环境配置：

系统：

Centos 7.3

ubuntu 16.04

节点：

IP 角色 主机名

192.168.1.104 namenode test

192.168.1.102 datanode1 test02

192.168.1.143 datanode2 wiki-ubuntu

**jdk-8u144-linux-x64.tar.gz (**[**Java**](http://lib.csdn.net/base/javaee)**)**

**hadoop-2.7.4.tar.gz (Hadoop 包)**

**安装步骤：**

**1.修改各个虚拟机的hostname和host**

**2.创建用户组和用户**

**3、配置虚拟机网络，使虚拟机系统之间以及和host主机之间可以通过相互ping通。**

**4.安装jdk和配置环境变量，检查是否配置成功**

**5、配置ssh，实现节点间的无密码登录 ssh node1指令验证时候成功**

**7、master配置hadoop，并将hadoop文件传输到node节点**

**8、配置环境变量，并启动hadoop，检查是否安装成功，执行wordcount检查是否成功。**

**1.修改虚拟机的hostname和hosts文件**

以master上机器为例，打开终端，执行如下的操作，把hostname修改成master，hosts修改成如下所示的样子：

#修改hostname的指令：

sudo vim /etc/hostname

#修改hosts指令：

sudo vim /etc/hosts

#将以下内容添加到hosts中

192.168.1.104 test

192.168.1.102 test02

192.168.1.143 wiki-ubuntu

192.168.1.142 gitlab

如下图所示:



**同样地，在test和test02 wiki-ubuntu gitlab机器上做相似的操作，分别更改主机名为test和test02 wiki-ubuntu、gitlab。**

**3.创建用户和用户组（二台机器上都要操作）**

1. 创建hadoop用户组

**sudo addgroup hadoop**

2. 创建hadoop用户

**sudo adduser -ingroup hadoop hadoop**

3. 给hadoop用户添加权限，打开/etc/sudoers文件

**sudo vim /etc/sudoers**

按回车键后就会打开/etc/sudoers文件了，给hadoop用户赋予root用户同样的权限  
在root  ALL=(ALL:ALL)  ALL下添加hadoop  ALL=(ALL:ALL)  ALL

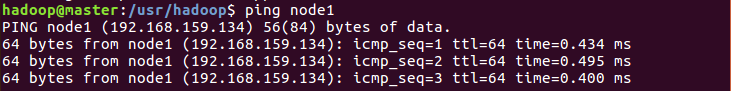
**4.检验各个主机之间能否连通**

分别以刚刚创建的hadoop用户重新登录系统，以后的操作都以hadoop用户登录。

**ping +主机名**

分别在各个主机上执行上述指令，看是否能与其他主机连通。

出现下图代表能够连通：



如果都成功ping通，进行下面的操作。

**5.安装jdk和配置环境变量**

分别在每台主机上安装jdk，并配置环境变量。（嫌麻烦的前面可以安装完jdk后再克隆）

1）下载jdk安装包（自行百度），并将安装包拖入到虚拟机当中

2）通过cd命令进入到安装包的当前目录，利用如下命令进行解压缩。

**tar -zxvf jdk.....(安装包名称)**

3）利用如下命令将解压后的文件夹移到/usr目录下

#注意，这样移动到/usr以后就没有jdk1.8...这个目录了，是将这个目录下的所有文件全部移动到/usr/java下，

**mv jdk1.8...(文件夹名称) /usr/local**

4）配置环境变量

**sudo vim /etc/profile**

在末尾加上四行：

**[plain]** [view plain](http://blog.csdn.net/xiaoxiangzi222/article/details/52757168) [copy](http://blog.csdn.net/xiaoxiangzi222/article/details/52757168)

[print?](http://blog.csdn.net/xiaoxiangzi222/article/details/52757168)

1. #java
2. export JAVA\_HOME=/usr/java
3. export JRE\_HOME=/usr/java/jre
4. export CLASSPATH=$JAVA\_HOME/lib
5. export PATH=:$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

输入如下命令使配置生效：source /etc/profile

**查看配置是否成功，**



出现如上信息说明java配置成功。

6.配置SSH，实现节点间的无密码登录

本人在这一步经常出错，莫名其妙的错误，网上也找不到资料。需要自己多实验几次。

下面的 1.2在所有主机上都要做

**1.生成密钥Pair**

**ssh-keygen -t rsa**

输入之后一直选择enter即可。生成的秘钥位于 ~/.ssh文件夹下。可用cd 命令进入查看。

2.在master上，导入authorized\_keys

**cat /usr/local/hadoop/.ssh/id\_rsa.pub >> /usr/local/hadoop/.ssh/authorized\_keys**

3.远程无密码登录(把master上的authorized\_keys拷贝到其他主机的相应目录下)

#进入master的.ssh目录,执行复制操作

**ssh-copy-id hadoop@test**

**ssh-copy-id hadoop@test02**

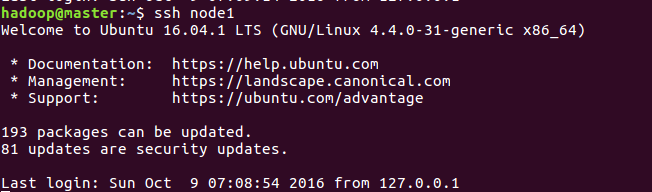
**ssh-copy-id hadoop@wiki-ubuntu**

修改各台主机上authorized\_keys文件的权限：  
  
所有机器上，均执行命令：

**chmod 600 .ssh/authorized\_keys**

完成之后，在master上执行下面操作，检查免密码登录是否成功。

**ssh test(test02 wiki-ubuntu)**

****

**7.master配置hadoop,然后将master的hadoop文件传送给node节点**

1）解包移动

**[plain]** [view plain](http://blog.csdn.net/xiaoxiangzi222/article/details/52757168) [copy](http://blog.csdn.net/xiaoxiangzi222/article/details/52757168)

[print?](http://blog.csdn.net/xiaoxiangzi222/article/details/52757168)

1. #解压hadoop包
2. tar -zxvf hadoop...
3. #将安装包移到/usr目录下
4. mv hadoop... /usr/local/hadoop

2）新建文件夹

**[plain]** [view plain](http://blog.csdn.net/xiaoxiangzi222/article/details/52757168) [copy](http://blog.csdn.net/xiaoxiangzi222/article/details/52757168)

[print?](http://blog.csdn.net/xiaoxiangzi222/article/details/52757168)

1. #在/usr/local/hadoop目录下新建如下目录(root)
2. mkdir /dfs
3. mkdir /dfs/name
4. mkdir /dfs/data
5. mkdir /tmp

3)配置文件：hadoop-env.sh(文件都在/usr/local/hadoop/etc/hadoop中)

修改JAVA\_HOME值（export JAVA\_HOME=/usr/local/jdk1.8.0\_144）

4)配置文件：yarn-env.sh

修改JAVA\_HOME值（export JAVA\_HOME=/usr/local/jdk1.8.0\_144）

5)配置文件：slaves

将内容修改为：

test02

wiki-ubuntu

### 6）修改 core-site.xml

打开 core-site.xml文件并对其进行编辑，如下图所示。

vi core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://node1:9000</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>file:/home/hadoop/hadoop-2.7.4/tmp</value>

</property>

</configuration>

### 修改 hdfs-site.xml

打开hdfs-site.xml文件并对其进行编辑，如下图所示。

vi hdfs-site.xml

<configuration>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>node1:50090</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/home/hadoop/hadoop-2.7.4/tmp/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/home/hadoop/hadoop-2.7.4/tmp/dfs/data</value>

</property>

</configuration>

### 修改 mapred-site.xml

目录下么没有这个文件,这有一个模板,我们需要先拷贝一份

cp mapred-site.xml.template mapred-site.xml

vi mapred-site.xml

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>node1:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>node1:19888</value>

</property>

</configuration>

### 修改 yarn-site.xml

vi yarn-site.xml

<configuration>

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

<property>

<name>yarn.resourcemanager.hostname</name>

<value>node1</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

</configuration>

[复制代码](javascript:void(0);)

13 创建对应的文件夹   mkdir -p logs （其实可以先创建好了文件夹再复制，文件夹多了不影响）

在每个节点上创建数据存储目录/home/hadoop-2.7.4/hdfs 用来存放集群数据。  
在主节点node上创建目录/home/hadoop-2.7.4/hdfs/name 用来存放文件系统元数据。  
在每个从节点上创建目录/home/hadoop-2.7.4/hdfs/data 用来存放真正的数据。  
所有节点上的日志目录为/home/hadoop-2.7.4/logs  
所有节点上的临时目录为/home/hadoop-2.7.4/tmp

14复制配置好的配置到slave节点

scp -r /home/hadoop-2.7.4 root@slave1:/home/hadoop-2.7.4

scp -r /home/hadoop-2.7.4 root@slave2:/home/hadoop-2.7.4

15 在master节点上配置hadoop salve配置文件 增加节点

vi /home/hadoop-2.7.4/etc/hadoop/slaves

增加

salve1

slave2

16格式化namenode和datanode并启动，（在master上执行就可以了 不需要在slave上执行）

/home/hadoop-2.7.4/bin/hadoop namenode -format

/home/hadoop-2.7.4/bin/hadoop datanode -format

/home/hadoop-2.7.4/sbin/start-all.sh

17 通过jps命令查看是否启动成功

[复制代码](javascript:void(0);)

[root@master ~]# ssh master

Last login: Sat Sep 2 00:47:50 2017 from slave1

[root@master ~]# jps

9187 Jps

3221 ResourceManager

3062 SecondaryNameNode

2856 NameNode

[root@master ~]# ssh slave1

Last login: Sat Sep 2 00:25:55 2017 from master

[root@slave1 ~]# jps

6044 Jps

2685 NodeManager

2590 DataNode

[root@slave1 ~]# ssh slave2

Last login: Wed Aug 30 21:34:38 2017 from master

j[root@slave2 ~]# jps

2679 NodeManager

5994 Jps

2590 DataNode

[root@slave2 ~]#

[复制代码](javascript:void(0);)

如果启动异常，一定要仔细看log并修正配置

**8.向hadoop集群系统提交第一个mapreduce任务(wordcount)**

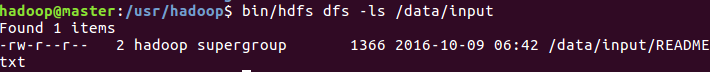
进入本地hadoop目录（/usr/local/hadoop）

1、  bin/hdfs dfs -mkdir -p /data/input在虚拟分布式文件系统上创建一个测试目录/data/input

2、  hdfs dfs -put README.txt  /data/input  将当前目录下的README.txt 文件复制到虚拟分布式文件系统中

3、  bin/hdfs dfs -ls /data/input    查看文件系统中是否存在我们所复制的文件

如图操作：



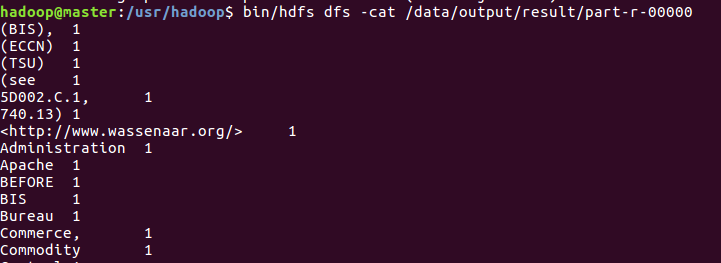
3、  运行如下命令向hadoop提交单词统计任务

进入jar文件目录，执行下面的指令。

hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar wordcount /data/input /data/output/result

查看result，结果在result下面的part-r-00000中

**hdfs dfs -cat /data/output/result/part-r-00000**



自此，hadoop集群搭建成功！