搭建伪分布模式

伪分布模式的安装与单机模式前面过程相同，在此基础上需要对hadoop进行一些额外的相关配置才可以正常运行。

1. 参照单机环境模式准备环境，并保证能够正常运行验证程序。
2. 编辑修改 etc/hadoop/core-site.xml，增加指定使用hdfs分布式文件系统，这里我们将它指向本机的9000端口。

```
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://localhost:9000</value>
</property>
```

3. hdfs-site.xml 增加设置分布式文件系统中文件存储数据的副本数，因为在单机上运行，只有一个数据节点，所以参数值为1。

   注意：当然单机模式加入的参数可保留不变

```
<property>
   <name>dfs.replication</name>
  <value>1</value>
</property>
```

4. hdfs-site.xml增加namenode数据存放目录，如果不指定参数，默认情况下，在进行HDFS格式化时，会将namenode数据放到/tmp目录下。

```
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:///data/hdfs/name</value>
 </property>
```

5. hdfs-site.xml增加设置分布式文件系统数据存放目录，同样，如果不指参数，默认情况下会将数据存入到/tmp目录下。

```
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:///data/hdfs/data</value>
</property>
```

6.修改mapred-site.xml文件

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <!-- 原2.9.x版本可以下设置  -->
    <property>
        <name>mapreduce.application.classpath</name>
<value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
    </property>

</configuration>
```

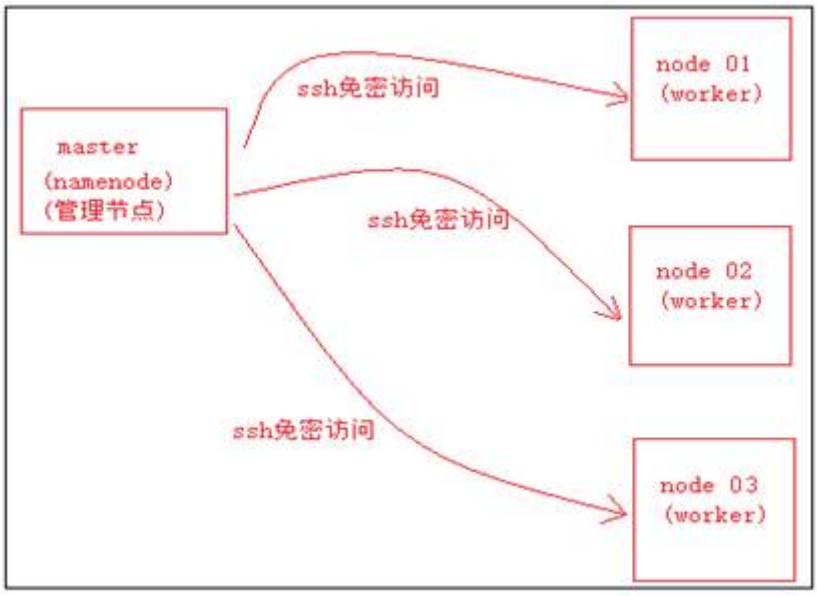
7.修改yarn-site.xml

```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>

    <!-- 原2.9.x版本可以下设置  -->
    <property>
        <name>yarn.nodemanager.env-whitelist</name>

 <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
    </property>
</configuration>
```

8.配置ssh免密登录

在使用分布式环境时，当Hadoop启动以后，NameNode节点会通过ssh方式来启动和停止各个DataNode上的各种守护进程的。默认情况下，ssh远程进行连接时需要使用密码来进行验证，如果结点较多，一个一个输入密码是不现实的方式，所以需要配置ssh免密码访问。

虽然伪分布式环境只有一台机器，但其工作原理与完全分布式环境基本相同，也需要配置ssh免密登录。

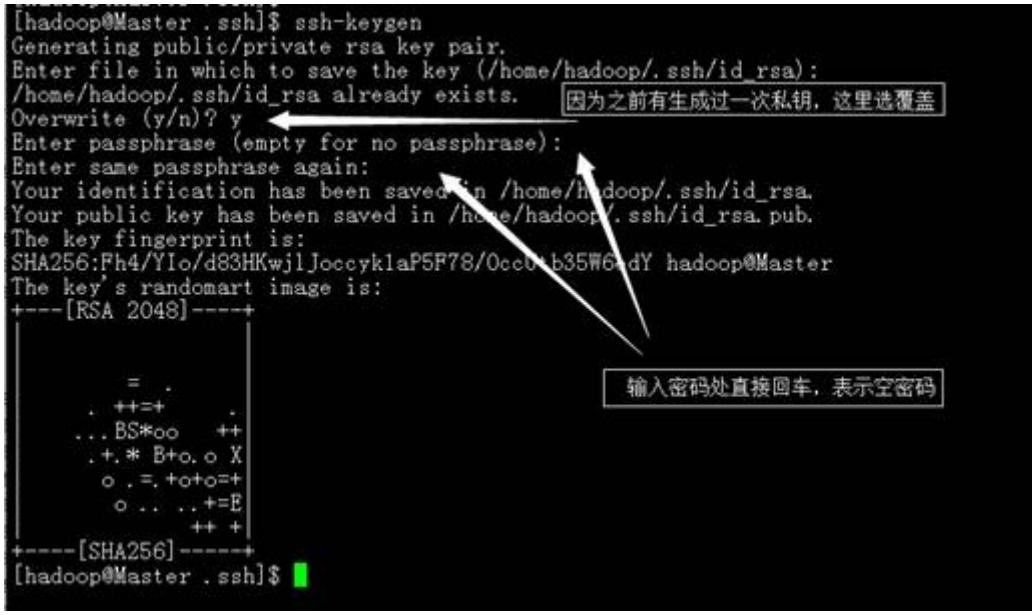


a.) 检查/etc/hosts中localhost对应IP为127.0.0.1，这是系统默认配置

```
[hadoop@Master ~]$
[hadoop@Master ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[hadoop@Master ~]$
```

b.) 对用户hadoop生成一对公私钥，在介绍putty使用时，生成的私钥带有密码，不满足要求，故这里需要重新生成一份。

```
1 [hadoop@Master .ssh]$ ssh-keygen
```



c.) 将生成的公钥传到相关机器对应用户名目录下authorized_keys文件中，公钥可以手工拷文件，也可以使用openssh自带的分发工具ssh-copy-id来完成，在这里因为只有一台机器，用户也是同一个hadoop帐号，可直接将公钥追加到hadoop帐号的authorized_keys文件中即可。

```
[hadoop@Master ~] cd ~/.ssh
[hadoop@Master .ssh] cat id_rsa.pub >> authorized_keys #或者
[hadoop@Master .ssh] ssh-copy-id    hadoop@localhost
[hadoop@Master .ssh] chmod 600 authorized_keys
```

d.) 验证ssh免密码登录

9.格式化hdfs分布式文件系统，格式化时，输出信息较多，需要注意是否有错误信息，要保证格式化是成功完成的。

```
[hadoop@Master ~] cd  /opt/hadoop
[hadoop@Master hadoop] bin/hdfs  namenode -format
```

格式化后的目录结构如下：

```
[hadoop@Master hadoop]$ find /data/hdfs/name
/data/hdfs/name
/data/hdfs/name/current
/data/hdfs/name/current/VERSION
/data/hdfs/name/current/seen_txid
/data/hdfs/name/current/fsimage_0000000000000000000.md5
/data/hdfs/name/current/fsimage_0000000000000000000
[hadoop@Master hadoop]$

```

10.修改start-dfs.sh中的JAVA_HOME变量

```
[hadoop@Master ~] cd /opt/hadoop
[hadoop@Master hadoop] vi etc/hadoop/hadoop-env.sh
```

将其中的JAVA_HOME设置为正确路径，结果如下图:

11.启动hdfs分布式文件系统

```
[hadoop@Master hadoop] sbin/start-dfs.sh
```

如果在启动过程中，有任何问题，可以查看/opt/hadoop/logs目录下的日志文件，从中找出错误原因并进行修正。

检查JAVA进程：

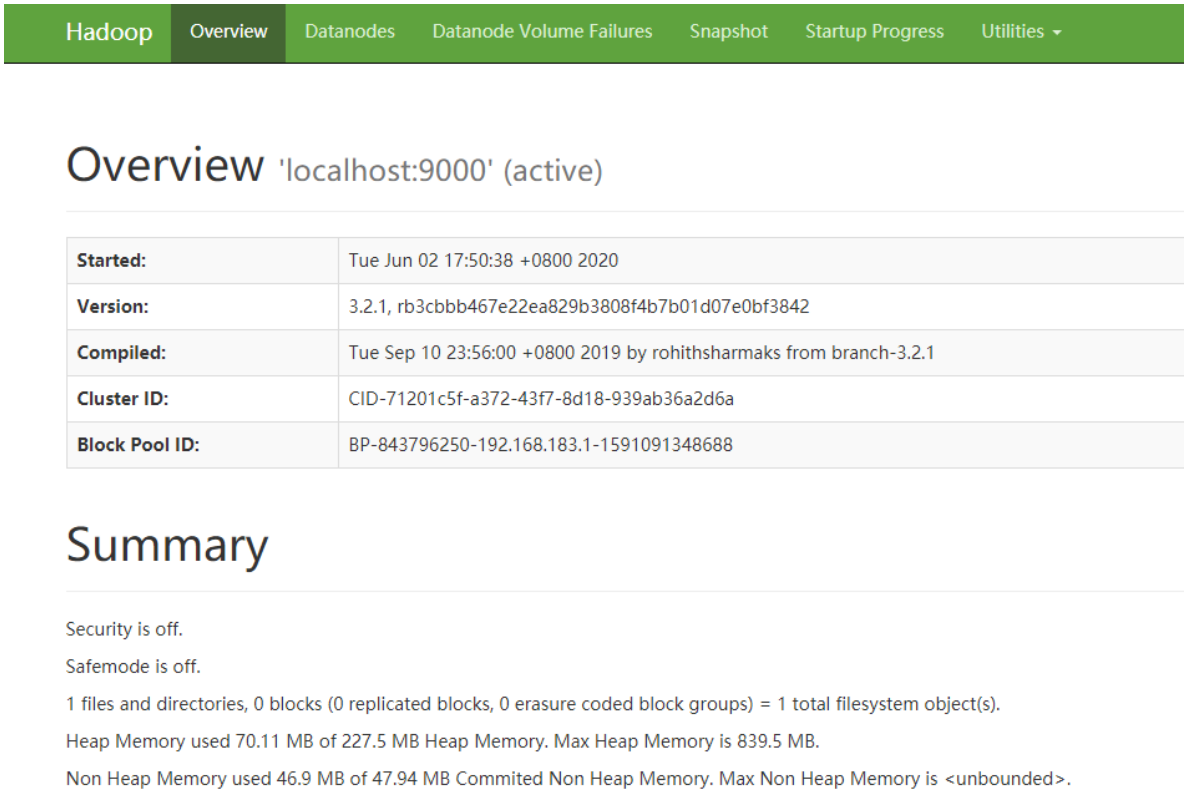
```
[hadoop@Master hadoop]$ jps
12549 NameNode
13029 Jps
12698 DataNode
12891 SecondaryNameNode
[hadoop@Master hadoop]$

```

12. 启动资源管理器

```
[hadoop@Master hadoop] sbin/start-yarn.sh
```

13.访问hadoop分布式文件系统和资源管理器

使用浏览器打开http://192.168.74.11:9870即可看到hfds的运行状态，因为CentOS在这次的环境中使用的是minimal方式安装，没有安装图形化的浏览器，所以只能用Windows系统下的浏览器进行访问。



资源管理器：

使用浏览器打开http://192.168.74.11:8088

image-20200602180337027

14.测试创建用户目录并查看结果

```
[hadoop@Master hadoop]$ bin/hdfs dfs -mkdir /user
[hadoop@Master hadoop]$ bin/hdfs dfs -ls /
[hadoop@Master hadoop]$ bin/hdfs dfs -mkdir /user/hwadee
[hadoop@Master hadoop]$ bin/hdfs dfs -find /
```

或者

```
[hadoop@Master hadoop]$ bin/hadoop fs -mkdir /user
[hadoop@Master hadoop]$ bin/hadoop fs -ls /
[hadoop@Master hadoop]$ bin/hadoop fs -mkdir /user/hwadee
[hadoop@Master hadoop]$ bin/hadoop fs -find /

```

即hdfs dfs命令可以使用hadoop fs代替

15.编辑一个文件，用于测试文件中单词数量，内容不限

```
vi    hwadee-word.txt
```

16.将此文件上传到分布式文件系统中

```
[hadoop@Master hadoop]$ bin/hdfs dfs -put hwadee-word.txt /user/hwadee
[hadoop@Master hadoop]$ bin/hdfs dfs -ls /user/hwadee
[hadoop@Master hadoop]$ bin/hdfs dfs -cat  /user/hwadee/hwadee-word.txt
```

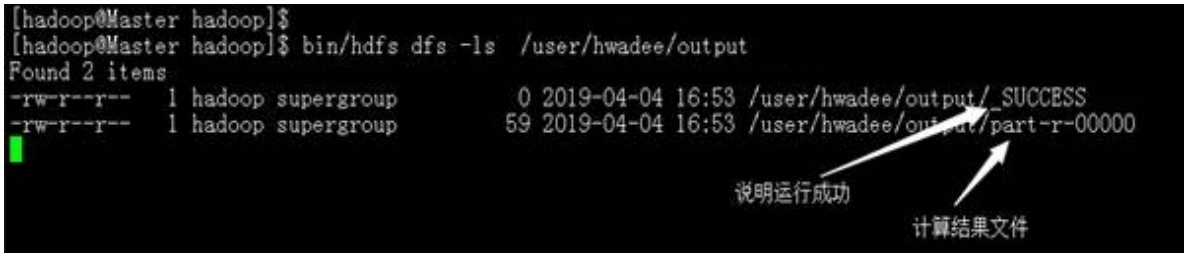
上传测试文件到 HDFS

17.运行自带example程序，统计其中的单词数

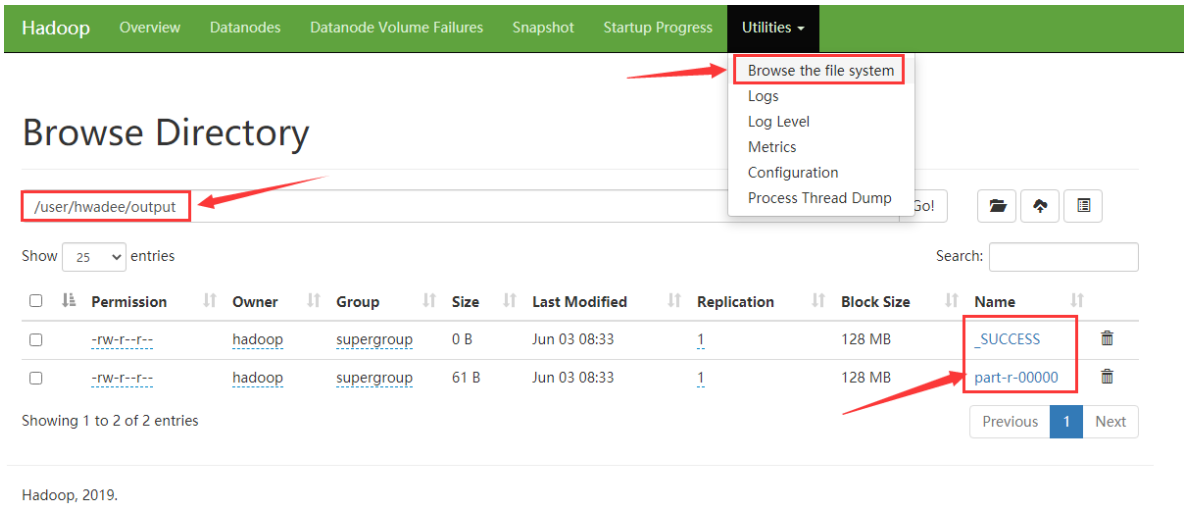
```
[hadoop@Master hadoop]$ bin/hadoop jar \
 share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar wordcount \
 /user/hwadee/hwadee-word.txt /user/hwadee/output
```

17.使用命令查看结果：

```
[hadoop@Master hadoop]$ bin/hdfs dfs -ls /user/hwadee/output
```



18.使用浏览器查看结果：



19.完善浏览器访问(可选)：

在上图中，如果在浏览器里点击文件出现文件详细信息，如果此时想要下载文件，会出现不成功的情况。

不能下载的原因是Download指向的链接会跳转为使用Linux的主机名或localhost，如果我们的Windows机器不能解析此主机名称，当然就不能下载，同时这里下载时会使用到另一个tcp端口9864，所在要检查 Linux系统中的防火墙是否打开此端口，允许其它机器连入(建议测试时直接关闭防火墙服务)。

解决方法（总体思想就是将使用本机可对外访问的IP来运行服务）：

/etc/hosts中增加当前主机名设置

```
sudo echo “192.168.74.11 master” >> /etc/hosts
```

重启dfs文件系统

```
$sbin/stop-dfs.sh
$sbin/start-dfs.sh
```

编辑文件C:\windows\system32\drivers\etc\hosts

增加一行

```
192.168.74.11    master
```

请将IP和Linux机器名称替换为你自己设置，然后就可以在浏览器中下载文件了。

pi 计算显示内存分配问题，调整以下参数

yarn-site.xml

```
<property>
  <name>yarn.scheduler.minimum-allocation-mb</name>
  <value>2048</value>
</property>

```