0 环境

本教程使用 Ubuntu 14.04 64位 作为系统环境，请自行安装系统。

本教程基于原生 Hadoop 2，在 Hadoop 2.9.2 (stable) 版本下验证通过，可适合任何 Hadoop 2.x.y 版本，例如 Hadoop 2.4.1。

Hadoop版本

Hadoop 有两个主要版本，Hadoop 1.x.y 和 Hadoop 2.x.y 系列，比较老的教材上用的可能是 0.20 这样的版本。Hadoop 2.x 版本在不断更新，本教程均可适用。如果需安装 0.20，1.2.1这样的版本，本教程也可以作为参考，主要差别在于配置项，配置请参考官网教程或其他教程。

新版是兼容旧版的，书上旧版本的代码应该能够正常运行（我自己没验证，欢迎验证反馈）。

装好了 Ubuntu 系统之后，在安装 Hadoop 前还需要做一些必备工作。

设定：

master:192.168.22.100

slave1:192.168.22.77

主机192.168.22.100为master ，192.168.22.77为slave1。本文以介绍Hadoop在master上的安装与配置为主，兼顾介绍在slave1的安装和配置，其他担任slave角色的主机安装与配置同slave1。

1 创建hadoop用户

如果你安装 Ubuntu 的时候不是用的 “hadoop” 用户，那么需要增加一个名为 hadoop 的用户。

首先按 ctrl+alt+t 打开终端窗口，输入如下命令创建新用户 :

sudo useradd -m hadoop -s /bin/bash

Shell 命令

这条命令创建了可以登陆的 hadoop 用户，并使用 /bin/bash 作为 shell。

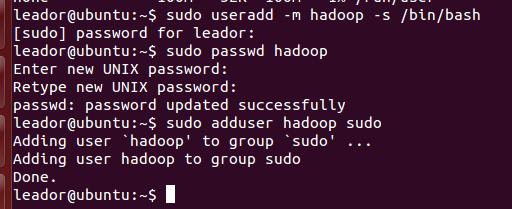
Tips: 在Ubuntu终端窗口中，复制粘贴的快捷键需要加上 shift，即粘贴是 ctrl+shift+v

接着使用如下命令设置密码，可简单设置为 hadoop，按提示输入两次密码：

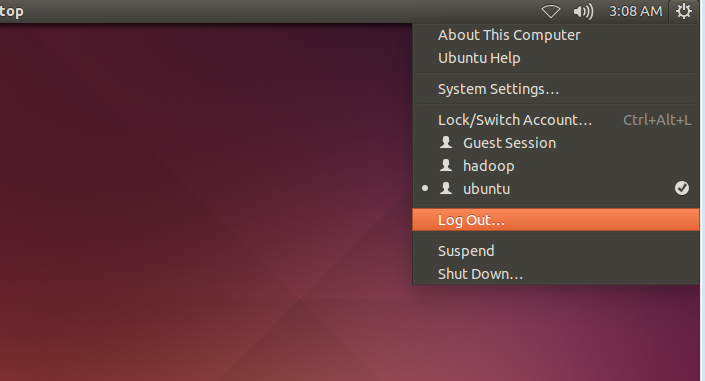
sudo passwd hadoop

可为 hadoop 用户增加管理员权限，方便部署，避免一些对新手来说比较棘手的权限问题：

sudo adduser hadoop sudo



最后注销当前用户（点击屏幕右上角的齿轮，选择注销），在登陆界面使用刚创建的 hadoop 用户进行登陆。



2 更新apt

用 hadoop 用户登录后，我们先更新一下 apt，后续我们使用 apt 安装软件，如果没更新可能有一些软件安装不了。按 ctrl+alt+t 打开终端窗口，执行如下命令：

sudo apt-get update

若出现如下 “Hash校验和不符” 的提示，可通过更改软件源来解决。若没有该问题，则不需要更改。

后续需要更改一些配置文件，我比较喜欢用的是 vim（vi增强版，基本用法相同），建议安装一下（如果你实在还不会用 vi/vim 的，请将后面用到 vim 的地方改为 gedit，这样可以使用文本编辑器进行修改，并且每次文件更改完成后请关闭整个 gedit 程序，否则会占用终端）：

sudo apt-get install vim

安装软件时若需要确认，在提示处输入 y 即可。

3 安装SSH、配置SSH免密码登陆

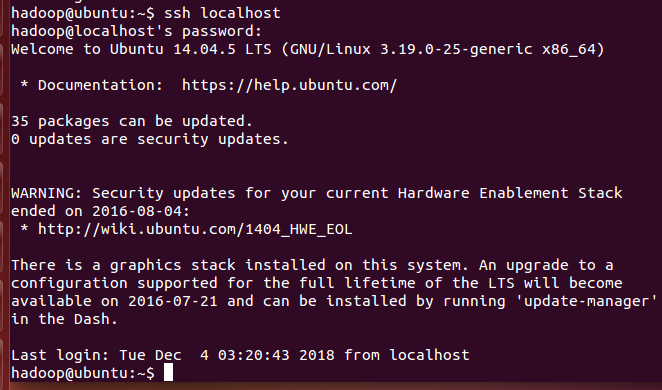
3.1 安装SSH server

集群、单节点模式都需要用到 SSH 登陆（类似于远程登陆，你可以登录某台 Linux 主机，并且在上面运行命令），Ubuntu 默认已安装了 SSH client，此外还需要安装 SSH server：

sudo apt-get install openssh-server

安装后，可以使用如下命令登陆本机：

ssh localhost



但这样登陆是需要每次输入密码的，我们需要配置成SSH免密码登陆比较方便。

3.2 配置集群中Master与Salve 互相免密登录

**(1) SSH免密码原理**

Master（NameNode | JobTracker）作为客户端，要实现免密码公钥认证，连接到服务器Salve（DataNode | Tasktracker）上时，需要在Master上生成一个密钥对，包括一个公钥和一个私钥，而后将公钥复制到所有的Slave上。当Master通过SSH连接Salve时，Salve就会生成一个随机数并用Master的公钥对随机数进行加密，并发送给Master。Master收到加密数之后再用私钥解密，并将解密数回传给Slave，Slave确认解密数无误之后就允许Master进行连接了。这就是一个公钥认证过程，其间不需要用户手工输入密码。重要过程是将客户端Master复制到Slave上。

**(2) Master机器上生成密码对**

在Master节点上执行以下命令：

ssh-keygen -t rsa

这条命是生成其免密码密钥对，询问其保存路径时直接回车采用默认路径。生成的密钥对：id\_rsa和id\_rsa.pub，默认存储在"/home/hadoop/.ssh"目录下。

首先退出刚才的 ssh，就回到了我们原先的终端窗口，然后利用 ssh-keygen 生成密钥，并将密钥加入到授权中：

第一步：

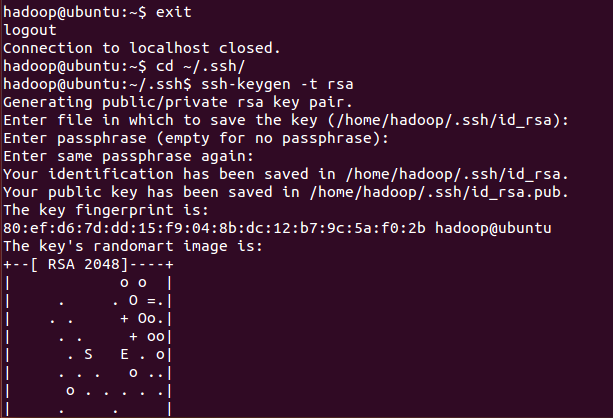
exit # 退出刚才的 ssh localhost

第二步：

cd ~/.ssh/ # 若没有该目录，请先执行一次ssh localhost

第三步：

ssh-keygen -t rsa # 会有提示，都按回车就可以



第四步：

cat ./id\_rsa.pub >> ./authorized\_keys # 加入授权



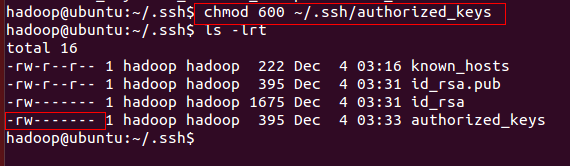
Tips: 在 Linux 系统中，~ 代表的是用户的主文件夹，即 “/home/用户名” 这个目录，如你的用户名为 hadoop，则 ~ 就代表 “/home/hadoop/”。 此外，命令中的 # 后面的文字是注释。

3.3 修改权限与配置

在验证前，需要做两件事儿。第一件事儿是修改文件"authorized\_keys"权限（权限的设置非常重要，因为不安全的设置安全设置，会让你不能使用RSA功能），另一件事儿是用root用户设置"/etc/ssh/sshd\_config"的内容。使其免密码登录有效。

**(1) 修改文件"authorized\_keys"权限**

chmod 600 ~/.ssh/authorized\_keys



**(2) 设置SSH配置**

用root用户登录服务器修改SSH配置文件"/etc/ssh/sshd\_config"的下列内容。

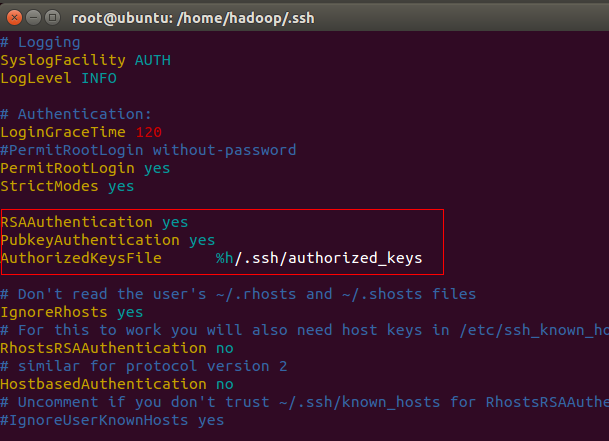
vim /etc/ssh/sshd\_config

RSAAuthentication yes # 启用 RSA 认证

PubkeyAuthentication yes # 启用公钥私钥配对认证方式

AuthorizedKeysFile .ssh/authorized\_keys # 公钥文件路径（和上面生成的文件同）



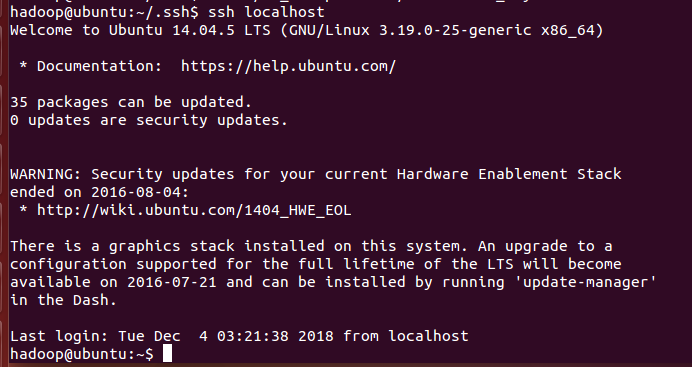


设置完之后记得重启SSH服务，才能使刚才设置有效。

sudo /etc/init.d/ssh restart

退出root登录，使用hadoop普通用户验证是否成功。

此时再用 ssh localhost 命令，无需输入密码就可以直接登陆了，如下图所示。



到目前为止Master.Hadoop的公钥也有了，文件夹".ssh"也有了，且权限也修改了。下一步就是设置master免密连接slave。

从上图中得知免密码登录本级已经设置完毕，接下来的事儿是把**公钥**复制**所有**的Slave机器上。

**(3) master免密连接Slave**

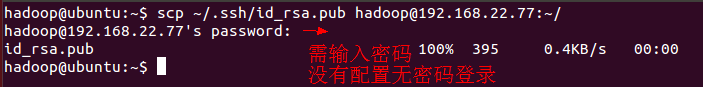
在master上使用下面的命令复制公钥到slave上：

scp ~/.ssh/id\_rsa.pub 远程用户名@远程服务器IP:~/

例如：

scp ~/.ssh/id\_rsa.pub hadoop@192.168.22.77:~/

　　上面的命令是**复制**文件"**id\_rsa.pub**"到服务器IP为"**192.168.22.77**"的用户为"**hadoop**"的"**/home/hadoop/**"下面。



输入密码后，可以到192.168.22.77上相应目录下有没有id\_rsa.pub文件。下图是远程访问查看截图，文件已成功拷贝到目标主机上。



从上图中我们得知，已经把文件"id\_rsa.pub"传过去了，因为并没有建立起免密码连接，所以在连接时，仍然要提示输入输入192.168.22.77服务器用户hadoop的密码。为了确保确实已经把文件传过去了，用XShell登录192.168.22.77服务器，查看"/home/hadoop/"下是否存在这个文件。从上面得知我们已经成功把公钥复制过去了。

下面对192.168.22.77做一些配置，使得master免密访问以192.168.22.77为代表的slave。

在"/home/hadoop/"下创建".ssh"文件夹。这一步并不是必须的，如果在Slave1.Hadoop的"/home/hadoop"已经存在就不需要创建了，因为我们之前并没有对Slave机器做过免密码登录配置，所以该文件是不存在的。用下面命令进行创建。（备注：用hadoop登录系统，如果不涉及系统文件修改，一般情况下都是用我们之前建立的普通用户hadoop进行执行命令。）

mkdir ~/.ssh

然后是修改文件夹".ssh"的用户权限，把他的权限修改为"700"，用下面命令执行：

chmod 700 ~/.ssh

重要提醒：如果不进行，即使你按照前面的操作设置了"authorized\_keys"权限，并配置了"/etc/ssh/sshd\_config"，还重启了sshd服务，在Master能用"ssh localhost"进行免密码登录，但是对192.168.22.77.Hadoop进行登录仍然需要输入密码，就是因为".ssh"文件夹的权限设置不对。这个文件夹".ssh"在配置SSH免密码登录时系统自动生成时，权限自动为"700"，如果是自己手动创建，它的组权限和其他权限都有，这样就会导致RSA免密码远程登录失败。

把从Master.Hadoop上scp过来的的公钥**追加**到Slave1.Hadoop的授权文件"authorized\_keys"中去。使用下面命令进行追加并修改"authorized\_keys"文件权限：

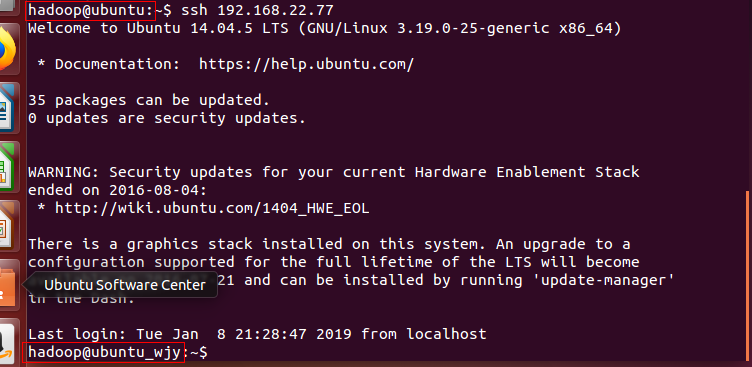
在slave1上执行如下命令：

cat ~/id\_rsa.pub >> ~/.ssh/authorized\_keys # id\_rsa.pub文件是之前从master上scp过来的。

chmod 600 ~/.ssh/authorized\_keys

接下来很重要但容易被忽略的一步，像在master上一样，用root用户设置"/etc/ssh/sshd\_config"的内容并重启ssh服务，使slave免密码登录有效。操作参考上文的步骤。

验证master免密登录slave1



master.hadoop免密登录slave1成功。最后，记得删除slave1.hadoop的~/ id\_rsa.pub文件。可以参考以上步骤使得master可以免密登录其他的slave。

**（4）配置所有Slave免密码登录Master**

　　和Master免密码登录所有Slave原理一样，就是把Slave的公钥追加到Master的".ssh"文件夹下的"authorized\_keys"中，记得是追加（>>）。

为了说明情况，我们现在就以"Slave1.Hadoop"免密码登录"Master.Hadoop"为例，进行一遍操作，也算是巩固一下前面所学知识，剩余的"Slave2.Hadoop"和"Slave3.Hadoop"就按照这个示例进行就可以了。

　　首先创建"Slave1.Hadoop"自己的公钥和私钥，并把自己的公钥追加到"authorized\_keys"文件中。用到的命令如下：

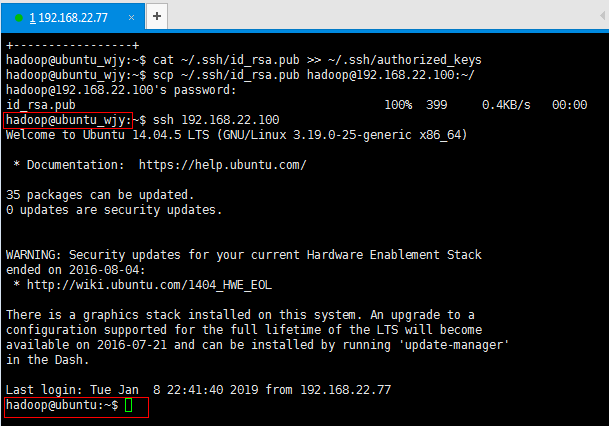
ssh-keygen –t rsa –P ''

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

接着是用命令"scp"复制"Slave1.Hadoop"的公钥"id\_rsa.pub"到"Master.Hadoop"的"/home/hadoop/"目录下，并追加到"Master.Hadoop"的"authorized\_keys"中。

然后删除掉刚才复制过来的"id\_rsa.pub"文件。

最后是测试从"Slave1.Hadoop"到"Master.Hadoop"免密码登录。



至此"Master.Hadoop"与"Slave1.Hadoop"之间可以互相免密码登录了，剩下的就是按照上面的步骤把剩余的"Slave2.Hadoop"和"Slave3.Hadoop"与"Master.Hadoop"之间建立起免密码登录。这样，Master能免密码验证登录每个Slave，每个Slave也能免密码验证登录到Master。

4 安装Java环境

**(1)官网下载JDK**

地址: http://www.oracle.com/technetwork/articles/javase/index-jsp-138363.html

选择相应的 .gz包下载

**（2）安装**

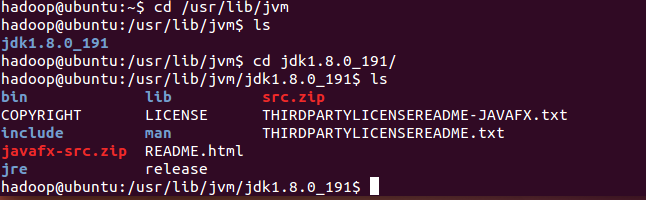
解压缩,放到指定目录(以jdk-8u191-linux-x64.tar.gz为例)

创建目录:

sudo mkdir /usr/lib/jvm

加压缩到该目录:

sudo tar -zxvf /data/jdk-8u191-linux-x64.tar.gz -C /usr/lib/jvm



**（3）修改环境变量**

在"/etc/profile"文件的尾部添加以下内容：

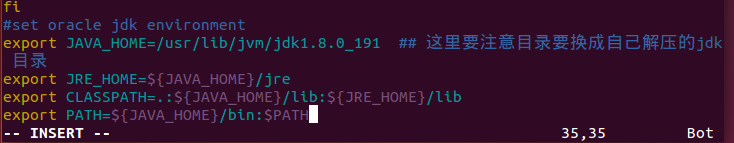
#set oracle jdk environment

export JAVA\_HOME=/usr/lib/jvm/jdk1.8.0\_191 ## 这里要注意目录要换成自己解压的jdk 目录

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

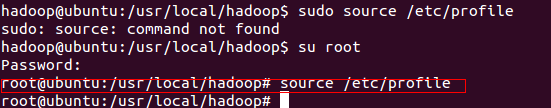
export PATH=${JAVA\_HOME}/bin:$PATH



使配置生效

保存并退出，执行下面命令使其配置立即生效。切换到root用户使用source命令

source /etc/profile

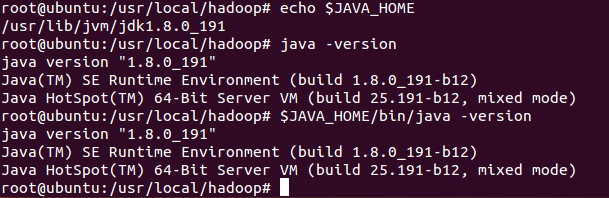


**（4）测试java环境**

echo $JAVA\_HOME # 检验变量值

java -version

$JAVA\_HOME/bin/java -version # 与直接执行 java -version 一样

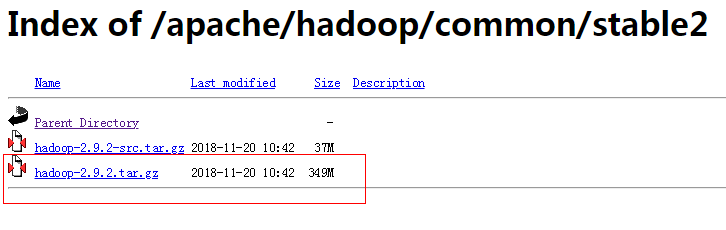


5 安装hadoop

Hadoop 2 可以通过 http://mirror.bit.edu.cn/apache/hadoop/common/ 或者 http://mirrors.cnnic.cn/apache/hadoop/common/ 下载，一般选择下载最新的稳定版本，即下载 “stable” 下的hadoop-2.x.y.tar.gz 这个格式的文件，这是编译好的，另一个包含 src 的则是 Hadoop 源代码，需要进行编译才可使用。

本文涉及的文件放在/data下（若不是请自行更改 tar 命令后的相应目录）。另外，本教程选择的是 2.9.2 版本，如果你用的不是 2.9.2 版本，则将所有命令中出现的 2.9.2.0更改为你所使用的版本。





我们选择将 Hadoop 安装至 /usr/local/ 中：

sudo tar -zxf /data/hadoop-2.9.2.tar.gz -C /usr/local # 解压到/usr/local中

cd /usr/local/

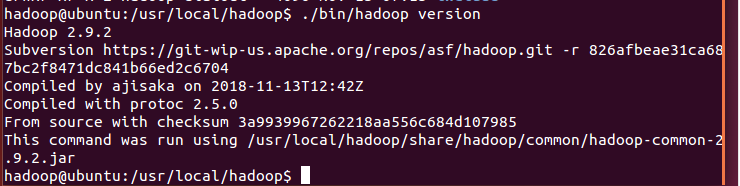
sudo mv ./hadoop-2.9.2/ ./hadoop # 将文件夹名改为hadoop

sudo chown -R hadoop ./hadoop # 修改文件权限（非常重要）

Hadoop 解压后即可使用。输入如下命令来检查 Hadoop 是否可用，成功则会显示 Hadoop 版本信息：

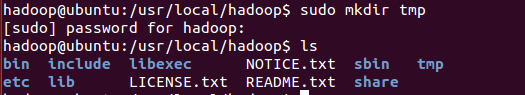
cd /usr/local/hadoop

./bin/hadoop version



最后在"/usr/local/hadoop"下面创建tmp文件夹，把Hadoop的安装路径添加到"/etc/profile"中，修改"/etc/profile"文件（配置java环境变量的文件），将以下语句添加到末尾，并使其有效：

sudo mkdir tmp



sudo vim /etc/profile

文件末添加以下内容

# set hadoop path

export HADOOP\_HOME=/usr/local/hadoop

export PATH=$PATH:$HADOOP\_HOME/bin



使配置生效。切换到root用户执行下面的命令。

source /etc/profile

6 配置hadoop

6.1 创建需要的目录与修改权限

在配置之前，我们先预先创建几个目录，这些目录需要被写入到配置文件中。

sudo mkdir -p /data/hadoop/hdfs/nn

sudo mkdir -p /data/hadoop/hdfs/dn

sudo mkdir -p /data/hadoop/hdfs/snn

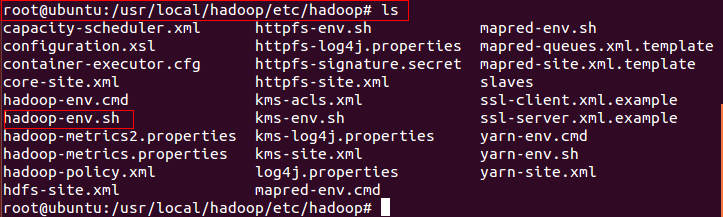
sudo mkdir -p /data/hadoop/yarn/nm

修改权限（非常重要）

sudo chmod -R 777 /data

6.2 配置hadoop-env.sh

该"hadoop-env.sh"文件位于"/usr/local/hadoop/etc/hadoop"目录下。



在文件的末尾添加下面内容。

# set java environment

export JAVA\_HOME=/usr/lib/jvm/jdk1.8.0\_191 ## 这里要注意目录要换成自己解压的jdk 目录

6.3 配置core-site.xml文件

　　修改Hadoop核心配置文件core-site.xml，这里配置的是HDFS的地址和端口号。(-备注：请先在 /usr/local/hadoop 目录下建立 tmp 文件夹)

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://192.168.22.100:9000</value>

</property>

<property>

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

<value>/usr/local/hadoop/tmp</value>

<description>A base for other temporary directories.</description>

</property>

</configuration>

备注：如没有配置hadoop.tmp.dir参数，此时系统默认的临时目录为：/tmp/hadoo-hadoop。而这个目录在每次重启后都会被删除，必须重新执行format才行，否则会出错。

6.4 配置hdfs-site.xml文件

修改Hadoop中HDFS的配置，配置的备份方式默认为3。

<configuration>

<property>

<name>dfs.nameservices</name>

<value>hadoop-cluster</value>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>file:///data/hadoop/hdfs/nn</value>

</property>

<property>

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

<value>file:///data/hadoop/hdfs/snn</value>

</property>

<property>

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

<value>file:///data/hadoop/hdfs/snn</value>

</property>

<property>

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

<value>file:///data/hadoop/hdfs/dn</value>

</property>

</configuration>

一个问题：

你可能会发现，name标签下都是以dfs.\*的形式，只有一个fs.datanode.data.dir是以fs开头。这里我在配置的时候遇到一个问题，就是当使用dfs.datanode.data.dir的时候，从节点就是起不来。当改成fs.datanode.data.dir就有用了。但是官方给出的文档确实就是这个呀！所以很邪乎。因为只有2.0版本之前是fs。

6.5 配置mapred-site.xml文件

注意：如果在刚解压之后，是没有这个文件的，需要将mapred-site.xml.template复制为mapred-site.xml。

cd /usr/local/hadoop/etc/hadoop

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

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

6.6 配置yarn-site.xml

<configuration>

<property>

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

<value>192.168.22.100</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property>

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

<value>file:///data/hadoop/yarn/nm</value>

</property>

</configuration>

6.7 配置slaves文件

有两种方案：

（1）第一种

去掉"localhost"，每行只添加一个主机名，把剩余的Slave主机名都填上。

例如：添加形式如下

Slave1.Hadoop

Slave2.Hadoop

Slave3.Hadoop

（2）第二种（推荐使用，本文也是使用这一种）

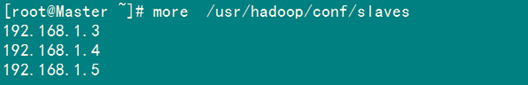
去掉"localhost"，加入集群中所有Slave机器的IP，也是每行一个。

例如：添加形式如下

192.168.1.3

192.168.1.4

192.168.1.5



7 分发Master主机上Hadoop安装目录到所有Slave

到这一步，一台Maeter主机安装配置好了，剩下的工作是完成其他hadoop集群主机上hadoop的安装配置。请参考1到4完成集群主机创建hadoop用户、ssh免密互相登录和java环境的安装，第5到6步略过，然后进行下面的操作。

将 Master上配置好的hadoop所在文件夹"/usr/local/hadoop"复制到所有的Slave的"/usr/local"目录下（实际上Slave机器上的slavers文件是不必要的， 复制了也没问题）。这种方式使得配置只在一台master主机上进行，其他主机只需修改少量配置甚至无需修改（如果java安装目录不同，修改6.1中的配置，相同则无需修改）。用下面命令格式进行。（备注：此时用户可以为hadoop也可以为root）

scp -r /usr/local/hadoop root@服务器IP:/usr/local/

hadoop文件夹确实已经复制了，但是我们发现hadoop权限是root，所以我们现在要给slave主机上的用户hadoop添加对"/usr/hadoop"的权限(非常重要)。

sudo chown -R hadoop ./hadoop

接着在slave上修改"/etc/profile"文件（配置 java 环境变量的文件），将以下语句添加到末尾，并使其有效（source /etc/profile）：

# set hadoop environment

export HADOOP\_HOME=/usr/hadoop

export PATH=$PATH :$HADOOP\_HOME/bin

最后别忘了，直接从master主机上scp过来的hadoop文件夹中配置文件中一些目录还没有创建。按照6.1目录创建和修改权限。按照这个步骤将hadoop文件夹分发到所有集群主机并配置好后，hadoop集群基本安装配置算是完成了。

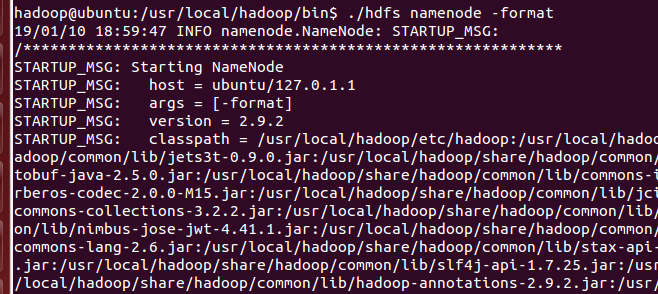
8 启动hadoop

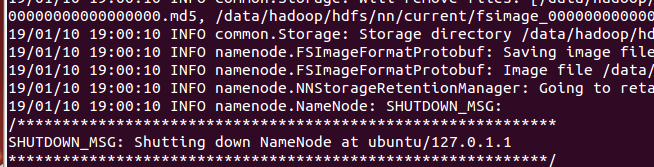
在master节点启动hadoop服务，各个从节点会自动启动。

（1）初始化，

cd /usr/local/hadoop/bin

./hdfs namenode -format

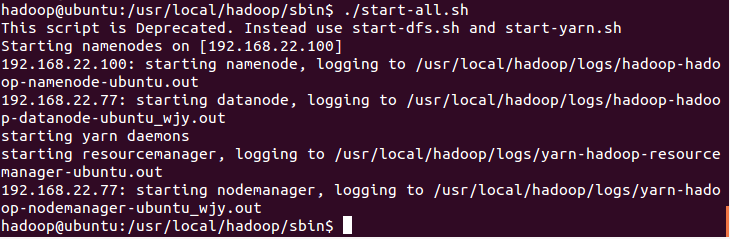




（2）启动

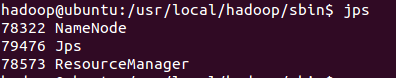
cd /usr/local/hadoop/sbin

./start-all.sh

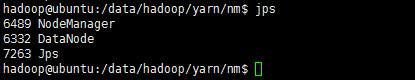


（3）检查启动是否成功

输入jps命令查看相关信息，master上截图如下：

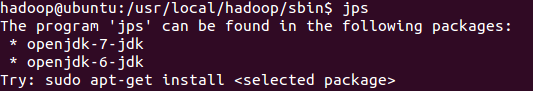


slave节点上输入jps查看：



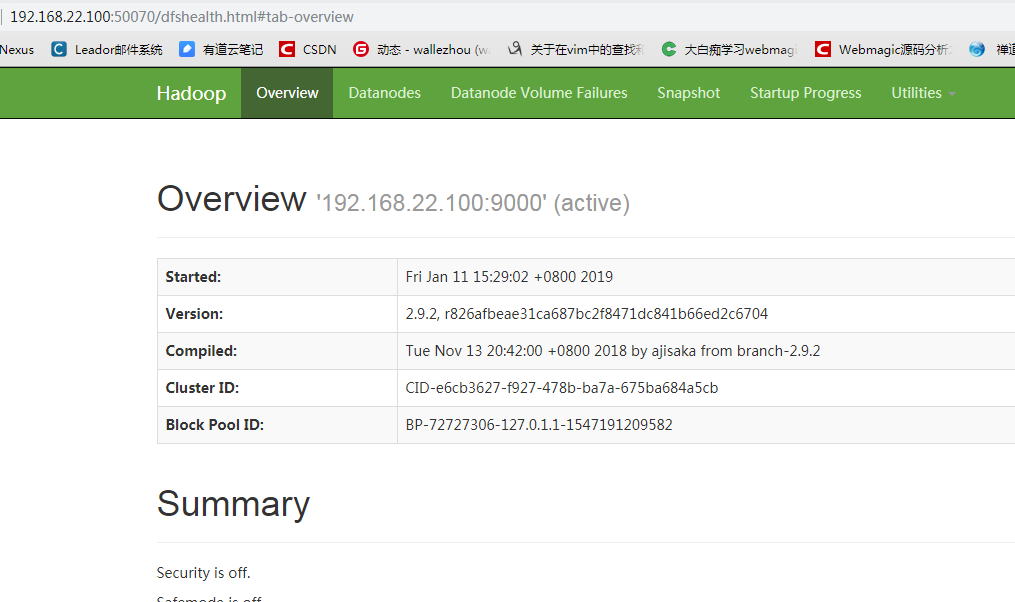
一个问题：

如果你运行jps出现如下情况

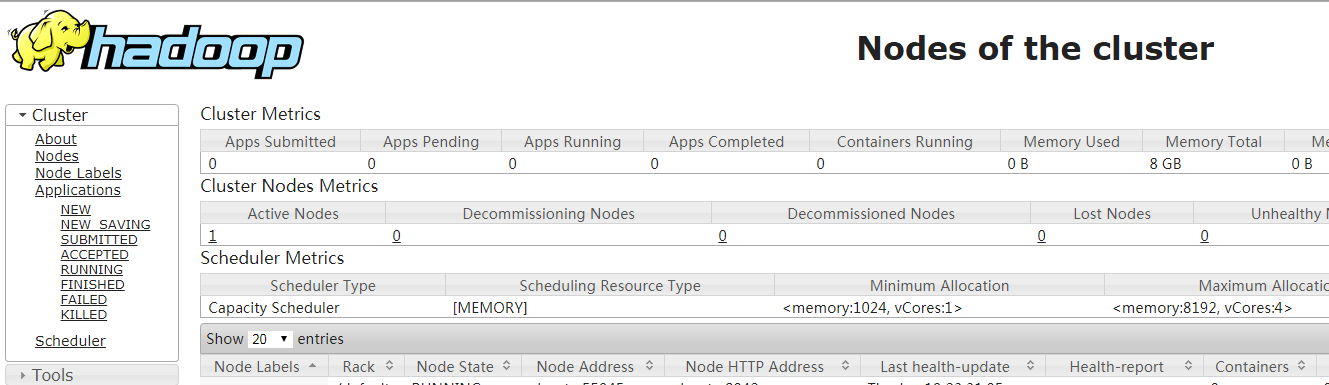


请另开一个terminal，然后执行jps就正常了。这很奇怪，jps是java自带的java进程管理工具，按道理java环境在正确配置后是可以识别jps命令的，但是终端在开启hadoop后就无法识别了。另开一个终端又可以使用jps工具了。原因不知，希望有知道的大神可以补充。

（4）访问，输入http://192.168.10.128:50070，看到如下界面：



访问http://192.168.22.100:8088，看到如下界面：



好了。如果以上都成功，那么基本上完成了hadoop集群的搭建；

（5）关闭hadoop

cd /usr/local/hadoop/sbin

./stop-all.sh