Hadoop 安装

安装准备

1.两台装有 ubuntu16.04 或者更新版本操作系统的物理机或虚拟机

主机名	主机 ip
master	172.19.5.33
slave01	172.19.5.34

其中 master 节点作为 hadoop 集群中 namenode,slave01 节点作为 hadoop 集群中 datanode.

2.Xshell,Xftp

3.本次安装是 ubuntu16.04+ hadoop-2.9.1.tar.gz+jdk-8u74-linux-x64.tar.gz。如果版本不同，参考 Hadoop 官网 http://hadoop.apache.org/docs/

练习一：更改主机名等信息

Step1:更改主机名

1.在 master 节点中添加集群中主机 ip，域名

```
root@master:~# vim /etc/hosts
172.19.5.33 master
172.19.5.34 slave01
```

2.在 master 节点中更改主机名，将原来的主机名更改为 master。

```
root@master:~# vim /etc/hostname
master
```

3.在 slave01 节点中添加集群中主机 ip，域名

```
root@slave01:~# vim /etc/hosts
172.19.5.33 master
172.19.5.34 slave01
```

4. 在 slave01 节点中更改主机名，将原来的主机名更改为 slave01。

```
root@slave01:~# vim /etc/hostname
slave01
```

5.重启 master,slave01 节点

练习二：ssh 免密登录

Step1:在 master 和 slave01 节点设置 root 登录密码

1.在 master 节点设置登录密码

```
root@master: ~# sudo passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Root 用户密码设置成功

2.在 master 节点配置文档，允许其它节点通过免密登录方式登录 master 节点

```
root@master: ~ # vim /etc/ssh/sshd_config
PermitRootLogin yes
```

配置成功后，重启 ssh 服务

```
root@master: ~# /etc/init.d/ssh restart
[ ok ] Restarting ssh (via systemctl): ssh.service.
```

3.在 slave01 节点重复 1,2 步骤。

Step2:master 节点免密登录

1.在 su 状态下设置免密登录

```
root@master:~# su
```

2.生成秘钥对

```
root@master:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:hfPzND40zums9eU1vAqnDjEyZyD1MsRW9GG1iJKktpY root@master
The key's randomart image is:
+---[RSA 2048]----+
|          .+oo o..     |
|          =oo.+ o .    |
|         +.*oo.o .     |
|        . + =+         |
```

```
|        E oS=o =       |
|      .     = oO +.    |
|             . .O. oo|
|               .++o o+|
|               o+o.o..|
+----[SHA256]-----+
```

3.添加 master 需要 ssh 免密登录访问的主机，也就是 slave01

```
root@master:~# ssh-copy-id slave01
```

```
/id_rsa.pub"
The authenticity of host 'slave01 (172.19.5.34)' can't be established.
eA.
Are you sure you want to continue connecting (yes/no)? yes
filter out any that are already installed
 prompted now it is to install the new keys
root@slave01's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'slave01'"
and check to make sure that only the key(s) you wanted were added.
```

4.验证是否可以免密登录 slave01

```
root@master:~# ssh slave01
```

```
root@master:~# ssh slave01
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-127-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

16 packages can be updated.
11 updates are security updates.


Last login: Mon Jun 11 04:56:06 2018 from 172.19.5.33
root@slave01:~# exit
logout
Connection to slave01 closed.
```

登录成功。

通过 exit 退出登录

Stp3:slave01 节点免密登录

1.在 su 状态下设置免密登录

```
root@slave01:~# su
```

2. 生成秘钥对

```
root@slave01:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
```

```
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:rUtiW73upps0M2cHliTf5mcMVhRNFU16i8teSuKCN8o root@slave01
The key's randomart image is:
+---[RSA 2048]----+
|                  o*B|
|                 . .o|
|            . .    o .|
|             = o ....|
|            S * =. . |
|             + =.o.   |
|         o X +.o++. |
|        ..B %oo+oo   |
|         .EB**. o    |
+----[SHA256]-----+
```

3. 添加 slave01 需要 ssh 免密登录访问的主机

```
root@slave01:~# ssh-copy-id master
a.pub"
  out any that are already installed
ted now it is to install the new keys
root@master's password:

Number of key(s) added: 1

Now try logging into the machine, with:      "ssh 'master'"
and check to make sure that only the key(s) you wanted were added.
```

4.验证 slave01 节点是否可以免密登录 master 节点

```
root@slave01:~# ssh master
```

```
root@slave01:~# ssh master
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

16 packages can be updated.
11 updates are security updates.


Last login: Mon Jun 11 04:56:37 2018 from 172.19.5.34
root@master:~# exit
logout
Connection to master closed.
```

Slave01 节点可以免密登录 master 节点。

通过 exit 退出。

练习三：安装 java

Step1:master 节点安装 java

Java 下载地址：

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

1.通过 xftp 将 jdk 安装包发送到 master 节点，记住 jdk 安装包的路径，第 3 步需要用.

2.在/usr 中创建 java 目录

```
root@master:/usr# mkdir java
```

3.将 jdk 安装包解压到/usr/java 路径下

```
root@master:/home/ubuntu# ls
hadoop-2.9.1.tar.gz    jdk-8u74-linux-x64.tar.gz
root@master:/home/ubuntu# tar -zxvf jdk-8u74-linux-x64.tar.gz -C /usr/java
```

4.在 /etc/profile 文档末尾中添加 java 环境变量

```
root@master:~# vim /etc/profile
```

在 profile 中，添加以下信息

export JAVA_HOME=/usr/java/jdk1.8.0_74

export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib

export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "$PS1" ]; then
  if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
export JAVA_HOME=/usr/java/jdk1.8.0_74
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    30,52         All
```

5.执行 /etc/profile，使环境变量生效

```
root@master:~# source /etc/profile
```

6.验证 java 环境是否配置成功

```
root@master:~# java -version
java version "1.8.0_74"
Java(TM) SE Runtime Environment (build 1.8.0_74-b02)
Java HotSpot(TM) 64-Bit Server VM (build 25.74-b02, mixed mode)
root@master:~# javac -version
javac 1.8.0_74
```

Java 环境配置成功。

Step2:在 slave01 节点配置 java 环境

1.在 slave01 节点中，在/usr 中创建 java 目录

```
root@slave01:/usr# mkdir java
```

2.在 master 节点将/usr/java 中 jdk1.8.0_74 文件发送到 slave01 节点

```
root@master:/usr/java# scp -r jdk1.8.0_74 root@slave01:/usr/java
root@master:/usr/java# scp -r jdk1.8.0_74 root@slave01:/usr/java
jni_md.h                                      100%  824     0.8KB/s   00:00
jawt_md.h                                     100%  995     1.0KB/s   00:00
jawt.h                                        100% 8690     8.5KB/s   00:00
jvmticmlr.h                                   100% 3774     3.7KB/s   00:00
jdwpTransport.h                               100% 6209     6.1KB/s   00:00
classfile_constants.h                         100%   20KB  19.7KB/s   00:00
jvmti.h                                       100%   76KB  75.6KB/s   00:00
jni.h                                         100%   72KB  72.0KB/s   00:00
NOTICE                                        100%   13KB  13.0KB/s   00:00
RELEASE-NOTES.html                            100%   64KB  64.5KB/s   00:00
derbynet.jar                                  100%  260KB 260.2KB/s   00:00
derbytools.jar                                100%  224KB 224.2KB/s   00:00
derby.jar                                     100% 3032KB   3.0MB/s   00:00
derbyLocale_hu.jar                            100%   93KB  93.1KB/s   00:00
derbyclient.jar                               100%  572KB 572.0KB/s   00:00
derbyoptionaltools.jar                        100%   49KB  48.6KB/s   00:00
derbyLocale_fr.jar                            100%  109KB 109.5KB/s   00:00
derbyLocale_pl.jar                            100%   91KB  91.1KB/s   00:00
derbyLocale_pt_BR.jar                         100%   88KB  88.4KB/s   00:00
derbyLocale_ru.jar                            100%  118KB 118.4KB/s   00:00
derby.war                                     100% 1490     1.5KB/s   00:00
derbyLocale_es.jar                            100%  103KB 103.4KB/s   00:00
derbyLocale_ja_JP.jar                         100%  121KB 120.7KB/s   00:00
derbyLocale_it.jar                            100%  103KB 103.3KB/s   00:00
derbyLocale_ko_KR.jar                         100%  115KB 114.7KB/s   00:00
```

确认 master 中 jdk1.8.0_74 文件成功发送到 slave01 /usr/java 路径下

```
root@slave01:/usr/java/jdk1.8.0_74# ls
bin            db            javafx-src.zip  lib          man              release
THIRDPARTYLICENSEREADME-JAVAFX.txt
COPYRIGHT  include  jre                  LICENSE  README.html  src.zip
THIRDPARTYLICENSEREADME.txt
```

3.在/etc/profile 文档末尾添加 java 环境变量

```
root@slave01:~# vim /etc/profile
```

在 profile 中，添加以下信息

```
export JAVA_HOME=/usr/java/jdk1.8.0_74
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
```

export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "$PS1" ]; then
  if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
export JAVA_HOME=/usr/java/jdk1.8.0_74
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    30,52         All
```

执行 /etc/profile，试环境变量生效

```
root@slave01:~# source /etc/profile
```

4. 验证 java 环境是否配置成功

```
root@slave01:~# java -version
java version "1.8.0_74"
Java(TM) SE Runtime Environment (build 1.8.0_74-b02)
Java HotSpot(TM) 64-Bit Server VM (build 25.74-b02, mixed mode)
root@slave01:~# javac -version
javac 1.8.0_74
```

Java 环境配置成功

练习四 master 节点安装、配置 hadoop

Hadoop 软件可以在官网上下载：https://archive.apache.org/dist/hadoop/common/
里面可以选择下载版本，然后下载 .tar.gz 文件

Step1:master 安装 hadoop

1.通过 xftp 将 hadoop-2.9.1.tar.gz 从本地发送到 master 节点，记住 hadoop-2.9.1.tar.gz 的路径。

2.将 hadoop-2.9.1.tar.gz 解压到 /usr/local 路径下

```
root@master:/home/ubuntu# ls
hadoop-2.9.1.tar.gz    jdk-8u74-linux-x64.tar.gz
root@master:/home/ubuntu# tar -zxvf hadoop-2.9.1.tar.gz -C /usr/local
```

解压后

```
root@master:/usr/local/hadoop-2.9.1# ls
bin   etc   include   lib   libexec   LICENSE.txt   NOTICE.txt   README.txt   sbin   share
```

3.在/etc/profile 中配置 hadoop 环境变量

```
root@slave01:~# vim /etc/profile
```

在 profile 中添加以下内容：

export HADOOP_HOME=/lib/hadoop/hadoop2.9.1

export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH

因为 profile 中含有 java 环境变量，将两个配置合在一起之后添加到 profile 中：

export JAVA_HOME=/usr/java/jdk1.8.0_74

export HADOOP_HOME=/usr/local/hadoop-2.9.1

export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib

export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "$PS1" ]; then
  if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
export JAVA_HOME=/usr/java/jdk1.8.0_74
export HADOOP_HOME=/usr/local/hadoop-2.9.1
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

执行 /etc/profile，试环境变量生效

```
root@master:~# source /etc/profile
```

4.修改 hadoop 的 hadoop-env.sh，修改 JAVA_HOME 的路径

```
root@master:/usr/local/hadoop-2.9.1# vim etc/hadoop/hadoop-env.sh
```

```
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME.  All others are
# optional.  When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
#export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME=/usr/java/jdk1.8.0_74

# 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}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}

# Extra Java CLASSPATH elements.  Automatically insert capacity-scheduler.
for f in $HADOOP_HOME/contrib/capacity-scheduler/*.jar; do
  if [ "$HADOOP_CLASSPATH" ]; then
    export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
  else
"etc/hadoop/hadoop-env.sh" 118L, 5009C                        21,1          Top
```

5.验证 hadoop 环境变量是否安装成功，查看 bin/hadoop 命令，bin/hadoop 必须在 /usr/local/hadoop-2.9.1 路径下执行

```
root@master:/usr/local/hadoop-2.9.1# bin/hadoop
Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]
  CLASSNAME                 run the class named CLASSNAME
 or
  where COMMAND is one of:
  fs                        run a generic filesystem user client
  version                   print the version
  jar <jar>                 run a jar file
                              note: please use "yarn jar" to launch
                                    YARN applications, not this command.
  checknative [-a|-h]       check native hadoop and compression libraries availability
  distcp <srcurl> <desturl> copy file or directories recursively
  archive -archiveName NAME -p <parent path> <src>* <dest> create a hadoop archive
  classpath                 prints the class path needed to get the
                              Hadoop jar and the required libraries
  credential                interact with credential providers
  daemonlog                 get/set the log level for each daemon
  trace                     view and modify Hadoop tracing settings

Most commands print help when invoked w/o parameters.
```

6.查看 hadoop 版本

```
root@master:/usr/local/hadoop-2.9.1# bin/hadoop version

Hadoop 2.9.1
Subversion https://github.com/apache/hadoop.git -r
e30710aea4e6e55e69372929106cf119af06fd0e
Compiled by root on 2018-04-16T09:33Z
Compiled with protoc 2.5.0
From source with checksum 7d6d2b655115c6cc336d662cc2b919bd
This command was run using /usr/local/hadoop-2.9.1/share/hadoop/common/hadoop-
common-2.9.1.jar
```

Step2:在 master 节点配置相关文档

1.在 master 中的/home 下创建以下目录 hadoop-data/tmp、 hadoop-data/hdfs 、hadoop-data/hdfs/data、hadoop-data/hdfs/name

```
root@master:/home# mkdir hadoop-data
root@master:/home# cd hadoop-data/
root@master:/home/hadoop-data# mkdir tmp
root@master:/home/hadoop-data# ls
tmp
root@master:/home/hadoop-data# mkdir hdfs
root@master:/home/hadoop-data# ls
hdfs    tmp
root@master:/home/hadoop-data# cd hdfs
root@master:/home/hadoop-data/hdfs# mkdir data
root@master:/home/hadoop-data/hdfs# mkdir name
root@master:/home/hadoop-data/hdfs# ls
data    name
```

2.修改/usr/local/hadoop-2.9.1/etc/hadoop 中 的 core-site.xml 文件

```
root@master:/usr/local/hadoop-2.9.1/etc/hadoop# vim core-site.xml
```

修改 core-site.xml，加上

```
<configuration>
<property>
        <name>fs.defaultFS</name>
        <value>hdfs://master:9000</value>
</property>
<property>
        <name>hadoop.tmp.dir</name>
        <value>file:/home/hadoop-data/tmp</value>
</property>
<property>
```

<name>io.file.buffer.size</name>
<value>131702</value>
</property>
</configuration>

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://master:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>file:/home/hadoop-data/tmp</value>
</property>
<property>
<name>io.file.buffer.size</name>
<value>131702</value>
</property>
</configuration>
~
~
~
~
~
~
~
~
"etc/hadoop/core-site.xml" 32L, 1028C                         1,1           All
```

3.修改/usr/local/hadoop-2.9.1/etc/hadoop 中 的 hdfs-site.xml 文件

```
root@master:/usr/local/hadoop-2.9.1/etc/hadoop# vim core-site.xml
```

修改 hdfs-site.xml,加上

<configuration>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/home/hadoop-data/hdfs/name</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/home/hadoop-data/hdfs/data</value>
</property>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
</configuration>

</property>
<property>
<name>dfs.namenode.secondary.http-address</name>
<value>master:9001</value>
</property>
<property>
<name>dfs.webhdfs.enabled</name>
<value>true</value>
</property>
</configuration>

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/home/hadoop-data/hdfs/name</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/home/hadoop-data/hdfs/data</value>
</property>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>dfs.namenode.secondary.http-address</name>
<value>master:9001</value>
</property>
<property>
<name>dfs.webhdfs.enabled</name>
<value>true</value>
</property>
</configuration>
"etc/hadoop/hdfs-site.xml" 40L, 1230C                                  1,1           All
```

4.修改/usr/local/hadoop-2.9.1/etc/hadoop 中 的 mapred-site.xml 文件：
Hadoop 文件中可能没有 mapred-site.xml, 但是包含 mapred-site.xml. template,需要将 mapred-site.xml. template 另存为 mapred-site.xml。

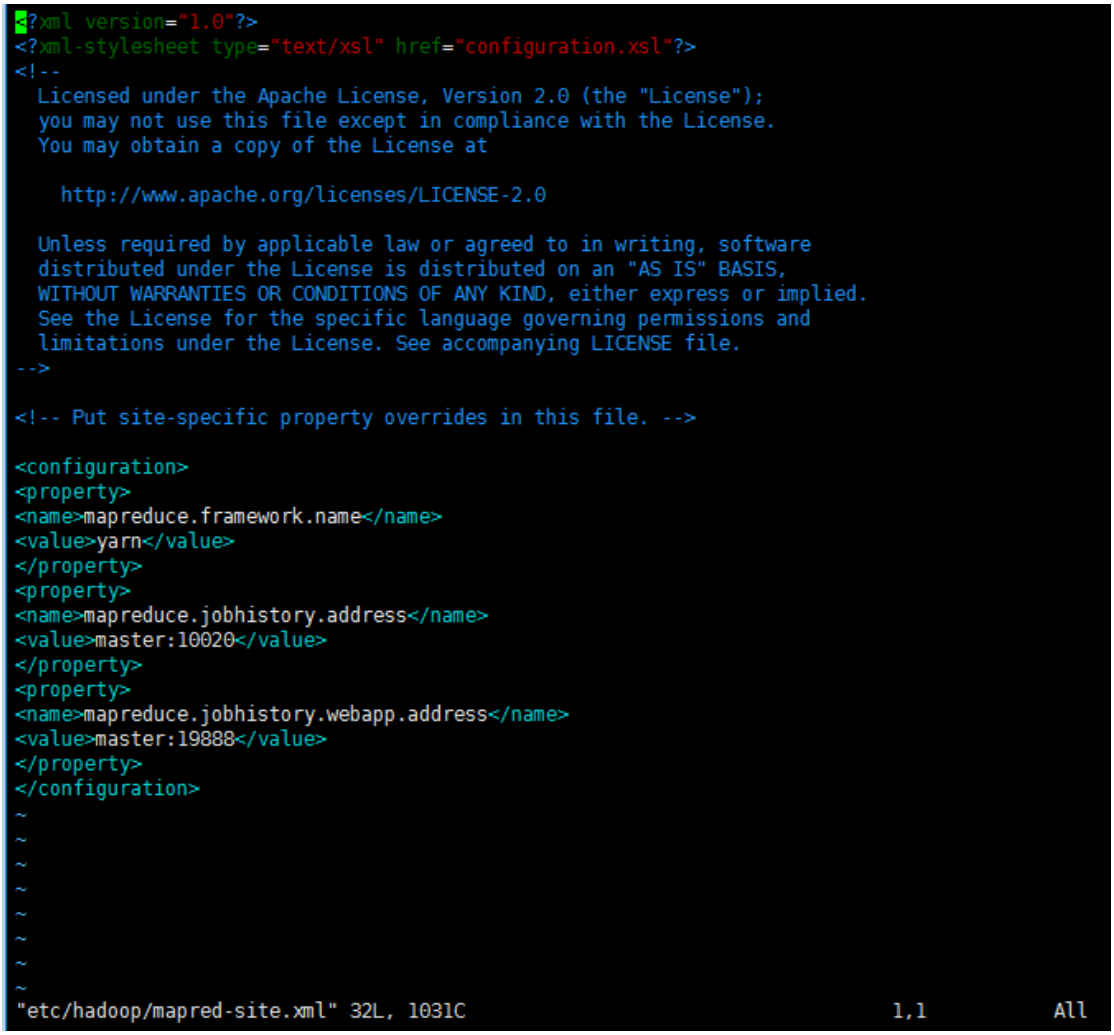
```
root@master:/usr/local/hadoop-2.9.1/etc/hadoop# cp mapred-site.xml. template mapred-site.xml
```

修改 mapred-site.xml

```
root@master:/usr/local/hadoop-2.9.1/etc/hadoop# vim mapred-site.xml
```

在 mapred-site.xml 中，添加以下内容：

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



5.修改/usr/local/hadoop-2.9.1/etc/hadoop 中 的 yarn-site.xml 文件：

```
root@master:/usr/local/hadoop-2.9.1/etc/hadoop# vim yarn-site.xml
```

在 yarn-site.xml 中，添加以下内容：

```
<configuration>
<property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
</property>
<property>
        <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<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.resource.memory-mb</name>
        <value>768</value>
</property>
</configuration>
```

```
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<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.resource.memory-mb</name>
<value>768</value>
</property>
</configuration>
                                                              50,1          Bot
```

6. 修改/usr/local/hadoop-2.9.1/etc/hadoop 中 的 slaves 文件：

配置 datanode 的主机名，每行一个，默认为 localhost，所以在伪分布式配置时，节点即作为 namenode 也作为 datanode。分布式配置可以保留 localhost，也可以删掉，让 master 节点仅作为 namenode 使用。

现配置一个 datanode,则在该文件中编辑如下字段：

```
root@master:/usr/local/hadoop-2.9.1/etc/hadoop# vim slaves
```

在 slaves 中添加如下内容

slave01

注：如果是两个或者多个节点的话，每行写一个：比如

练习五 slave01 节点安装、配置 hadoop

Step1:slave01 节点上配置 hadoop 环境变量

Slave01 节点上和 master 节点上完全一致

1. 在 master 节点上，将/usr/local 下 的 hadoop-2.9.1 发送到 slave01 节点上的 /usr/local 路径下

```
root@master:/usr/local# scp -r hadoop-2.9.1 root@slave01:/usr/local
```

在 slave01 节点上确认已经收到 hadoop-2.9.1：

```
root@slave01:/usr/local/hadoop-2.9.1# ls
bin         lib              logs         sbin
etc         libexec          NOTICE.txt   share
include     LICENSE.txt      README.txt
```

2.修改 hadoop 环境变量，

```
root@slave01:~# vim /etc/profile
```

在 profile 中添加以下内容：

export HADOOP_HOME=/lib/hadoop/hadoop2.9.1

export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH

因为 profile 中含有 java 环境变量，将两个配置合在一起之后添加到 profile 中：

export JAVA_HOME=/usr/java/jdk1.8.0_74

export HADOOP_HOME=/usr/local/hadoop-2.9.1

export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib

export

PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "$PS1" ]; then
  if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
export JAVA_HOME=/usr/java/jdk1.8.0_74
export HADOOP_HOME=/usr/local/hadoop-2.9.1
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

3. 执行 /etc/profile，试环境变量生效

```
root@slave01:~# source /etc/profile
```

4. 验证 hadoop 环境变量是否安装成功，查看 bin/hadoop 命令，bin/hadoop 必须在 /usr/local/hadoop-2.9.1 路径下执行。

```
root@slave01:/usr/local/hadoop-2.9.1# bin/hadoop
Usage: hadoop [--config confdir] [COMMAND | CLASSNAME]
    CLASSNAME                 run the class named CLASSNAME
  or
    where COMMAND is one of:
    fs                        run a generic filesystem user client
    version                   print the version
    jar <jar>                 run a jar file
                                note: please use "yarn jar" to launch
                                      YARN applications, not this command.
    checknative [-a|-h]       check native hadoop and compression libraries availability
    distcp <srcurl> <desturl> copy file or directories recursively
    archive -archiveName NAME -p <parent path> <src>* <dest> create a hadoop archive
    classpath                 prints the class path needed to get the
                                Hadoop jar and the required libraries
```

```
  credential            interact with credential providers
  daemonlog             get/set the log level for each daemon
  trace                 view and modify Hadoop tracing settings

Most commands print help when invoked w/o parameters.
```

5. 查看 hadoop 版本

```
root@slave01:/usr/local/hadoop-2.9.1# bin/hadoop version

Hadoop 2.9.1
Subversion https://github.com/apache/hadoop.git -r
e30710aea4e6e55e69372929106cf119af06fd0e
Compiled by root on 2018-04-16T09:33Z
Compiled with protoc 2.5.0
From source with checksum 7d6d2b655115c6cc336d662cc2b919bd
This command was run using /usr/local/hadoop-2.9.1/share/hadoop/common/hadoop-
common-2.9.1.jar
```

补充：修改 etc/hadoop/hadoop-env.sh 中的环境变量

Step2:slave01 节点进行配置 hadoop 集群相关文件

1.在 slave01 中的/home 下创建以下目录 hadoop-data/tmp、 hadoop-data/hdfs 、hadoop-data/hdfs/data、hadoop-data/hdfs/name

```
root@ slave01:/home# mkdir hadoop-data
root@ slave01:/home# cd hadoop-data/
root@ slave01:/home/hadoop-data# mkdir tmp
root@ slave01:/home/hadoop-data# ls
tmp
root@ slave01:/home/hadoop-data# mkdir hdfs
root@ slave01:/home/hadoop-data# ls
hdfs    tmp
root@ slave01:/home/hadoop-data# cd hdfs
root@ slave01:/home/hadoop-data/hdfs# mkdir data
root@ slave01:/home/hadoop-data/hdfs# mkdir name
root@ slave01:/home/hadoop-data/hdfs# ls
data    name
```

2.因为练习 5 的 step1 已经将 master 节点中 hadoop-2.9.1 文档拷贝到 slave01 节点处，所以不用再配置 core-site.xml 等文件

练习六：启动并验证集群

以下步骤均是在 master 节点上执行

Step1:启动集群

1.启动集群

```
root@master:/usr/local/hadoop-2.9.1# bin/hdfs namenode -format
```

```
18/06/11 08:22:47 INFO blockmanagement.BlockManager: maxNumBlocksToLog          = 1000
18/06/11 08:22:47 INFO namenode.FSNamesystem: Append Enabled: true
18/06/11 08:22:47 INFO util.GSet: Computing capacity for map INodeMap
18/06/11 08:22:47 INFO util.GSet: VM type       = 64-bit
18/06/11 08:22:47 INFO util.GSet: 1.0% max memory 889 MB = 8.9 MB
18/06/11 08:22:47 INFO util.GSet: capacity      = 2^20 = 1048576 entries
18/06/11 08:22:47 INFO namenode.FSDirectory: ACLs enabled? false
18/06/11 08:22:47 INFO namenode.FSDirectory: XAttrs enabled? true
18/06/11 08:22:47 INFO namenode.NameNode: Caching file names occurring more than 10 times
yChange: false
18/06/11 08:22:47 INFO util.GSet: Computing capacity for map cachedBlocks
18/06/11 08:22:47 INFO util.GSet: VM type       = 64-bit
18/06/11 08:22:47 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
18/06/11 08:22:47 INFO util.GSet: capacity      = 2^18 = 262144 entries
18/06/11 08:22:47 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
18/06/11 08:22:47 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
18/06/11 08:22:47 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
18/06/11 08:22:47 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
piry time is 600000 millis
18/06/11 08:22:47 INFO util.GSet: Computing capacity for map NameNodeRetryCache
18/06/11 08:22:47 INFO util.GSet: VM type       = 64-bit
18/06/11 08:22:47 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
18/06/11 08:22:47 INFO util.GSet: capacity      = 2^15 = 32768 entries
18/06/11 08:22:47 INFO namenode.FSImage: Allocated new BlockPoolId: BP-681919662-172.19.5.33-1528719767598
matted.
simage.ckpt_0000000000000000000 using no compression
ckpt_0000000000000000000 of size 321 bytes saved in 0 seconds .
18/06/11 08:22:47 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
18/06/11 08:22:47 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at master/172.19.5.33
************************************************************/
```

2.启动 NameNode：

```
root@master:/usr/local/hadoop-2.9.1# sbin/start-dfs.sh
```

```
root@master:/usr/local/hadoop-2.9.1# sbin/start-dfs.sh
Starting namenodes on [master]
The authenticity of host 'master (172.19.5.33)' can't be established.
ECDSA key fingerprint is SHA256:AdQSPg5J0PCoEML1+OL9rQicOCgoPek4EXIWJd69yeA.
Are you sure you want to continue connecting (yes/no)? yes
master: Warning: Permanently added 'master,172.19.5.33' (ECDSA) to the list of known hosts.
root@master's password:
master: starting namenode, logging to /usr/local/hadoop-2.9.1/logs/hadoop-root-namenode-master.out
slave01: starting datanode, logging to /usr/local/hadoop-2.9.1/logs/hadoop-root-datanode-slave01.out
Starting secondary namenodes [master]
root@master's password:
er.out
```

注：箭头处指向的是 master 节点 root 用户的密码,也就是在免密登录设置的密码

3.启动 ResourceManager

```
root@master:/usr/local/hadoop-2.9.1# sbin/start-yarn.sh
```

```
root@master:/usr/local/hadoop-2.9.1# sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop-2.9.1/logs/yarn-root-resourcemanager-master.out
slave01: starting nodemanager, logging to /usr/local/hadoop-2.9.1/logs/yarn-root-nodemanager-slave01.out
```

4.以下两步是用来启动 MapReduce JobHistory Server

```
root@master:/usr/local/hadoop-2.9.1/sbin# mr-jobhistory-daemon.sh start historyserver
```

```
root@master:/usr/local/hadoop-2.9.1/sbin# mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /usr/local/hadoop-2.9.1/logs/mapred-root-historyserver-master.out
```

```
root@master:/usr/local/hadoop-2.9.1/sbin# yarn-daemon.sh start timelineserver
```

```
root@master:/usr/local/hadoop-2.9.1/sbin# yarn-daemon.sh start timelineserver
starting timelineserver, logging to /usr/local/hadoop-2.9.1/logs/yarn-root-timelineserver-master.out
```

Step2:本地验证

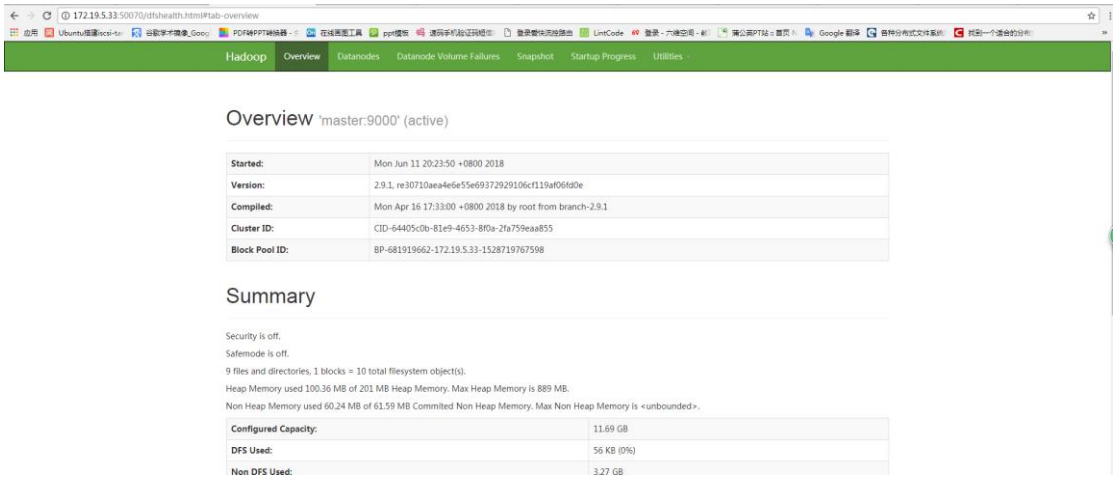
在本地浏览器中输入不同的端口就可以看到不同的服务

NameNode　http://172.19.5.33:50070/

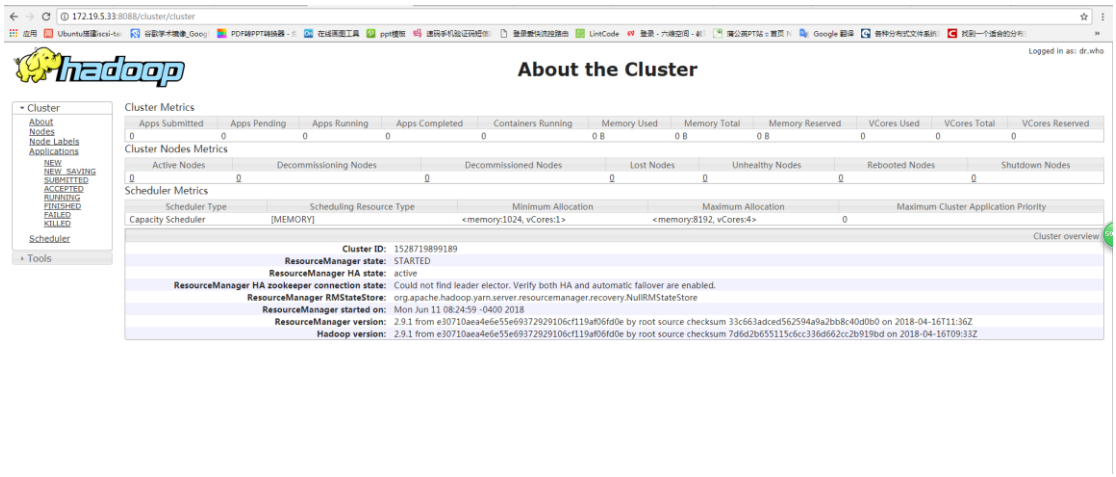
ResourceManager http:// 172.19.5.33:8088/

MapReduce JobHistory Server　http:// 172.19.5.33:19888/

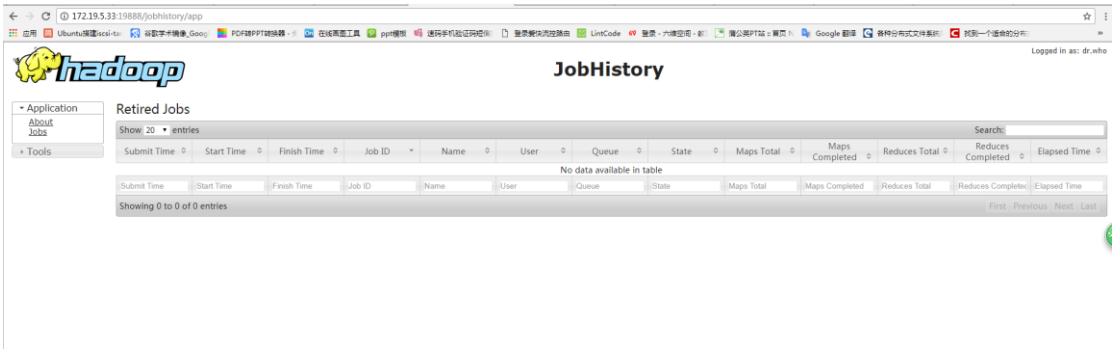
NameNode　http://172.19.5.33:50070/



ResourceManager http:// 172.19.5.33:8088/



MapReduce JobHistory Server　http:// 172.19.5.33:19888/



Step3:在 hadoop 集群上传文档

1.上传到 hdfs

首先要为上传的文件在 HDFS 上创建文件夹 hdfs dfs －mkdir [指定目录和文件夹名]。如：hdfs dfs –mkdir /test。

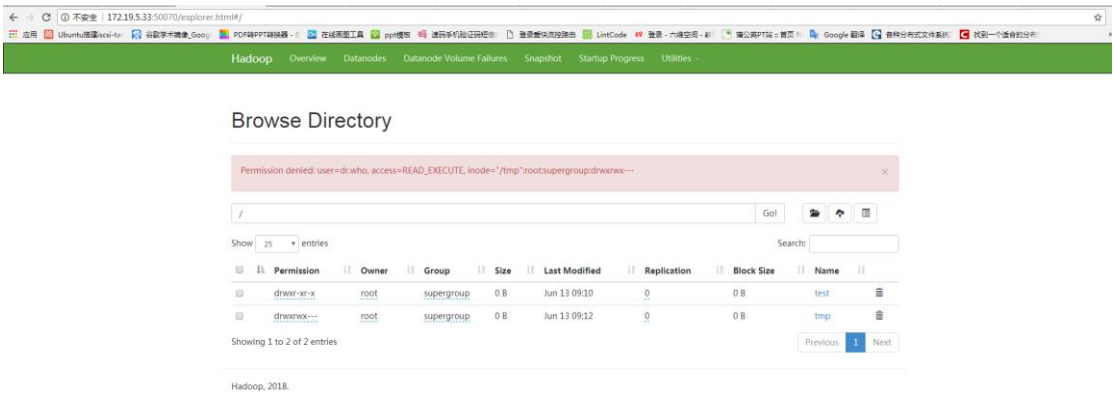
把节点文件夹上传到 hdfs 上 hdfs dfs －put [本地文件] [新创建的 hdfs 上文件]

如：hdfs dfs －put LICENSE.txt /test/

2. 查看 hdfs 上的文件信息

hadoop fs －ls / (查看根目录下所有已经上传的文件)或者进入 http://master:50070，Utilities->Browse the file system 中查看。查看具体内容如下：

```
root@master:/usr/local/hadoop-2.9.1# bin/hadoop fs -ls /
Found 2 items
drwxr-xr-x   - root supergroup          0 2018-06-12 21:10 /test
drwxrwx---   - root supergroup          0 2018-06-12 21:12 /tmp
```



3. 从 hdfs 上下载文件到本地

从 HDFS 文件系统中下载文件到本地，有 2 种方法，一种可以直接进入 WebUI 下载，一种是通过命令获取。

a.进入 http://master:50070 后，Utilities->Browse the file system 可以点击 name 下载。

b.在本地某个目录下创建一个新文件夹，用于接收从 hdfs 下载下来的文件, mkdir newfile

获取文件存放到本地指定路径 hadoop fs －get [hdfs 上的文件] [本地文件路径]

如：hadoop fs －get /test /home/ubuntu/newfile

```
root@master:/usr/local/hadoop-2.9.1# bin/hadoop fs -get /test /home/ubuntu/newfile
root@master:/usr/local/hadoop-2.9.1# cd /home/ubuntu/newfile/
```

```
root@master:~/newfile# ls
test
```

4. 删除 hdfs 上指定文件

hadoop fs –rm [指定文件]，如果是文件夹里包含多个子文件的话，可以使用 hadoop fs –rm –r [指定文件]

```
root@master:/usr/local/hadoop-2.9.1# bin/hadoop fs -rm -r /test
Deleted /test
root@master:/usr/local/hadoop-2.9.1# bin/hadoop fs -ls /
Found 1 items
drwxrwx---   - root supergroup          0 2018-06-12 21:12 /tmp
```

5. 查看 HDFS 块的信息

使用命令 hdfs fsck [指定文件] –blosks

```
root@master:/usr/local/hadoop-2.9.1# bin/hdfs fsck /tmp -blocks
Connecting to namenode via http://master:50070/fsck?ugi=root&blocks=1&path=%2Ftmp
FSCK started by root (auth:SIMPLE) from /172.19.5.33 for path /tmp at Tue Jun 12 21:33:41 EDT 2018
.
/tmp/LICENSE.txt:  Under replicated BP-681919662-172.19.5.33-1528719767598:blk_1073741827_1003. Targ
et Replicas is 2 but found 1 live replica(s), 0 decommissioned replica(s), 0 decommissioning replica
(s).
Status: HEALTHY
 Total size:    106210 B
 Total dirs:    6
 Total files:   1
 Total symlinks:                0
 Total blocks (validated):      1 (avg. block size 106210 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       1 (100.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    2
 Average block replication:     1.0
 Corrupt blocks:                0
 Missing replicas:              1 (50.0 %)
 Number of data-nodes:          1
 Number of racks:               1
FSCK ended at Tue Jun 12 21:33:41 EDT 2018 in 3 milliseconds


The filesystem under path '/tmp' is HEALTHY
```