思考题：hadoop 完全分布式(3 台虚拟机）步骤：

一 部署多节点

二 启动 hadoop 完全分布式

三 wordcount 测试

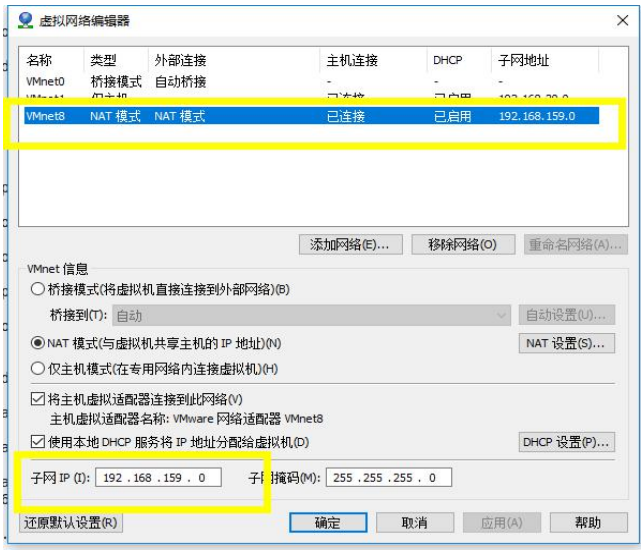
四 日志查看

以下详细说明：

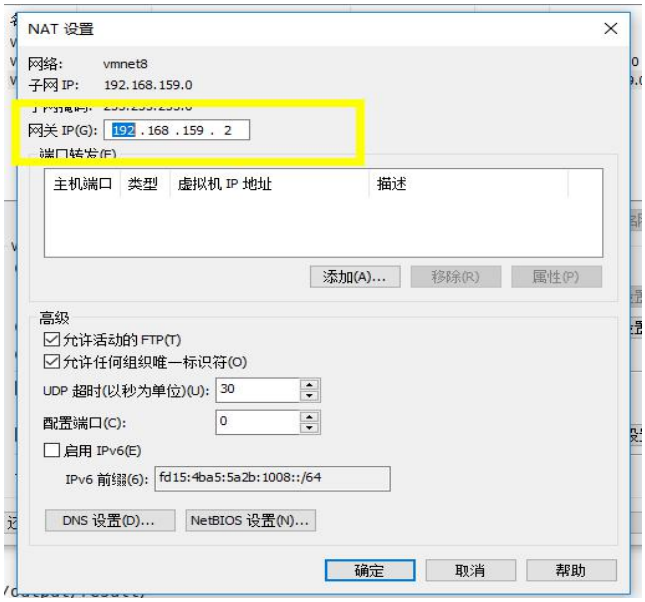
一、部署多节点

① NAT 静态 IP 配置：

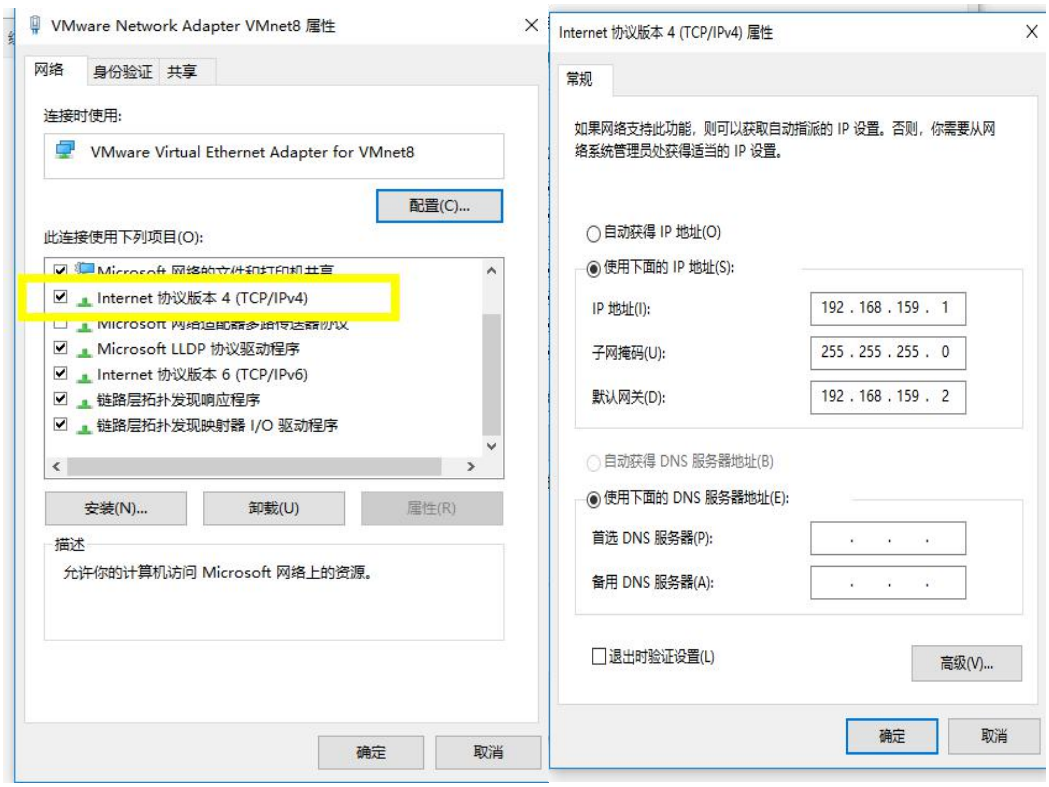
- 点击编辑-虚拟网络编辑器，点击 VMnet8，子网 IP 设置成如下：192.168.159.0



- 点击 Net 设置，网关 IP 设置成如下：192.168.159.2. 点击确定。



② 对主机的虚拟网卡进行设置：网络-属性-更改适配器设置，右键属性，选择 TCP/IPv4，配置如下：

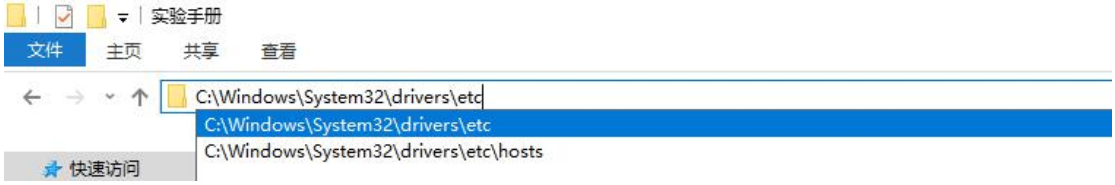


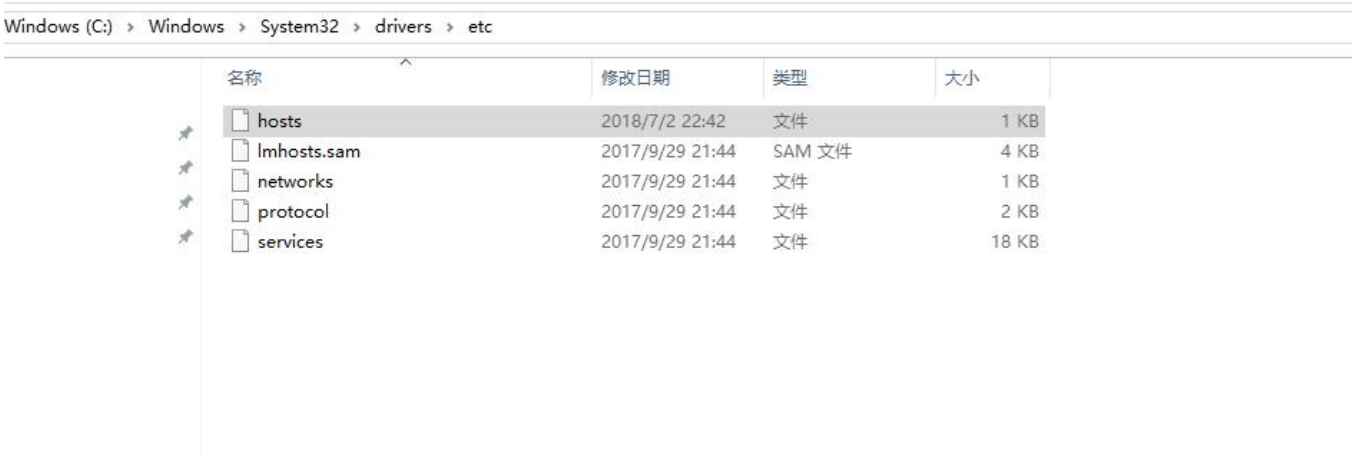
③ 修改主机 hosts 文件：在路径框中输入:C:\Windows\System32\drivers\etc，将 hosts 文件复制到桌面（因为权限原因无法直接修改），打开在其中添加：

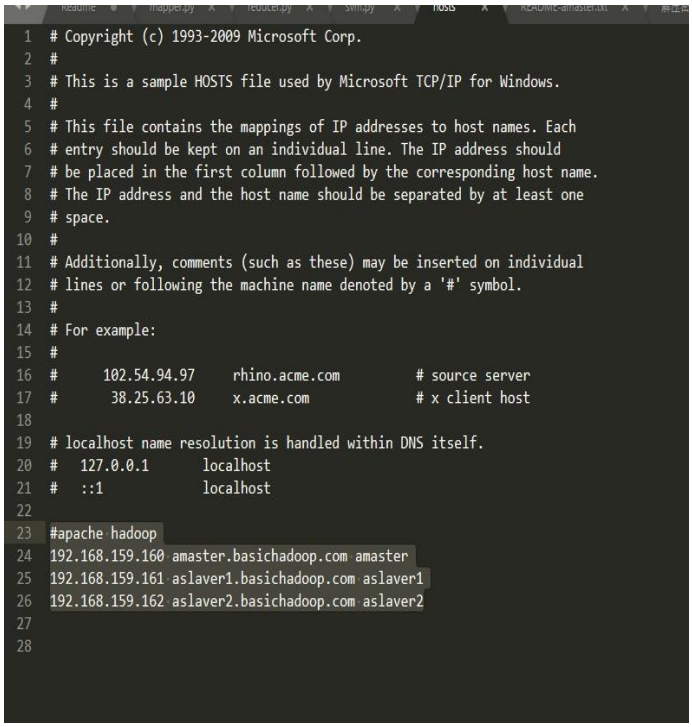
#apache hadoop

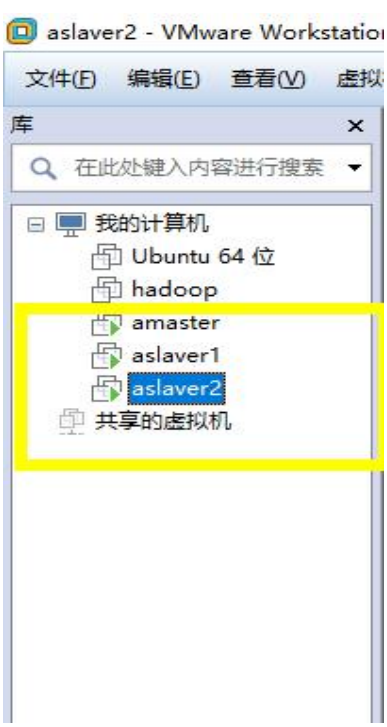
192.168.159.160 amaster.basichadoop.com amaster

192.168.159.161 aslaver1.basichadoop.com aslaver1
192.168.159.162 aslaver2.basichadoop.com aslaver2
保存，复制回原路径覆盖即可。









此时 IP 配置完成，开始启动 hadoop 三个节点。

二、启动 hadoop 完全分布式

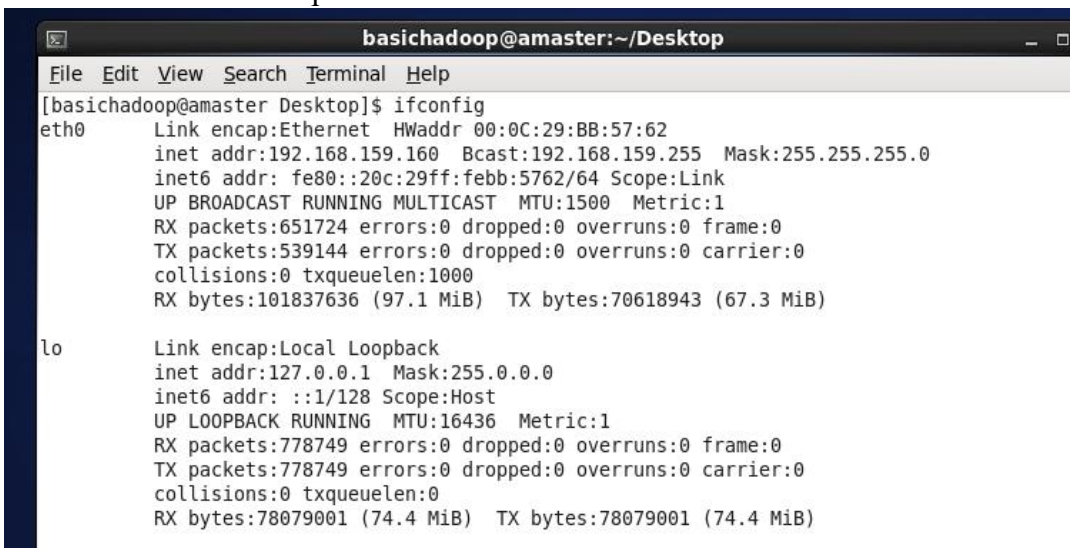
① 因为是三台虚拟机组成的完全分布式，我们第一步当然是测试相互间是否通信正常：

■ 查看三台虚拟机的 IP：$ ifconfig

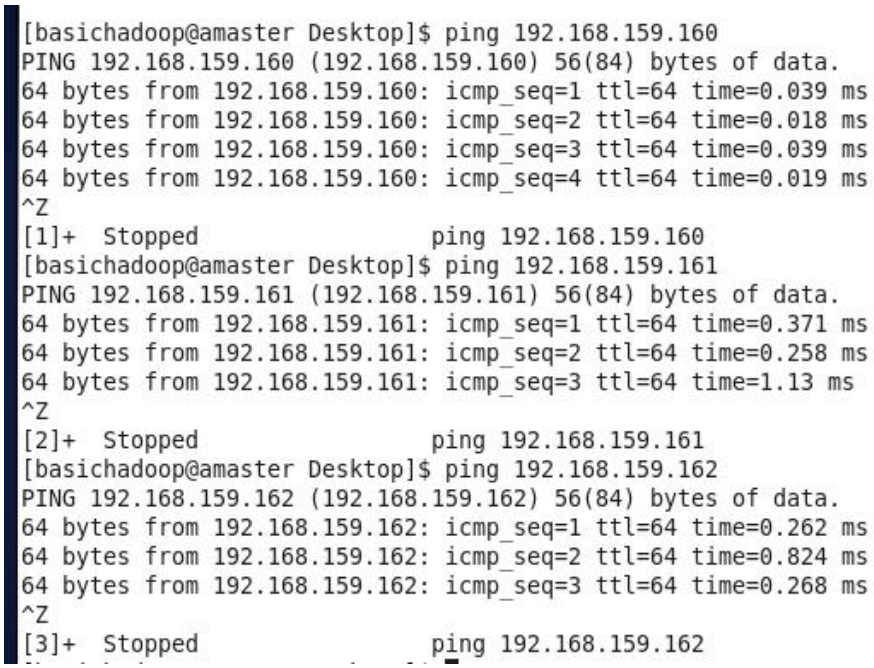
我们发现：amaster ip:192.168.159.160

aslaver1 ip:192.168.159.161

aslaver2 ip:192.168.159.162



■ 我们在主节点 amaster 中分别 ping 一下这三个 IP：皆可 ping 通，相互通信正常。



■ 继续在本地命令行中 ping 一下，查看通信是否正常：

ping amaster.basichadoop.com

ping aslaver1.basichadoop.com

ping aslaver2.basichadoop.com

```
C:\Users\gsz>ping amaster.basichadoop.com

正在 Ping amaster.basichadoop.com [192.168.159.160] 具有 32 字节的数据:
来自 192.168.159.160 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.159.160 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.159.160 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.159.160 的回复: 字节=32 时间<1ms TTL=64

192.168.159.160 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms

C:\Users\gsz>ping aslaver1.basichadoop.com

正在 Ping aslaver1.basichadoop.com [192.168.159.161] 具有 32 字节的数据:
来自 192.168.159.161 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.159.161 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.159.161 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.159.161 的回复: 字节=32 时间<1ms TTL=64

192.168.159.161 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Users\gsz>ping aslaver2.basichadoop.com

正在 Ping aslaver2.basichadoop.com [192.168.159.162] 具有 32 字节的数据:
来自 192.168.159.162 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.159.162 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.159.162 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.159.162 的回复: 字节=32 时间<1ms TTL=64

192.168.159.162 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms
```

通信一切正常后，我们开始启动节点。

② 第一步，先启动 zookeeper。

■ 三个节点都执行一遍：

```
$ cd /opt/modules/zookeeper-3.4.5
$ bin/zkServer.sh start
```

```
[basichadoop@amaster Desktop]$ cd /opt/modules/zookeeper-3.4.5
[basichadoop@amaster zookeeper-3.4.5]$
[basichadoop@amaster zookeeper-3.4.5]$
```

```
[basichadoop@aslaver1 Desktop]$ cd /opt/modules/zookeeper-3.4.5
[basichadoop@aslaver1 zookeeper-3.4.5]$
[basichadoop@aslaver1 zookeeper-3.4.5]$
[basichadoop@aslaver1 zookeeper-3.4.5]$ bin/zkServer.sh start
JMX enabled by default
Using config: /opt/modules/zookeeper-3.4.5/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
```

■ 三个节点都查看一下节点启动状态：

```
$ bin/zkServer.sh status
```

```
[basichadoop@aslaver1 zookeeper-3.4.5]$ bin/zkServer.sh status
JMX enabled by default
Using config: /opt/modules/zookeeper-3.4.5/bin/../conf/zoo.cfg
Mode: leader
```

zk 集群启动完毕.

③ 第二步，启动 hadoop 的 hdfs 与 yarn:

- **节点 amaster** 中，进入 hadoop 目录，启动 hdfs。注意每一个节点启动了不同任务。

```
$ cd /opt/modules/hadoop-2.6.5
$ sbin/start-dfs.sh
```

```
[basichadoop@amaster zookeeper-3.4.5]$ cd /opt/modules/hadoop-2.6.5
[basichadoop@amaster hadoop-2.6.5]$
[basichadoop@amaster hadoop-2.6.5]$
[basichadoop@amaster hadoop-2.6.5]$
[basichadoop@amaster hadoop-2.6.5]$ sbin/start-dfs.sh

Starting namenodes on [amaster aslaver1]
amaster: starting namenode, logging to /opt/modules/hadoop-2.6.5/logs/hadoop-bas
                                                    com.out
aslaver1: startin  namenode, logging t  /opt/modules/hadoop-2.6.5/logs/hadoop-ba
sichadoop namenod -aslaver1.basichadoc  .com.out
aslaver1: startin  datanode, logging t  /opt/modules/hadoop-2.6.5/logs/hadoop-ba
sichadoop datanod -aslaver1.basichadoc  .com.out
amaster:  tarting datanode, logging tc  /opt/modules/hadoop-2.6.5/logs/hadoop-bas
ichadoop- atanode amaster.basichadoop.com.out
aslaver2: startin  datanode, logging t  /opt/modules/hadoop-2.6.5/logs/hadoop-ba
sichadoop datanod -aslaver2.basichadoc  .com.out
Starting  ournal  odes [amaster aslave 1 aslaver2]
amaster:  tarting journalnode, logging to /opt/modules/hadoop-2.6.5/logs/hadoop-
basichado p-journ lnode-amaster.basich doop.com.out
aslaver2: startin  journalnode, loggi  to /opt/modules/hadoop-2.6.5/logs/hadoop
-basichad op-jour alnode-aslaver2.basi hadoop.com.out
aslaver1: startin  journalnode, loggi  to /opt/modules/hadoop-2.6.5/logs/hadoop
-basichad op-jour alnode-aslaver1.basi hadoop.com.out
Starting  K Failo er Controllers on NM hosts [amaster aslaver1]
amaster:  tarting zkfc, logging to /op /modules/hadoop-2.6.5/logs/hadoop-basicha
doop-zkfc amaster basichadoop.com.out
aslaver1: startin  zkfc, logging to /c t/modules/hadoop-2.6.5/logs/hadoop-basich
adoop-zkf -aslave 1.basichadoop.com.ou
```

- **节点 aslaver2 中，**进入 hadoop 目录，启动 MR 的 historyserver 进程和 yarn。

```
$ cd /opt/modules/hadoop-2.6.5
$ sbin/mr-jobhistory-daemon.sh start historyserver
$ sbin/start-yarn.sh
```

此时 hadoo 的历史日志进程和 yarn 资源节点管理都成功启动。

```
[basichadoop@aslaver2 zookeeper-3.4.5]$
[basicha oop@aslaver2 hadoop-2.6.5]$ sbin/mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /opt/modules/hadoop-2.6.5/logs/mapred-basichadoop-historyserver-a
slaver2.basichadoop.com.out
[basichado p@aslaver2 hadoop-2.6.5]$ sbin/start-yarn.sh
starting y rn daemons
starting r source anager, logging  o /opt/modules/hadoop-2.6.5/logs/yarn-basichadoop-resourcemanager-aslaver
2.basichad op.com out
aslaver1:  tartin  nodemanager, lo ging to /opt/modules/hadoop-2.6.5/logs/yarn-basichadoop-nodemanager-aslav
er1.basich doop.c m.out
amaster: s arting nodemanager, log ing to /opt/modules/hadoop-2.6.5/logs/yarn-basichadoop-nodemanager-amaste
r.basichad op.com out
aslaver2:  tartin  nodemanager, lo ging to /opt/modules/hadoop-2.6.5/logs/yarn-basichadoop-nodemanager-aslav
er2.basich doop.c m.out
```

④ **现在我们查看一下 jps，看一下各节点启动情况。**

- amaster 中: $ jps

```
[basichadoop@amaster hadoop-2.6.5]$ jps
3448 JournalNode
3160 NameNode
3917 Jps
3262 DataNode
3619 DFSZKFailoverController
2857 QuorumPeerMain
3768 NodeManager
```

■ **aslaver1** 中: $ jps

```
[basichadoop@aslaver1 zookeeper-3.4.5]$ jps
2656 QuorumPeerMain
2748 NameNode
2984 DFSZKFailoverController
2812 DataNode
2891 JournalNode
3185 NodeManager
3349 Jps
```

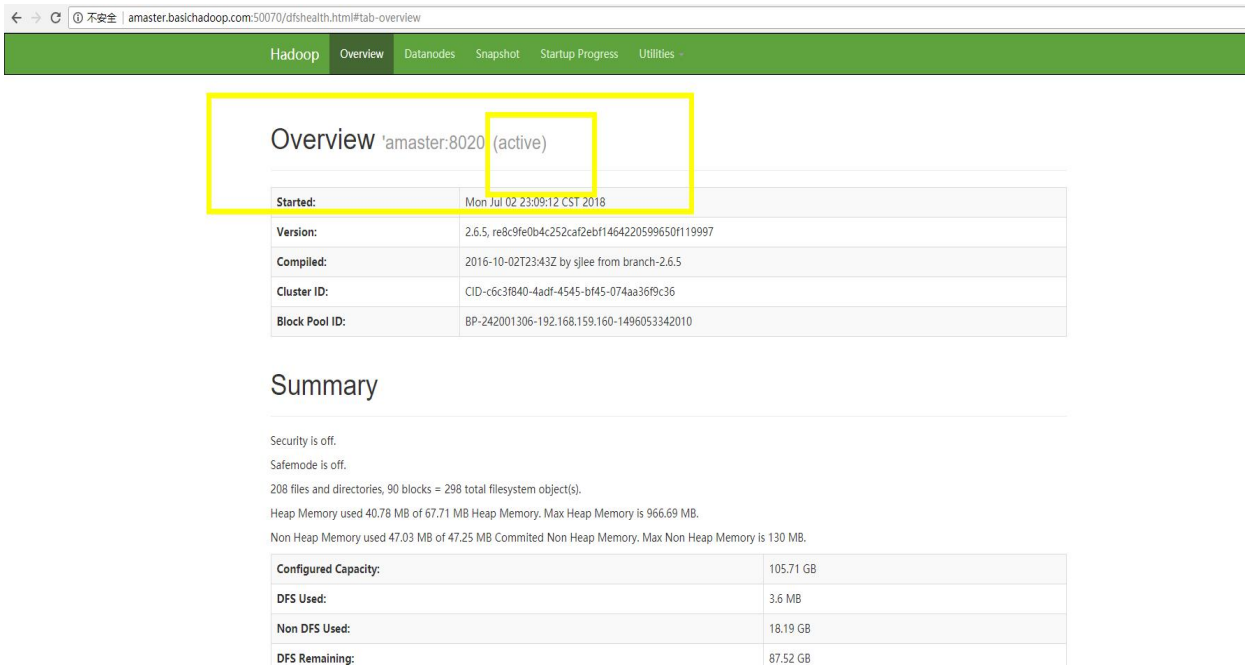
■ **aslaver2** 中: $ jps

```
[basichadoop@aslaver2 hadoop-2.6.5]$ jps
2624 QuorumPeerMain
2792 JournalNode
3133 NodeManager
2922 JobHistoryServer
3471 Jps
3030 ResourceManager
2722 DataNode
```

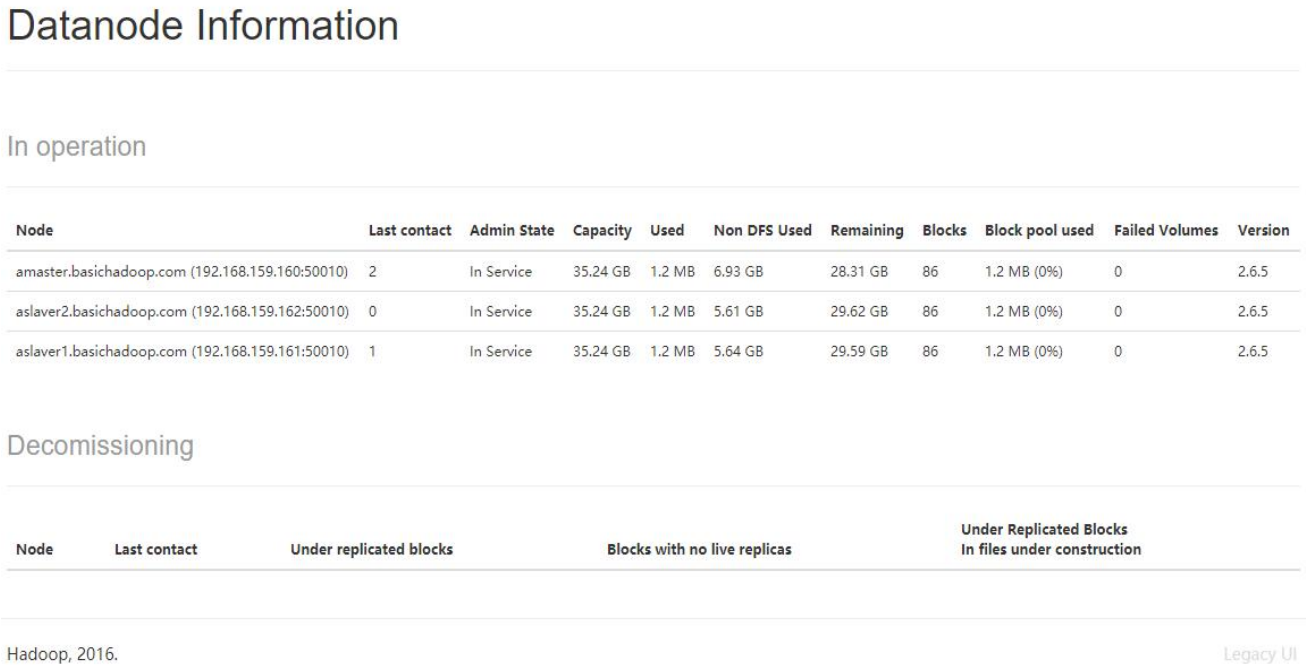
我们对照一下应该启动起来的说明，发现全部启动完成。

	192.168.159.160	192.168.159.161	192.168.159.162	
	amaster	**aslaver1**	**aslaver2**	
zookeeper	QuorumPeerMain	QuorumPeerMain	QuorumPeerMain	
hdfs	namenode	namenode		zk选举
hdfs	datanode	datanode	datanode	
hdfs		secondnamenode		
hdfs-HA-Log	JournalNode	JournalNode	JournalNode	
hdfs-HA-Controller	ZKFC	ZKFC		
mapreduce-Log			JobHistoryServer	
yarn			ResourceManager	
yarn	NodeManager	NodeManager	NodeManager	
hbase-master	master	backup		zk选举
hbase-regionserver	regionserver	regionserver	regionserver	
hive	metastore-server			
hive	hiveserver2-server			

⑤ 浏览器查看，打开主机（外部）浏览器：输入：http://amaster.basichadoop.com:50070，可以查看 hadoop 集群信息。

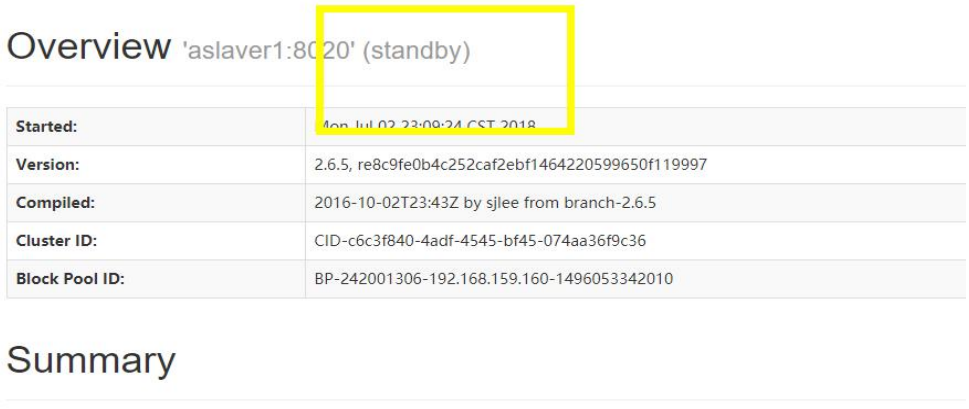


点击 Live Nodes:三个节点信息都在此：

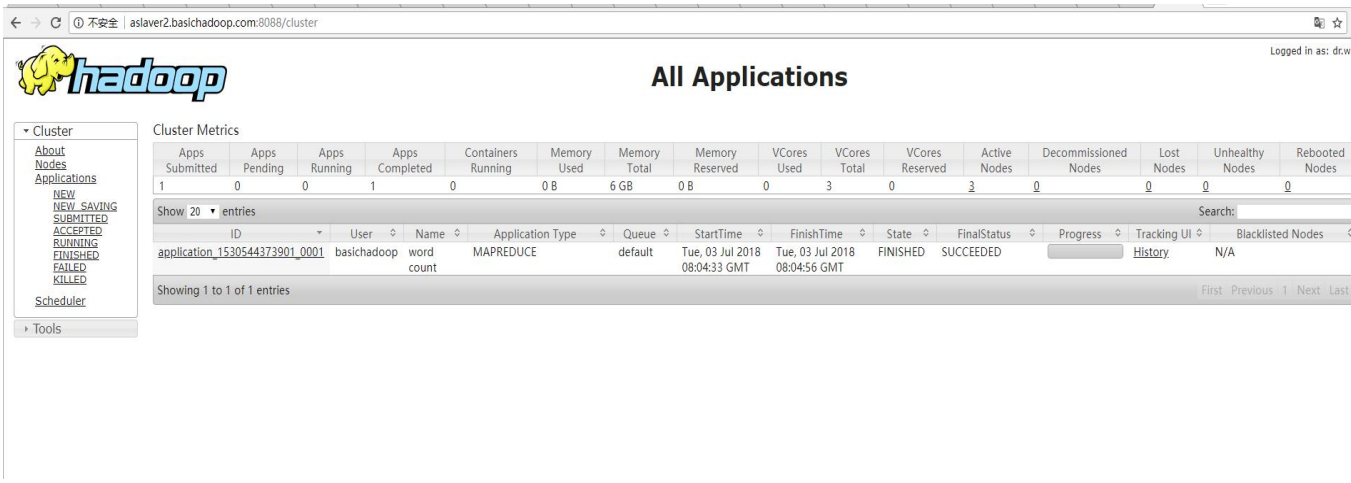


- 输入：http://aslaver1.basichadoop.com:50070

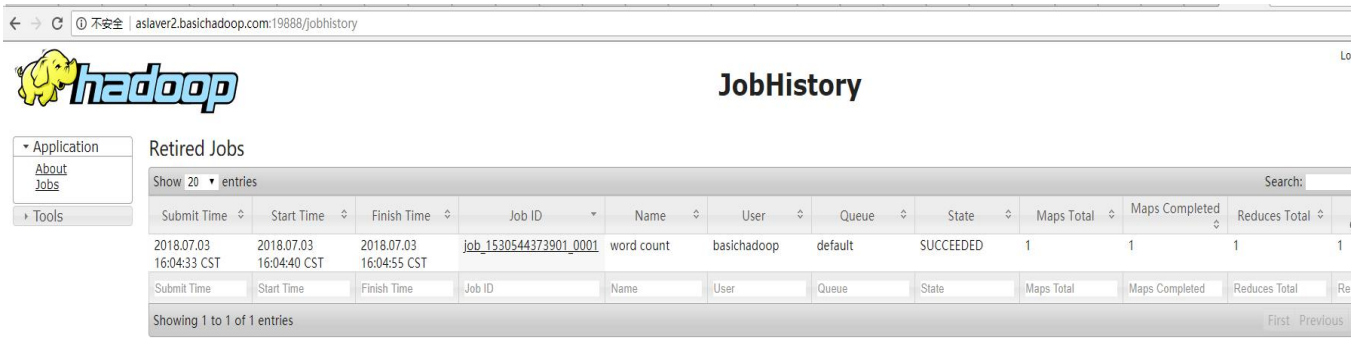
表示两个 namenode
amaster 启动着（active）
aslaver1 等待状态（standby）
一旦 amaster 出现故障，aslaver1 自动起来。



- 查看 aslaver2 节点：yarn 资源管理，监控 task



- yarn 的 task 历史:http://aslaver2.basichadoop.com:19888



三、wordcount 测试

① 进入 hadoop 目录下：

$ cd /opt/modules/hadoop-2.6.5

② 与伪分布式思路一致，同样按照创建输入文件的文件夹，放入待处理的输入文件，执行运行命令，输出，查看结果的思路来：

■ 创建输入文件夹：

$ bin/hdfs dfs -mkdir -p /data/input

```
[basichadoop@amaster hive-1.1.0]$
[basichadoop@amaster hive-1.1.0]$ cd /opt/modules/hadoop-2.6.5
[basichadoop@amaster hadoop-2.6.5]$
[basichadoop@amaster hadoop-2.6.5]$ bin/hdfs dfs -mkdir -p /data/input
[basichadoop@amaster hadoop-2.6.5]$
```

■ 放入待处理文件：

$ bin/hdfs dfs -put README.txt /data/input

```
[basichadoop@amaster hadoop-2.6.5]$ bin/hdfs dfs -put README.txt  /data/input
[basichadoop@amaster hadoop-2.6.5]$
```

■ 查看输入文件夹：

bin/hdfs dfs -ls /data/input

```
[basichadoop@amaster hadoop-2.6.5]$ bin/hdfs dfs -ls /data/input
Found 1 items
-rw-r--r--   3 basichadoop supergroup       1366 2018-07-03 16:01 /data/input/README.txt
[basichadoop@amaster hadoop-2.6.5]$
[basichadoop@amaster hadoop-2.6.5]$
```

■ 在 /share/hadoop/mapreduce 文件夹下有测试用例用的 jar 包：hadoop-mapreduce-examples-2.6.5.jar：

```
[basichadoop@amaster mapreduce]$
[basichadoop@amaster mapreduce]$ ls
hadoop-mapreduce-client-app-2.6.5.jar        hadoop-mapreduce-client-jobclient-2.6.5-tests.jar
hadoop-mapreduce-client-common-2.6.5.jar     hadoop-mapreduce-client-shuffle-2.6.5.jar
hadoop-mapreduce-client-core-2.6.5.jar       hadoop-mapreduce-examples-2.6.5.jar
hadoop-mapreduce-client-hs-2.6.5.jar         lib
hadoop-mapreduce-client-hs-plugins-2.6.5.jar lib-examples
hadoop-mapreduce-client-jobclient-2.6.5.jar  sources
```

■ 我们输入指令进行运算：

$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.5.jar wordcount /data/input /data/output/result

```
[basichadoop@amaster hadoop-2.6.5]$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.5.jar wordcount /data/input /data/output/result
18/07/03 16:04:32 INFO client.RMProxy: Connecting to ResourceManager at aslaver2/192.168.159.162:8032
18/07/03 16:04:32 INFO mapreduce.JobSubmissionFiles: Permissions on staging directory /tmp/hadoop-yarn/staging/basichadoop/.staging are incorrect: rwxrwxrwx. Fixing permissions to correct value rwx------
18/07/03 16:04:33 INFO input.FileInputFormat: Total input paths to process : 1
18/07/03 16:04:33 INFO mapreduce.JobSubmitter: number of splits:1
18/07/03 16:04:33 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1530544373901_0001
18/07/03 16:04:33 INFO impl.YarnClientImpl: Submitted application application_1530544373901_0001
18/07/03 16:04:33 INFO mapreduce.Job: The url to track the job: http://aslaver2:8088/proxy/application_1530544373901_0001/
18/07/03 16:04:33 INFO mapreduce.Job: Running job: job_1530544373901_0001
18/07/03 16:04:42 INFO mapreduce.Job: Job job_1530544373901_0001 running in uber mode : false
18/07/03 16:04:42 INFO mapreduce.Job:  map 0% reduce 0%
18/07/03 16:04:51 INFO mapreduce.Job:  map 100% reduce 0%
18/07/03 16:04:57 INFO mapreduce.Job:  map 100% reduce 100%
18/07/03 16:04:58 INFO mapreduce.Job: Job job_1530544373901_0001 completed successfully
18/07/03 16:04:58 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=1836
                FILE: Number of bytes written=222107
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1468
                HDFS: Number of bytes written=1306
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6501
                Total time spent by all reduces in occupied slots (ms)=3171
                Total time spent by all map tasks (ms)=6501
                Total time spent by all reduce tasks (ms)=3171
                Total vcore-milliseconds taken by all map tasks=6501
                Total vcore-milliseconds taken by all reduce tasks=3171
                Total megabyte-milliseconds taken by all map tasks=6657024
                Total megabyte-milliseconds taken by all reduce tasks=3247104
        Map-Reduce Framework
                Total megabyte-milliseconds taken by all reduce tasks=32
        Map-Reduce Framework
                Map input records=31
                Map output records=179
                Map output bytes=2055
                Map output materialized bytes=1836
                Input split bytes=102
                Combine input records=179
                Combine output records=131
                Reduce input groups=131
                Reduce shuffle bytes=1836
                Reduce input records=131
                Reduce output records=131
                Spilled Records=262
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=83
                CPU time spent (ms)=1510
                Physical memory (bytes) snapshot=329129984
                Virtual memory (bytes) snapshot=1687105536
                Total committed heap usage (bytes)=196153344
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=1366
        File Output Format Counters
                Bytes Written=1306
basichadoop@amaster hadoop-2.6.5]$ hdfs dfs -cat /data/output/result/p
```

③ 查看结果：

- $ bin/hdfs dfs -ls /data/output

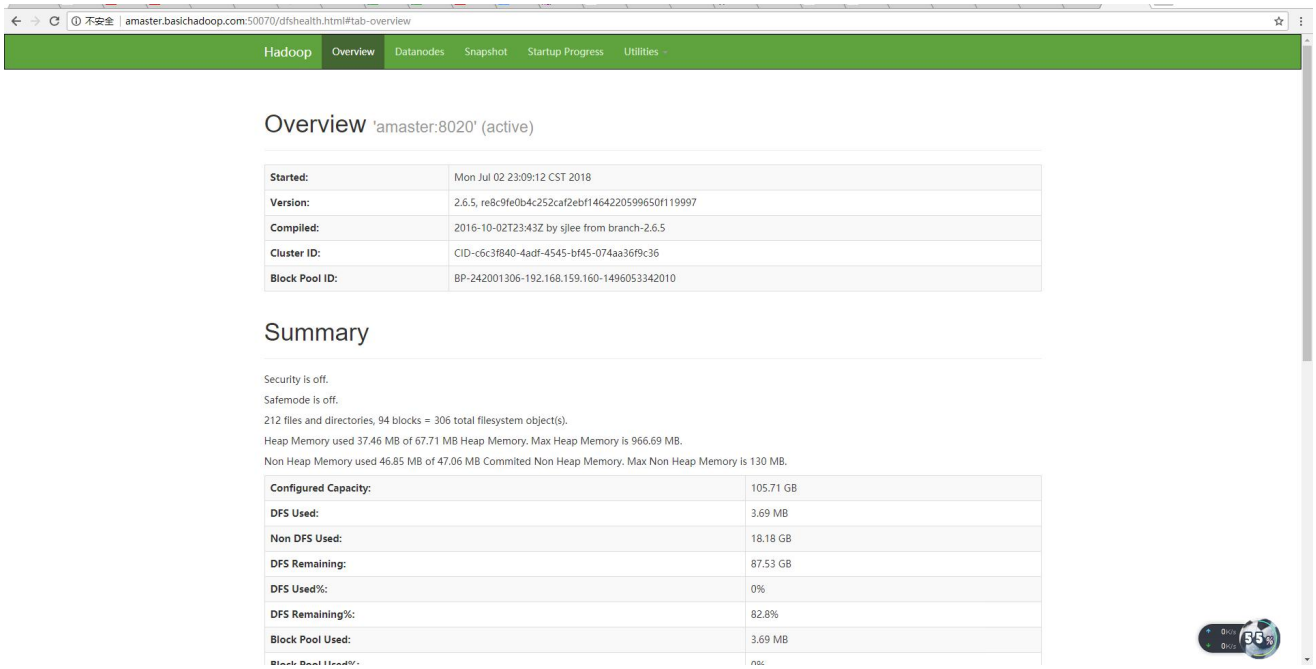
```
/data/output: Unknown command
[basichadoop@amaster hadoop-2.6.5]$ bin/hdfs dfs -ls /data/output
Found 1 items
drwxr-xr-x   - basichadoop supergroup          0 2018-07-03 16:04 /data/output/result
[basichadoop@amaster hadoop-2.6.5]$
```

- $ bin/hdfs dfs -cat /data/output/result/part-r-00000

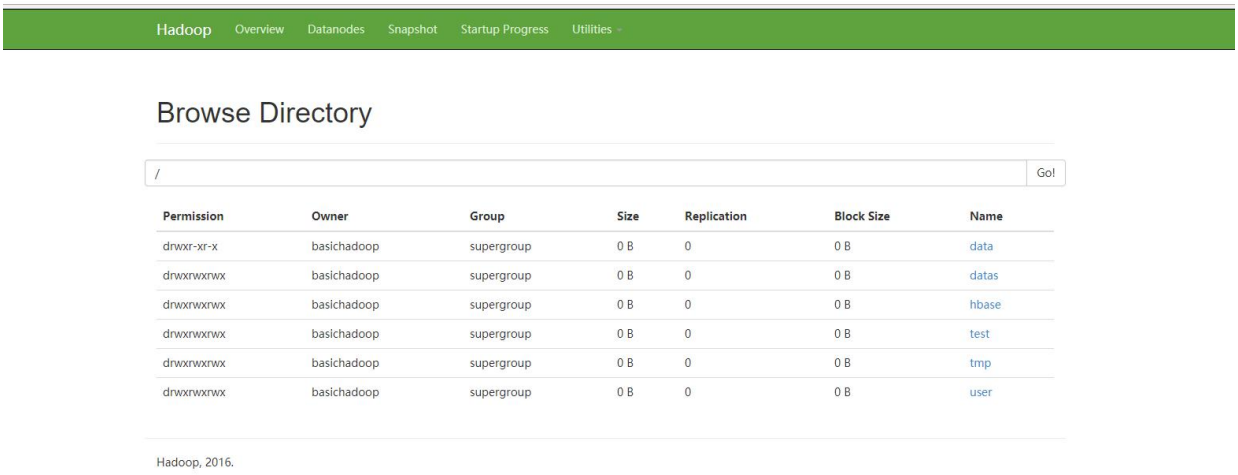
```
[basichadoop@amaster hadoop-2.6.5]$ bin/hdfs dfs -cat /data/output/result/part-r-00000
(BIS),  1
(ECCN)  1
(TSU)   1
(see    1
5D002.C.1,      1
740.13) 1
<http://www.wassenaar.org/>     1
Administration  1
Apache  1
BEFORE  1
BIS     1
Bureau  1
Commerce,       1
Commodity       1
Control 1
Core    1
Department      1
ENC     1
Exception       1
Export  2
For     1
Foundation      1
Government      1
Hadoop  1
Hadoop, 1
Industry        1
Jetty   1
License 1
Number  1
Regulations,    1
SSL     1
Section 1
Security        1
See     1
Software        2
Technology      1
The     4
This    1
U.S.    1
Unrestricted    1
about   1
algorithms.     1
```

四、日志查看

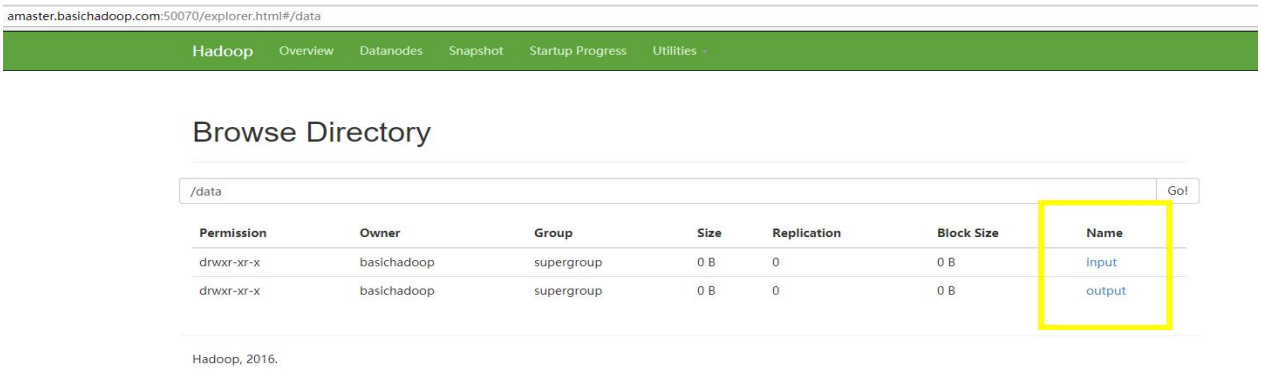
① 此时在本地浏览器输入：http://amaster.basichadoop.com:50070/



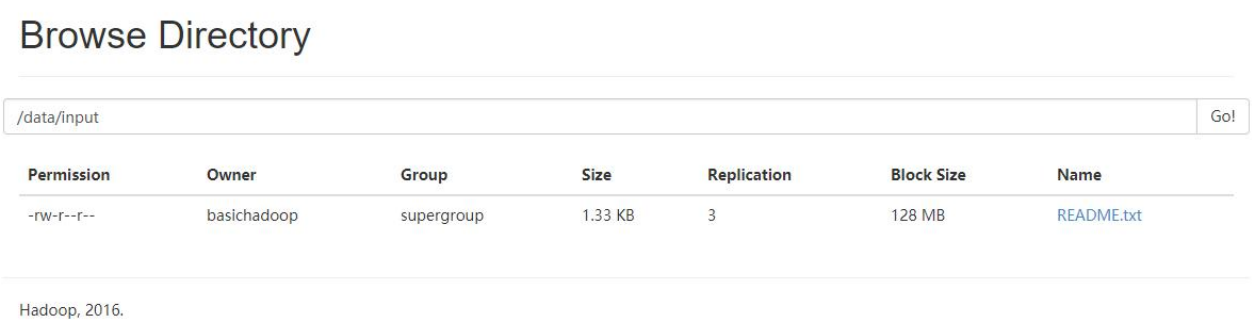
② 查看刚刚 wordcount 测试的日志：

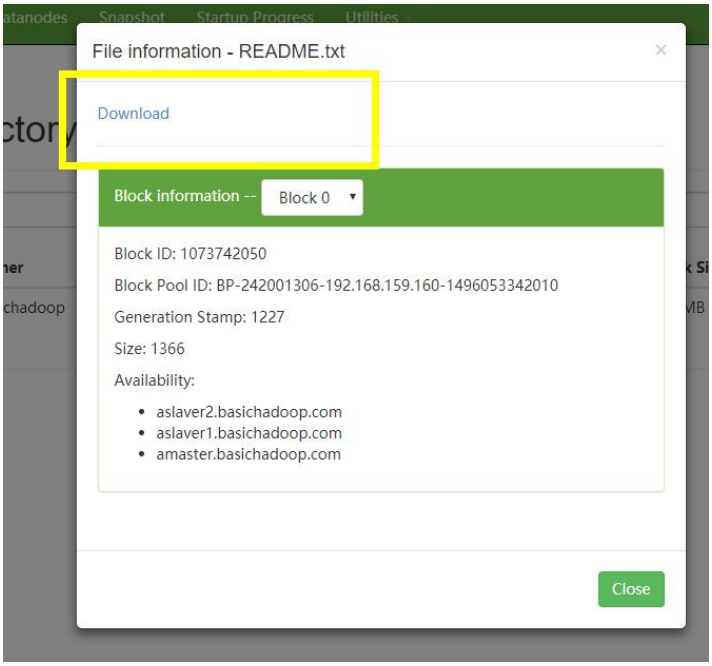


- 点击”data”,因为我们刚刚新建的输入输出文件都在 data 文件夹下面。input 和 output 文件都在该目录下面



- 我们尝试点击”input”:发现我们刚刚提交的输入文件在这，继续点击，发现可以下载。因此这就涉及到了权限问题了，在真正的企业级项目中，日志的查看只有部分权限。





- 点击 output,发现输出结果：继续点击：

Browse Directory

/data/output Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	basichadoop	supergroup	0 B	0	0 B	result

Hadoop, 2016.

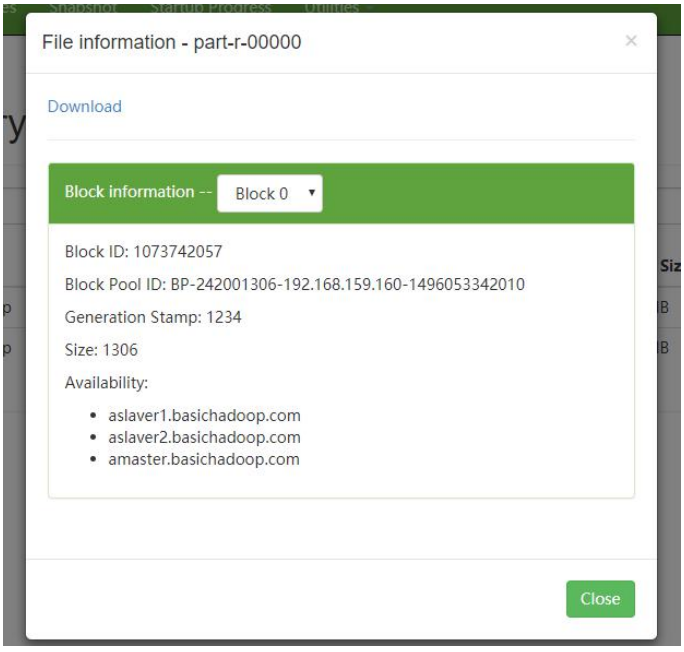
Browse Directory

/data/output/result Go!

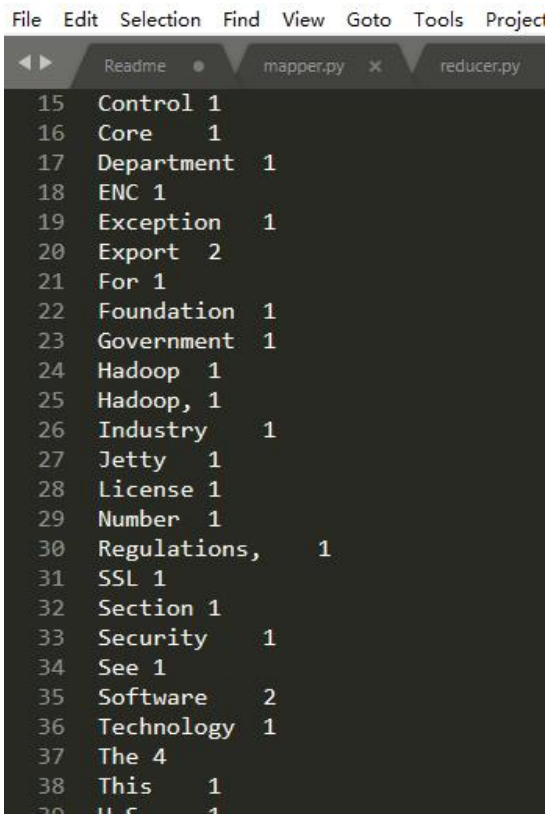
Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	basichadoop	supergroup	0 B	3	128 MB	_SUCCESS
-rw-r--r--	basichadoop	supergroup	1.28 KB	3	128 MB	part-r-00000

Hadoop, 2016.

- 点击 part-r-00000,发现结果可下载：



- 我们下载一下,打开：和我们在命令行里所见一致：


File Edit Selection Find View Goto Tools Project
Readme
mapper.py
reducer.py
15 Control 1
16 Core 1
17 Department 1
18 ENC 1
19 Exception 1
20 Export 2
21 For 1
22 Foundation 1
23 Government 1
24 Hadoop 1
25 Hadoop, 1
26 Industry 1
27 Jetty 1
28 License 1
29 Number 1
30 Regulations, 1
31 SSL 1
32 Section 1
33 Security 1
34 See 1
35 Software 2
36 Technology 1
37 The 4
38 This 1