

# Hadoop完全分布式集群搭建

hadoop运行的三种模式	
单机模式	<ol style="list-style-type: none"><li>1、默认模式。</li><li>2、不对配置文件进行修改。</li><li>3、使用本地文件系统，而不是分布式文件系统。</li><li>4、Hadoop不会启动NameNode、DataNode、JobTracker、TaskTracker等守护进程，Map()和Reduce()任务作为同一个进程的不同部分来执行的。</li><li>5、用于对MapReduce程序的逻辑进行调试，确保程序的正确。</li></ol>
伪分布式模式	<ol style="list-style-type: none"><li>1、在一台主机模拟多主机。</li><li>2、Hadoop启动NameNode、DataNode、JobTracker、TaskTracker这些守护进程都在同一台机器上运行，是相互独立的Java进程。</li><li>3、在这种模式下，Hadoop使用的是分布式文件系统，各个作业也是由JobTracker服务，来管理的独立进程。在单机模式之上增加了代码调试功能，允许检查内存使用情况，HDFS输入输出，以及其他的守护进程交互。类似于完全分布式模式，因此，这种模式常用来开发测试Hadoop程序的执行是否正确。</li><li>4、修改3个配置文件：core-site.xml（Hadoop集群的特性，作用于全部进程及客户端）、hdfs-site.xml（配置HDFS集群的工作属性）、mapred-site.xml（配置MapReduce集群的属性）</li><li>5、格式化文件系统</li></ol>

## Hadoop运行的三种模式

### 完全分布式模式

- 1、Hadoop的守护进程运行在由多台主机搭建的集群上，是真正的生产环境。
- 2、在所有的主机上安装JDK和Hadoop，组成相互连通的网络。
- 3、在主机间设置SSH免密码登录，把各从节点生成的公钥添加到主节点的信任列表。
- 4、修改3个配置文件：core-site.xml、hdfs-site.xml、mapred-site.xml，指定NameNode和JobTracker的位置和端口，设置文件的副本等参数
- 5、格式化文件系统

## hadoop完全分布式搭建流程

1、创建hadoop用户组

2、配置host

3、配置ssh

4、配置jdk

5、配置hadoop

## Linux的基础知识

Linux系统的文件组织结构	树状结构，没有windows的磁盘概念
Linux一些命令解释	sudo获取管理员权限、cd切换目录、ls -la列出当前目录中的文件、mkdir创建一个文件夹
Linux下常用的编辑器	Vim古老但功能强大，gedit图形界面编辑使用方便但只在gtk框架下的图形界面可用

## 00创建Hadoop用户组

<code>sudo useradd hadoop</code>	创建一个hadoop的用户
<code>sudo passwd hadoop</code>	为hadoop用户配置密码
<code>sudo mkdir /home/hadoop</code>	创建hadoop用户的家目录
<code>sudo chown hadoop /home/hadoop</code>	把hadoop家目录的管理权限赋给hadoop用户
<code>sudo adduser hadoop sudo</code>	把管理员权限给hadoop用户

# 01配置host

## 1、修改hostname

hostname	查看主机名称
sudo gedit /etc/hostname	修改主机名称，使其与节点名称相互一致。

## 2、修改hosts

ifconfig	查看本机的ip地址
sudo gedit /etc/hosts	修改hosts，使本机ip与hostname建立对应关系

3、将集群中其他机器的ip与hostname的映射关系写入本节点的/etc/hosts文件中

4、将本机的ip与hostname的映射关系写入集群其他节点的/etc/hosts文件中

## 集群机器配置详情

机器1	maser	ip
机器2	node1	ip
机器3	node2	ip
机器4	node3	ip

## 02安装ssh

```
sudo apt-get install ssh
```

ubuntu系统安装ssh服务

ssh的使用方法：

```
ssh user@ip
```

如：ssh pika@192.168.0.111  
pika是用户名，后面是此机器  
对应的ip地址

由于我们已经把所有节点ip与hostname映射关系，写到了每台机器的/etc/hosts中：

从而可以这么使用ssh：ssh hadoop@master 从而不用输入master节点机器对应的ip地址



# 1. ssh免密钥登录配置

1.1在终端命令行输入`ssh-keygen -t rsa -P ''` 产生一对无密码密钥`id_rsa`和`id_rsa.pub`

```
hadoop@master:~$ ssh-keygen -t rsa -P ''
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Em0QwF20AmEUvPJYzzGUjXmE3F496ndBGuef0056M1A hadoop@master
The key's randomart image is:
+---[RSA 2048]---+
|=*o+.@o  .      |
|.o. X++  .  + o  |
|  o..o=.  .  B   |
|....o..o.  . o E  |
|. + o o..S      + .|
|. . o  ..  . o o  |
|          . . . .  |
|                  o*|
|                  .o.+|
+---[SHA256]-----+
```

这对密钥会存放在一个新产生的.ssh文件夹下，该文件夹的目录结构是/home/hadoop/.ssh

下面这张图就是这个目录下的文件情况。

```
hadoop@master:~$ cd ~/.ssh
hadoop@master:~/.ssh$ ls -la
总用量 20
drwx----- 2 hadoop hadoop 4096 3月 12 00:25 .
drwxr-xr-x 20 hadoop root 4096 3月 11 23:44 ..
-rw----- 1 hadoop hadoop 1675 3月 12 00:25 id_rsa
-rw-r--r-- 1 hadoop hadoop 395 3月 12 00:25 id_rsa.pub
-rw-r--r-- 1 hadoop hadoop 888 3月 12 00:12 known_hosts
```

1.2接着在`master`节点上做如下配置，把`id_rsa.pub`追加到授权的`key`里面去。

命令：`cat id_rsa.pub >> authorized_keys`

注意：原来文件夹中没有`authorized_keys`这个文件，但现在有了。

```
hadoop@master:~/.ssh$ cat id_rsa.pub >> authorized_keys
hadoop@master:~/.ssh$ ls -la
总用量 24
drwx----- 2 hadoop hadoop 4096 3月 12 00:30 .
drwxr-xr-x 20 hadoop root 4096 3月 11 23:44 ..
-rw-r--r-- 1 hadoop hadoop 395 3月 12 00:30 authorized_keys
-rw----- 1 hadoop hadoop 1675 3月 12 00:25 id_rsa
-rw-r--r-- 1 hadoop hadoop 395 3月 12 00:25 id_rsa.pub
-rw-r--r-- 1 hadoop hadoop 888 3月 12 00:12 known_hosts
```

### 1.3重启ssh服务器: `sudo service ssh restart`

```
hadoop@master:~/.ssh$ sudo service ssh restart
[sudo] hadoop 的密码:
```

### 1.4使用`ssh-copy-id`命令将公钥传送到远程主机上, 如下图所示: `ssh-copy-id hadoop@node1 ; ssh-copy-id hadoop@node2`

```
hadoop@master:~/.ssh$ ssh-copy-id hadoop@node1
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
hadoop@node1's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'hadoop@node1'"
and check to make sure that only the key(s) you wanted were added.

hadoop@master:~/.ssh$ ssh-copy-id hadoop@node2
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
hadoop@node2's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'hadoop@node2'"
and check to make sure that only the key(s) you wanted were added.
```

1.5在master节点上验证能否免密码登陆node1和node2

```
hadoop@master:~/.ssh$ ssh hadoop@node1
```

```
hadoop@master:~/.ssh$ ssh hadoop@node2
```

1.6分别在node1和node2上，重复上述过程。这样三台机器两两之间就能实现免密钥登陆。

## 2. 配置jdk

2.1解压jdk: `tar -zxf *****.tar.gz`

```
hadoop@master:~$ cd ./下载
hadoop@master:~/下载$ ls -la
总用量 185320
drwxr-xr-x  2 hadoop hadoop      4096 3月  12 01:06 .
drwxr-xr-x 23 hadoop root        4096 3月  12 00:53 ..
-rw-r--r--  1 hadoop hadoop 189756259 3月  12 01:06 jdk-8u161-linux-x64.tar.gz
hadoop@master:~/下载$ tar -zxf jdk-8u161-linux-x64.tar.gz
hadoop@master:~/下载$ ls -la
总用量 185324
drwxr-xr-x  3 hadoop hadoop      4096 3月  12 01:08 .
drwxr-xr-x 23 hadoop root        4096 3月  12 00:53 ..
drwxr-xr-x  8 hadoop hadoop      4096 12月 20 08:24 jdk1.8.0_161
-rw-r--r--  1 hadoop hadoop 189756259 3月  12 01:06 jdk-8u161-linux-x64.tar.gz
```

2.2把jdk文件复制到`~/java`中。`cp`命令有复制内容并创建文件夹的功能，所以事先不用在`~/`中创建`java`文件夹

```
hadoop@master:~/下载$ cp -r jdk1.8.0_161 ~/.java
```

2.3添加jdk环境变量，用gedit编辑器在~/目录下新建.bashrc文件

```
hadoop@master:~$ gedit ~/.bashrc
```

在.bashrc中写如下配置文档：

```
export JAVA_HOME=/home/hadoop/.java  
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/bin  
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin
```

2.4source命令更新刚写好的配置，让/bin/bash解释器加载配置

```
hadoop@master:~$ source ~/.bashrc
```

2.5验证jdk是否安装成功。不成功的话，检查上述步骤。

```
hadoop@master:~$ javac -version  
javac 1.8.0_161
```

2.6用ssh的分发功能，把写好的.bashrc分发到node1和node2。  
使用scp命令实现，如图。

```
hadoop@master:~$ scp -r ./ .bashrc hadoop@node1:
.bashrc                                100% 152   124.3KB/s   00:00
hadoop@master:~$ scp -r ./ .bashrc hadoop@node2:
.bashrc                                100% 152   293.5KB/s   00:00
```

2.7同样把.java分发到其他节点上

```
hadoop@master:~$ scp -r ./ .java hadoop@node1:
```

```
hadoop@master:~$ scp -r ./ .java hadoop@node2:█
```



2.8从master登陆到node2，验证java是否能运行：

先输入`source ~/.bashrc` 再输入`javac -version`验证

```
hadoop@master:~$ ssh hadoop@node2
```

可能出现的问题：输入`source ~/.bashrc`后提示`source`命令不存在，这`ssh`登陆`node2`时自动启用的`shell`不是`/bin/bash`。可以用如下图中命令解决问题：

```
$ /bin/bash
hadoop@node2:~$ source ~/.bashrc
hadoop@node2:~$ javac -version
javac 1.8.0_161
```

对于`ssh`登陆节点时，默认`shell`不是`/bin/bash`时，可以用下面命令解决。

```
$ sudo usermod -s /bin/bash hadoop
```

### 3. Hadoop的安装

3.1解压下载好的hadoop，这里选择的hadoop版本是2.7.5

```
hadoop@master:~/下载$ tar -zxf hadoop-2.7.5.tar.gz
hadoop@master:~/下载$ ls -la
总用量 397180
drwxr-xr-x  4 hadoop hadoop    4096 3月 12 08:59 .
drwxr-xr-x 25 hadoop root      4096 3月 12 08:21 ..
drwxr-xr-x  9 hadoop hadoop    4096 12月 16 09:12 hadoop-2.7.5
-rw-r--r--  1 hadoop hadoop 216929574 3月 12 08:58 hadoop-2.7.5.tar.gz
drwxr-xr-x  8 hadoop hadoop    4096 12月 20 08:24 jdk1.8.0_161
-rw-r--r--  1 hadoop hadoop 189756259 3月 12 01:06 jdk-8u161-linux-x64.tar.gz
```

3.2把hadoop-2.7.5文件复制到~/.hadoop中

```
hadoop@master:~/下载$ cp -r hadoop-2.7.5 ~/.hadoop
```

### 3.3切换到hadoop的配置目录下

```
hadoop@master:~$ cd ~/.hadoop/etc/hadoop
hadoop@master:~/.hadoop/etc/hadoop$ ls -la
```

先配置hadoop的运行环境

```
hadoop@master:~/.hadoop/etc/hadoop$ gedit hadoop-env.sh
```

在hadoop-env.sh文件中找到下面这一行

```
export JAVA_HOME=${JAVA_HOME}
```

把这一行的值改为下图

```
export JAVA_HOME=/home/hadoop/.java
```

验证hadoop以及yarn能否启动

```
hadoop@master:~$ cd ~/.hadoop/bin
hadoop@master:~/.hadoop/bin$ ./hadoop
```

补充:配置hadoop要配置两类文件：运行环境配置和节点设置配置。运行环境由三个文件配置hadoop-env.sh、yarn-env.sh和mapred-env.sh  
节点配置文件是core-site.xml、hdfs-site.xml、yarn-site.xml

这里只使用最小化的配置，所以运行环境只配置这么多。

更多的配置详情还要阅读官方文档。

```
hadoop@master:~/hadoop/etc/hadoop$ gedit core-site.xml
```

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop@master:9000</value>
  </property>
  <property>
    <name>io.file.buffer.size</name>
    <value>131072</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/hadoop/.hadata/tmp</value>
  </property>
</configuration>
```

```
hadoop@master:~/hadoop/etc/hadoop$ mkdir /home/hadoop/.hadata
hadoop@master:~/hadoop/etc/hadoop$ mkdir /home/hadoop/.hadata/tmp
```

### 3.4配置core-site.xml

在core-site.xml中写配置文档。

参数含义下文会简要介绍。

注意：配置文档中指定了hadoop.tmp的位置。这个文档没有创建，因此在下面用mkdir命令创建了一下。

```
hadoop@master:~/hadoop/etc/hadoop$ gedit hdfs-site.xml
```

```
<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/home/hadoop/.hadata/name</value>
  </property>
  <property>
    <name>dfs.hosts</name>
    <value>hadoop@master,hadoop@node1,hadoop@node2</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/home/hadoop/.hadata/data</value>
  </property>
</configuration>
```

```
hadoop@master:~/hadoop/etc/hadoop$ mkdir /home/hadoop/.hadata/name
hadoop@master:~/hadoop/etc/hadoop$ mkdir /home/hadoop/.hadata/data
```

### 3.5配置hdfs-site.xml

在hdfs-site.xml中写配置文档。

注意：配置文档中指定了dfs.hadoop.namenode的位置。这个文档没有创建，因此在下面用mkdir命令创建了一下。

```
hadoop@master:~/hadoop/etc/hadoop$ gedit yarn-site.xml
```

```
<configuration>
<!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>hadoop@master</value>
  </property>
  <property>
    <name>yarn.resourcemanager.nodes.include-path</name>
    <value>hadoop@master,hadoop@node1,hadoop@node2</value>
  </property>
  <property>
    <name>yarn.nodemanager.local-dirs</name>
    <value>/home/hadoop/.hadata/yarn-local</value>
  </property>
  <property>
    <name>yarn.nodemanager.log-dirs</name>
    <value>/home/hadoop/.hadata/yarn-log</value>
  </property>
</configuration>
```

### 3.6配置yarn-site.xml

在yarn-site.xml中写配置文档。

注意：配置文档中指定了两个默认目录的位置。这个文档没有创建，因此在下面用mkdir命令创建了一下。



```
hadoop@master:~/.hadoop/etc/hadoop$ gedit slaves
```

```
hadoop@master  
hadoop@node1  
hadoop@node2
```

### 3.7配置slaves文件

关于slaves文件的作用，官网文档给出描述是：

List all slave hostnames or IP addresses in your etc/hadoop/slaves file, one per line. Helper scripts (described below) will use the etc/hadoop/slaves file to run commands on many hosts at once. It is not used for any of the Java-based Hadoop configuration. In order to use this functionality, ssh trusts (via either passphraseless ssh or some other means, such as Kerberos) must be established for the accounts used to run Hadoop.

```
hadoop@master:~$ scp -r ~/.hadoop hadoop@node1:
```

```
hadoop@master:~$ scp -r ~/.hadoop hadoop@node2:
```

```
hadoop@master:~$ scp -r ~/.hadata hadoop@node1:
```

```
hadoop@master:~$ scp -r ~/.hadata hadoop@node2:
```

3.6注意刚才所有操作都是在**master节点**中配置的。所以要分发到各个节点中去。

请问：为什么几台机器之间的用户名都为hadoop而hostname配置为master、node1、node2？

主要目的就是，保持这几台机器的文件目录一致。可以在一台机器上完成所有操作后，直接分发到各节点。



```
hadoop@master:~/hadoop/bin$ ./hdfs namenode -format
```

```
hadoop@master:~/hadoop/bin$ cd ..  
hadoop@master:~/hadoop$ cd ./sbin
```

```
hadoop@master:~/hadoop/sbin$ ./start-all.sh
```

```
hadoop@master:~/hadoop/sbin$ jps  
28048 SecondaryNameNode  
28386 NodeManager  
28453 Jps  
28236 ResourceManager  
27805 DataNode  
27646 NameNode
```

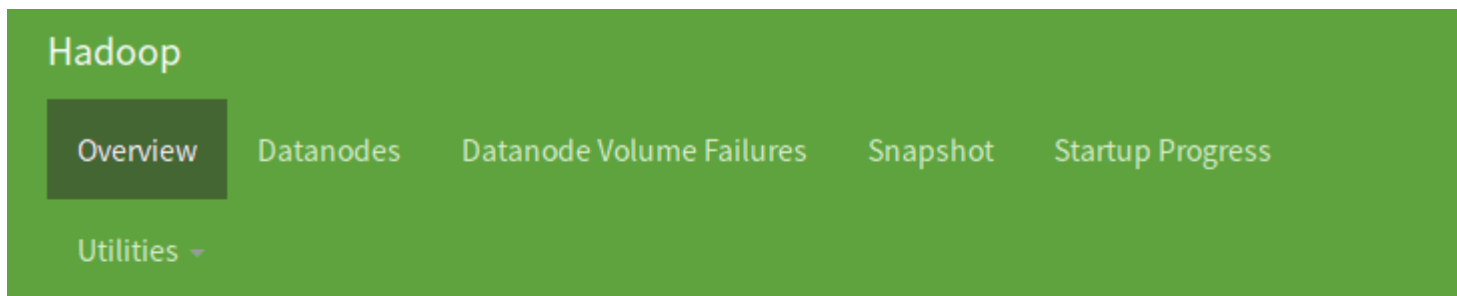
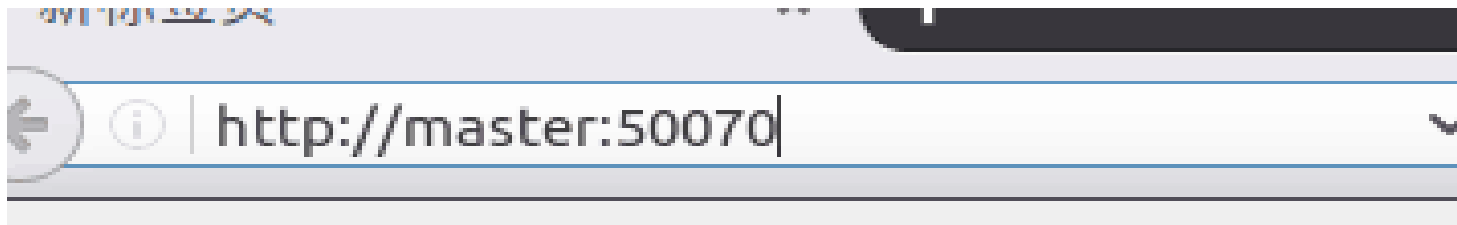
```
hadoop@node1:~$ jps  
17877 Jps  
17098 DataNode  
17311 NodeManager
```

## 4. 启动hadoop集群

启动方式是：先对文件系统进行格式化，然后切换到hadoop的sbin目录下，运行start-all.sh。

运行完毕后，输入jps命令查看个节点运行的hadoop进程。

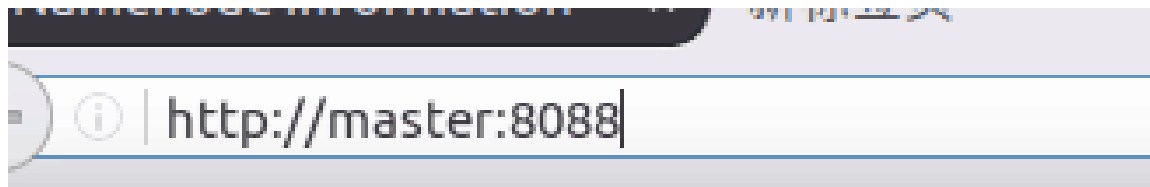
```
hadoop@node2:~$ jps  
17300 NodeManager  
18015 Jps  
17087 DataNode
```



在webui上查看hdfs  
文件系统

## Overview 'master:9000' (active)

Started:	Mon Mar 12 10:33:01 CST 2018
Version:	2.7.5, r18065c2b6806ed4aa6a3187d77cbe21bb3dba075
Compiled:	2017-12-16T01:06Z by kshvachk from branch-2.7.5



在webui上查看  
resourcemanager的运行。



## All Applications

### Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total
0	0	0	0	0	0 B	24 GB	0 B	0	24

### Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation
Fair Scheduler	[MEMORY]	<memory:1024, vCores:1

v 20 entries							
User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalS