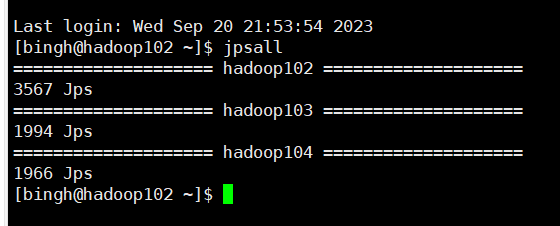
If you're facing a situation in XShell where it can't connect to the server (virtual machine), open the terminal in the VMware's Hadoop102 virtual machine and enter the following command line:

systemctl stop NetworkManager

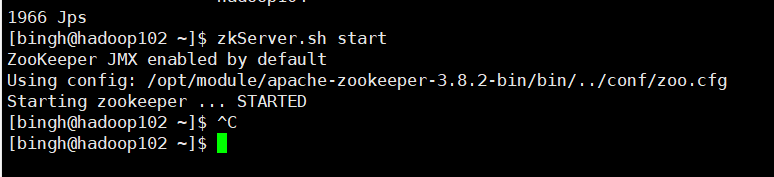
After successfully running the command,

systemctl restart network



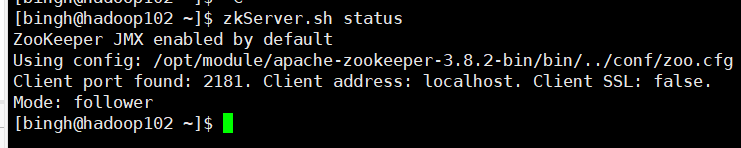
Check server running status:

jpsall can be used to see what is running inside the server, and it can be used each time a new program is started to check.



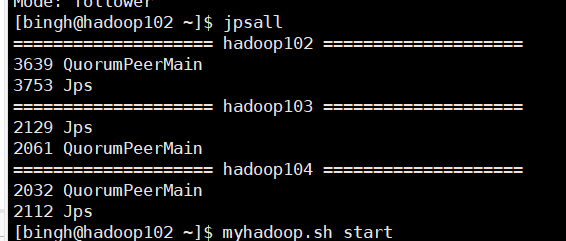
Open Zookeeper:

zkServer.sh start

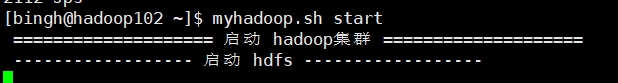


Check leader follower status:

zkServer.sh status



Check if Zookeeper is already open. If it is open, it should display 'QuorumPeerMain'.



Start the Hadoop cluster:

myhadoop.sh start



Start the HBase service:

start-hbase.sh

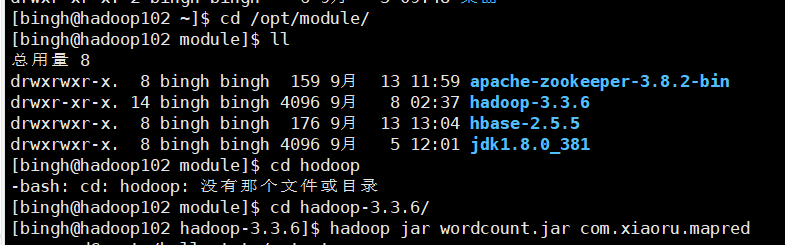
1. **Open the ResourceManager web page: http://hadoop103:8088/cluster**
2. **Open the HDFS web page:** [**http://hadoop102:9870/explorer.html#/**](http://hadoop102:9870/explorer.html#/)

***Complete the above actions before proceeding with the following steps.***

All the following operations need to be performed in the path below:



Use the following command to navigate there.



* Note: The pom.xml file must be consistent.



The command to run a MapReduce JAR file on Hadoop is:

* Every time when you want to upload the JAR file
  + Run 'clean' in the Maven tab once.
  + Only after cleaning, you can click 'package' (under lifecycle) to package it into a single JAR file.
  + Drag the JAR to the desktop.
  + In XShell, use the 'll' command to check for duplicate filenames. If there are any, use the following command to delete them.



* + Drag the JAR file into XShell.
  + Enter the command to run the JAR file as mentioned above.
  + For 'com.xiaoru.', right-click on the 'public class' in the 'driver' file, go to 'Copy/Paste Special', and select 'Copy Reference' to obtain it.
  + '/hello.txt' is the input file.
  + '/output1' is to create the output file.
* After the program execution, proceed to step 2. to check the running status and whether it has completed. You can also check the execution time (optimization suggestions can be considered).
* Once confirmed that it has completed, go back to step 1. to check the output file. In this example, it is the 'part...' file inside the 'output1' directory.

Conclusion: To implement the secondary index:

1) Write code in IDEA.

2) After completion, connect to the server via XShell and perform all the mentioned operations.

3) Note: The input file in the command to run the JAR file should be 'part-r-00000', and the output file can be customized.



SecondaryIndex.SecondaryIndexDriver