# Hadoop集群安装配置手册

## 系统基本配置

### 主机名修改

修改/etc/sysconfig/network文件中HOSTNAME=hadoop1

### **IP地址配置**

修改/etc/sysconfig/network-scripts/ifcfg-eth0，修改后的信息如下：

DEVICE="eth0"

#BOOTPROTO="dhcp"

BOOTPROTO="static"

HWADDR="00:0C:29:59:35:85"

IPV6INIT="no"

IPADDR=10.254.231.185

NETMASK=255.255.0.0

GATEWAY=10.254.0.1

DNS1=10.254.0.1

NM\_CONTROLLED="yes"

ONBOOT="yes"

TYPE="Ethernet"

UUID="9a604f00-87f1-4d6d-8cb3-6eac05a8ac01"

说明：IP地址、子网淹码、网管、DNS要根据自己网络的实际情况进行配置

### **主机名与ip地址做映射**

修改/etc/hosts文件，修改后的信息如下：

#127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

#::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

10.254.231.185 hadoop1

### **防火墙关闭**

chkconfig --level 0123456 iptables off

### **selinux关闭**

**（**1）检查selinux 状态

执行getenforce命令，如果结果显示 permissive 或disabled，你可以跳过这一步

（2）打开/etc/selinux/config文件，修改SELINUX=enforcing 为SELINUX=permissive

（3）重启系统

### **JDK安装**

从网上下载jdk1.7以上64位linux版本。在此，我们安装jdk-7u60-linux-x64.rpm版本，下载后执行rpm -ivh jdk-7u60-linux-x64.rpm命令进行安装。然后用java -version 查看，如果显示下边信息，说明安装成功。

[root@hadoop1 ~]# java -version

java version "1.7.0\_60"

Java(TM) SE Runtime Environment (build 1.7.0\_60-b19)

Java HotSpot(TM) 64-Bit Server VM (build 24.60-b09, mixed mode)

在/etc/profile里添加JDK的环境变量

export JAVA\_HOME=/usr/java/default

并在PATH里添加:$JAVA\_HOME/bin

### **yum安装配置**

本次我们选择安装cdh5.10.1版本的Hadoop，我们先下载cdh yum源库文件repo，yum源库文件地址为：

<https://archive.cloudera.com/cdh5/redhat/6/x86_64/cdh/cloudera-cdh5.repo。>

(1) 在系统终端，执行下边命令下载reop:

wget https://archive.cloudera.com/cdh5/redhat/6/x86\_64/cdh/cloudera-cdh5.repo -P /etc/yum.repos.d/

1. 因为我们安装的是cdh5.10.1版本的Hadoop，所以我们需要修改cloudera-cdh5.repo文件中baseurl=https://archive.cloudera.com/cdh5/redhat/6/x86\_64/cdh/5/

为baseurl=https://archive.cloudera.com/cdh5/redhat/6/x86\_64/cdh/5.10.1/

1. 清理缓存残留的yum信息，执行yum clean all。
2. 确认一下是否是我们需要的Hadoop版本，执行yum list|grep hadoop命令后，结果如下：

[root@hadoop1 yum.repos.d]# yum list|grep hadoop

hadoop.x86\_64 2.6.0+cdh5.10.1+2141-1.cdh5.10.1.p0.10.el6

hadoop-0.20-conf-pseudo.x86\_64 2.6.0+cdh5.10.1+2141-1.cdh5.10.1.p0.10.el6

hadoop-0.20-mapreduce.x86\_64 2.6.0+cdh5.10.1+2141-1.cdh5.10.1.p0.10.el6

hadoop-0.20-mapreduce-jobtracker.x86\_64

hadoop-0.20-mapreduce-jobtrackerha.x86\_64

hadoop-0.20-mapreduce-tasktracker.x86\_64

hadoop-0.20-mapreduce-zkfc.x86\_64 2.6.0+cdh5.10.1+2141-1.cdh5.10.1.p0.10.el6

hadoop-client.x86\_64 2.6.0+cdh5.10.1+2141-1.cdh5.10.1.p0.10.el6

hadoop-conf-pseudo.x86\_64 2.6.0+cdh5.10.1+2141-1.cdh5.10.1.p0.10.el6

hadoop-debuginfo.x86\_64 2.6.0+cdh5.10.1+2141-1.cdh5.10.1.p0.10.el6

## **伪分布式安装**

伪分布式其实就是把所有的进程安装在同一台服务器上，在同一台服务器上拉起所有的进程。

### **安装**

### 我们需要安装Namenode、Seconderay Namenode、Datanode、Resourcemanager、Nodemanager、Jobhistoryserver

yum install -y hadoop-hdfs-namenode.x86\_64

yum install -y hadoop-hdfs-secondarynamenode.x86\_64

yum install -y hadoop-hdfs-datanode.x86\_64

yum install -y hadoop-yarn-resourcemanager.x86\_64

yum install -y hadoop-yarn-nodemanager.x86\_64

yum install -y hadoop-mapreduce-historyserver.x86\_64

### **重要配置**

如果是通过rpm包安装，配置文件一般都在/etc/hadoop/conf目录下

（1）core-site.xml

<property>

<name>fs.defaultFS</name>

<value>hdfs://hadoop1:8020</value>

</property>

（2）hdfs-site.xml

<property>

<name>dfs.permissions.superusergroup</name>

<value>hadoop</value>

</property>

### 本地存储路径配置

（1）hdfs-site.xml on the NameNode:

<property>

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

<value>/data/nn</value>

</property>

（2）hdfs-site.xml on each DataNode:

<property>

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

<value>/data/dn</value>

</property>

（3）在所有的namenode上创建/data/nn，在所有的datanode上创建/data/dn。归属hdfs用户。

chown -R hdfs:hdfs /data/nn /data/dn

1. 修改两个目录权限为700，只允许hdfs用户进行读写执行

chmod 700 /data/nn /data/dn

### Secondary NameNode配置

在hdfs-site.xml文件中做一下配置

<property>

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

<value>hadoop1:50070</value>

<description>

The address and the base port on which the dfs NameNode Web UI will listen.

</description>

</property>

<property>

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

<value>hadoop2:50070</value>

<description>

The address and the base port on which the dfs NameNode Web UI will listen.

</description>

</property>

### Yarn和MapReduce配置

（1）Yarn-site.xml配置

<property>

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

<value>hadoop1</value>

</property>

<property>

<description>Classpath for typical applications.</description>

<name>yarn.application.classpath</name>

<value>

$HADOOP\_CONF\_DIR,

$HADOOP\_COMMON\_HOME/\*,$HADOOP\_COMMON\_HOME/lib/\*,

$HADOOP\_HDFS\_HOME/\*,$HADOOP\_HDFS\_HOME/lib/\*,

$HADOOP\_MAPRED\_HOME/\*,$HADOOP\_MAPRED\_HOME/lib/\*,

$HADOOP\_YARN\_HOME/\*,$HADOOP\_YARN\_HOME/lib/\*

</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.local-dirs</name>

<value>/data/yarn/local</value>

</property>

<property>

<name>yarn.nodemanager.log-dirs</name>

<value>/data/yarn/logs</value>

</property>

<property>

<name>yarn.log.aggregation-enable</name>

<value>true</value>

</property>

<property>

<description>Where to aggregate logs</description>

<name>yarn.nodemanager.remote-app-log-dir</name>

<value>/var/log/hadoop-yarn/apps</value>

</property>

（2）mapred-site.xml

<property>

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

<value>yarn</value>

</property>

在所有的nodemanager上创建路径/data/yarn/local /data/yarn/logs

修改路径归属yarn用户

chown yarn:yarn /data/yarn/local /data/yarn/logs

修改权限为755

chmod 755 /data/yarn/local /data/yarn/logs

### 配置hadoop-env.sh

添加export JAVA\_HOME=/usr/java/default变量

### 配置JobHistory Server

（1）在mapred-site.xml中配置

<property>

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

<value>hadoop1:10020</value>

</property>

<property>

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

<value>hadoop1:19888</value>

</property>

### 格式化Namenode

$ sudo -u hdfs hdfs namenode -format

### 启动NameNode服务

Service hadoop-hdfs-namenode start

在浏览器里输入http://hadoop1:50070，可以看到集群的详细信息。

### 启动resourcemanager 服务

service hadoop-yarn-resourcemanager start

在浏览器里输入http://10.254.231.185:8088/，可以查看资源管理信息。

### 启动nodemanager服务

service hadoop-yarn-nodemanager start

### 启动secondarynamenode服务

service hadoop-hdfs-secondarynamenode start

### 启动historyserver服务

service hadoop-mapreduce-historyserver start

### 检查集群是否成功

/usr/java/default/bin/jps

结果如下：

2847 SecondaryNameNode

2683 DataNode

2983 JobHistoryServer

3294 ResourceManager

2754 NameNode

3076 NodeManager

## **分布式安装配置**

分布式安装就是把所有的进程安装在不同的服务武器上，在不同的服务器上拉起所有的进程，进程之间通过网络来进行通信。此处，我们通过三台虚拟机来搭建分布式环境。

### 集群规划

说明：本次实验我们定义的三台主机名分别为hadoop1、hadoop2、hadoop3对应的ip地址是根据我自己的环境来设置的，大家可以根据自己的实际情况来设置ip地址。

|  |  |  |
| --- | --- | --- |
| hadoop1(192.168.1.151) | hadoop2(192.168.1.152) | hadoop3(192.168.1.153) |
| NameNode | SecondaryNameNode | ResourceManager |
| DataNode | DataNode | DataNode |
| NodeManager | NodeManager | NodeManager |
| JobHistoryServer |  |  |

### 创建新的虚拟机

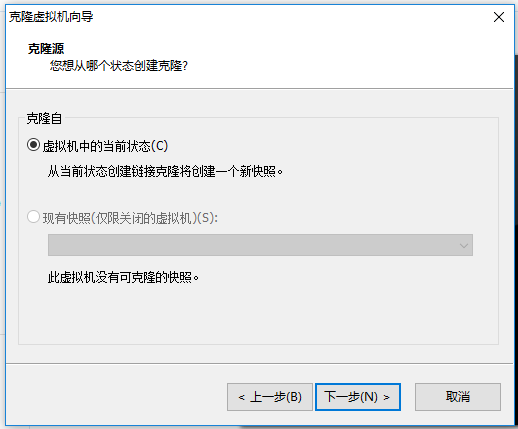
在前边的伪分布式中我们已经有一台虚拟机hadoop1，接下来我们可以在hadoop1的基础之上克隆两台虚拟机。废话不多说我们可是克隆新的虚拟机，克隆前先关闭hadoop1。



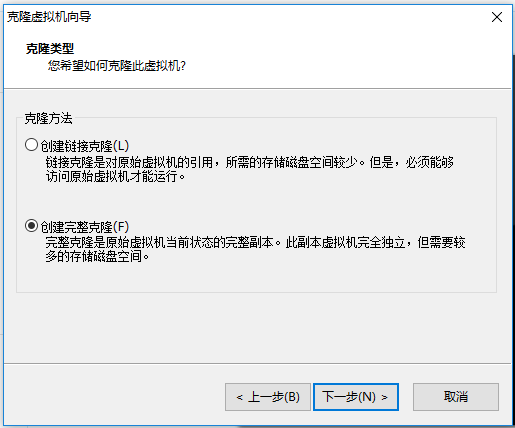
1. 右击上图中的hadoop1,在下拉框中点击“管理”，然后再点击“克隆”，进入克隆虚拟机向导界面：



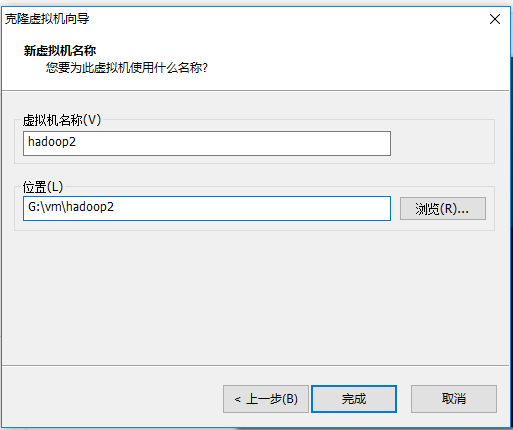
1. 点击“下一步”按钮，结果如下图：



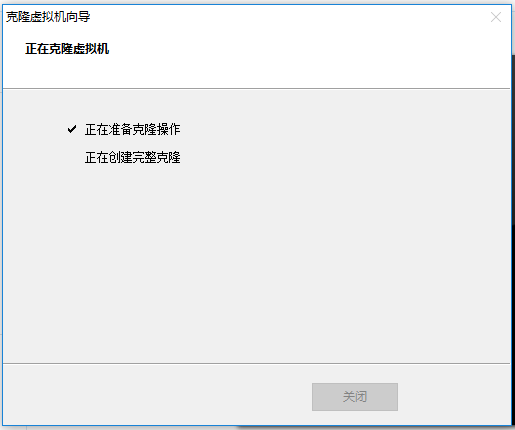
1. 点击“下一步”按钮，结果如下图：



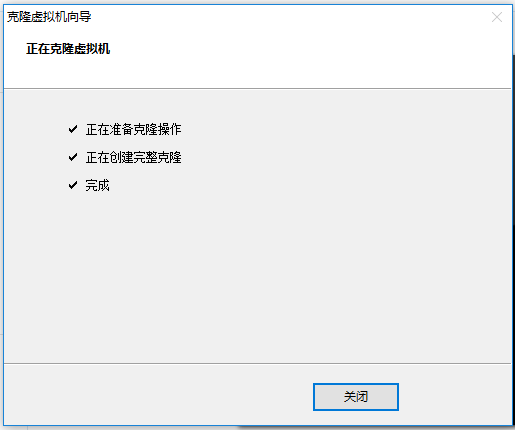
1. 选择“创建完整克隆”，然后点击“下一步按钮”，结果如下图：



1. 在虚拟机名称栏，设置虚拟机名称，此处我们设置为hadoop2,位置栏设置虚拟机创建的位置，此处我们设置为G:\vm\hadoop2，然后点击“完成”按钮，等待创建。结果如下图。



1. 创建完成后，如下图：



1. 点击“关闭”按钮。这个时候我们进入到G:\vm目录会看到多了一个hadoop2目录，此目录就是我们刚才创建的hadoop2虚拟机。

### hadoop2虚拟机配置和整理

（1）设置hadoop2的内存为2G（根据自己电脑配置进行设置，因为我的电脑内存为10G，所以hadoop1配置3G,hadoop2和hadoop3分别配置2G）。详细设置可参考vmware虚拟机创建手册第11步。

（2）进入hadoop2虚拟机，需要清除DataNode数据写入目录信息（因为目录里有DataNode的编号信息，如不清除，DataNode只能启动一个）。登陆hadoop2后切换到root账号，root密码为：cloudera(我们在创建hadoop1时设定的密码)。然后执行删除命令:

|  |
| --- |
| rm -rf /data/dn/\* |

（3）修改hadoop2系统内部的主机名为hadoop2，详细步骤参考本文档第一大项中的第1小项。

（4）设置hadoop2的mac地址

因为是克隆hadoop1，克隆后的eth0的MAC地址跟hadoop1，所以我们需要重新设定mac地址。详细步骤如下：

执行下边命令删除/etc/udev/rules.d/70-persistent-net.rules，然后重启hadoop2。

|  |
| --- |
| rm -rf /etc/udev/rules.d/70-persistent-net.rules |

Hadoop2重启后，修改新生成/etc/udev/rules.d/70-persistent-net.rules 中NAME="eth0"。

修改/etc/sysconfig/network-scripts/ifcfg-eth0 中HWADDR

跟/etc/udev/rules.d/70-persistent-net.rules中的ATTR{address}一样。也就修改mac地址一样。

（5）修改主机IP地址为我们前边规划的192.168.1.152，详细修改方法参考本文档第一大项中第2小项。

**同样的方法克隆hadoop3。**

### SSH免密码登陆配置

（1）在hadoop1、hadoop2和hadoop3的root账户下分别执行

|  |
| --- |
| ssh-keygen -t rsa -P '' |

（2）在hadoop1上执行以下命令

|  |
| --- |
| *cat /root/.ssh/id\_rsa.pub>>/root/.ssh/authorized\_keys* |

（3）把hadoop1上/root/.ssh/authorized\_keys文件拷贝到hadoop2 /root/.ssh/目录里

在hadoop1上执行以下命令，回车后输入密码：cloudera

|  |
| --- |
| *scp /root/.ssh/authorized\_keys root@192.168.1.152:/root/.ssh* |

（4）hadoop2上执行以下命令

|  |
| --- |
| *cat /root/.ssh/id\_rsa.pub>>/root/.ssh/authorized\_keys* |

（5）把hadoop2上的/root/.ssh/authorized\_keys再拷贝到hadoop3上，在hadoop2上行下边命令，回车后输入密码：cloudera

|  |
| --- |
| *scp /root/.ssh/authorized\_keys root@192.168.1.153:/root/.ssh* |

（6）hadoop3上执行以下命令

|  |
| --- |
| *cat /root/.ssh/id\_rsa.pub>>/root/.ssh/authorized\_keys* |

（7）把hadoop3上的*/root/.ssh/authorized\_keys文件分别拷贝到hadoop1和hadoop2上，在hadoop3上执行以下命令*

|  |
| --- |
| *scp /root/.ssh/authorized\_keys [root@192.168.1.151:/root/.ssh](mailto:root@192.168.1.151:/root/.ssh)* |

|  |
| --- |
| *scp /root/.ssh/authorized\_keys [root@192.168.1.152:/root/.ssh](mailto:root@192.168.1.151:/root/.ssh)* |

### 配置主机名和IP地址映射

hadoop是以主机名进行通信的，所以我们需要配置主机名与IP地址的映射。

在hadoop1、hadoop2和hadoop3的/etc/hosts文件里分别添加以下信息：

192.168.1.151 hadoop1

192.168.1.152 hadoop2

192.168.1.153 hadoop3

### 进程禁用

1. 前边我们在hadoop1上搭建了伪分布式，上边安装了hadoop集群所有的进程。根据我们分布式规划，我们需要关闭SecondaryNameNode和ResourceManager进程。

在hadoop1执行下边命令：

|  |
| --- |
| *chkconfig hadoop-hdfs-secondarynamenode off* |

|  |
| --- |
| *chkconfig hadoop-yarn-resourcemanager off* |

1. hadoop2是从hadoop1上克隆过来的，根据我们分布式规划，我们需要关闭NameNode、ResourceManager和JobHistoryServer。

|  |
| --- |
| *chkconfig hadoop-hdfs-namenode off* |

|  |
| --- |
| *chkconfig hadoop-yarn-resourcemanager off* |

|  |
| --- |
| *chkconfig hadoop-mapreduce-historyserver off* |

1. hadoop3也是从hadoop1上克隆过来的，根据我们分布式规划，我们需要关闭NameNode、SecondaryNameNode和JobHistoryServer。

|  |
| --- |
| *chkconfig hadoop-hdfs-namenode off* |

|  |
| --- |
| *chkconfig hadoop-hdfs-secondarynamenode off* |

|  |
| --- |
| *chkconfig hadoop-mapreduce-historyserver off* |

### 修改resourcemanager的主机名

修改/etc/hadoop/conf/yarn-site.xml中的yarn.resourcemanager.hostname为hadoop3(因为我们的resourcemanager在hadoop3上部署着)。

<property>

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

<value>hadoop3</value>

</property>

### 格式化Namode

在hadoop1上执行下边命令：

|  |
| --- |
| *sudo -u hdfs hdfs namenode -format* |

### 删除DataNode本地路径数据

在hadoop1、hadoop2和hadoop3上分别删除/data/dn/current，执行以下命令

|  |
| --- |
| *rm -rf /data/dn/current* |

### 启动服务

根据我们前面的规划在各节点启动相应服务

|  |
| --- |
| *service hadoop-hdfs-namenode restart* |

|  |
| --- |
| *service hadoop-hdfs-secondarynamenode restart* |

|  |
| --- |
| *service hadoop-hdfs-datanode restart* |

|  |
| --- |
| *service hadoop-yarn-resourcemanager restart* |

|  |
| --- |
| *service hadoop-yarn-nodemanager restart* |

|  |
| --- |
| *service hadoop-mapreduce-historyserver restart* |