南昌大学实验报告

姓名：Qing Liu

学号：6130116184

邮箱地址：1119637652@qq.com

专业班级：Class 164 of Computer Science and Technology

实验日期：THU，May 16th, 2019

课程名称：Cloud Computing Technology Experiments

实验项目名称

Hadoop And MapReduce

实验目的

- Understanding the concept of MapReduce Model
- Building a file system in a distributed way that store large data trunks
- Merging all things together into a docker image
- Writing some sample demo on running MapReduce models

实验基础

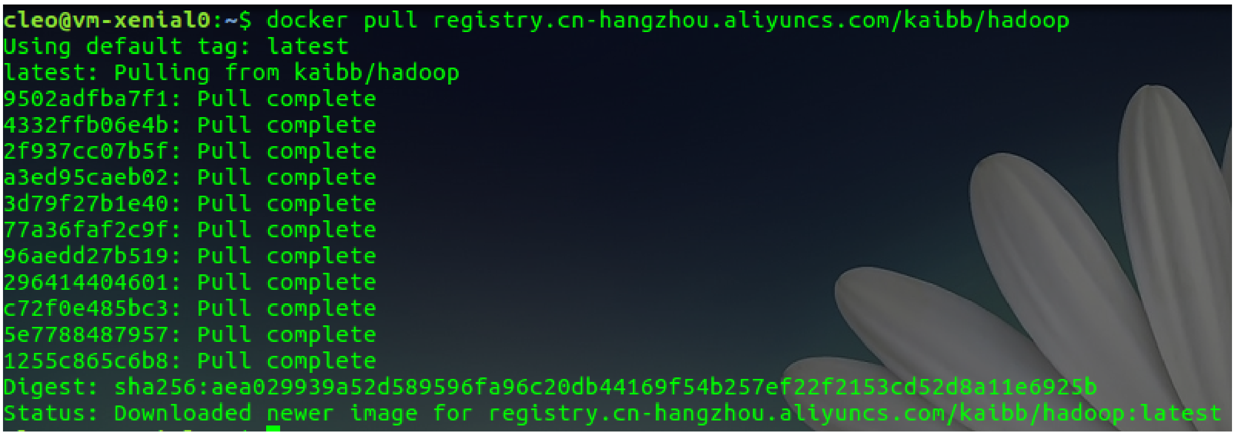
- Hadoop
- MapReduce
- MongoDB

实验步骤

Word Count

- Pull the hadoop image from Docker Hub

```
docker pull registry.cn-hangzhou.aliyuncs.com/kaibb/hadoop
```



```
cleo@vm-xenial0:~$ docker pull registry.cn-hangzhou.aliyuncs.com/kaibb/hadoop
Using default tag: latest
latest: Pulling from kaibb/hadoop
9502adfba7f1: Pull complete
4332ffb06e4b: Pull complete
2f937cc07b5f: Pull complete
a3ed95caeb02: Pull complete
3d79f27b1e40: Pull complete
77a36faf2c9f: Pull complete
96aedd27b519: Pull complete
296414404601: Pull complete
c72f0e485bc3: Pull complete
5e7788487957: Pull complete
1255c865c6b8: Pull complete
Digest: sha256:aea029939a52d589596fa96c20db44169f54b257ef22f2153cd52d8a11e6925b
Status: Downloaded newer image for registry.cn-hangzhou.aliyuncs.com/kaibb/hadoop:latest
```

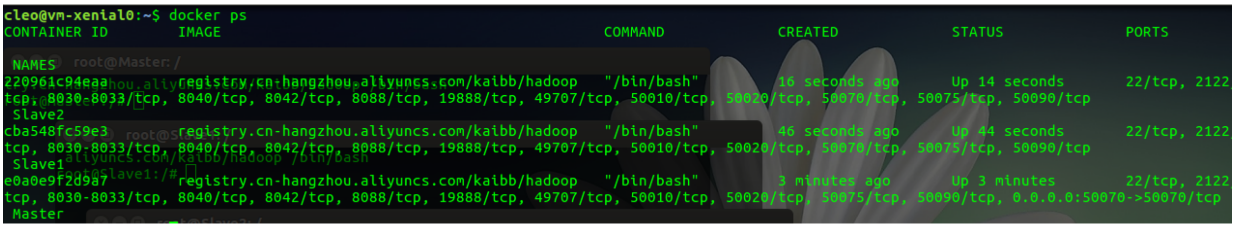
- Run contaniers

  After pull the docker image, we need to create three containers, one work as master, and others work as slaves.

  ```
  docker run -i -t --name Master -h Master -p 50070:50070 registry.cn-hangzhou
  ```

  ```
  docker run -i -t --name Slave1 -h Slave1 registry.cn-hangzhou.aliyuncs.com/k
  ```

  ```
  docker run -i -t --name Slave2 -h Slave2 registry.cn-hangzhou.aliyuncs.com/k
  ```

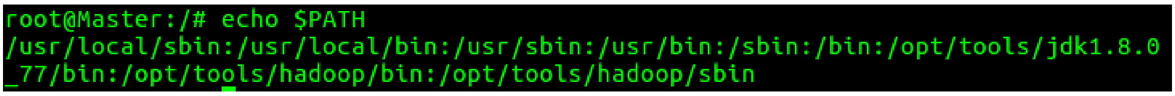


```
cleo@vm-xenial0:~$ docker ps
CONTAINER ID        IMAGE                                            COMMAND             CREATED             STATUS              PORTS
 NAMES
220961c94eaa        registry.cn-hangzhou.aliyuncs.com/kaibb/hadoop   "/bin/bash"         16 seconds ago      Up 14 seconds       22/tcp, 2122
tcp, 8030-8033/tcp, 8040/tcp, 8042/tcp, 8088/tcp, 19888/tcp, 49707/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp
 Slave2
cba548fc59e3        registry.cn-hangzhou.aliyuncs.com/kaibb/hadoop   "/bin/bash"         46 seconds ago      Up 44 seconds       22/tcp, 2122
tcp, 8030-8033/tcp, 8040/tcp, 8042/tcp, 8088/tcp, 19888/tcp, 49707/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp
 Slave1
e0a0e9f2d9a7        registry.cn-hangzhou.aliyuncs.com/kaibb/hadoop   "/bin/bash"         3 minutes ago       Up 3 minutes        22/tcp, 2122
tcp, 8030-8033/tcp, 8040/tcp, 8042/tcp, 8088/tcp, 19888/tcp, 49707/tcp, 50010/tcp, 50020/tcp, 50075/tcp, 50090/tcp, 0.0.0.0:50070->50070/tcp
 Master
```

- Confugure SSH (In Master and Slaves)

  - check the PATH environment variable

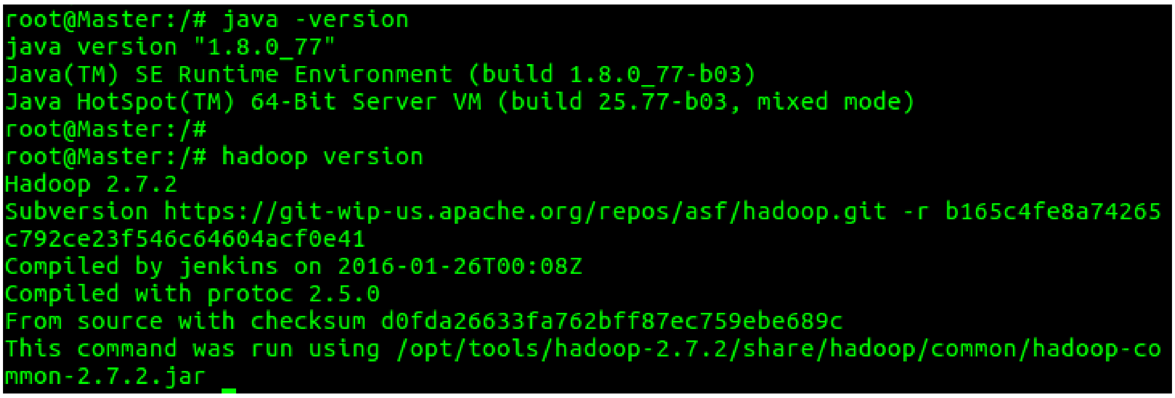
    ```
    echo $PATH
    ```

    ```
    root@Master:/# echo $PATH
    /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/opt/tools/jdk1.8.0
    _77/bin:/opt/tools/hadoop/bin:/opt/tools/hadoop/sbin
    ```

    We can see that the path of java and hadoop had been added into the PATH.

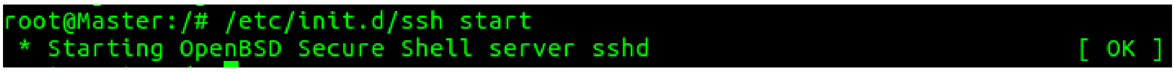
    ```
    java -version
    hadoop version
    ```



```
root@Master:/# java -version
java version "1.8.0_77"
Java(TM) SE Runtime Environment (build 1.8.0_77-b03)
Java HotSpot(TM) 64-Bit Server VM (build 25.77-b03, mixed mode)
root@Master:/#
root@Master:/# hadoop version
Hadoop 2.7.2
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r b165c4fe8a74265
c792ce23f546c64604acf0e41
Compiled by jenkins on 2016-01-26T00:08Z
Compiled with protoc 2.5.0
From source with checksum d0fda26633fa762bff87ec759ebe689c
This command was run using /opt/tools/hadoop-2.7.2/share/hadoop/common/hadoop-co
mmon-2.7.2.jar
```

- Run SSH

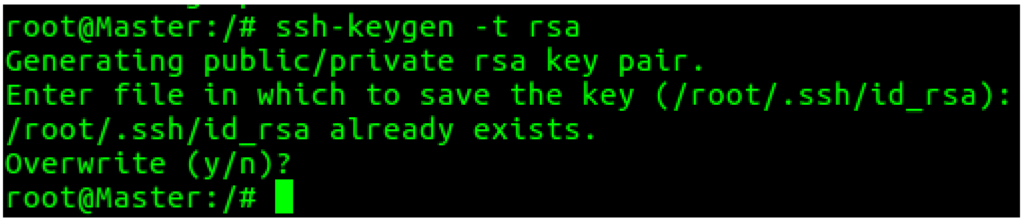
```
/etc/init.d/ssh start
```



```
root@Master:/# /etc/init.d/ssh start
 * Starting OpenBSD Secure Shell server sshd                              [ OK ]
```

Generate the key:

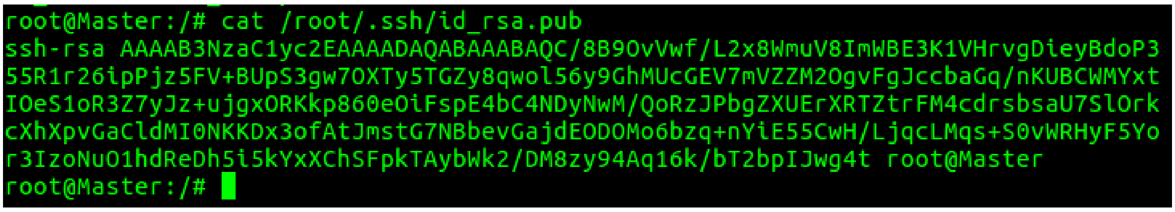
```
ssh-keygen -t rsa
```



```
root@Master:/# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)?
root@Master:/#
```

Check the key in `/root/.ssh/`

```
cat /root/.ssh/id_rsa.pub
```



```
root@Master:/# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC/8B9OvVwf/L2x8WmuV8ImWBE3K1VHrvgDieyBdoP3
55R1r26ipPjz5FV+BUpS3gw7OXTy5TGZy8qwol56y9GhMUcGEV7mVZZM2OgvFgJccbaGq/nKUBCWMYxt
IOeS1oR3Z7yJz+ujgxORKkp860eOiFspE4bC4NDyNwM/QoRzJPbgZXUErXRTZtrFM4cdrsbsaU7SlOrk
cXhXpvGaCldMI0NKKDx3ofAtJmstG7NBbevGajdEODOMo6bzq+nYiE55CwH/LjqcLMqs+S0vWRHyF5Yo
r3IzoNuO1hdReDh5i5kYxXChSFpkTAybWk2/DM8zy94Aq16k/bT2bpIJwg4t root@Master
root@Master:/#
```

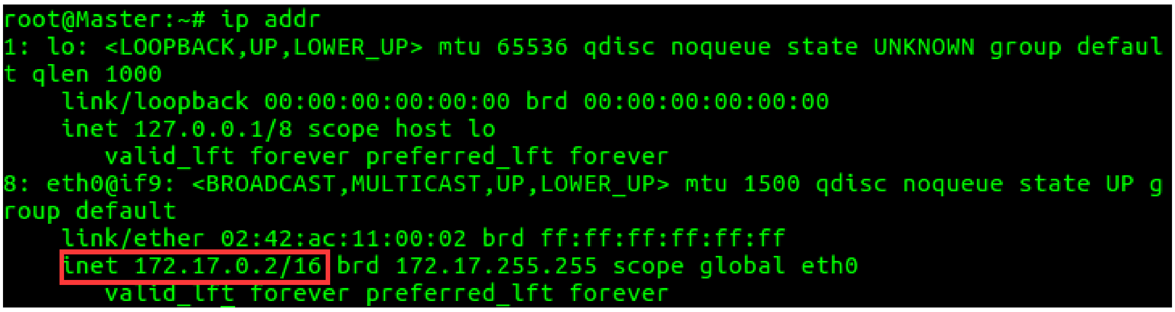
We need to copy the key to other two machines in every node and save into `authorized_keys`

```
cd /root/.ssh
vim master_key
```

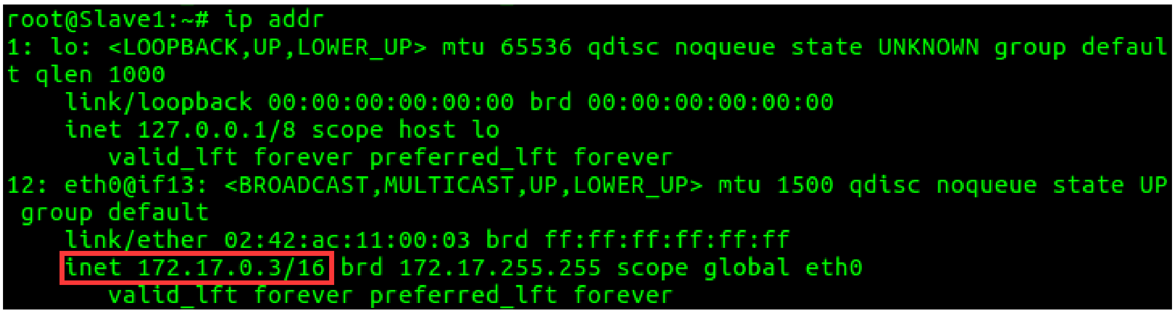
```
cat master_key >> authorized_keys
```

- Check the ip address of master and slaves

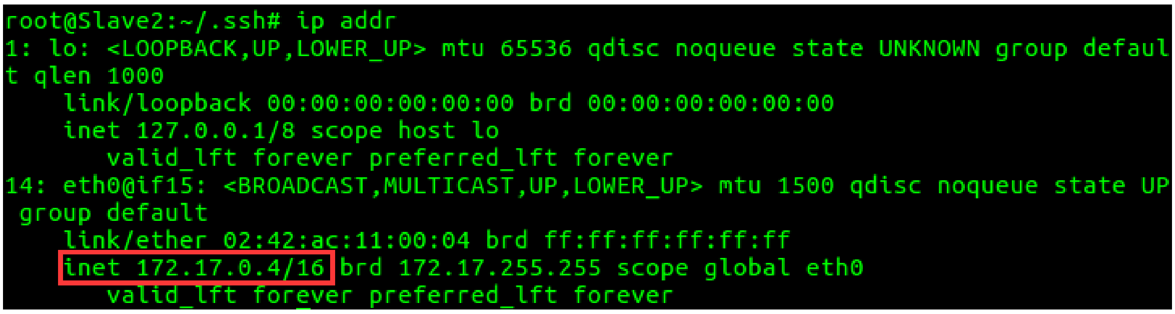
  ```
  ip addr
  ```



```
root@Master:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
8: eth0@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP g
roup default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
```



```
root@Slave1:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
12: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
```

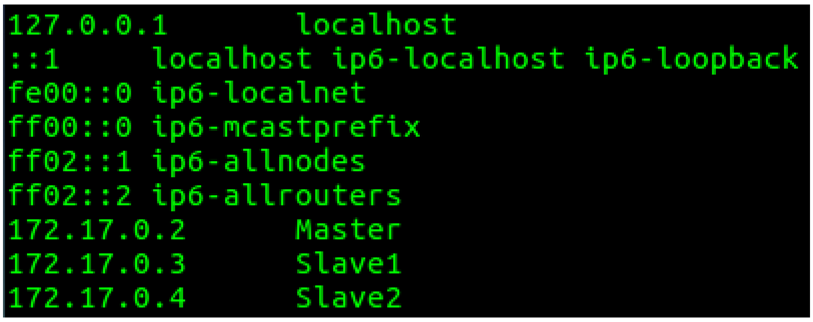


```
root@Slave2:~/.ssh# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
14: eth0@if15: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:ac:11:00:04 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.4/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
```

- Configure the `/etc/hosts`

  Add the hostname and ip address into `/etc/hosts`

  ```
  ...

  172.17.0.2         Master
  172.17.0.3         Slave1
  172.17.0.4         Slave2
  ```

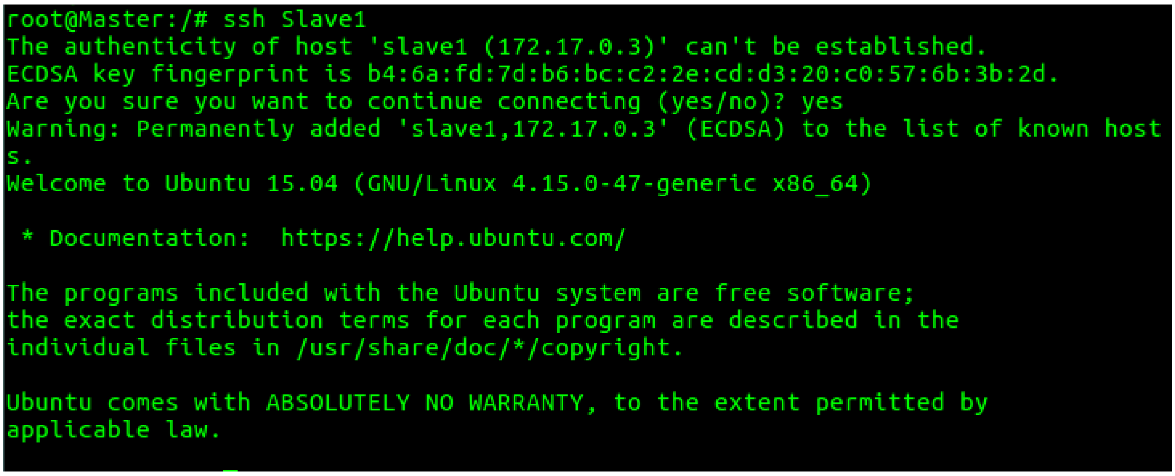


```
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      Master
172.17.0.3      Slave1
172.17.0.4      Slave2
```

- Run ssh

  In master:

  ```
  ssh Slave1
  ```



```
root@Master:/# ssh Slave1
The authenticity of host 'slave1 (172.17.0.3)' can't be established.
ECDSA key fingerprint is b4:6a:fd:7d:b6:bc:c2:2e:cd:d3:20:c0:57:6b:3b:2d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'slave1,172.17.0.3' (ECDSA) to the list of known host
s.
Welcome to Ubuntu 15.04 (GNU/Linux 4.15.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

  We connect to Slave1 from Master successfully by ssh without password.

- Configure Hadoop

  - New some directory and send to two slaves by `scp`

    ```
    cd /
    mkdir -p hadoop/data
    mkdir -p hadoop/name
    mkdir -p hadoop/tmp
    scp -r hadoop root@Slave1:/
    scp -r hadoop root@Slave2:/
    ```

  - Configure some files in `/opt/tools/hadoop/etc/hadoop/` and send them into slaves

    `hadoop-env.sh`:

    ```
    export JAVA_HOME=/opt/tools/jdk1.8.0_77
    ```

`core-site.xml`:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://Master:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/hadoop/tmp</value>
  </property>
</configuration>
```

`hdfs-site.xml`:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/hadoop/data</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/hadoop/name</value>
  </property>
</configuration>
```

`mapred-site.xml`:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

`yarn-site.xml`:

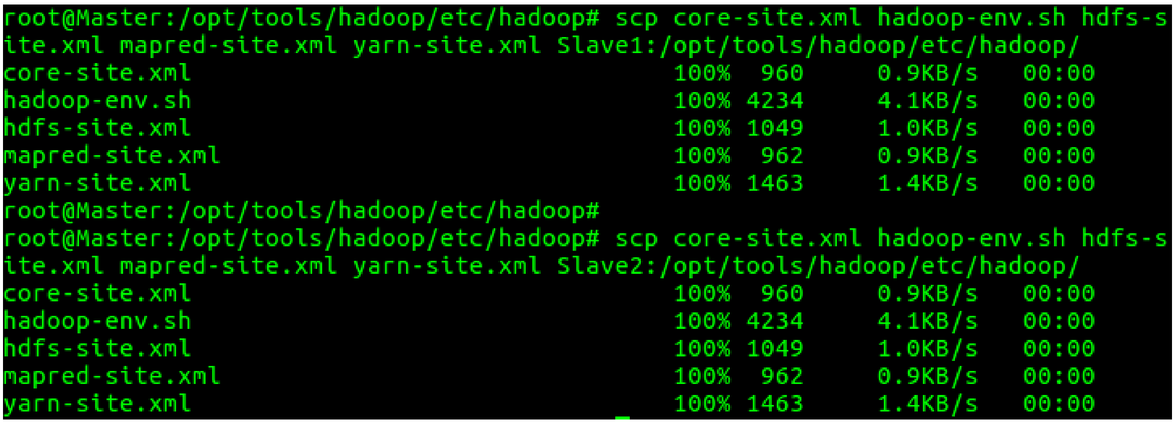
```
<configuration>
  <property>
  <name>yarn.resourcemanager.address</name>
  <value>Master:8032</value>
  </property>
  <property>
      <name>yarn.resourcemanager.scheduler.address</name>
      <value>Master:8030</value> </property> <property>
      <name>yarn.resourcemanager.resource-tracker.address</name>
      <value>Master:8031</value>
  </property>
  <property>
  <name>yarn.resourcemanager.admin.address</name>
  <value>Master:8033</value>
  </property>
  <property>
      <name>yarn.resourcemanager.webapp.address</name>
      <value>Master:8088</value>
  </property>
  <property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
  </property>
  <property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

`slaves`:

```
Master
Slave1
Slave2
```

Send to slaves:

```
scp core-site.xml hadoop-env.sh hdfs-site.xml mapred-site.xml yarn-site.
scp core-site.xml hadoop-env.sh hdfs-site.xml mapred-site.xml yarn-site.
```

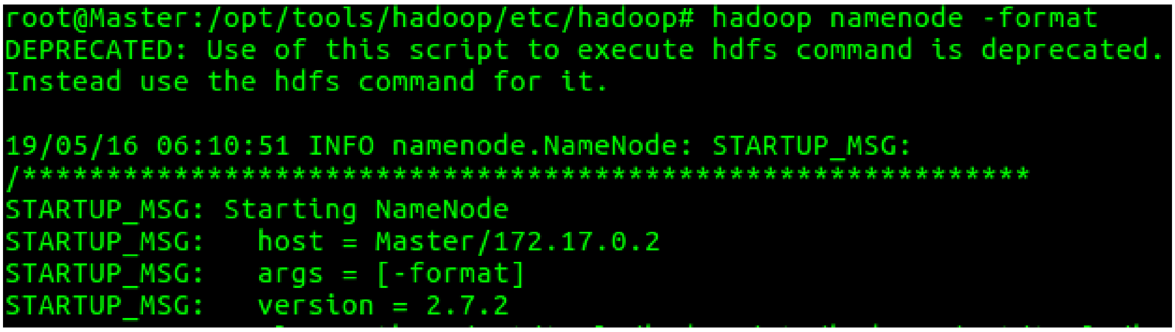


```
root@Master:/opt/tools/hadoop/etc/hadoop# scp core-site.xml hadoop-env.sh hdfs-s
ite.xml mapred-site.xml yarn-site.xml Slave1:/opt/tools/hadoop/etc/hadoop/
core-site.xml                                 100%  960     0.9KB/s   00:00
hadoop-env.sh                                 100% 4234     4.1KB/s   00:00
hdfs-site.xml                                 100% 1049     1.0KB/s   00:00
mapred-site.xml                               100%  962     0.9KB/s   00:00
yarn-site.xml                                 100% 1463     1.4KB/s   00:00
root@Master:/opt/tools/hadoop/etc/hadoop#
root@Master:/opt/tools/hadoop/etc/hadoop# scp core-site.xml hadoop-env.sh hdfs-s
ite.xml mapred-site.xml yarn-site.xml Slave2:/opt/tools/hadoop/etc/hadoop/
core-site.xml                                 100%  960     0.9KB/s   00:00
hadoop-env.sh                                 100% 4234     4.1KB/s   00:00
hdfs-site.xml                                 100% 1049     1.0KB/s   00:00
mapred-site.xml                               100%  962     0.9KB/s   00:00
yarn-site.xml                                 100% 1463     1.4KB/s   00:00
```

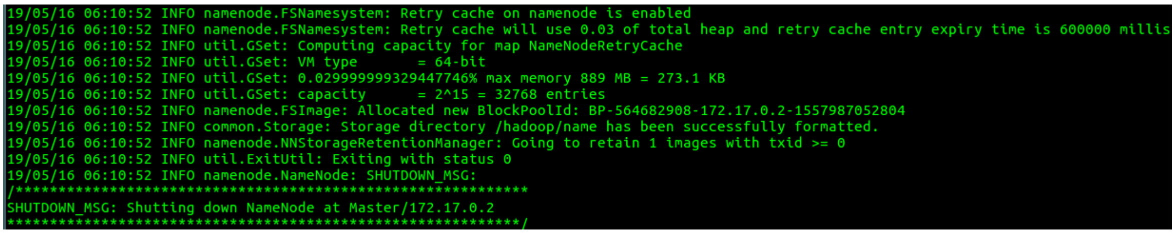
- Run Hadoop

- Format operation

  ```
  hadoop namenode -format
  ```

  ```
  root@Master:/opt/tools/hadoop/etc/hadoop# hadoop namenode -format
  DEPRECATED: Use of this script to execute hdfs command is deprecated.
  Instead use the hdfs command for it.

  19/05/16 06:10:51 INFO namenode.NameNode: STARTUP_MSG:
  /************************************************************
  STARTUP_MSG: Starting NameNode
  STARTUP_MSG:   host = Master/172.17.0.2
  STARTUP_MSG:   args = [-format]
  STARTUP_MSG:   version = 2.7.2
  ```

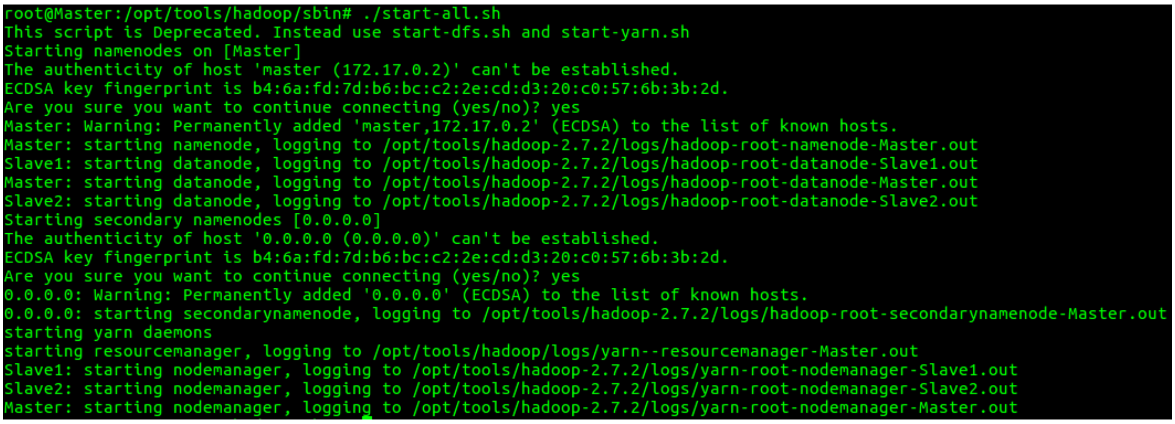
  

  ```
  19/05/16 06:10:52 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
  19/05/16 06:10:52 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
  19/05/16 06:10:52 INFO util.GSet: Computing capacity for map NameNodeRetryCache
  19/05/16 06:10:52 INFO util.GSet: VM type       = 64-bit
  19/05/16 06:10:52 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
  19/05/16 06:10:52 INFO util.GSet: capacity      = 2^15 = 32768 entries
  19/05/16 06:10:52 INFO namenode.FSImage: Allocated new BlockPoolId: BP-564682908-172.17.0.2-1557987052804
  19/05/16 06:10:52 INFO common.Storage: Storage directory /hadoop/name has been successfully formatted.
  19/05/16 06:10:52 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
  19/05/16 06:10:52 INFO util.ExitUtil: Exiting with status 0
  19/05/16 06:10:52 INFO namenode.NameNode: SHUTDOWN_MSG:
  /************************************************************
  SHUTDOWN_MSG: Shutting down NameNode at Master/172.17.0.2
  ************************************************************/
  ```

- Start the cluster

  ```
  cd /opt/tools/hadoop/sbin
  ./start-all.sh
  ```

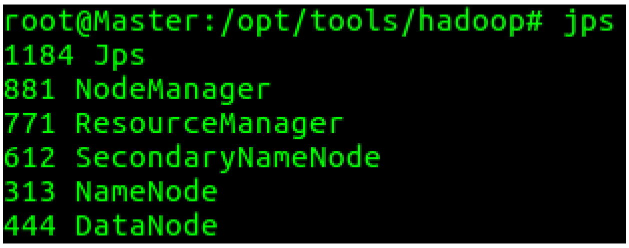
  In this period, if the password was asked, press `123465`.

  ```
  root@Master:/opt/tools/hadoop/sbin# ./start-all.sh
  This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
  Starting namenodes on [Master]
  The authenticity of host 'master (172.17.0.2)' can't be established.
  ECDSA key fingerprint is b4:6a:fd:7d:b6:bc:c2:2e:cd:d3:20:c0:57:6b:3b:2d.
  Are you sure you want to continue connecting (yes/no)? yes
  Master: Warning: Permanently added 'master,172.17.0.2' (ECDSA) to the list of known hosts.
  Master: starting namenode, logging to /opt/tools/hadoop-2.7.2/logs/hadoop-root-namenode-Master.out
  Slave1: starting datanode, logging to /opt/tools/hadoop-2.7.2/logs/hadoop-root-datanode-Slave1.out
  Master: starting datanode, logging to /opt/tools/hadoop-2.7.2/logs/hadoop-root-datanode-Master.out
  Slave2: starting datanode, logging to /opt/tools/hadoop-2.7.2/logs/hadoop-root-datanode-Slave2.out
  Starting secondary namenodes [0.0.0.0]
  The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
  ECDSA key fingerprint is b4:6a:fd:7d:b6:bc:c2:2e:cd:d3:20:c0:57:6b:3b:2d.
  Are you sure you want to continue connecting (yes/no)? yes
  0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
  0.0.0.0: starting secondarynamenode, logging to /opt/tools/hadoop-2.7.2/logs/hadoop-root-secondarynamenode-Master.out
  starting yarn daemons
  starting resourcemanager, logging to /opt/tools/hadoop/logs/yarn--resourcemanager-Master.out
  Slave1: starting nodemanager, logging to /opt/tools/hadoop-2.7.2/logs/yarn-root-nodemanager-Slave1.out
  Slave2: starting nodemanager, logging to /opt/tools/hadoop-2.7.2/logs/yarn-root-nodemanager-Slave2.out
  Master: starting nodemanager, logging to /opt/tools/hadoop-2.7.2/logs/yarn-root-nodemanager-Master.out
  ```

- Check the status with `jps` command

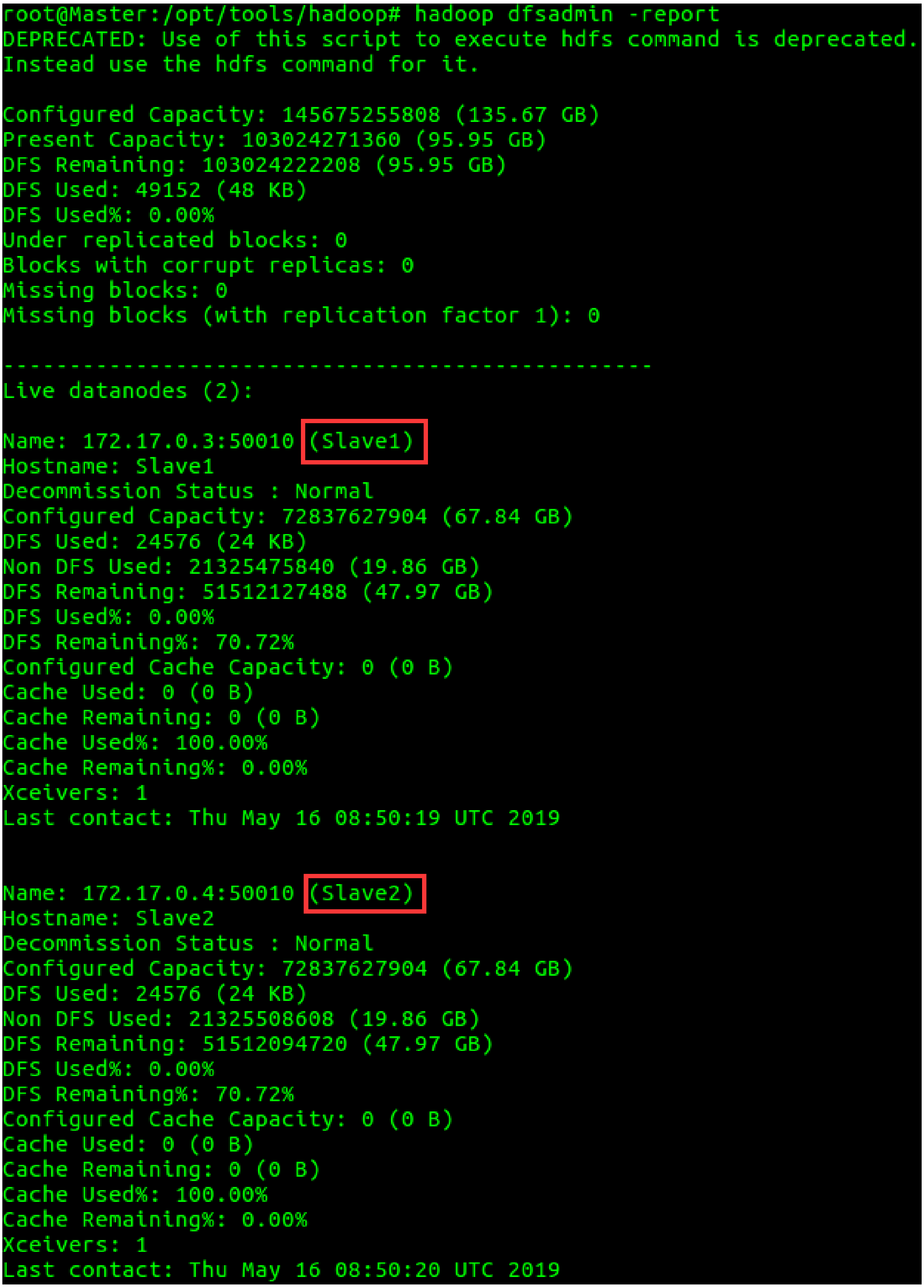
  ```
  cd /opt/tools/hadoop
  jps
  ```



```
root@Master:/opt/tools/hadoop# jps
1184 Jps
881 NodeManager
771 ResourceManager
612 SecondaryNameNode
313 NameNode
444 DataNode
```

- Check the information of every node:

```
hadoop dfsadmin -report
```



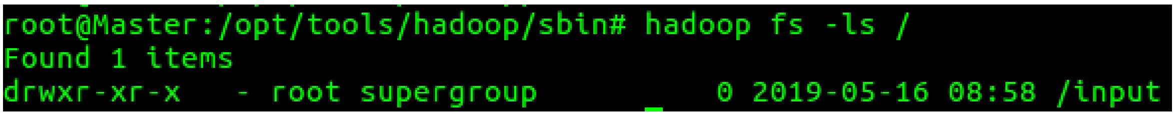
```
root@Master:/opt/tools/hadoop# hadoop dfsadmin -report
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

Configured Capacity: 145675255808 (135.67 GB)
Present Capacity: 103024271360 (95.95 GB)
DFS Remaining: 103024222208 (95.95 GB)
DFS Used: 49152 (48 KB)
DFS Used%: 0.00%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
Live datanodes (2):

Name: 172.17.0.3:50010 (Slave1)
Hostname: Slave1
Decommission Status : Normal
Configured Capacity: 72837627904 (67.84 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 21325475840 (19.86 GB)
DFS Remaining: 51512127488 (47.97 GB)
DFS Used%: 0.00%
DFS Remaining%: 70.72%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Thu May 16 08:50:19 UTC 2019


Name: 172.17.0.4:50010 (Slave2)
Hostname: Slave2
Decommission Status : Normal
Configured Capacity: 72837627904 (67.84 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 21325508608 (19.86 GB)
DFS Remaining: 51512094720 (47.97 GB)
DFS Used%: 0.00%
DFS Remaining%: 70.72%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Thu May 16 08:50:20 UTC 2019
```

- Word Count
  - New a directory in hdfs

    ```
    hadoop fs -mkdir /input
    hadoop fs -mkdir /output
    ```

Check the directory you created

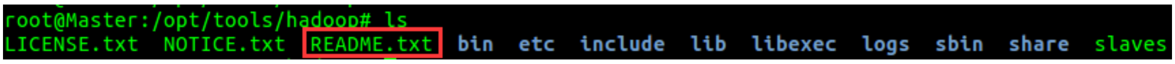
```
hadoop fs -ls /
```



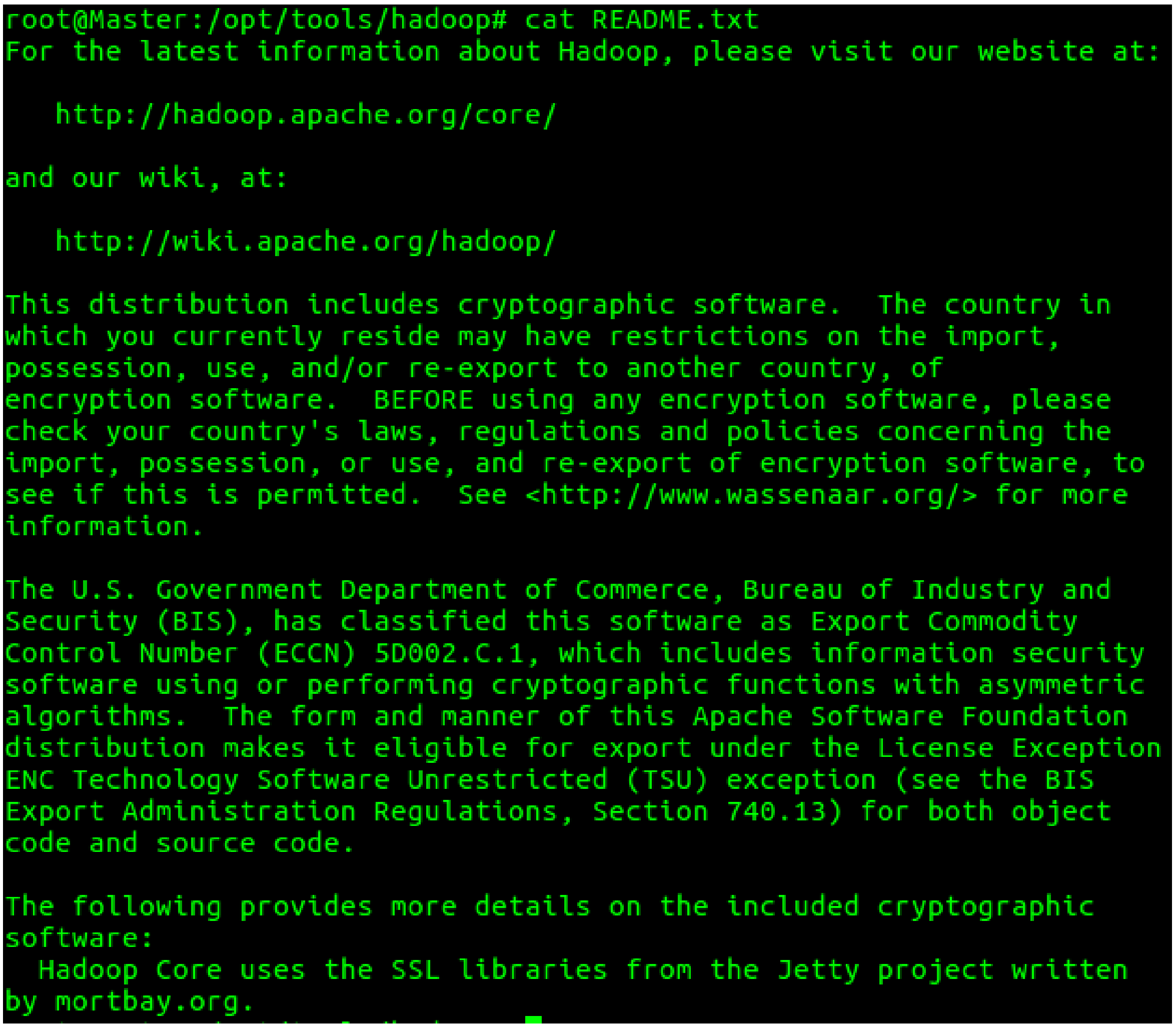
```
root@Master:/opt/tools/hadoop/sbin# hadoop fs -ls /
Found 1 items
drwxr-xr-x   - root supergroup          0 2019-05-16 08:58 /input
```

- Upload file into hdfs

  In this step, there is a file `README.txt` in current directory, so I upload this file into hdfs to test the word count.

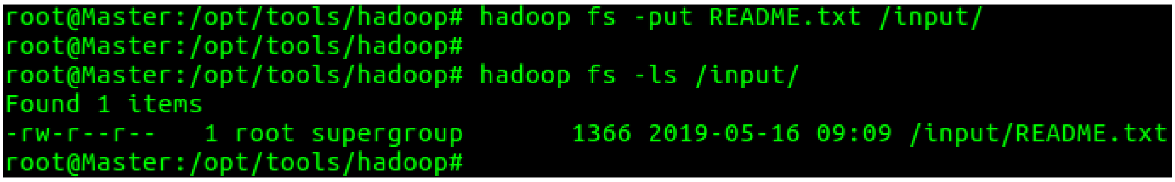


```
root@Master:/opt/tools/hadoop# ls
LICENSE.txt  NOTICE.txt  README.txt  bin  etc  include  lib  libexec  logs  sbin  share  slaves
```



```
root@Master:/opt/tools/hadoop# cat README.txt
For the latest information about Hadoop, please visit our website at:

   http://hadoop.apache.org/core/

and our wiki, at:

   http://wiki.apache.org/hadoop/

This distribution includes cryptographic software.  The country in
which you currently reside may have restrictions on the import,
possession, use, and/or re-export to another country, of
encryption software.  BEFORE using any encryption software, please
check your country's laws, regulations and policies concerning the
import, possession, or use, and re-export of encryption software, to
see if this is permitted.  See <http://www.wassenaar.org/> for more
information.

The U.S. Government Department of Commerce, Bureau of Industry and
Security (BIS), has classified this software as Export Commodity
Control Number (ECCN) 5D002.C.1, which includes information security
software using or performing cryptographic functions with asymmetric
algorithms.  The form and manner of this Apache Software Foundation
distribution makes it eligible for export under the License Exception
ENC Technology Software Unrestricted (TSU) exception (see the BIS
Export Administration Regulations, Section 740.13) for both object
code and source code.

The following provides more details on the included cryptographic
software:
  Hadoop Core uses the SSL libraries from the Jetty project written
by mortbay.org.
```

```
hadoop fs -put README.txt /input/
```

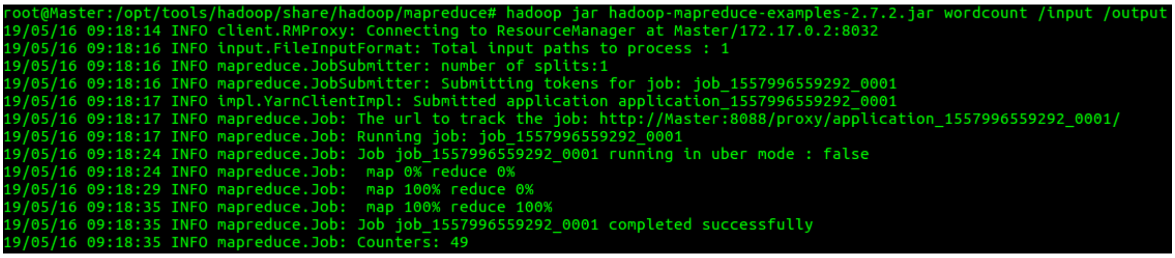


```
root@Master:/opt/tools/hadoop# hadoop fs -put README.txt /input/
root@Master:/opt/tools/hadoop#
root@Master:/opt/tools/hadoop# hadoop fs -ls /input/
Found 1 items
-rw-r--r--   1 root supergroup       1366 2019-05-16 09:09 /input/README.txt
root@Master:/opt/tools/hadoop#
```

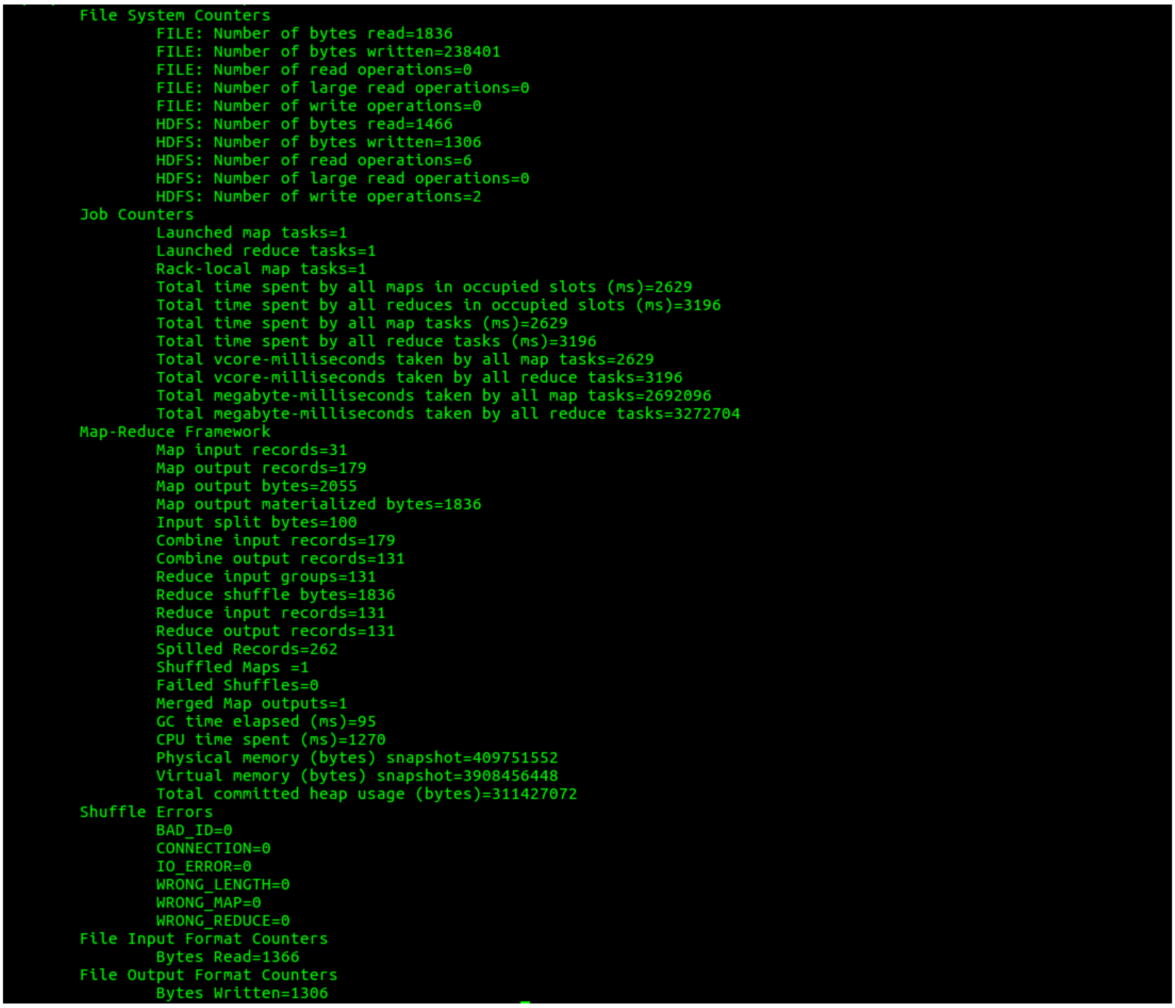
- Do the word count

```
cd /opt/tools/hadoop/share/hadoop/mapreduce
hadoop jar hadoop-mapreduce-examples-2.7.2.jar wordcount /input /output
```

Here are the output informations:



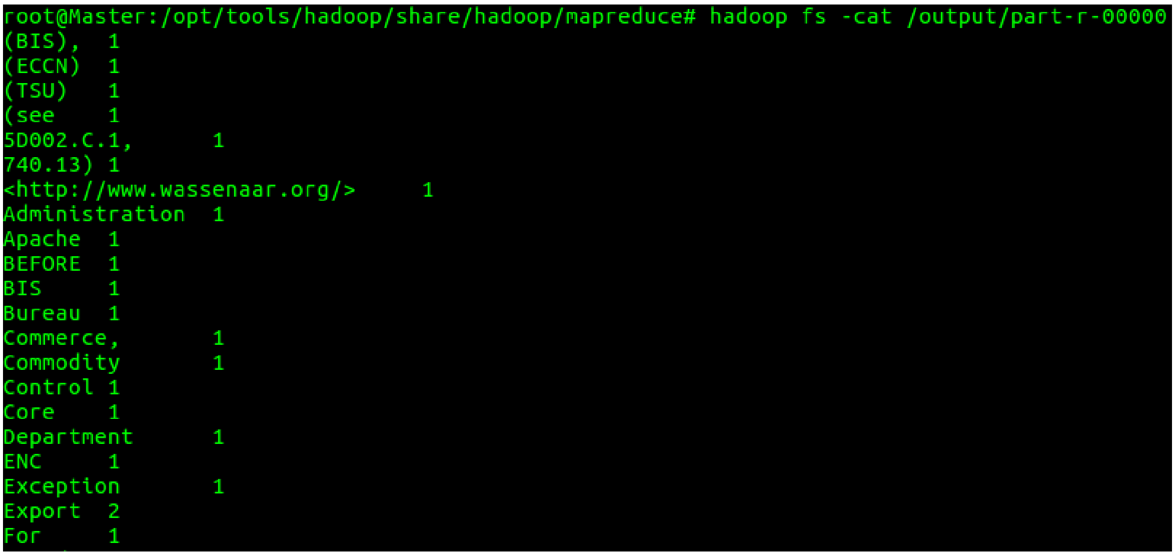
```
root@Master:/opt/tools/hadoop/share/hadoop/mapreduce# hadoop jar hadoop-mapreduce-examples-2.7.2.jar wordcount /input /output
19/05/16 09:18:14 INFO client.RMProxy: Connecting to ResourceManager at Master/172.17.0.2:8032
19/05/16 09:18:16 INFO input.FileInputFormat: Total input paths to process : 1
19/05/16 09:18:16 INFO mapreduce.JobSubmitter: number of splits:1
19/05/16 09:18:16 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1557996559292_0001
19/05/16 09:18:17 INFO impl.YarnClientImpl: Submitted application application_1557996559292_0001
19/05/16 09:18:17 INFO mapreduce.Job: The url to track the job: http://Master:8088/proxy/application_1557996559292_0001/
19/05/16 09:18:17 INFO mapreduce.Job: Running job: job_1557996559292_0001
19/05/16 09:18:24 INFO mapreduce.Job: Job job_1557996559292_0001 running in uber mode : false
19/05/16 09:18:24 INFO mapreduce.Job:  map 0% reduce 0%
19/05/16 09:18:29 INFO mapreduce.Job:  map 100% reduce 0%
19/05/16 09:18:35 INFO mapreduce.Job:  map 100% reduce 100%
19/05/16 09:18:35 INFO mapreduce.Job: Job job_1557996559292_0001 completed successfully
19/05/16 09:18:35 INFO mapreduce.Job: Counters: 49
```



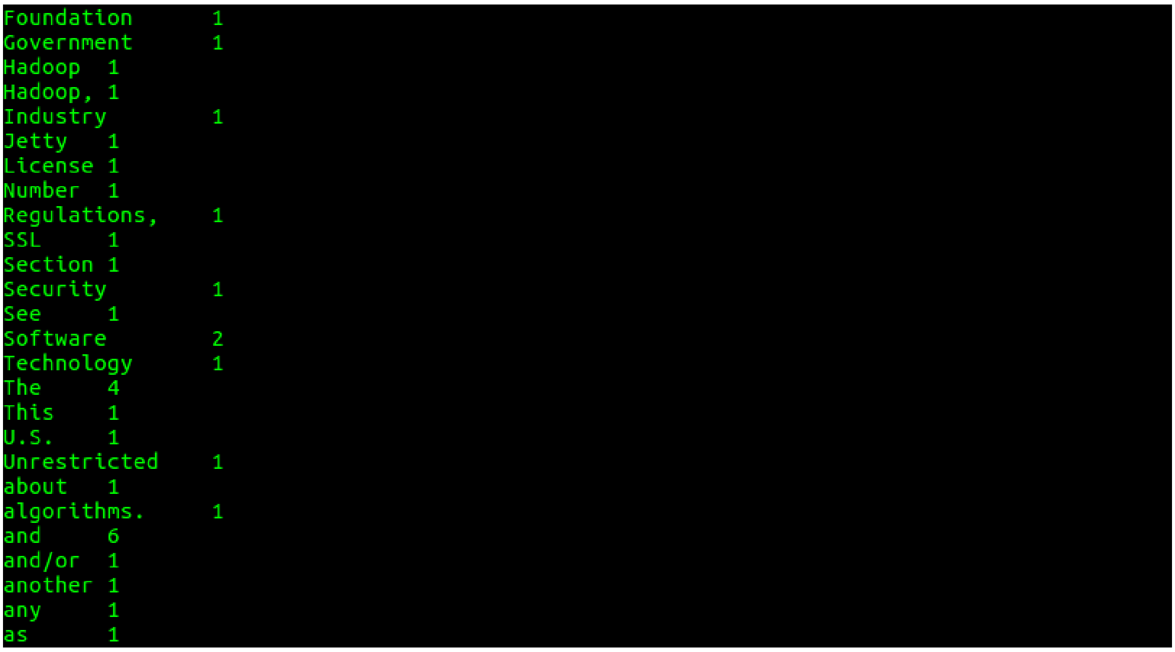
```
        File System Counters
                FILE: Number of bytes read=1836
                FILE: Number of bytes written=238401
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1466
                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
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=2629
                Total time spent by all reduces in occupied slots (ms)=3196
                Total time spent by all map tasks (ms)=2629
                Total time spent by all reduce tasks (ms)=3196
                Total vcore-milliseconds taken by all map tasks=2629
                Total vcore-milliseconds taken by all reduce tasks=3196
                Total megabyte-milliseconds taken by all map tasks=2692096
                Total megabyte-milliseconds taken by all reduce tasks=3272704
        Map-Reduce Framework
                Map input records=31
                Map output records=179
                Map output bytes=2055
                Map output materialized bytes=1836
                Input split bytes=100
                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)=95
                CPU time spent (ms)=1270
                Physical memory (bytes) snapshot=409751552
                Virtual memory (bytes) snapshot=3908456448
                Total committed heap usage (bytes)=311427072
        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
```

- Check the result of word count

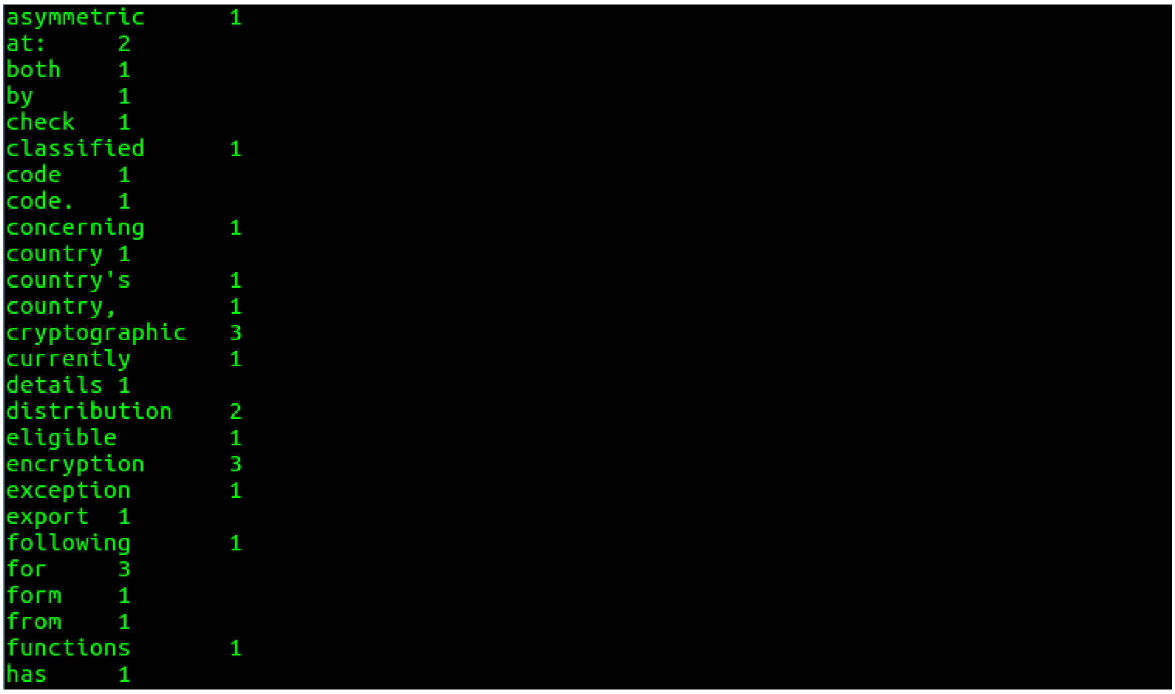
```
hadoop fs -cat /output/
```



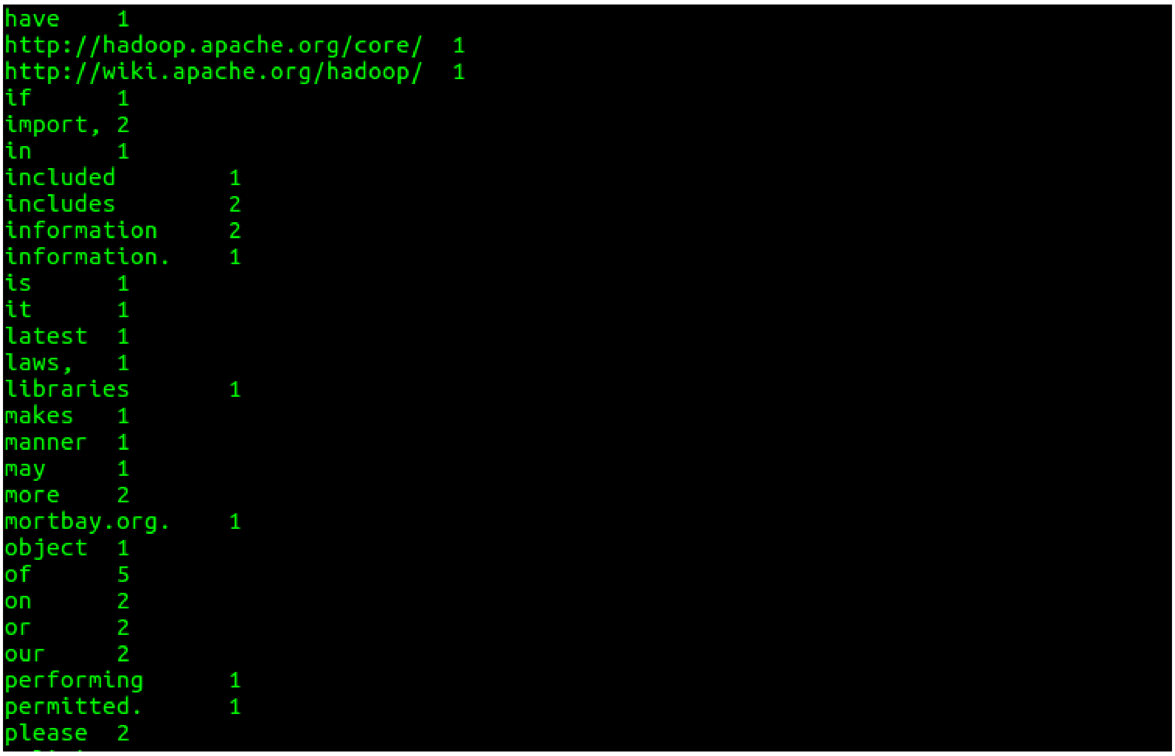
```
root@Master:/opt/tools/hadoop/share/hadoop/mapreduce# hadoop fs -cat /output/part-r-00000
(BIS),  1
(ECCN)  1
(TSU)   1
(see    1
5D002.C.1,      1
740.13) 1
<http://www.wassenaar.org/>     1
Administration  1
Apache  1
BEFORE  1
BIS     1
Bureau  1
Commerce,       1
Commodity       1
Control 1
Core    1
Department      1
ENC     1
Exception       1
Export  2
For     1
```



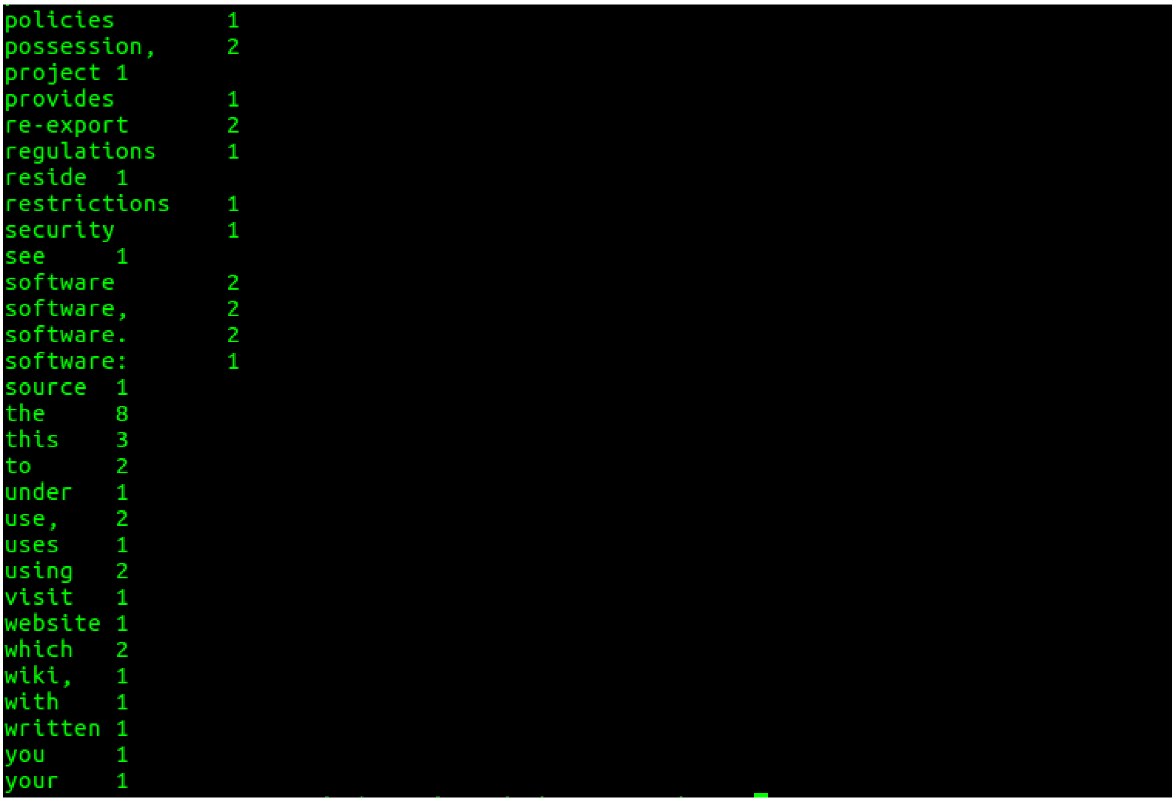
```
Foundation      1
Government      1
Hadoop  1
Hadoop, 1
Industry        1
Jetty   1
License 1
Number  1
Regulations,    1
SSL     1
Section 1
Security        1
See     1
Software        2
Technology      1
The     4
This    1
U.S.    1
Unrestricted    1
about   1
algorithms.     1
and     6
and/or  1
another 1
any     1
as      1
```



```
asymmetric      1
at:     2
both    1
by      1
check   1
classified      1
code    1
code.   1
concerning      1
country 1
country's       1
country,        1
cryptographic   3
currently       1
details 1
distribution    2
eligible        1
encryption      3
exception       1
export  1
following       1
for     3
form    1
from    1
functions       1
has     1
```



```
have    1
http://hadoop.apache.org/core/  1
http://wiki.apache.org/hadoop/  1
if      1
import, 2
in      1
included        1
includes        2
information     2
information.    1
is      1
it      1
latest  1
laws,   1
libraries       1
makes   1
manner  1
may     1
more    2
mortbay.org.    1
object  1
of      5
on      2
or      2
our     2
performing      1
permitted.      1
please  2
```

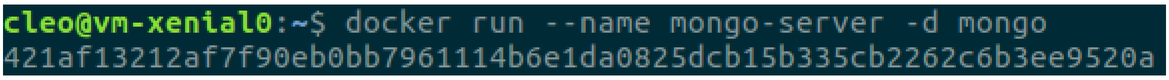


```
policies        1
possession,     2
project 1
provides        1
re-export       2
regulations     1
reside  1
restrictions    1
security        1
see     1
software        2
software,       2
software.       2
software:       1
source  1
the     8
this    3
to      2
under   1
use,    2
uses    1
using   2
visit   1
website 1
which   2
wiki,   1
with    1
written 1
you     1
your    1
```

Quantity Analysis

- Start the mongo container

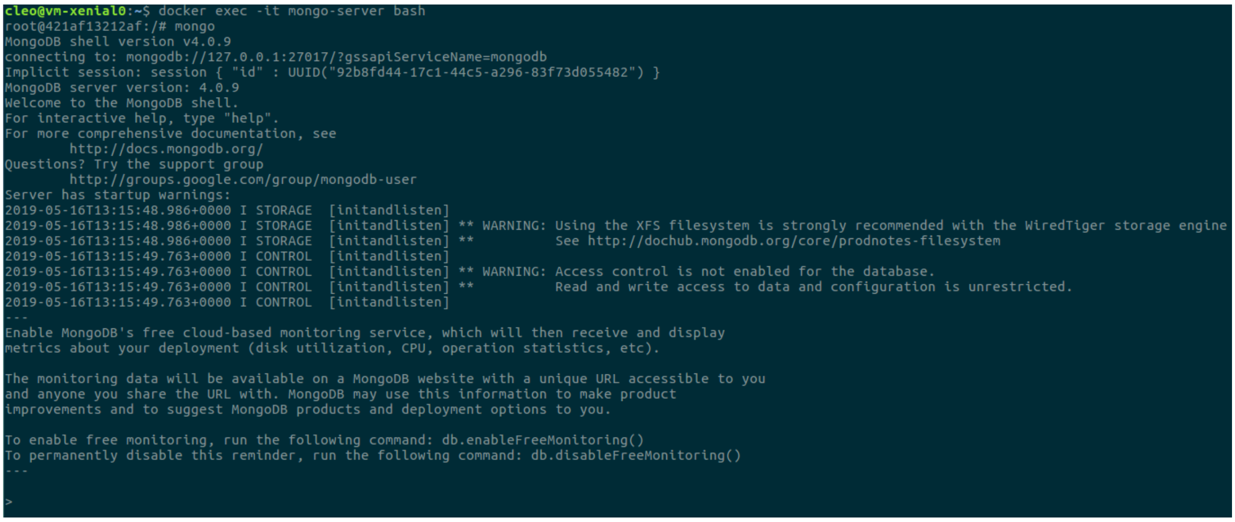
```
docker run --name mongo-server -d mongo
```



```
cleo@vm-xenial0:~$ docker run --name mongo-server -d mongo
421af13212af7f90eb0bb7961114b6e1da0825dcb15b335cb2262c6b3ee9520a
```

- Enter the command line of `mongo-server` container

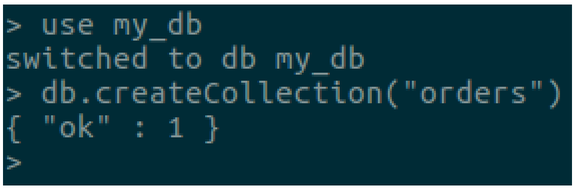
  ```
  docker exec -it mongo-server bash
  mongo
  ```

  ```
  cleo@vm-xenial0:~$ docker exec -it mongo-server bash
  root@421af13212af:/# mongo
  MongoDB shell version v4.0.9
  connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
  Implicit session: session { "id" : UUID("92b8fd44-17c1-44c5-a296-83f73d055482") }
  MongoDB server version: 4.0.9
  Welcome to the MongoDB shell.
  For interactive help, type "help".
  For more comprehensive documentation, see
          http://docs.mongodb.org/
  Questions? Try the support group
          http://groups.google.com/group/mongodb-user
  Server has startup warnings:
  2019-05-16T13:15:48.986+0000 I STORAGE  [initandlisten]
  2019-05-16T13:15:48.986+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
  2019-05-16T13:15:48.986+0000 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
  2019-05-16T13:15:49.763+0000 I CONTROL  [initandlisten]
  2019-05-16T13:15:49.763+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
  2019-05-16T13:15:49.763+0000 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
  2019-05-16T13:15:49.763+0000 I CONTROL  [initandlisten]
  ---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
  metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
  and anyone you share the URL with. MongoDB may use this information to make product
  improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
  ---

  >
  ```

- Create a new database and a collection named `orders`

  ```
  use my_db
  db.createCollection("orders")
  ```

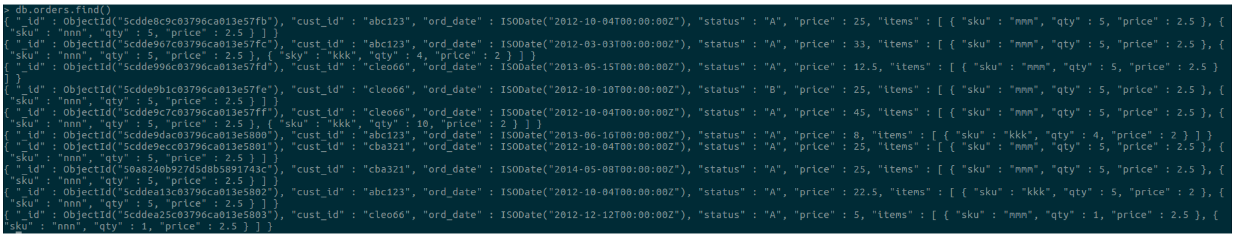
  ```
  > use my_db
  switched to db my_db
  > db.createCollection("orders")
  { "ok" : 1 }
  >
  ```

- Insert some documents into collection `orders`

  The document are following prototype:

  ```
  {
     _id: ObjectId("50a8240b927d5d8b5891743c"),
     cust_id: "abc123",
     ord_date: new Date("Oct 04, 2012"),
     status: 'A',
     price: 25,
     items: [ { sku: "mmm", qty: 5, price: 2.5 },
              { sku: "nnn", qty: 5, price: 2.5 } ]
  }
  ```

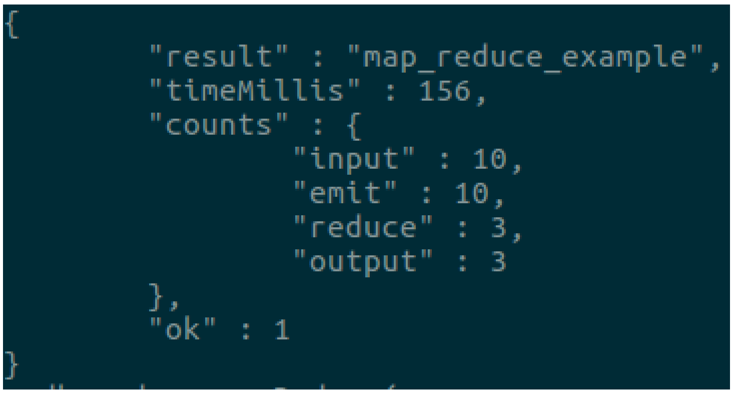
  Here, I just insert 10 documents into collection `orders`:



```
> db.orders.find()
{ "_id" : ObjectId("5cdde8c9c03796ca013e57fb"), "cust_id" : "abc123", "ord_date" : ISODate("2012-10-04T00:00:00Z"), "status" : "A", "price" : 25, "items" : [ { "sku" : "mmm", "qty" : 5, "price" : 2.5 }, {
 "sku" : "nnn", "qty" : 5, "price" : 2.5 } ] }
{ "_id" : ObjectId("5cdde967c03796ca013e57fc"), "cust_id" : "abc123", "ord_date" : ISODate("2012-03-03T00:00:00Z"), "status" : "A", "price" : 33, "items" : [ { "sku" : "mmm", "qty" : 5, "price" : 2.5 }, {
 "sku" : "nnn", "qty" : 5, "price" : 2.5 }, { "sky" : "kkk", "qty" : 4, "price" : 2 } ] }
{ "_id" : ObjectId("5cdde996c03796ca013e57fd"), "cust_id" : "cleo66", "ord_date" : ISODate("2013-05-15T00:00:00Z"), "status" : "A", "price" : 12.5, "items" : [ { "sku" : "mmm", "qty" : 5, "price" : 2.5 }
] }
{ "_id" : ObjectId("5cdde9b1c03796ca013e57fe"), "cust_id" : "cleo66", "ord_date" : ISODate("2012-10-10T00:00:00Z"), "status" : "B", "price" : 25, "items" : [ { "sku" : "mmm", "qty" : 5, "price" : 2.5 }, {
 "sku" : "nnn", "qty" : 5, "price" : 2.5 } ] }
{ "_id" : ObjectId("5cdde9c7c03796ca013e57ff"), "cust_id" : "cleo66", "ord_date" : ISODate("2012-10-04T00:00:00Z"), "status" : "A", "price" : 45, "items" : [ { "sku" : "mmm", "qty" : 5, "price" : 2.5 }, {
 "sku" : "nnn", "qty" : 5, "price" : 2.5 }, { "sku" : "kkk", "qty" : 10, "price" : 2 } ] }
{ "_id" : ObjectId("5cdde9dac03796ca013e5800"), "cust_id" : "abc123", "ord_date" : ISODate("2013-06-16T00:00:00Z"), "status" : "A", "price" : 8, "items" : [ { "sku" : "kkk", "qty" : 4, "price" : 2 } ] }
{ "_id" : ObjectId("5cdde9ecc03796ca013e5801"), "cust_id" : "cba321", "ord_date" : ISODate("2012-10-04T00:00:00Z"), "status" : "A", "price" : 25, "items" : [ { "sku" : "mmm", "qty" : 5, "price" : 2.5 }, {
 "sku" : "nnn", "qty" : 5, "price" : 2.5 } ] }
{ "_id" : ObjectId("50a8240b927d5d8b5891743c"), "cust_id" : "cba321", "ord_date" : ISODate("2014-05-08T00:00:00Z"), "status" : "A", "price" : 25, "items" : [ { "sku" : "mmm", "qty" : 5, "price" : 2.5 }, {
 "sku" : "nnn", "qty" : 5, "price" : 2.5 } ] }
{ "_id" : ObjectId("5cddea13c03796ca013e5802"), "cust_id" : "abc123", "ord_date" : ISODate("2012-10-04T00:00:00Z"), "status" : "A", "price" : 22.5, "items" : [ { "sku" : "kkk", "qty" : 5, "price" : 2 }, {
 "sku" : "nnn", "qty" : 5, "price" : 2.5 } ] }
{ "_id" : ObjectId("5cddea25c03796ca013e5803"), "cust_id" : "cleo66", "ord_date" : ISODate("2012-12-12T00:00:00Z"), "status" : "A", "price" : 5, "items" : [ { "sku" : "mmm", "qty" : 1, "price" : 2.5 }, {
"sku" : "nnn", "qty" : 1, "price" : 2.5 } ] }
```

- Get the Total Price Per Customer

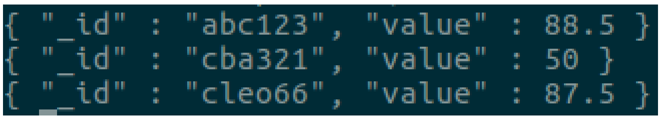
```
var mapFunction1 = function() {
                     emit(this.cust_id, this.price);
                 };

var reduceFunction1 = function(keyCustId, valuesPrices) {
                        return Array.sum(valuesPrices);
                    };

db.orders.mapReduce(
                   mapFunction1,
                   reduceFunction1,
                   { out: "map_reduce_example" }
                 )
```



```
{
        "result" : "map_reduce_example",
        "timeMillis" : 156,
        "counts" : {
                "input" : 10,
                "emit" : 10,
                "reduce" : 3,
                "output" : 3
        },
        "ok" : 1
}
```

Check the documents of result:

```
db.orders.mapReduce(
                   mapFunction1,
                   reduceFunction1,
                   { out: "map_reduce_example" }
                 ).find()
```



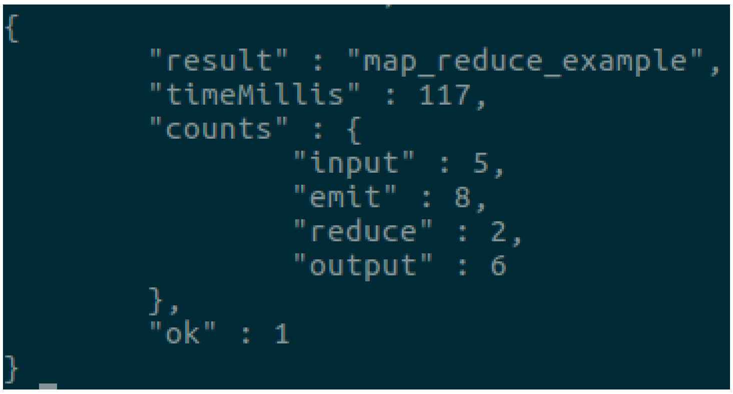
```
{ "_id" : "abc123", "value" : 88.5 }
{ "_id" : "cba321", "value" : 50 }
{ "_id" : "cleo66", "value" : 87.5 }
```

In the process of MapReduce, there are 10 documents were maped and reduced, and the result was seperated into 3 groups.

- Calculate order and total quantity with average quantity per item

  In this example, you will perform a map-reduce operation on the orders collection for all documents that have an ord_date value greater than 10/05/2012. The operation groups by the item.sku field, and calculates the number of orders and the total quantity ordered for each sku. The operation concludes by calculating the average quantity per order for each sku value.

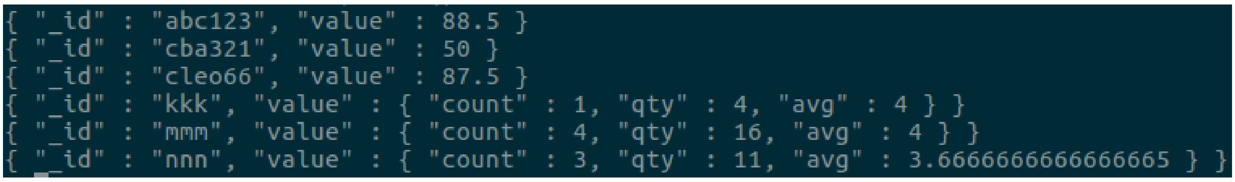
```
var mapFunction2 = function() {
                     for (var idx = 0; idx < this.items.length; idx++) {
                         var key = this.items[idx].sku;
                         var value = {
                                       count: 1,
                                       qty: this.items[idx].qty
                                     };
                         emit(key, value);
                     }
                  };

var reduceFunction2 = function(keySKU, countObjVals) {
                   reducedVal = { count: 0, qty: 0 };

                   for (var idx = 0; idx < countObjVals.length; idx++) {
                       reducedVal.count += countObjVals[idx].count;
                       reducedVal.qty += countObjVals[idx].qty;
                   }

                   return reducedVal;
                };

var finalizeFunction2 = function (key, reducedVal) {

                     reducedVal.avg = reducedVal.qty/reducedVal.count;

                     return reducedVal;

                  };

db.orders.mapReduce( mapFunction2,
                   reduceFunction2,
                   {
                     out: { merge: "map_reduce_example" },
                     query: { ord_date:
                                { $gt: new Date('10/05/2012') }
                            },
                     finalize: finalizeFunction2
                   }
                 )
```



```
{
        "result" : "map_reduce_example",
        "timeMillis" : 117,
        "counts" : {
                "input" : 5,
                "emit" : 8,
                "reduce" : 2,
                "output" : 6
        },
        "ok" : 1
}
```

Check the documents of result:

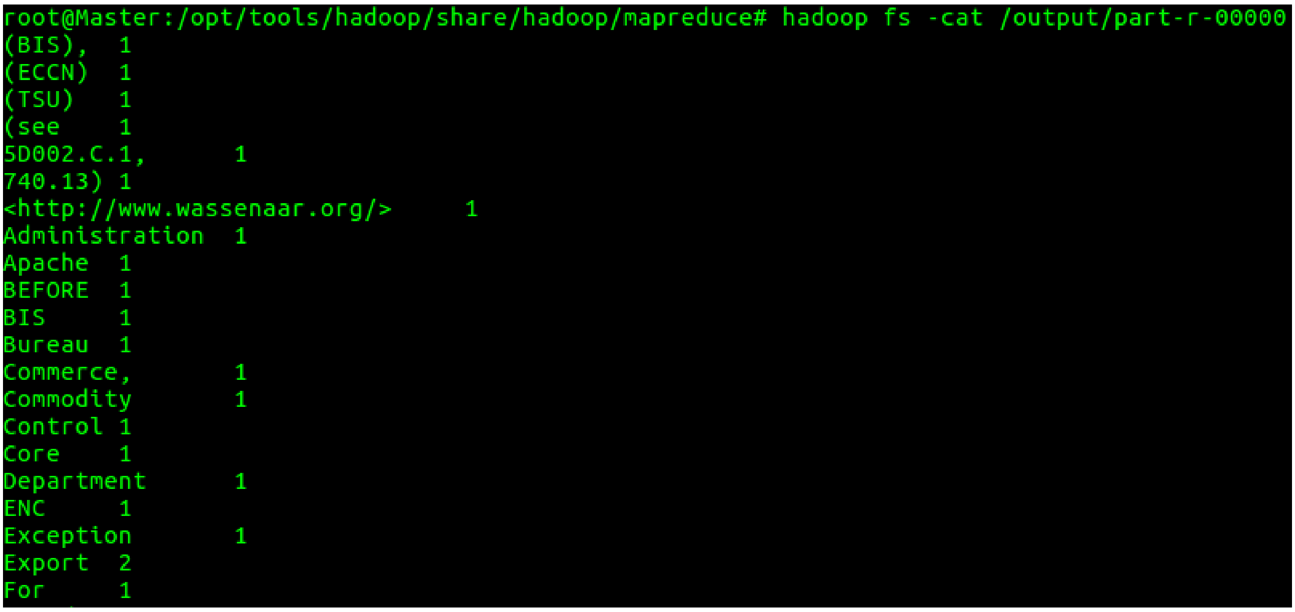
```
db.orders.mapReduce( mapFunction2,
                     reduceFunction2,
                     {
                       out: { merge: "map_reduce_example" },
                       query: { ord_date:
                                  { $gt: new Date('10/05/2012') }
                              },
                       finalize: finalizeFunction2
                     }
                   ).find()
```



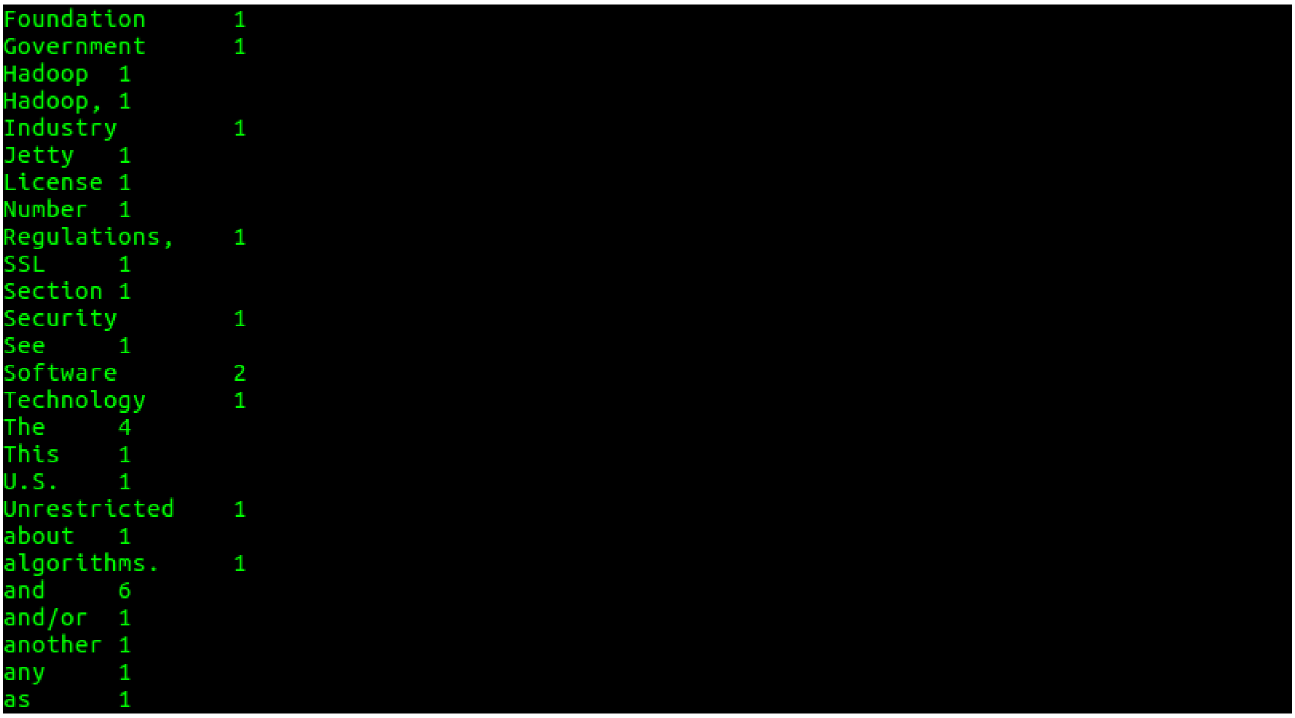
```
{ "_id" : "abc123", "value" : 88.5 }
{ "_id" : "cba321", "value" : 50 }
{ "_id" : "cleo66", "value" : 87.5 }
{ "_id" : "kkk", "value" : { "count" : 1, "qty" : 4, "avg" : 4 } }
{ "_id" : "mmm", "value" : { "count" : 4, "qty" : 16, "avg" : 4 } }
{ "_id" : "nnn", "value" : { "count" : 3, "qty" : 11, "avg" : 3.6666666666666665 } }
```

实验数据或结果

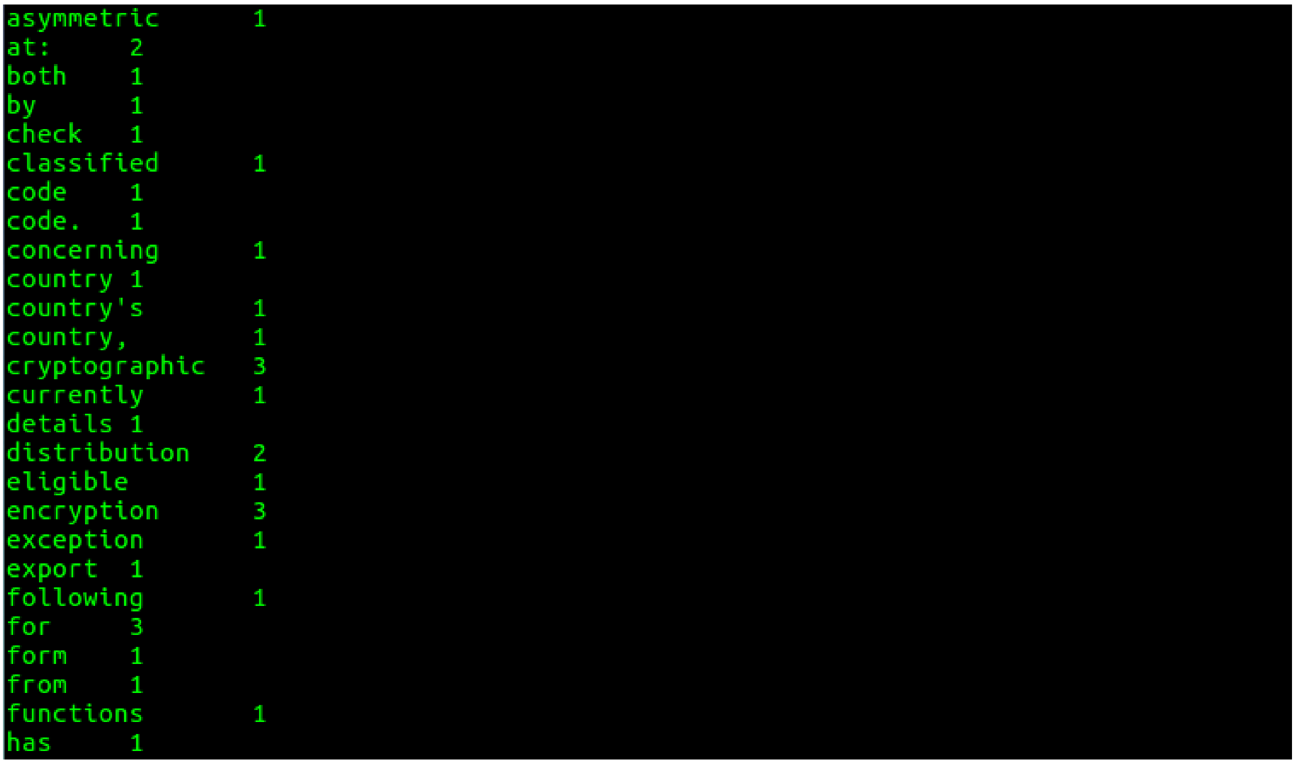
Word Count



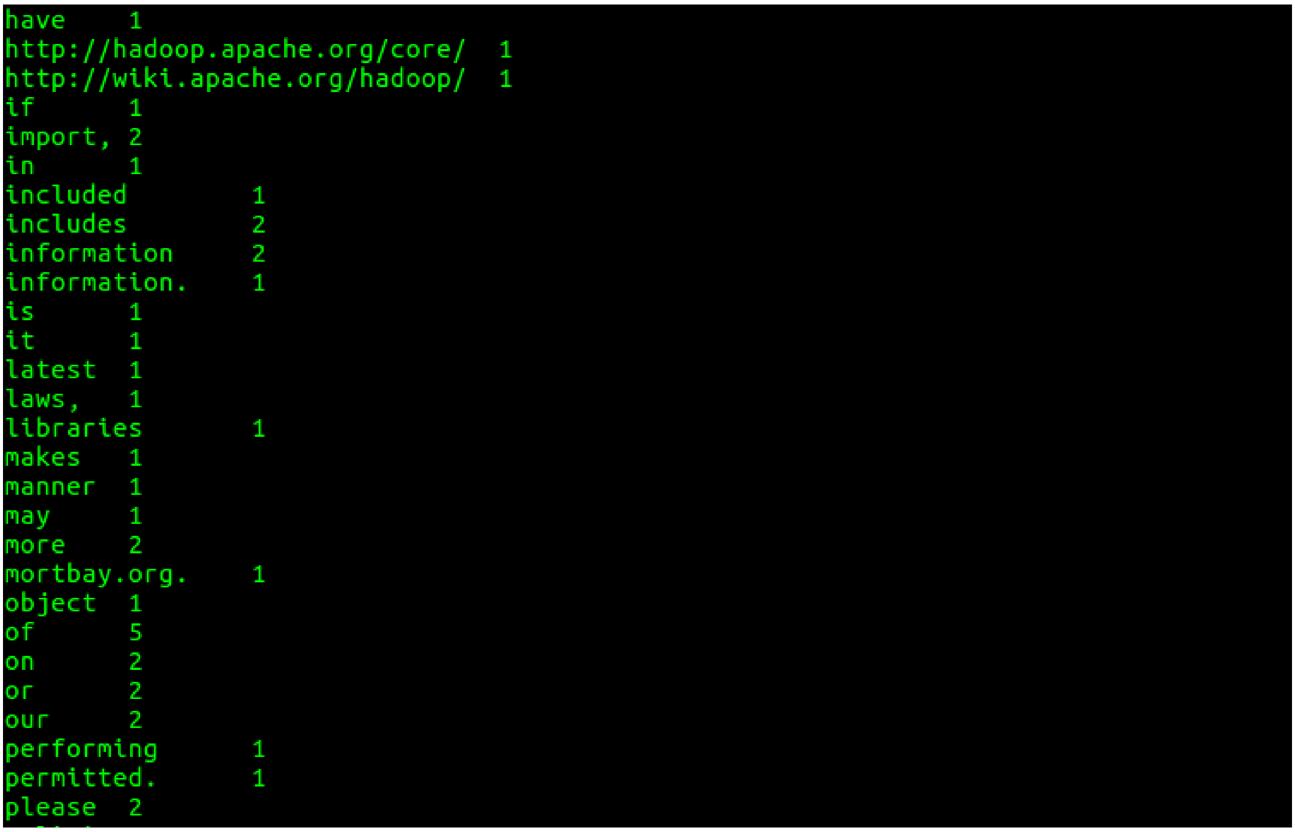
```
root@Master:/opt/tools/hadoop/share/hadoop/mapreduce# hadoop fs -cat /output/part-r-00000
(BIS),  1
(ECCN)  1
(TSU)   1
(see    1
5D002.C.1,      1
740.13) 1
<http://www.wassenaar.org/>     1
Administration  1
Apache  1
BEFORE  1
BIS     1
Bureau  1
Commerce,       1
Commodity       1
Control 1
Core    1
Department      1
ENC     1
Exception       1
Export  2
For     1
```



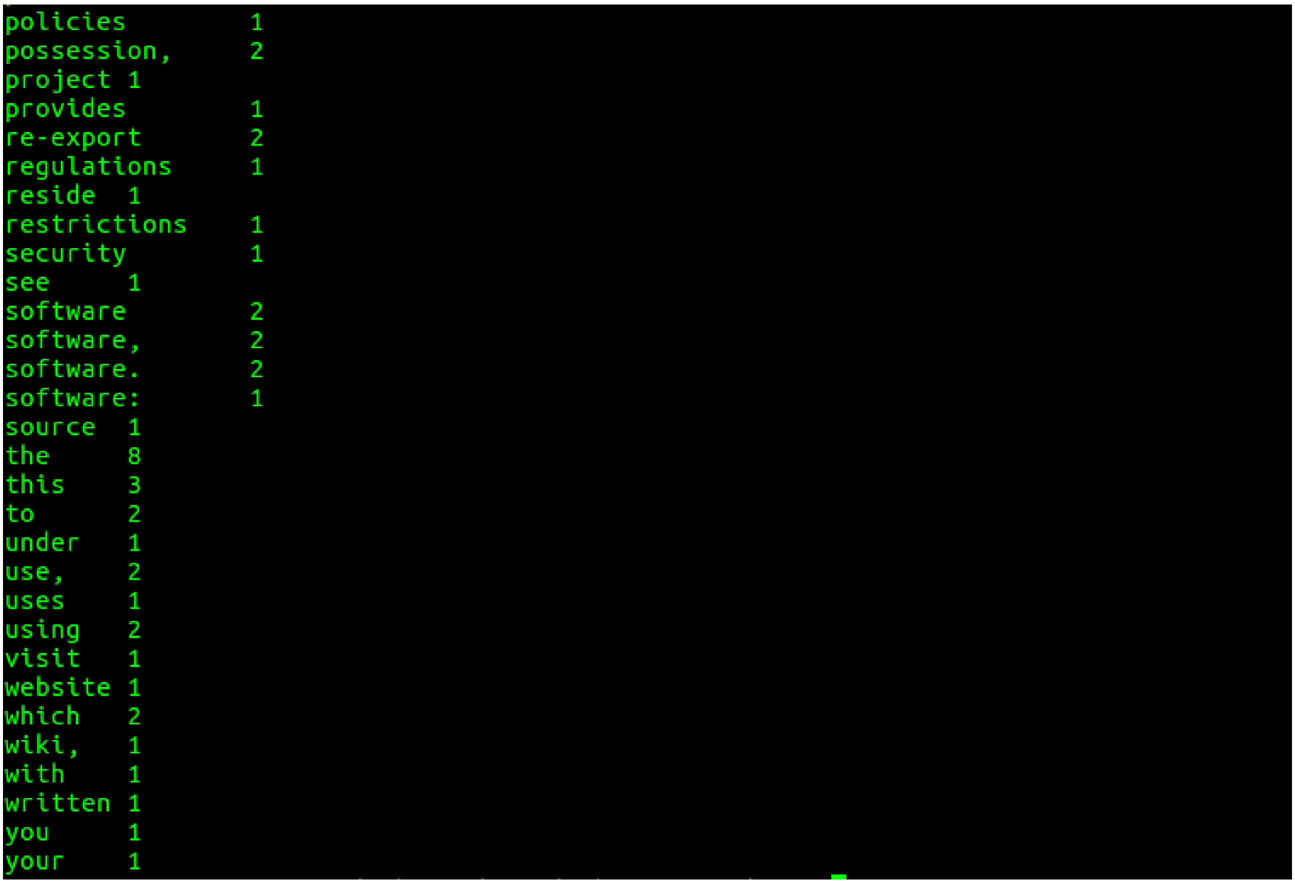
```
Foundation      1
Government      1
Hadoop  1
Hadoop, 1
Industry        1
Jetty   1
License 1
Number  1
Regulations,    1
SSL     1
Section 1
Security        1
See     1
Software        2
Technology      1
The     4
This    1
U.S.    1
Unrestricted    1
about   1
algorithms.     1
and     6
and/or  1
another 1
any     1
as      1
```



```
asymmetric      1
at:     2
both    1
by      1
check   1
classified      1
code    1
code.   1
concerning      1
country 1
country's       1
country,        1
cryptographic   3
currently       1
details 1
distribution    2
eligible        1
encryption      3
exception       1
export  1
following       1
for     3
form    1
from    1
functions       1
has     1
```

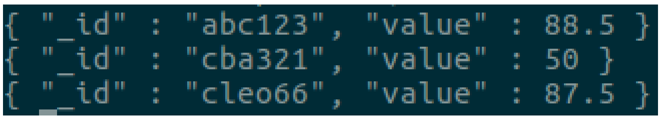


```
have    1
http://hadoop.apache.org/core/  1
http://wiki.apache.org/hadoop/  1
if      1
import, 2
in      1
included        1
includes        2
information     2
information.    1
is      1
it      1
latest  1
laws,   1
libraries       1
makes   1
manner  1
may     1
more    2
mortbay.org.    1
object  1
of      5
on      2
or      2
our     2
performing      1
permitted.      1
please  2
```

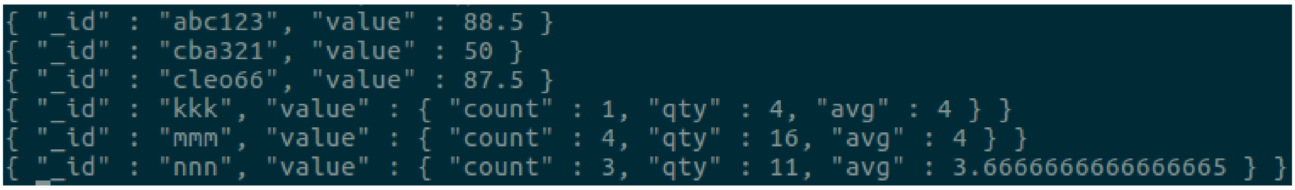


```
policies        1
possession,     2
project 1
provides        1
re-export       2
regulations     1
reside  1
restrictions    1
security        1
see     1
software        2
software,       2
software.       2
software:       1
source  1
the     8
this    3
to      2
under   1
use,    2
uses    1
using   2
visit   1
website 1
which   2
wiki,   1
with    1
written 1
you     1
your    1
```

Quantity Analysis



```
{ "_id" : "abc123", "value" : 88.5 }
{ "_id" : "cba321", "value" : 50 }
{ "_id" : "cleo66", "value" : 87.5 }
```



```
{ "_id" : "abc123", "value" : 88.5 }
{ "_id" : "cba321", "value" : 50 }
{ "_id" : "cleo66", "value" : 87.5 }
{ "_id" : "kkk", "value" : { "count" : 1, "qty" : 4, "avg" : 4 } }
{ "_id" : "mmm", "value" : { "count" : 4, "qty" : 16, "avg" : 4 } }
{ "_id" : "nnn", "value" : { "count" : 3, "qty" : 11, "avg" : 3.6666666666666665 } }
```

实验思考

Hadoop

Hadoop is an open source framework that can write and run distributed applications to process large-scale data. It is designed for offline and large-scale data analysis. It is not suitable for the online transaction processing mode of random reading and writing of several records. Hadoop = HDFS (file system, data storage technology related) + Mapreduce (data processing), Hadoop's data source can be any form, in processing semi-structured and unstructured data than relational databases have better performance, more flexible processing capacity, regardless of any data form will eventually be converted into key/value, key/value is the basic data unit. Functional Mapreduce instead of SQL, SQL is a query statement, while Mapreduce uses scripts and code. For relational databases, Hadoop, which is used to SQL, has an open source tool called Hive instead.

HDFS

HDFS is the primary distributed storage used by Hadoop applications. A HDFS cluster primarily consists of a NameNode that manages the file system metadata and DataNodes that store the actual data. The HDFS Architecture Guide describes HDFS in detail. This user guide primarily deals with the interaction of users and administrators with HDFS clusters. The HDFS architecture diagram depicts basic interactions among NameNode, the DataNodes, and the clients. Clients contact NameNode for file metadata or file modifications and perform actual file I/O directly with the DataNodes.

The following are some of the salient features that could be of interest to many users.

- Hadoop, including HDFS, is well suited for distributed storage and distributed processing using commodity hardware. It is fault tolerant, scalable, and extremely simple to expand. MapReduce, well known for its simplicity and applicability for large set of distributed applications, is an integral part of Hadoop.

- HDFS is highly configurable with a default configuration well suited for many installations. Most of the time, configuration needs to be tuned only for very large clusters.

- Hadoop is written in Java and is supported on all major platforms.

- Hadoop supports shell-like commands to interact with HDFS directly.

- The NameNode and Datanodes have built in web servers that makes it easy to check current status of the cluster.

- New features and improvements are regularly implemented in HDFS. The following is a subset of useful features in HDFS:

  - File permissions and authentication.
  - Rack awareness: to take a node’s physical location into account while scheduling tasks and allocating storage.
  - Safemode: an administrative mode for maintenance.
  - fsck: a utility to diagnose health of the file system, to find missing files or blocks.
  - fetchdt: a utility to fetch DelegationToken and store it in a file on the local system.
  - Balancer: tool to balance the cluster when the data is unevenly distributed among DataNodes.
  - Upgrade and rollback: after a software upgrade, it is possible to rollback to HDFS’ state before the upgrade in case of unexpected problems.
  - Secondary NameNode: performs periodic checkpoints of the namespace and helps keep the size of file containing log of HDFS modifications within certain limits at the NameNode.
  - Checkpoint node: performs periodic checkpoints of the namespace and helps minimize the size of the log stored at the NameNode containing changes to the HDFS. Replaces the role previously filled by the Secondary NameNode, though is not yet battle hardened. The NameNode allows multiple Checkpoint nodes simultaneously, as long as there are no Backup nodes registered with the system.
  - Backup node: An extension to the Checkpoint node. In addition to checkpointing it also receives a stream of edits from the NameNode and maintains its own in-memory copy of the namespace, which is always in sync with the active NameNode namespace state. Only one Backup node may be registered with the NameNode at once.

MapReduce

MapReduce is a software architecture proposed by Google for parallel computing of large data sets (larger than 1TB). The concepts "Map" and "Reduce" and their main ideas are borrowed from functional programming languages, as well as features borrowed from vector programming languages.

Current software implementations specify a Map function to map a set of key-value pairs to a new set of key-value pairs and a concurrent Reduce (induction) function to ensure that each of the key-value pairs of all mappings shares the same key group.

Simply put, a mapping function is to specify each element of a conceptual list of independent elements (for example, a list of test scores). (For example, someone found that all students'scores were overestimated, and he could define a "minus one" mapping function to correct the error.) In fact, each element is operated on independently, and the original list is not changed because a new list is created here to hold the new answers. That is to say, Map operations can be highly parallel, which is very useful for applications requiring high performance and for requirements in the field of parallel computing.

The inductive operation refers to the appropriate merging of elements in a list (continue to look at the previous example, if someone wants to know what to do with the average score of the class? He can define an inductive function to halve the list by adding odd or even elements in the list to their adjacent elements, so that the recursive operation until there is only one element in the list, and then divide the element by the number, and get the average score. Although it is not as parallel as mapping functions, induction functions are also useful in highly parallel environments because induction always has a simple answer and large-scale operations are relatively independent.

MapReduce achieves reliability by distributing large-scale operations to each node on the network; each node periodically reports back the completed work and status updates. If a node stays silent for more than a preset time interval, the primary node (similar to the primary server in the Google file system) records the status of the node as dead and sends the data allocated to the node to other nodes. Each operation uses the indivisible operation of named files to ensure that no collision between parallel threads occurs; when files are renamed, the system may copy them to another name other than the task name.

Inductive operations work in a similar way, but because inductive operations are poorly parallel, the primary node will try to schedule inductive operations on one node, or as close as possible to the data needed to operate on the node; this feature can meet the needs of Google, because they have enough bandwidth, their internal network does not have so many machines.

参考资料

- https://www.jianshu.com/p/30604c820e9d?tdsourcetag=s_pctim_aiomsg
- https://docs.mongodb.com/manual/tutorial/map-reduce-examples/
- http://www.runoob.com/mongodb/mongodb-map-reduce.html
- https://www.jianshu.com/p/30604c820e9d?tdsourcetag=s_pctim_aiomsg
- http://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html
- https://juejin.im/post/5af3f57cf265da0b7e0c49c7