**Hadoop安装说明**

## Linux环境搭建及配置

### **软件环境**

* CentOS 7 64位
* hadoop-2.6.4.tar.gz
* jdk-7u79-linux-x64.gz
* VMware Workstation Pro 12

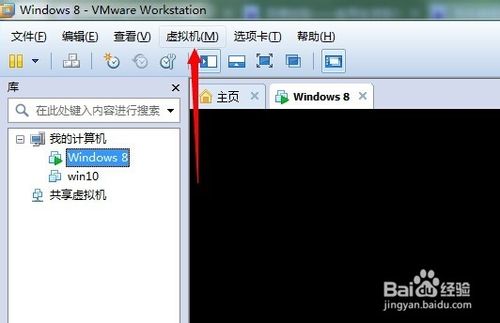
### 安装Linux

在windows上安装VMware，然后在Vmware上新建虚拟机，安装CentOS系统，安装过程略过，安装后要开启ssh，ftp服务，我们需要准备三台虚拟机，可以新建好一个虚拟机，然后通过克隆当前虚拟机的方式来进行快速建立三台虚拟机，克隆操作可以在一些前期安装配置完成后在进行，以免后面要进行重复配置。

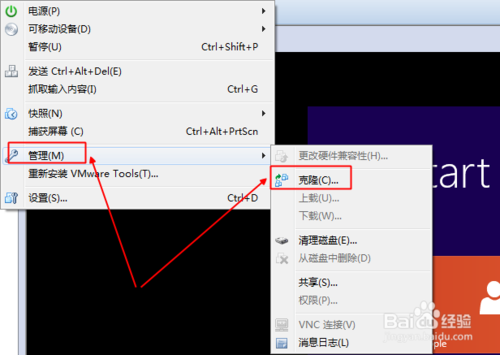
#### 克隆虚拟机

配图安装的是windows系统，不过没有关系，只需要关注操作步骤即可。

##### 导航栏中找到‘虚拟机’并进入。

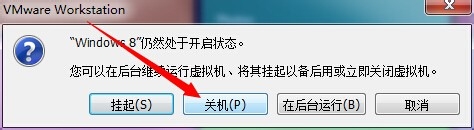
[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=2)

##### 在‘管理’中找到‘克隆’，点击进去。

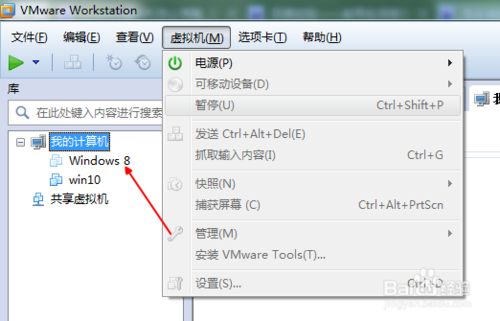
[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=3)

##### 很多朋友安装的时候都会出现无法使用克隆的问题。其实在克隆的时候把虚拟机系统关机就可以克隆了。

[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=4)

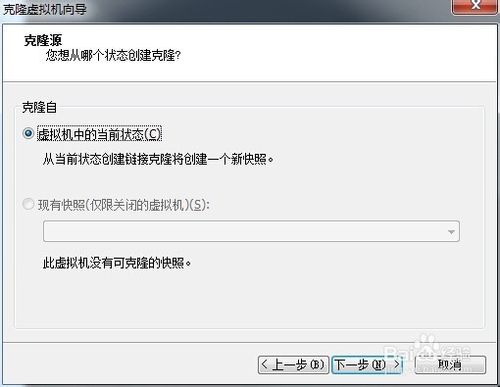
[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=5)

##### 无法选择，要选中被克隆的系统。

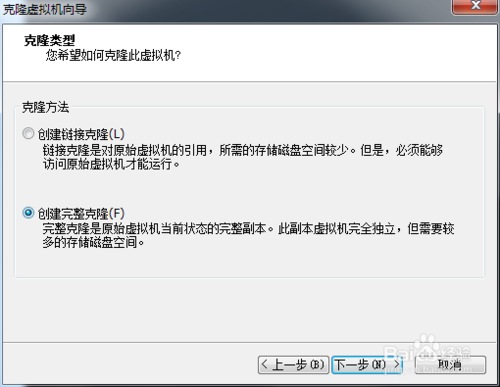
[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=6)

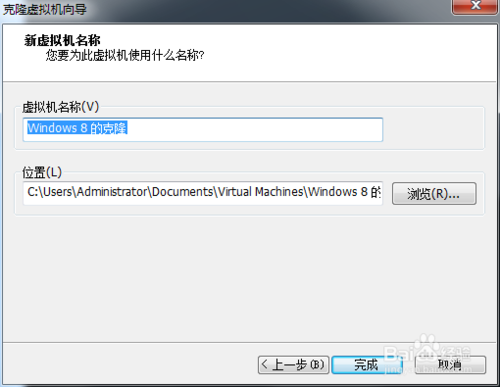
##### 正式开始克隆咯，点击下一步。进行创建克隆。

[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=7)

[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=8)

##### 你可以选择使用完整克隆和q链接克隆，图上都有详细说明。最后等待克隆完成。

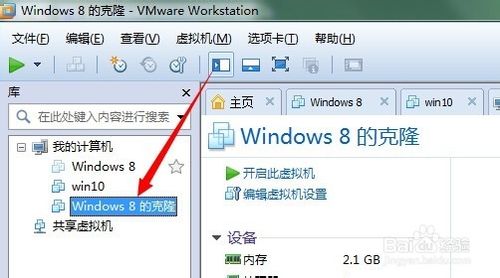
[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=9)

[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=10)

[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=11)

##### 安装完成点击‘关闭’，克隆已经完成，点击开启此虚拟机。

[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=12)

[](http://jingyan.baidu.com/album/6b97984d9798f11ca2b0bfcd.html?picindex=13)

##### 好的，已经克隆成功，进入系统。

### 修改虚拟机hostname

将三台虚拟机主机名称分别改为master 、slave01、slaver02

首先切换到root用户，然后执行如下命令：

[root@master ~]# hostname //查看当前主机名

[root@master ~]# hostnamectl set-hostname master //将主机名改为master

[root@master ~]# hostnamectl //查看更改是否成功

注：hostname不要命名为“xxx.01,xxx.02”之类以“.数字”结尾，否则到最后hadoop的NameNode服务将启动失败。

### 修改hosts文件

三台虚拟机hosts文件中都加入ip与主机名映射。

[root@master ~]# vi /etc/hosts //打开hosts文件

打开后看到如下内容

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

注释掉第一行，加入

192.168.64.130 master master

192.168.64.128 slave01 slave01

193.168.64.129 slave02 slave02

完成后hosts文件内容如下：

#127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

192.168.64.130 master master

192.168.64.128 slave01 slave01

193.168.64.129 slave02 slave02

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

保存后使用

[root@master ~]# ping master //测试连接

应该就能ping通

### 安装JDK

#### 卸载OpenJDK

[root@master ~]# rpm -qa | grep java //查看JDK信息

javapackages-tools-3.4.1-6.el7\_0.noarch

python-javapackages-3.4.1-6.el7\_0.noarch

java-1.7.0-openjdk-1.7.0.75-2.5.4.2.el7\_0.x86\_64

tzdata-java-2015a-1.el7.noarch

java-1.7.0-openjdk-headless-1.7.0.75-2.5.4.2.el7\_0.x86\_64

执行以下操作卸载OpenJDK，

[root@master ~]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.75-2.5.4.2.el7\_0.x86\_64

[root@master ~]# rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.75-2.5.4.2.el7\_0.x86\_64

[root@master ~]# rpm -e --nodeps tzdata-java-2015a-1.el7.noarch

#### 安装JDK

##### 建立一个目录，上传jdk安装文件到目录,然后建立/usr/java目录

[root@master ~]# mkdir /project

[root@master ~]# mkdir /usr/java

[root@master ~]# cd /project/

[root@master project]# tar -zxvf jdk-7u79-linux-x64.gz

[root@master project]# mv ./jdk1.7.0\_79/ /usr/java/jdk1.7.0\_79

##### 配置环境变量

[root@master project]# vi /etc/profile //打开配置文件

在proflie文件结尾添加

export JAVA\_HOME=/usr/java/jdk1.7.0\_79  
export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar   
export PATH=$PATH:$JAVA\_HOME/bin

##### 使修改的内容在当前shell窗口中生效。（永久生效，重启系统或注销用户）

[root@master project]# source /etc/profile

测试是否生效

[root@master project]# java –version

java version "1.7.0\_79"

Java(TM) SE Runtime Environment (build 1.7.0\_79-b15)

Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)

##### 拓展

编写脚本一次完成整个JDK安装

<!-- lang: shell -->

#!/bin/bash

# shell script to install jdk

# 1. remove openjdk if exists.

for i in $(rpm -qa | grep openjdk | grep -v grep)

do

echo "Deleting rpm -> "$i

rpm -e --nodeps $i

done

if [[ ! -z $(rpm -qa | grep jdk | grep -v grep) ]];

then

echo "-->Failed to remove the defult Jdk."

else

# 2.tar and install JDK(jdk-7u79-linux-x64.gz)

mkdir /usr/java

cd /project/

tar -zxvf jdk-7u79-linux-x64.gz

mv ./jdk1.7.0\_79/ /usr/java/jdk1.7.0\_79

# 3. config /etc/profile

echo "export JAVA\_HOME=/usr/java/jdk1.7.0\_79" >>/etc/profile

echo -e 'export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar'>>/etc/profile

echo -e 'export PATH=$PATH:$JAVA\_HOME/bin'>>/etc/profile

source /etc/profile

fi

### 创建hadoop用户

在三台虚拟机上创建hadoop用户，密码最好也设置成一样

[root@master ~]# useradd hadoop

[root@master ~]# passwd hadoop

更改用户 hadoop 的密码 。

新的 密码：

无效的密码： 密码少于 8 个字符

重新输入新的 密码：

passwd：所有的身份验证令牌已经成功更新。

将hadoop加入root用户组

[root@master ~]# usermod -g root hadoop

[root@master ~]# id hadoop //测试是否加入root组

uid=1001(hadoop) gid=0(root) 组=0(root) //出现类似说明则成功

### 配置ssh免密码登录

hadoop工作时，各节点要相互通讯，正常情况下linux之间通讯要提供用户名、密码（目的是保证通讯安全），如果需要人工干预输入密码，显然不方便，做这一步的目的，是让各节点能自动通过安全认证，不影响正常通讯。

#### 先在master上，生成公钥、私钥对

以hadoop身份登录到系统,执行如下命令

[hadoop@master ~]$ cd //(进入个人主目录，默认为/home/hadoop)

[hadoop@master ~]$ ssh-keygen -t rsa -P '' //(注：最后是二个单引号)

即：以rsa算法，生成公钥、私钥对，-P ''表示空密码。该命令运行完后，会在个人主目录下生成.ssh目录，里面会有二个文件id\_rsa（私钥） ,id\_rsa.pub(公钥)

#### 导入公钥

[hadoop@master ~]$ cat .ssh/id\_rsa.pub >> .ssh/authorized\_keys

执行完以后，可以在本机上测试下，用ssh连接自己，即：ssh localhost (或ssh master)，如果不幸还是提示要输入密码，说明还没起作用，还有一个关键的操作

[hadoop@master ~]$ chmod 600 .ssh/authorized\_keys //(修改文件权限，否则不起作用)

然后再测试下 ssh localhost ，如果不需要输入密码，就连接成功，表示ok，一台机器已经搞定了。

#### 在其它机器上生成公钥、密钥，并将公钥文件复制到master

##### 以hadoop身份登录其它二台机器 slave01、slave02，执行 ssh-keygen -t rsa -P '' 生成公钥、密钥

##### 然后用scp命令，把公钥文件发放给master（即：刚才已经搞定的那台机器）

slave01上：

scp .ssh/id\_rsa.pub hadoop@master:/home/hadoop/id\_rsa\_01.pub

slave02上：

scp .ssh/id\_rsa.pub [hadoop@master:/home/hadoop/id\_rsa\_02.pub](mailto:hadoop@master:/home/hadoop/id_rsa_02.pub)

这个操作可能需要确认，并输入密码

[hadoop@slave02 ~]$ scp .ssh/id\_rsa.pub hadoop@master:/home/hadoop/id\_rsa\_02.pubThe authenticity of host 'master (192.168.64.130)' can't be established.

ECDSA key fingerprint is 07:3e:b8:fc:d2:9c:ff:9b:0b:3a:fa:d7:e2:af:15:ff.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'master,192.168.64.130' (ECDSA) to the list of known hosts.

hadoop@master's password:

id\_rsa.pub 100% 396 0.4KB/s 00:00

这二行执行完后，回到master中，查看下/home/hadoop目录，应该有二个新文件id\_rsa\_01.pub、id\_rsa\_02.pub，然后在master上，导入这二个公钥

[hadoop@master ~]$ cat id\_rsa\_01.pub >> .ssh/authorized\_keys

[hadoop@master ~]$ cat id\_rsa\_02.pub >> .ssh/authorized\_keys

这样，master这台机器上，就有所有3台机器的公钥了。

##### 将master上的“最全”公钥，复制到其它机器

继续保持在master上，

scp .ssh/authorized\_keys hadoop@slave01:/home/hadoop/.ssh/authorized\_keys

scp .ssh/authorized\_keys [hadoop@slave02:/home/hadoop/.ssh/authorized\_keys](mailto:hadoop@slave02:/home/hadoop/.ssh/authorized_keys)

修改其它机器上authorized\_keys文件的权限

slave01以及slave02机器上，均执行命令

chmod 600 .ssh/authorized\_keys

##### 验证

在每个虚拟机上，均用 ssh 其它机器的hostname 验证下，如果能正常无密码连接成功，表示ok,如果失败，请检查是否没有配置权限

## Hadoop安装

### 上传并解压Hadoop安装文件

上传hadoop安装文件到master，可以直接放在/home/hadoop下或者自己新建一个目录，解压

[hadoop@master project]$ cd /project/

[hadoop@master project]$ tar -zxvf hadoop-2.6.4.tar.gz

### 修改配置

hadoop集群中每个机器上面的配置基本相同，所以我们先在master上面进行配置部署，然后再复制到其他节点。所以这里的安装过程相当于在每台机器上面都要执行。

【注意】：master和slaves安装的hadoop路径要完全一样，用户和组也要完全一致

一共有7个文件要修改：

$HADOOP\_HOME/etc/hadoop/hadoop-env.sh

$HADOOP\_HOME/etc/hadoop/yarn-env.sh

$HADOOP\_HOME/etc/hadoop/core-site.xml

$HADOOP\_HOME/etc/hadoop/hdfs-site.xml

$HADOOP\_HOME/etc/hadoop/mapred-site.xml

$HADOOP\_HOME/etc/hadoop/yarn-site.xml

$HADOOP\_HOME/etc/hadoop/slaves

其中$HADOOP\_HOME表示hadoop根目录，本文中默认为/project/hadoop-2.6.4

#### hadoop-env.sh 、yarn-env.sh

这二个文件主要是修改JAVA\_HOME后的目录，改成实际本机jdk所在目录位置

[hadoop@master project]$ vi hadoop-2.6.4/etc/hadoop/hadoop-env.sh

[hadoop@master project]$ vi hadoop-2.6.4/etc/hadoop/yarn-env.sh

找到下面这行的位置，改成（jdk目录位置，大家根据实际情况修改）

export JAVA\_HOME=/usr/java/jdk1.7.0\_79

另外 hadoop-env.sh中 , 建议加上这句:

export HADOOP\_PREFIX=/home/hadoop/hadoop-2.6.0

#### slaves

这个文件里面保存所有slave节点

写入如下，即加入两个节点，也可以把master加入



#### core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://master:9000</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

</property>

<property>

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

<value>file:/project/tmp </value>

<description>Abase for other temporary directories.</description>

</property>

<property>

<name>hadoop.proxyuser.aboutyun.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.aboutyun.groups</name>

<value>\*</value>

</property>

</configuration>

注：/project/tmp 目录如不存在，则先mkdir手动创建

[hadoop@master hadoop]$ mkdir /project/tmp

#### hdfs-site.xml

<configuration>

<property>

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

<value>master:9001</value>

</property>

<property>

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

<value>file:/project/dfs/name</value>

</property>

<property>

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

<value>file:/project/dfs/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

</configuration>

注：dfs.replication表示数据副本数，一般不大于datanode的节点数。

/project/dfs/name、/project/dfs/data

目录如不存在，则先mkdir手动创建

[hadoop@master hadoop]$ cd /project

[hadoop@master project]$ mkdir dfs

[hadoop@master project]$ mkdir dfs/name

[hadoop@master project]$ mkdir dfs/data

#### mapred-site.xml

默认情况下这个文件不存在，从模板复制一个

[hadoop@master hadoop]$ cp mapred-site.xml.template mapred-site.xml

再打开进行编辑

<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>

#### yarn-site.xml

<configuration>

<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>

<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>

</configuration>

### ****复制到其他节点****

将master上的hadoop目录复制到slave01,slave02

[hadoop@master project]$ cd /project

[hadoop@master project]$ scp -r hadoop-2.6.4 hadoop@slave01:/project

[hadoop@master project]$ scp -r hadoop-2.6.4 hadoop@slave02:/project

注：slave01、slave02上的hadoop临时目录(tmp)及数据目录(dfs)，仍然要先手动创建。slave01、slave02上hadoop用户要拥有/project的权限

### 测试

#### 格式化namenode

[hadoop@master bin]$ cd /project/hadoop-2.6.4/bin/

[hadoop@master bin]$ ./hdfs namenode -format

15/02/12 21:29:53 INFO namenode.FSImage: Allocated new BlockPoolId: BP-85825581-192.168.187.102-1423747793784

15/02/12 21:29:53 INFO common.Storage: Storage directory /home/hadoop/tmp/dfs/name has been successfully formatted.

等看到类似这个时，表示格式化ok

#### ****启动hdf****s:

[hadoop@master bin]$ cd /project/hadoop-2.6.4/sbin

[hadoop@master sbin]$ ./start-dfs.sh

启动完成后，输入jps查看进程：

此时在master上面运行的进程有：

namenode

secondarynamenode

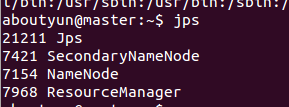
slave节点上面运行的进程有：datanode

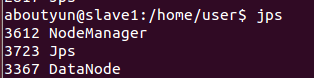
#### ****启动yarn****

[hadoop@master sbin]$ ./start-yarn.sh

我们看到如下效果：

master有如下进程：



slave1有如下进程  


此时hadoop集群已全部配置完成！！！

【注意】：而且所有的配置文件<name>和<value>节点处不要有空格，否则会报错！

然后我们输入<http://master:8088>、http://master:50070/

（这里如果没有配置hosts，所以输出master访问不到，如果访问不到输入ip地址即可）

<http://192.168.64.130:8088>

http://192.168.64.130:50070/

其它注意事项：

a) master(即：namenode节点)若要重新格式化，请先清空各datanode上的data目录（最好连tmp目录也一起清空），否则格式化完成后，启动dfs时，datanode会启动失败

b) 如果觉得master机器上只运行namenode比较浪费,想把master也当成一个datanode,直接在slaves文件里,添加一行master即可