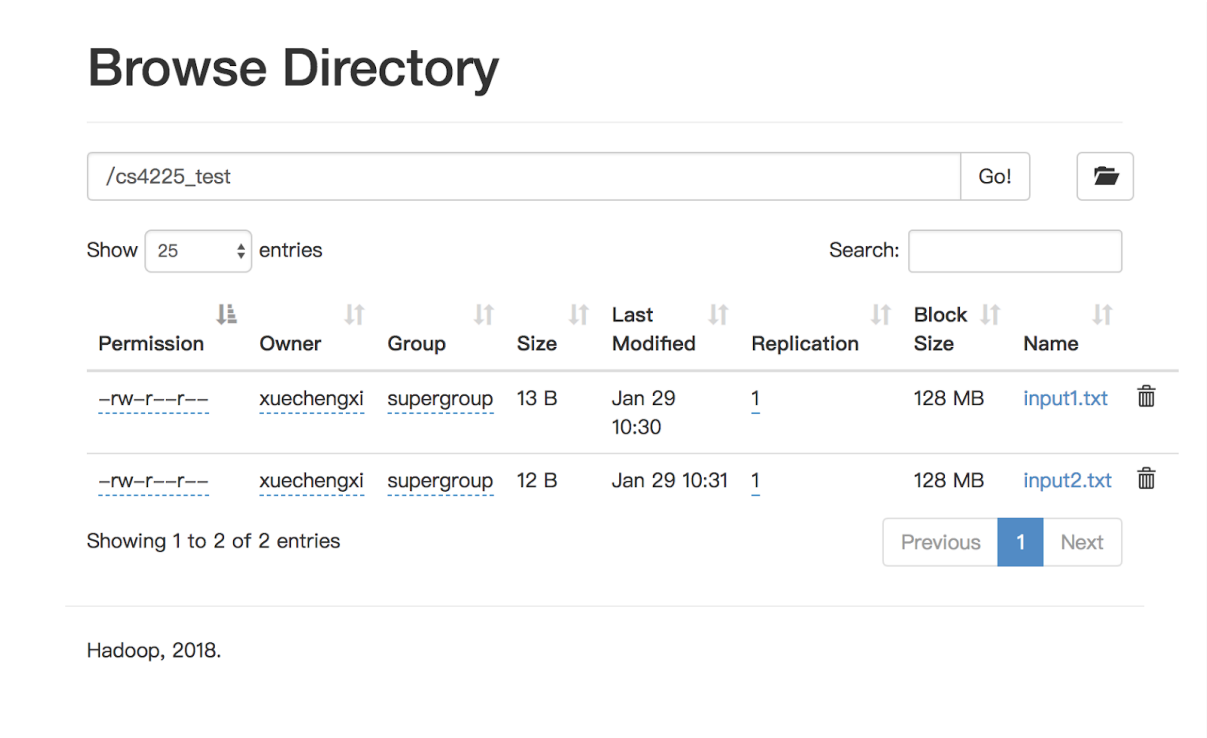
And you also can see the result using web-ui:



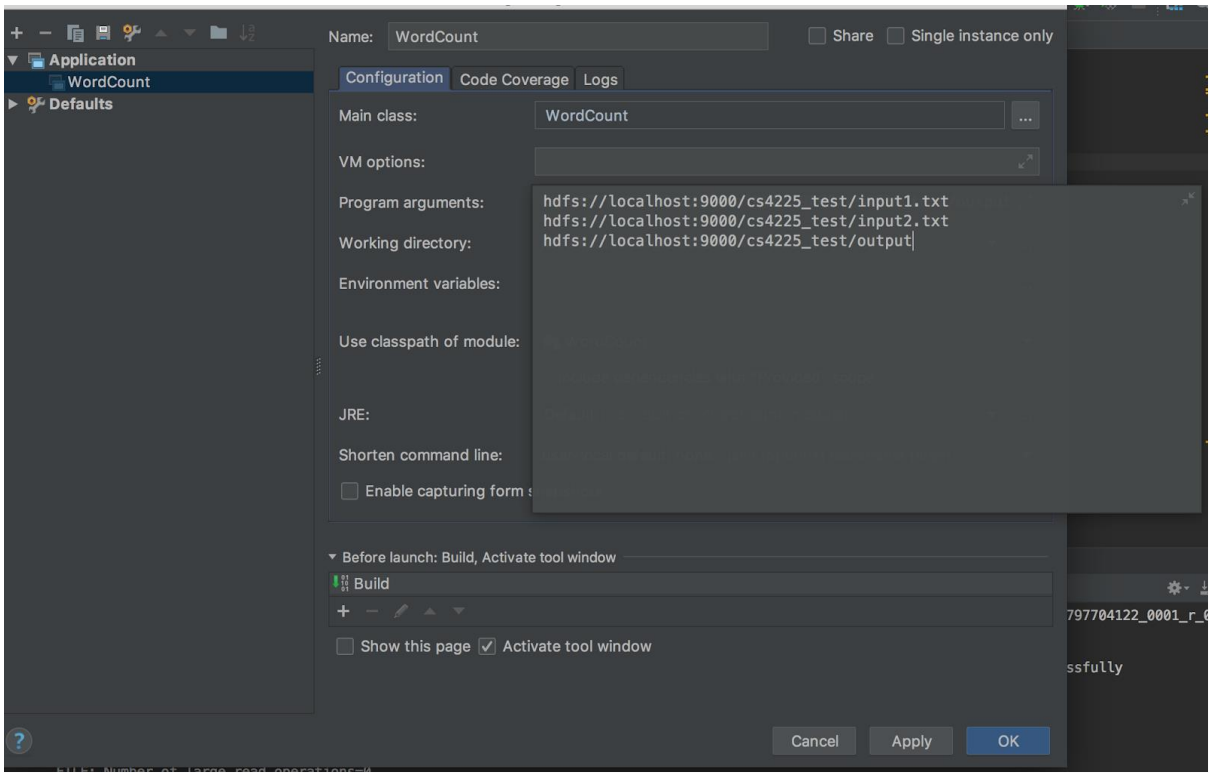
Then we can upload the files from local to HDFS.

"hdfs dfs -put sourcefile targetfile"

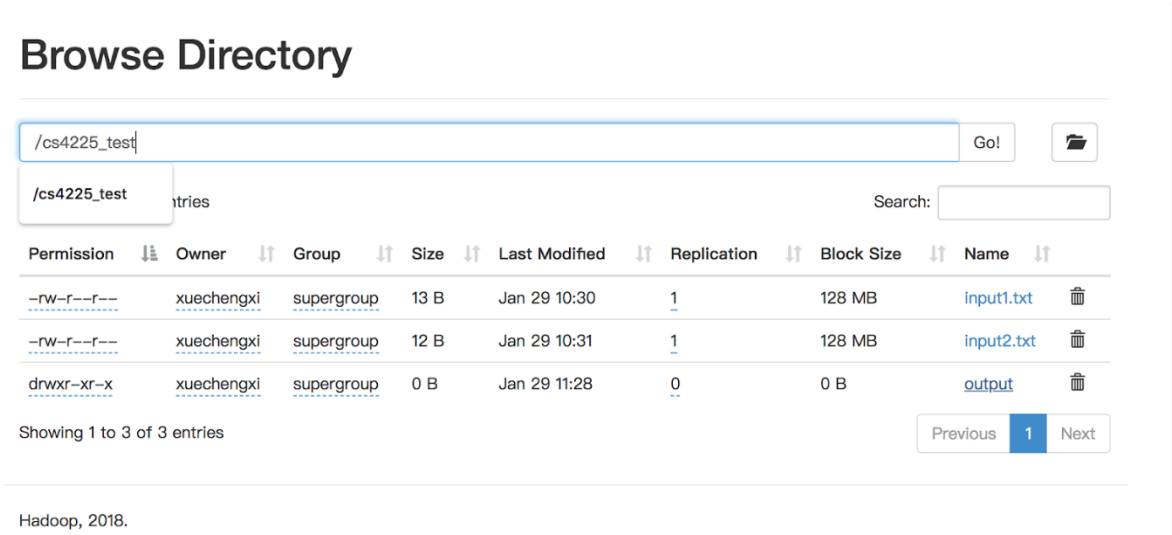
```
xuechengxi@r-54-98-25-172 [10:30:23] [/usr/local/hadoop-2.8.5/Big_data_test]
[-> % /usr/local/hadoop-2.8.5/bin/hdfs dfs -put ./input1.txt /cs4225_test
19/01/29 10:30:48 WARN util.NativeCodeLoader: Unable to load native-hadoop librar
platform... using builtin-java classes where applicable
xuechengxi@r-54-98-25-172 [10:30:50] [/usr/local/hadoop-2.8.5/Big_data_test]
[-> % /usr/local/hadoop-2.8.5/bin/hdfs dfs -put ./input2.txt /cs4225_test
19/01/29 10:31:10 WARN util.NativeCodeLoader: Unable to load native-hadoop librar
platform... using builtin-java classes where applicable
xuechengxi@r-54-98-25-172 [10:31:11] [/usr/local/hadoop-2.8.5/Big_data_test]
-> %
```

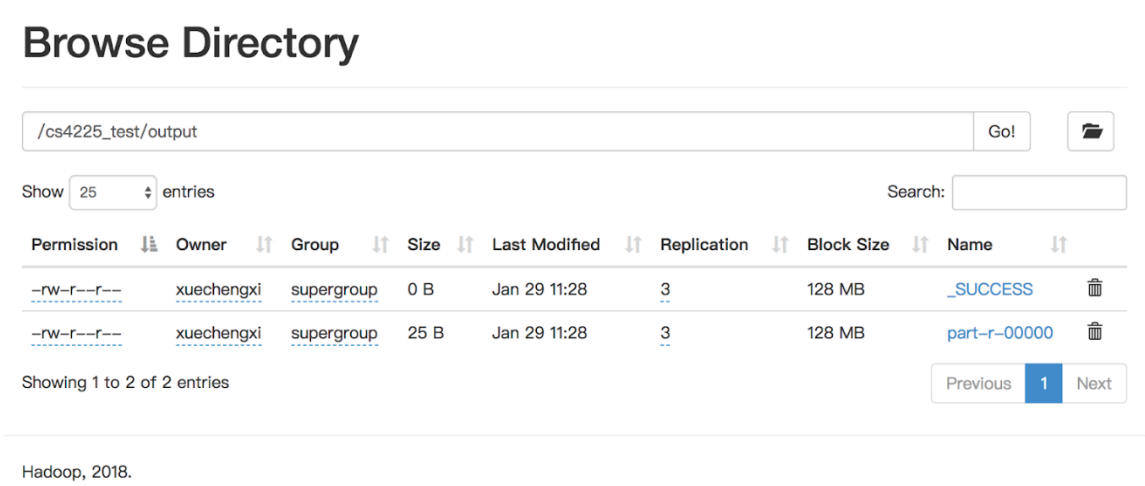


Then back to IDEA, you need to set the arguments.



Run the project.

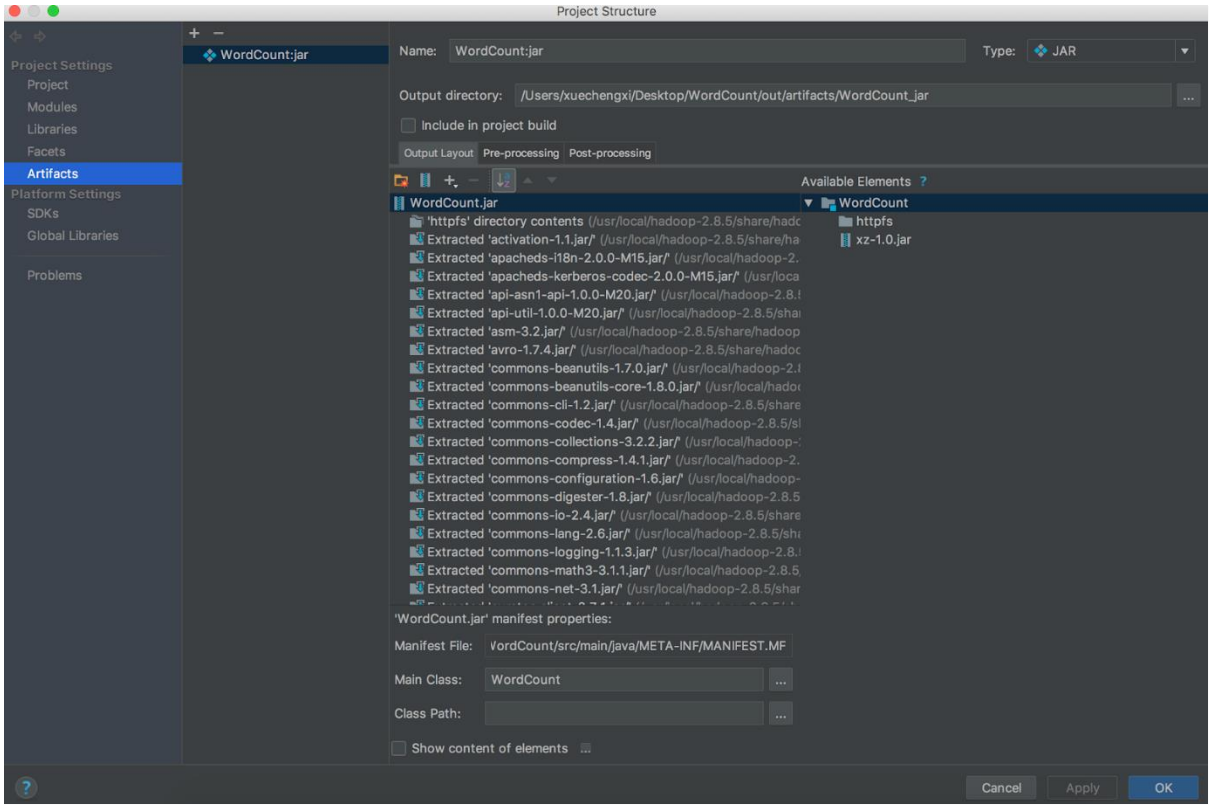

Browse Directory
/cs4225_test
Go!
/cs4225_test
ntries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
-rw-r--r--
xuechengxi
supergroup
13 B
Jan 29 10:30
1
128 MB
input1.txt
-rw-r--r--
xuechengxi
supergroup
12 B
Jan 29 10:31
1
128 MB
input2.txt
drwxr-xr-x
xuechengxi
supergroup
0 B
Jan 29 11:28
0
0 B
output
Showing 1 to 3 of 3 entries
Previous
1
Next
Hadoop, 2018.


Browse Directory
/cs4225_test/output
Go!
Show
25
entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
-rw-r--r--
xuechengxi
supergroup
0 B
Jan 29 11:28
3
128 MB
_SUCCESS
-rw-r--r--
xuechengxi
supergroup
25 B
Jan 29 11:28
3
128 MB
part-r-00000
Showing 1 to 2 of 2 entries
Previous
1
Next
Hadoop, 2018.

Done.

2. How to package your IDEA project and run it on the docker cluster.

1. Click File -> Project Structure->Artifacts, set the output directory and Main class, and click ok.



2. Click Build -> Build Project and Build Artifacts.

You will see jar file in your output directory.

3. Set up the docker cluster. (follow **the Installation&Configuration.docx**)

```
root@master: /usr/local/hadoop
er-slave02.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodeman
ager-master.out
root@master:/usr/local/hadoop# jps
2260 ResourceManager
2373 NodeManager
2101 SecondaryNameNode
1785 NameNode
1931 DataNode
2476 Jps
root@master:/usr/local/hadoop#
```

```
root@slave02: /
fordocker@fordocker-VirtualBox:~$ docker container exec -it slave02 /bin/bash
 * Starting OpenBSD Secure Shell server sshd                              [ OK ]
root@slave02:/# jps
48 Jps
root@slave02:/# vim /etc/hosts
root@slave02:/# jps
338 Jps
215 NodeManager
95 DataNode
root@slave02:/#
```

```
root@slave01: /
fordocker@fordocker-VirtualBox:~$ docker container exec -it slave01 /bin/bash
 * Starting OpenBSD Secure Shell server sshd                              [ OK ]
root@slave01:/# vim /etc/hosts
root@slave01:/# jps
85 DataNode
332 Jps
205 NodeManager
root@slave01:/#
```

4. Put your jar file into the docker. (e.g. put it into the master container)

"docker cp [OPTIONS] SRC_PATH|- CONTAINER:DEST_PATH"

5. Create the directory, upload files.

```
root@master:/usr/local/hadoop# ./bin/hdfs dfs -mkdir /cs4225_test
root@master:/usr/local/hadoop# ./bin/hdfs dfs -ls /cs4225_test
root@master:/usr/local/hadoop# ./bin/hdfs dfs -ls /
Found 3 items
drwxr-xr-x   - root supergroup          0 2019-01-29 05:14 /cs4225_test
drwxr-xr-x   - root supergroup          0 2019-01-25 12:50 /input
drwxr-xr-x   - root supergroup          0 2019-01-25 12:50 /output
root@master:/usr/local/hadoop#
```

```
root@master:/usr/local/hadoop# mkdir Bigdata_test
root@master:/usr/local/hadoop# cd Bigdata_test/
root@master:/usr/local/hadoop/Bigdata_test# vim input1.txt
root@master:/usr/local/hadoop/Bigdata_test# vim input2.txt
```

```
root@master:/usr/local/hadoop/Bigdata_test# /usr/local/hadoop/bin/hdfs dfs -put ./in
put1.txt /cs4225_test
root@master:/usr/local/hadoop/Bigdata_test# /usr/local/hadoop/bin/hdfs dfs -put ./in
put2.txt /cs4225_test
root@master:/usr/local/hadoop/Bigdata_test#
```

6. Run your jar file using Hadoop.

"hadoop jar [your jar file] [arguments]"

```
root@master:/usr/local/hadoop# ./bin/hadoop jar ./app/WordCount.jar WordCount hdfs:/
/master:9000/cs4225_test/input1.txt hdfs://master:9000/cs4225_test/input2.txt hdfs:/
/master:9000/cs4225_test/output
```

7. You will see the output.

```
root@master:/usr/local/hadoop# ./bin/hdfs dfs -ls /cs4225_test/
Found 3 items
-rw-r--r--   3 root supergroup         13 2019-01-29 05:19 /cs4225_test/input1.txt
-rw-r--r--   3 root supergroup         12 2019-01-29 05:19 /cs4225_test/input2.txt
drwxr-xr-x   - root supergroup          0 2019-01-29 05:28 /cs4225_test/output
root@master:/usr/local/hadoop#
```