Hadoop Cluster Installation Manual

2022.08.05

Chen Qi

a. Hadoop Preset parameters and build structure

i. The Hadoop cluster will be built with one master and two slaves. Obtain ip address through `/sbin/ifconfig | grep 'inet addr:'` for further operations.

ii. IP Address Summarization

```
linux-adev:~ # /sbin/ifconfig | grep 'inet addr:'
          inet addr:10.2.1.155  Bcast:10.2.1.255  Mask:255.255.255.0
          inet addr:127.0.0.1  Mask:255.0.0.0
```

1.1 Master-IP

```
linux-adev:~ # /sbin/ifconfig | grep 'inet addr:'
          inet addr:10.2.1.156  Bcast:10.2.1.255  Mask:255.255.255.0
          inet addr:127.0.0.1  Mask:255.0.0.0
```

1.2 Slave1-IP

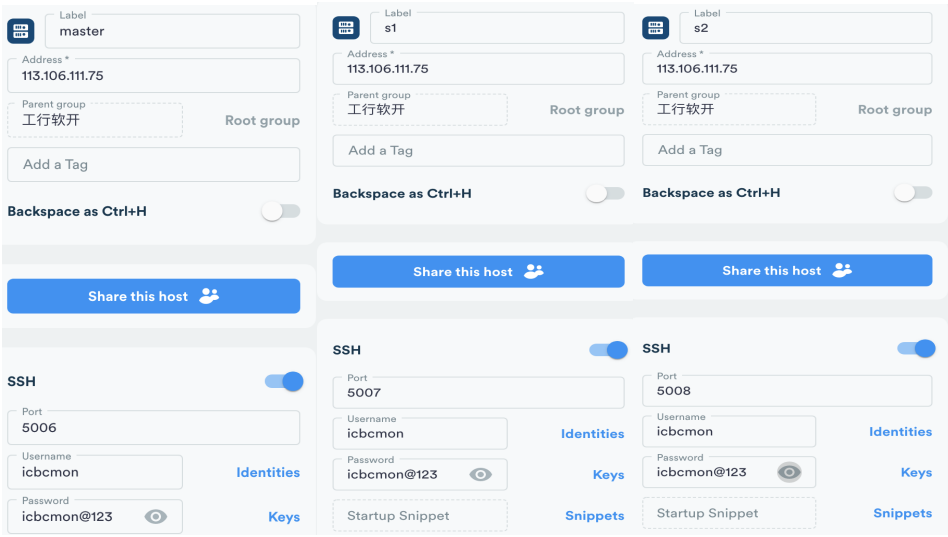
```
linux-adev:~ # /sbin/ifconfig | grep 'inet addr:'
          inet addr:10.2.1.157  Bcast:10.2.1.255  Mask:255.255.255.0
          inet addr:127.0.0.1  Mask:255.0.0.0
```

1.3 Slave2-IP 地址信息

iii. Hadoop Cluster Structure：

IP	Roles	Nodes
10.2.1.155	master	Namenode, Datanode
10.2.1.156	slave1	Datanode
10.2.1.157	slave2	Datanode

iv. Labeling each of these machines through SSH port; (Log in info is provided by ICBC)



2.1~2.3 SSH Configuration

b. Create an Hadoop user account

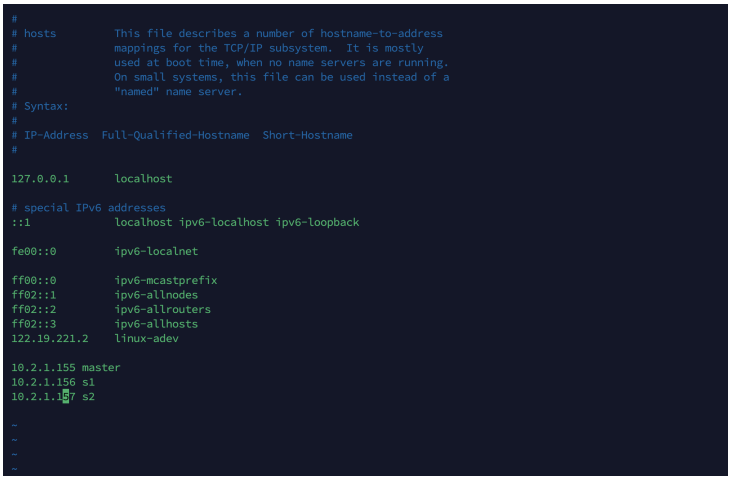
i. useradd -u 501 -g users -d /home/hadoop -s /bin/bash hadoop

ii. Under direction /home, create an Hadoop folder for further operation

1. mkdir /home/hadoop
2. chown -R hadoop:users /home/Hadoop
3. passwd hadoop (For connivence, password matches the username)

c. Matching Nodes

i. Assign nodes to three different IP address through via "# vi /etc/hosts;" Use INSERT commend to for editing. After edition, use ESC and wq commend to save the alteration and quit the vi editor.



```
#
# hosts         This file describes a number of hostname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.
# Syntax:
#
# IP-Address  Full-Qualified-Hostname  Short-Hostname
#

127.0.0.1       localhost

# special IPv6 addresses
::1             localhost ipv6-localhost ipv6-loopback

fe00::0         ipv6-localnet

ff00::0         ipv6-mcastprefix
ff02::1         ipv6-allnodes
ff02::2         ipv6-allrouters
ff02::3         ipv6-allhosts
122.19.221.2    linux-adev

10.2.1.155 master
10.2.1.156 s1
10.2.1.157 s2

~
~
~
~
```

3 Assignments of IP Nodes

d. Commuting Three Machines (Log in without passwords)

i. Through su hadoop command, make sure we are operating under the user "Hadoop"

ii. Through ssh-keygen -t rsa command, generates a local key in each of three machines.

```
master:~ # su - hadoop
hadoop@master:~> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Created directory '/home/hadoop/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:d14g+KkHXXoobTeVdrDky6nETSbuufSzSBFKp/vdcfc hadoop@master
The key's randomart image is:
+---[RSA 2048]----+
|              o  |
|         .   o + |
|        . o * O .|
|         = % @ + |
|        S % X *  |
|         * O *   |
|        . o B  .o|
|         . + =..=|
|            + +oE|
+----[SHA256]-----+
```

4.1 Results of Key Generation

iii. Under the direction of .ssh, create an folder named "authorized_keys" to store the keys：

1. cd ~/.ssh/
2. touch ~/.ssh/authorized_keys
3. cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

iv. Repeat the operations on S1 and S2; Then, send the local Keys of each machines to the Master machine：

1. Input following codes in S1 ：

scp -P 10155

~/.ssh/id_rsa.pub hadoop@master:~/.ssh/s1.id_rsa.pub

2. Input following codes in S2：

scp -P 10155

~/.ssh/id_rsa.pub hadoop@master:~/.ssh/s2.id_rsa.pub

v. Check if the keys are successfully sent to the Master machine. When succussed, store the keys in the authorized keys folder.

```
hadoop@master:~/.ssh> ls
authorized_keys  config  id_rsa  id_rsa.pub  s1.id_rsa.pub  s2.id_rsa.pub
```

```
hadoop@master:~/.ssh> cat ~/.ssh/s1.id_rsa.pub >> ~/.ssh/authorized_keys
hadoop@master:~/.ssh> cat ~/.ssh/s2.id_rsa.pub >> ~/.ssh/authorized_keys
```

4.2~4.3 Keys Storage Method

vi. Sent the keys of Master to S1 and S2 as well, so that Local keys in any of the three machines are all stored in all of our machines. Also, make sure the keys are redirected to the "authorized_keys" folder.

scp -P 10156

~/.ssh/id_rsa.pub hadoop@s1:~/.ssh/master.id_rsa.pub

scp -P 10157

~/.ssh/id_rsa.pub hadoop@s2:~/.ssh/master.id_rsa.pub

```
hadoop@master:~/.ssh> cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDm1HysSKcAfCRcBkBn2Lf/Y6FqmUzNq78WMFuYuwucHP2ca00eiNeSM4jyPqePOU4YDt6LvRb4jpmD+KnMoYt
Ag4Vy/+LX0Xu/Ji24iVSaAC9wl1mPNVRsWzruOR3VVBaJsBLMaSEwEE2F4Wc0dCDjK6EEsB0d8CRXDzE6mXAXOpie3xAigKgnxetZXV5E5pMNeibMPSYc1A4qLT
O2s12SFlVGix6UsVT6SzkW6iGRFPWonaYI8wIaAkxF5VrtDiGMZE5F7hxujOtuZtBAmM2tZOk7tw47us1oWB3nooXX1brAtrGSOEq8V6Nzw/ZqTpajULGm+M1Zm
HvbyV+twI2f hadoop@master
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDDW+Tz7PrvyuBZoKlmm8QQ/HuOV5BKgLBN7VijNQOkxTUa5uc5Q2eB+MdFybRbsECijUf+eqtP32JdhAdUgdy
3irR86dVyrbSCrjPjLwpc3rmBE4/E7Tp97tAMEmlBkMVJNpnUN93GFNhnj4rFjSCVVnSyfPzSpkbxThr9Tub3GdxC8XWGEOABWZ5UstNsC3F9LC3Iv+Ke+Ojm1k
oPOj1TAXw/86bSbKdUTpA/c+kyXAGH5IaMTzkyqqIeL0C7g0bk4l9nLnAo1QjJSPmXILt6Gwy6uA25f268J1pa1JFoqIGLRfNid1LrtDaFt6YqMroF/6Gzfwjpj
ctFEVwjKrgT hadoop@s1
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDNK+23SnuptjwMlf/wB7A6JWs39mJrjvYjhD7yZZCAuusgnXFW87s79hb3B4nQ78/gNov7UCRBtlUcgiWcPcs
slYIjJakNq1IUJwXY6XT/ZPoms0Gl36TmZ5DZEa2hUm2et2s74/d/UCjRurEaMC6lJlx7zESm7omE80Z09PsDdHNrreRZc3OMdZYyyrs2tB1ojyMnNoW9dLF0VS
6IegzDq+PkpqEb9cBQW+GNyXNWNR3fVhuE/iOAZ/GUSlCkDgRLpbiV2+GkH0/CX3oLZdQU0pGK3uZFFIbldaG88X6SVLyLoY738v/ShnOVQmK7fNiQHLIjHa58b
N16gPzSmbDR hadoop@s2 _
```

4.4 Keys Storage Results in S2

vii. Through ssh commands to make sure all machines can be logged in without password.

```
hadoop@master:~> ssh -p 10156 s1
-----------------------------------------------------------
ATTENTION:Please quit if maintainancing this system isnot
Your duty,thank you!
This system has been controled by the unified logging on system.
It is forbidden to access through other channels,plesase quit!
-----------------------------------------------------------
Last login: Sat Jul 30 06:39:45 2022 from 10.2.1.155
hadoop@s1:~> exit
logout
Connection to s1 closed.
hadoop@master:~> ssh -p 10157 s2
-----------------------------------------------------------
ATTENTION:Please quit if maintainancing this system isnot
Your duty,thank you!
This system has been controled by the unified logging on system.
It is forbidden to access through other channels,plesase quit!
-----------------------------------------------------------
Last login: Sat Jul 30 06:39:56 2022 from 10.2.1.155
hadoop@s2:~> exit
logout
Connection to s2 closed.
hadoop@master:~>
```

5.1-5.3 Successful Commuting all Machines

e. Load in Hadoop and JDK Packages

i. Download Installation Pacakgaes：

hadoop-2.7.7.tar.gz	218.7 MB	GZip archive
jdk-8u171-linux-x64.tar.gz	190.9 MB	GZip archive

6.1 Packages List

ii. Send installation packages to our target machines.

cd /home/hadoop

I used sftp tools for the file transaction here, the command which do the same jobs are listed below：

sudo scp

/Users/wcooper/Downloads/hadoop-2.7.7.tar.gz root@xxx.xxx.xxx.xx:/home/users/指令上传。

```
master:/home/hadoop # ls
.bash_history  .ssh  .viminfo  hadoop-2.7.7  hadoop-2.7.7.tar.gz  jdk-8u171-linux-x64.tar.gz
```

6.2 File List

iii. Unzip the folders in each machine with the following commands：

tar -zxvf hadoop-2.7.7.tar.gz

tar -zxvf jdk-8u171-linux-x64.tar.gz

iv. Configure the bash_profileas the picture shown below. Make sure to do it on every one of the machines.

touch .bash_profile

vim .bash_profile

```
master:/home/hadoop # cat .bash_profile
# Get the liases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export JAVA_HOME=/home/hadoop/jdk1.8.0_171
export PATH=$JAVA_HOME/bin:$PATH
```

7.1 .bash_profile Configuration

Use source .bash_profile to save and activate the configuration. Then, use java -version and echo $JAVA_HOME check if the configuration is successful：

```
master:/home/hadoop # source .bash_profile
master:/home/hadoop # java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
master:/home/hadoop # echo $JAVA_HOME
/home/hadoop/jdk1.8.0_171
master:/home/hadoop #
```

7.2 .bash_profile Results

f. Configuring Hadoop

i. First, redirect to folder hadoop-2.7.7. IMPORTANT: all of the following commands are excuted on the master machine, unless specify otherwise.

cd /home/hadoop/hadoop-2.7.7/etc/hadoop

```
master:/home/hadoop/hadoop-2.7.7/etc # cd /home/hadoop/hadoop-2.7.7/etc/hadoop
master:/home/hadoop/hadoop-2.7.7/etc/hadoop #
```

8.1 Hadoop Direction

ii. Create the following folders for further operations：

/home/hadoop/hadoop-2.7.7/tmp

/home/hadoop/hadoop-2.7.7/hdfs

/home/hadoop/hadoop-2.7.7/hdfs/name

/home/hadoop/hadoop-2.9.1/hdfs/data

iii. Through vim yarn-env.sh, configure $JAVA_HOME parameter to the local java path (where the Java is installed locally).

```
# some Java parameters
# export JAVA_HOME=/home/y/libexec/jdk1.6.0/
if [ "$JAVA_HOME" != "" ]; then
  #echo "run java in $JAVA_HOME"
  JAVA_HOME=/home/hadoop/jdk1.8.0_171
fi
```

9.1 yarn-env.sh Configuration

iv. Through vim hadoop-env.sh, configure as the picture below.

```
# The java implementation to use.
export JAVA_HOME=/home/hadoop/jdk1.8.0_171

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol.  Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}
```

9.2 hadoop-env.sh Configuration

v. Through vim core-site.xml, configure as the picture below.

```
<configuration>

        <property>
                 <name>fs.defaultFS</name>
                 <value>hdfs://10.2.1.155:8020</value>
        </property>
        <property>
                <name>hadoop.tmp.dir</name>
                <value>file:/home/hadoop/hadoop-2.7.7/tmp</value>
        </property>
        <property>
                <name>io.file.buffer.size</name>
                <value>131072</value>
        </property>

</configuration>
~
~
~
~
~
~
```

9.3 core-site.xml 配置

vi. Through vim hdfs-site.xml, configure as the picture below.

```
<configuration>

          <property>
                 <name>dfs.namenode.secondary.http-address</name>
                <value>10.2.1.155:50090</value>
        </property>
      <property>
              <name>dfs.namenode.name.dir</name>
              <value>file:/home/hadoop/hadooph/hdfs/name</value>
              <final>true</final>
        </property>
       <property>
               <name>dfs.datanode.data.dir</name>
               <value>file:/home/hadoop/hadoop/hdfs/data</value>
               <final>true</final>
        </property>
        <property>
                <name>dfs.replication</name>
                <value>2</value>
        </property>
         <property>
                 <name>dfs.blocksize</name>
                 <value>134217728</value>
                 <description>node2~V~G件系m~_HDFS~]~W大m~O为128M</description>
         </property>
         <property>
                  <name>dfs.webhdfs.enabled</name>
                   <value>true</value>
          </property>


<property>
    <name>dfs.client.use.datanode.hostname</name>
    <value>true</value>
    <description>only cofig in clients</description>
</property>


</configuration>
```

9.4 hdfs-site.xml Configuration

vii. Through vim yarn-site.xml, configure as the picture below.

```
<configuration>

<!-- Site specific YARN configuration properties -->

   <property>
           <name>yarn.resourcemanager.hostname</name>
           <value>master</value>
   </property>
   <property>
     <name>yarn.nodemanager.aux-services</name>
     <value>mapreduce_shuffle</value>
   </property>
   <property>
     <name>yarn.resourcemanager.scheduler.address</name>
     <value>10.2.1.155:8030</value>
   </property>
   <property>
     <name>yarn.resourcemanager.address</name>
     <value>10.2.1.155:8032</value>
   </property>
   <property>
     <name>yarn.resourcemanager.resource-tracker.address</name>
     <value>10.2.1.155:8031</value>
   </property>
   <property>
     <name>yarn.resourcemanager.admin.address</name>
     <value>10.2.1.155:8033</value>
   </property>
   <property>
      <name>yarn.resourcemanager.webapp.address</name>
      <value>10.2.1.155:8088</value>
   </property>

</configuration>
```

9.5 yarn-site.xml Configuration

viii. Through vim mapred-site.xml, configure as the picture below.

```
<configuration>

          <property>
                  <name>mapreduce.framework.name</name>
                  <value>yarn</value>
          </property>
          <property>
                  <name>mapreduce.jobhistory.address</name>
                  <value>10.2.1.155:10020</value>
          </property>
          <property>
                  <name>mapreduce.jobhistory.webapp.address</name>
                  <value>10.2.1.155:19888</value>
          </property>

</configuration>
~
~
~
```

9.6 mapred-site.xml 配置

ix. Through vim slaves, enter the nodes we need for the Hadoop cluster

```
master
s1
s2
```

9.6 slaves Configuration

g. Configuring Hadoopports

i. Ports of our machines

Machines	port
s1	10156
master	10155
s2	10157

ii. Under the hadoopfolder，through vim hadoop-env.s, assign ports to different machines so that all of them can interact correctly. Enter the following commands at the end of the hadoop-env.sh：

```
export JAVA_HOME=/home/hadoop/jdk1.8.0_171

export HADOOP_SSH_OPTS="-p 10155"
"hadoop-env.sh" 102L, 4303C
```

9.7 master Ports

```
export JAVA_HOME=/home/hadoop/jdk1.8.0_171

export HADOOP_SSH_OPTS="-p 10156"
```

9.8 s1 Ports

```
export JAVA_HOME=/home/hadoop/jdk1.8.0_171

export HADOOP_SSH_OPTS="-p 10157"
-- INSERT --
```

9.9 s2 Ports

h. Transporting configured Hadoop files

i. Through cd /home/hadoop to access the configured Hadoop files. Deleted them in the s1 and s2 machines. Then send the configured files in master to s1 and s2 by:

rm -rf /home/hadoop/hadoop-2.7.7/

scp -P 10156 -r /home/hadoop/hadoop-2.7.7 hadoop@s1:~/hadoop-2.7.7

scp -P 10157 -r /home/hadoop/hadoop-2.7.7 hadoop@s2:~/hadoop-2.7.7

```
s1:/home/hadoop # ls
.bash_history  .ssh      hadoop-2.7.7         jdk-8u171-linux-x64.tar.gz
.bash_profile  .viminfo  hadoop-2.7.7.tar.gz  jdk1.8.0_171
```

```
s2:~ # cd /home/hadoop
s2:/home/hadoop # ls
.bash_history  .ssh      hadoop-2.7.7         jdk-8u171-linux-x64.tar.gz
.bash_profile  .viminfo  hadoop-2.7.7.tar.gz  jdk1.8.0_171
```

10.1～10.2 Results of transportation

i. Set Hadoop Paths

i. In every one of the machines, under /home/Hadoop, create and configure .bashrc file as the following picture:

```
export JAVA_HOME=/home/hadoop/jdk1.8.0_171
export HADOOP_HOME=/home/hadoop/hadoop-2.7.7/
export PATH=$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

11.1 .bshrc Configuration

ii. Activate our configuration through source /.bashrc.

j. Formatting Hadoop system (only on master machine)

i. cd /home/hadoop/hadoop-2.7.7/etc/bin

ii. hdfs namenode -format

```
22/08/01 07:29:34 INFO namenode.FSNamesystem: fsOwner             = root (auth:SIMPLE)
22/08/01 07:29:34 INFO namenode.FSNamesystem: supergroup          = supergroup
22/08/01 07:29:34 INFO namenode.FSNamesystem: isPermissionEnabled = true
22/08/01 07:29:34 INFO namenode.FSNamesystem: HA Enabled: false
22/08/01 07:29:34 INFO namenode.FSNamesystem: Append Enabled: true
22/08/01 07:29:35 INFO util.GSet: Computing capacity for map INodeMap
22/08/01 07:29:35 INFO util.GSet: VM type       = 64-bit
22/08/01 07:29:35 INFO util.GSet: 1.0% max memory 889 MB = 8.9 MB
22/08/01 07:29:35 INFO util.GSet: capacity      = 2^20 = 1048576 entries
22/08/01 07:29:35 INFO namenode.FSDirectory: ACLs enabled? false
22/08/01 07:29:35 INFO namenode.FSDirectory: XAttrs enabled? true
22/08/01 07:29:35 INFO namenode.FSDirectory: Maximum size of an xattr: 16384
22/08/01 07:29:35 INFO namenode.NameNode: Caching file names occuring more than 10 times
22/08/01 07:29:35 INFO util.GSet: Computing capacity for map cachedBlocks
22/08/01 07:29:35 INFO util.GSet: VM type       = 64-bit
22/08/01 07:29:35 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
22/08/01 07:29:35 INFO util.GSet: capacity      = 2^18 = 262144 entries
22/08/01 07:29:35 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.99900000128746033
22/08/01 07:29:35 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
22/08/01 07:29:35 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension     = 30000
22/08/01 07:29:35 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
22/08/01 07:29:35 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
22/08/01 07:29:35 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
22/08/01 07:29:35 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
22/08/01 07:29:35 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 mill
is
22/08/01 07:29:35 INFO util.GSet: Computing capacity for map NameNodeRetryCache
22/08/01 07:29:35 INFO util.GSet: VM type       = 64-bit
22/08/01 07:29:35 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
22/08/01 07:29:35 INFO util.GSet: capacity      = 2^15 = 32768 entries
Re-format filesystem in Storage Directory /home/hadoop/hadoop-2.7.7/hdfs/name ? (Y or N) Y
22/08/01 07:29:38 INFO namenode.FSImage: Allocated new BlockPoolId: BP-944173238-10.2.1.155-1659310178846
22/08/01 07:29:38 INFO common.Storage: Storage directory /home/hadoop/hadoop-2.7.7/hdfs/name has been successfully formatted.
22/08/01 07:29:38 INFO namenode.FSImageFormatProtobuf: Saving image file /home/hadoop/hadoop-2.7.7/hdfs/name/current/fsimage.ckpt_0000
000000000000000 using no compression
22/08/01 07:29:38 INFO namenode.FSImageFormatProtobuf: Image file /home/hadoop/hadoop-2.7.7/hdfs/name/current/fsimage.ckpt_00000000000
00000000 of size 321 bytes saved in 0 seconds.
22/08/01 07:29:39 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
22/08/01 07:29:39 INFO util.ExitUtil: Exiting with status 0
22/08/01 07:29:39 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at master/10.2.1.155
************************************************************/
```

12.1 namnode formatted

k. Activat Hadoop platform through start-all.sh with the hadoop user. Use jps to check status.

```
hadoop@master:~> jps
9009 DataNode
11480 Jps
10937 NodeManager
10365 ResourceManager
9741 SecondaryNameNode
8398 NameNode
```

```
hadoop@s1:~> jps
9024 Jps
8000 DataNode
8476 NodeManager
```

```
hadoop@s2:~> jps
27170 DataNode
27628 NodeManager
28062 Jps
```

13.1～13.3 Result of Successful Installtion