# 系统

centos7-64位

# 软件

jdk-7u55-linux-x64.rpm

hadoop-2.7.1.tar.gz

备注:hadoop271必须是1.6以上JDK

# 电脑基本信息设置

## Hostname和ip地址域名映射设置。

对于腾讯云服务器申请的时候，CentOS服务器默认分配了内网ip和外网ip，在上面部署Hadoop的时候统一使用内网IP，对于集群中的每台电脑，通过如下命令编辑hosts文件

# vi /etc/hosts

按i键进入编辑模式，编辑完成后按esc键，进入命令模式，在命令模式下输入:wq 表示保存并退出。输入:q!表示退出不保存。hosts 配置文件是用来把主机名字映射到IP地址的方法，但这种映射只是本地机的映射，也就是说每台机器都是独立的。在hosts文件中编辑如下

10.104.139.222 hadoopmaster hadoopmaster

10.104.139.223 hadoopslave1 hadoopslave1

10.104.139.224 hadoopslave2 hadoopslave2

如果是云服务器，那么在云服务器上有外网ip和内网ip，此处是内网ip

每行为一个主机，每行由三部份组成，每个部份由空格隔开。其中#号开头的行是注释

第一部份：网络IP地址；第二部份：域名，第三部分是主机名。

如果域名和主机名相同，那么可以只写前两部分，如果不同，则必须全写。

主机名就机器本身的名字，域名是用来解析到IP的。但值得一说的是在局域网中，主机名也是可以解析到IP上的。局域网中的所有主机都有一个通用的，并且包含所有主机的/etc/hosts文件，就可以相互通过主机名访问了。

Linix可以通过hostname命令直接查看主机名， 如果要修改linix的主机名，可以通过编辑hostname文件实现。输入如下命令：

vi /etc/hostname

#编辑配置文件，输入后保存即可。

通过上面的设置，完成了每台电脑的域名、主机名配置，并且相互直接可以互访了。

# 部署步骤

## 安装JDK

1. **对每台电脑，在JDK的存放目录下执行如下命令，安装JDK：**

rpm -ivh jdk-7u55-linux-x64.rpm

1. **对每台电脑，编辑profile文件，配置环境变量**

vi /etc/profile

打开后，按i键进入编辑模式，在文件的最后添加如下变量。

**JAVA\_HOME=/usr/java/jdk1.7.0\_55**

**JRE\_HOME=/usr/java/jdk1.7.0\_55/jre**

**PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin**

**CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib**

**export JAVA\_HOME JRE\_HOME PATH CLASSPATH**

此时可以顺便，把HADOOP\_HOME变量添加进去，添加进去后，就可以直接调用hadoop的命令了，例如hdfs dfs haddop等方法。具体编辑如下如下：必须是hadoop的具体位置。

JAVA\_HOME=/usr/java/jdk1.7.0\_55

HADOOP\_HOME=/home/hadoop/hadoop-2.7.1

JRE\_HOME=/usr/java/jdk1.7.0\_55/jre

PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin:$HADOOP\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

export JAVA\_HOME HADOOP\_HOME JRE\_HOME PATH CLASSPATH

安装完java后，可以检查安装的java版本。

#java -version 和javac -verision 版本号如果不一致，则需要修改一致

**rm -rf /usr/bin/java**

**rm -rf /usr/bin/javac**

**rm –rf是删除文件夹的意思。**

**ln -s /usr/java/jdk1.7.0\_55/bin/java/usr/bin/java**

**ln -s /usr/java/jdk1.7.0\_55/bin/javac/usr/bin/javac**

**ln –s是建立文件连接的意思，类似于windows的快捷方式**

1. **#让/etc/profile文件修改后立即生效 ,可以使用如下命令:**

source /etc/profile

#上述生效命令，如果profile中有语法错误，则会出错。

1. **开启ssh**

查看SSH是否安装。（腾讯云CentOS服务器默认都已经安装并启用）

rpm -qa | grep ssh

注：若没安装SSH则可输入：yum install openssh-server安装。

启动SSH服务。

service sshd restart (重启SSH服务)

service sshd start 启动服务

service sshd stop 停止服务

netstat -antp | grep sshd 查看是否启动22端口

如何设置SSH服务为开机启动？

chkconfig sshd on

注：若是chkconfig sshd off则禁止SSH开机启动。

1. **生成本机密匙，然后合并SSH密匙**

**用如下命令，对集群中的每台电脑节点，生成本机密匙:**

**ssh-keygen -t rsa**

**#在执行上述命令的时候，会提示用户输入文件名等信息，这时直接按回车键，需要输入密码时也不输入密码, 敲回车键即可, 大约连续按3次回车键，生成完成。**

**对每台电脑都生成完成后，在master计算机上合并密匙(/root/.ssh目录):**

#可以执行下面的命令，将他们合并到到一起。

#先将master机的密钥添加到authorized\_keys文件，执行命令

**cat id\_rsa.pub>> authorized\_keys**

**#再将集群中每台电脑的密钥添加到authorized\_keys文件中。通过ssh 用户名@节点ip连接执行合并。**

**ssh root@192.168.0.183 cat ~/.ssh/id\_rsa.pub>> authorized\_keys**

**ssh root@192.168.0.184 cat ~/.ssh/id\_rsa.pub>> authorized\_keys**

**ssh root@** **119.29.14.79 cat ~/.ssh/id\_rsa.pub>> authorized\_keys**

**ssh root@** **119.29.36.87 cat ~/.ssh/id\_rsa.pub>> authorized\_keys**

**#把Master服务器的authorized\_keys、known\_hosts复制(**在/root/.ssh目录**)到每台Slave服务器的/root/.ssh目录。可以通过scp命令直接传送过去，具体方法是：**

**scp -r /root/.ssh/authorized\_keys 192.168.0.183:/root/.ssh/**

**scp -r /root/.ssh/known\_hosts 192.168.0.183:/root/.ssh/**

# 安装hadoop271

1. **创建用户和组**

给每个电脑创建hadoop用户:

adduser  hadoop

设置密码:

passwd hadoop

输入密码：hadoop，会提示再次输入密码

创建hadoop用户组:

addgroup hadoop

#也可以为groupadd命令， 如果还不行，切换到超级用户权限 su -

用户相关命令：

groups 查看当前登录用户的组内成员  
groups gliethttp 查看gliethttp用户所在的组,以及组内成员  
whoami 查看当前登录用户名

/etc/group文件包含所有组  
/etc/shadow和/etc/passwd系统存在的所有用户名

通过如下命令可以查看所有的用户组：cat /etc/group

把hadoop用户加到hadoop组(第一个hadoop是组,第二个是用户名)

usermod -G hadoop hadoop

修改hadoop目录

chown  hadoop:hadoop /home/hadoop(/home/hadoop目录所有者和组改为hadoop)

chmod -R 777 /home/hadoop /home/hadoop下所有的文件和文件夹及其子文件夹属性为可写可读可执行

1. **安装配置Hadoop(只需要配置name node即可,我的name node是192.168.141.10)**
2. 把hadoop-2.7.1.tar.gz 放/home/hadoop下面
3. mkdir tmp
4. mkdir hdfs
5. mkdir hdfs/data
6. mkdir hdfs/name
7. 解压tar -xzvf hadoop-2.7.0.tar.gz
8. 配置/home/hadoop/hadoop-2.7.1/etc/hadoop目录下的core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://192.168.141.10:9999</value>

</property>

<property>

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

<value>file:/home/hadoop/myhadoop/tmp</value>

</property>

<property>

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

<value>131702</value>

</property>

</configuration>

配置文件说明：

1. fs.defaultFS 是默认的Master主机节点访问地址和端口
2. hadoop.tmp.dir是hadoop处理数据的临时目录
3. hdfs://192.168.141.10是master机的地址，--------缓存大小,128KB
4. 配置/home/hadoop/hadoop-2.7.1/etc/hadoop目录下的hdfs-site.xml

<configuration>

<property>

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

<value>file:/home/hadoop/myhadoop/dfs/name</value>

</property>

<property>

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

<value>file:/home/hadoop/myhadoop/dfs/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value> -----datanode备份.默认备份3

</property>

<property>

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

<value>192.168.141.10:9001</value>----------namenode备份地址，

-----------此处和master机设置相同

</property>

<property>

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

<value>true</value> ---------------------开启web访问

</property>

<property>

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

<value>10.104.139.222:50070</value> ----web访问的ip地址，master机地址

</property>

<property>

<name>dfs.namenode.datanode.registration.ip-hostname-check</name>

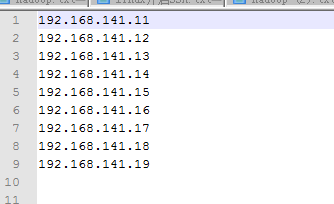
<value>false</value>

</property>

</configuration>

说明：dfs.namenode.datanode.registration.ip-hostname-check 设置为false，否则运行时可能出错。

1. 配置/home/hadoop/hadoop-2.7.1/etc/hadoop目录下的slaves文件,编辑datanode地址如下：



1. 配置/home/hadoop/hadoop-2.7.1/etc/hadoop目录下的yarn-site.xml

<configuration>

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

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>192.168.141.10:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>192.168.141.10:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>192.168.141.10:9031</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>192.168.141.10:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>192.168.141.10:8088</value>

</property>

<property>

<name>yarn.nodemanager.resource.memory-mb</name>

<value>3072</value>----java默认请求是1530M内存,如果不通过参数修改,此处必须不能小于1530

</property>

</configuration>

1. 配置/home/hadoop/hadoop-2.7.1/etc/hadoop目录下的mapred-site.xml

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>10.104.139.222:9001</value>

</property>

<property>

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

<value>yarn</value>

</property>

<property>

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

<value>10.104.139.222:10020</value>

</property>

<property>

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

<value>10.104.139.222:19888</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

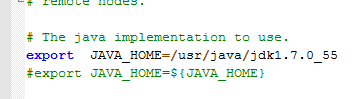
</property>

</configuration>

说明：上面红色dfs.permissions红色表示访问dfs不进行权限控制，方便远程开发的时候远程操作，实际部署的时候要删除。

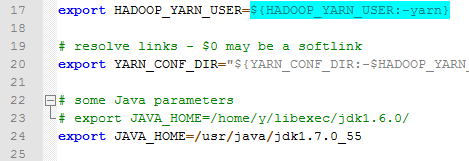
mapred.job.tracker表示任务配发服务器的地址和端口，在其他附属软件配置时经常需要。

1. 配置/home/hadoop/hadoop-2.7.1/etc/hadoop目录下hadoop-env.sh、yarn-env.sh的JAVA\_HOME



实际有时候需要修改Hadoop\_Conf\_dir的值为hadoop的安装位置，如下：

export HADOOP\_CONF\_DIR=/home/hadoop/hadoop-2.7.1/etc/hadoop



1. 将配置好的Hadoop复制到各个节点对应位置上，通过scp传送，通过scp传送之前，首先将前面复制过来的压缩安装包hadoop-2.7.0.tar.gz，删除或者移动到其它目录，安装完成后就可以删除了，否则它会被一起传送到其他的电脑节点，浪费传输时间。

scp -r /home/hadoop 192.168.141.11:/home/

scp -r /home/hadoop 192.168.141.12:/home/

1. 在Master服务器启动hadoop，从节点会自动启动，进入/home/hadoop/hadoop-2.7.1目录

(1)初始化bin/hdfs namenode -format

(2)全部启动sbin/start-all.sh

注意上面这个全部启动的脚本虽然可以用，但是hadoop官方已经弃用，建议使用下面的两个脚本命令执行。This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh

sbin/start-dfs.sh

sbin/start-yarn.sh

(3)停止sbin/stop-all.sh

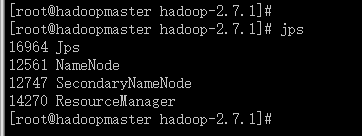
注意上面这个全部启动的脚本虽然可以用，但是hadoop官方已经弃用，建议使用下面的两个脚本命令执行。This script is Deprecated. Instead use stop-dfs.sh and stop-yarn.sh

sbin/stop-dfs.sh

sbin/stop-yarn.sh

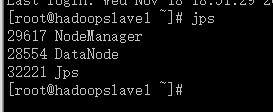
(4)输入jps命令,查看系统已经启动的进程是否有hadoop

如果master机有如下提示，表示成功：



因为master机并没有设置为datanode节点，所以上面没有datanode服务。

(5) 在datanode节点上输入jps，查看是否有datanode服务，显示如下表示成功：



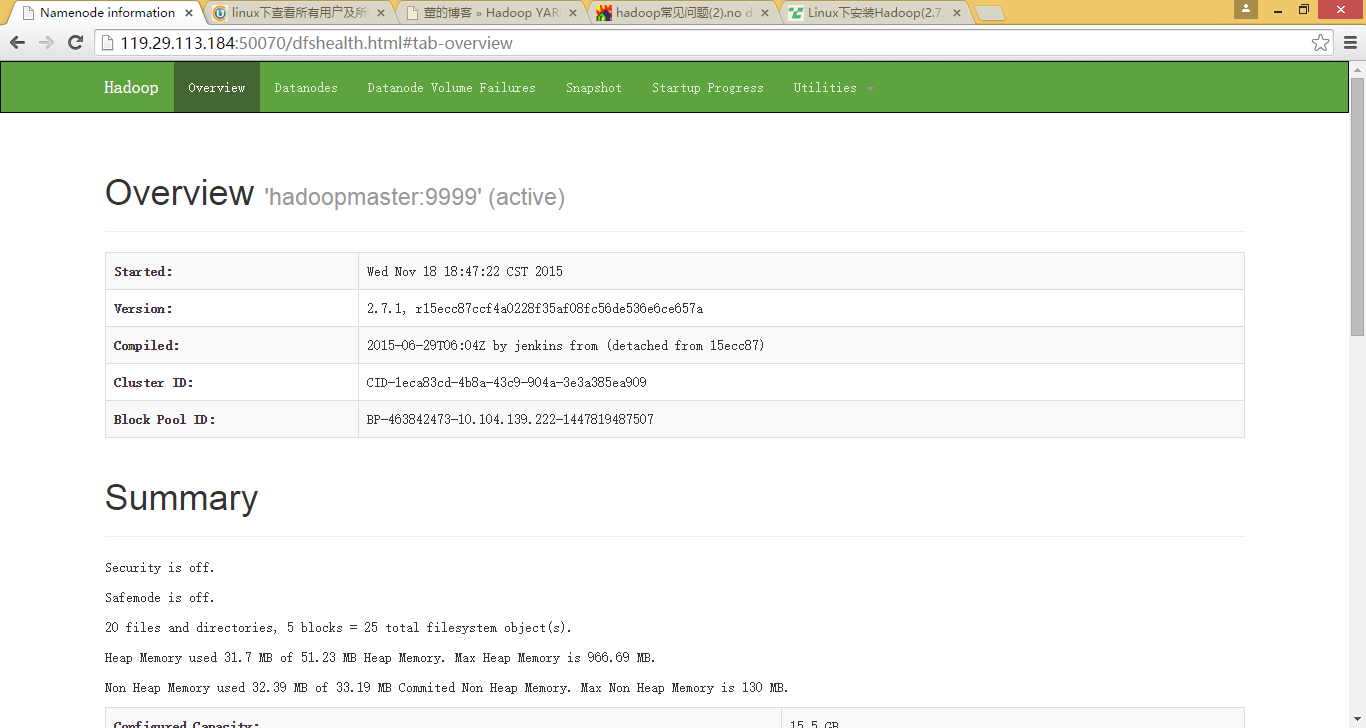
1. Web访问，要先开放端口或者直接关闭防火墙

(1)输入命令，systemctl stop firewalld.service

(2)浏览器打开http://192.168.141.10:8088/

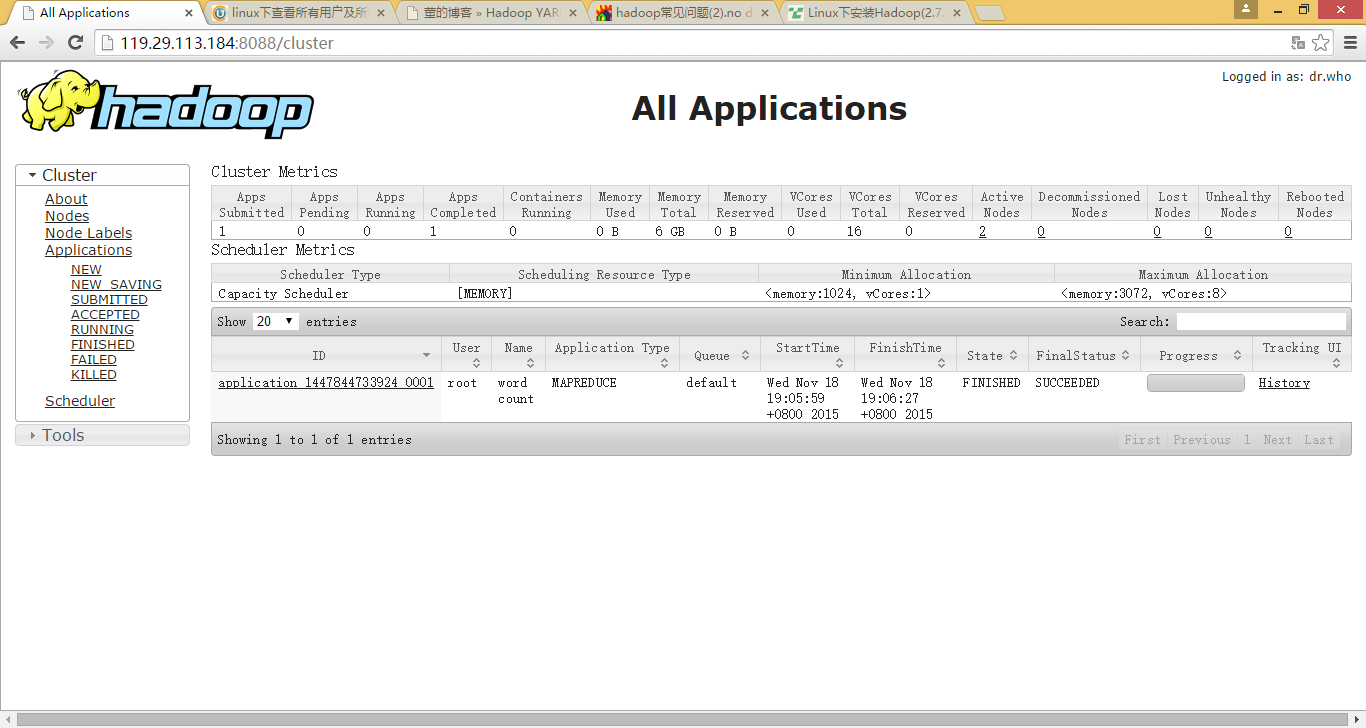
(3)浏览器打开<http://192.168.141.10:50070/>

浏览器打开后页面如下：





8088端口的页面如下：



遇到的问题：

今天配置完 Hadoop 运行Hadoop命令 hadoop fs -ls 时出现：

bash:haoop:command not found

后来经过检查，原来是配置文件中 HADOOP\_HOME 变量引用前少了个$，修改完毕后可以正常运行了。

vi /etc/profile

export JAVA\_HOME=/usr/java/jdk1.7.0\_75

export HADOOP\_HOME=/opt/hadoop-2.6.0

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

修改完后运行 source /etc/profile 让其立即生效！

在 start-all.sh 是出现异常：

Error:Cannot find configuration directory:/etc/hadoop

解决办法：

在 hadoop-env.sh  配置文件中将

export HADOOP\_CONF\_DIR=${HADOOP\_CONF\_DIR:-"etc/hadoop"}

替换成

export HADOOP\_CONF\_DIR=/usr/hadoop-2.6.0/etc/hadoop

修改完后运行 source hadoop-env.sh 让其立即生效！

**现象：**当停止Hadoop的时候发现no datanode to stop的信息。

**原因1：**每次namenode format会重新创建一个namenodeId，而tmp/dfs/data下包含了上次format下的id，namenode format清空了namenode下的数据，但是没有清空datanode下的数据，导致启动时失败，所要做的就是每次fotmat前，清空tmp一下的所有目录。

这里有两种解决方案：

1）删除“/usr/hadoop/tmp”里面的内容

       rm -rf /usr/hadoop/tmp/\*

2）删除“/tmp”下以“hadoop”开头的文件

       rm -rf /tmp/hadoop\*

3）重新格式化hadoop

       hadoop namenode -format

4）启动hadoop

       start-all.sh

这种方案的缺点是原来集群上的重要数据全没有了。因此推荐第二种方案：

1）修改每个Slave的namespaceID，使其与Master的namespaceID一致。

或者

2）修改Master的namespaceID使其与Slave的namespaceID一致。

Master的“namespaceID”位于“/usr/hadoop/tmp/dfs/name/current/VERSION”文件里面，Slave的“namespaceID”位于“/usr/hadoop/tmp/dfs/data/current/VERSION”文件里面。

**原因2：**问题的原因是hadoop在stop的时候依据的是datanode上的mapred和dfs进程号。而默认的进程号保存在/tmp下，linux 默认会每隔一段时间（一般是一个月或者7天左右）去删除这个目录下的文件。因此删掉hadoop-hadoop-jobtracker.pid和hadoop-hadoop-namenode.pid两个文件后，namenode自然就找不到datanode上的这两个进程了。

在配置文件hadoop\_env.sh中配置export HADOOP\_PID\_DIR可以解决这个问题。

在配置文件中，HADOOP\_PID\_DIR的默认路径是“/var/hadoop/pids”，我们手动在“/var”目录下创建一个“hadoop”文件夹，若已存在就不用创建，记得用chown将权限分配给hadoop用户。然后在出错的Slave上杀死Datanode和Tasktracker的进程(kill -9 进程号)，再重新start-all.sh，stop-all.sh时发现没有“no datanode to stop”出现，说明问题已经解决。

至此，Hadoop环境已经部署好，可以使用了。下面是其它常用软件的部署。

# yum设置

yum 主要功能是更方便的添加/删除/更新RPM 包，自动解决包的倚赖性问题，便于管理大量系统的更新问题。

在腾讯云服务器上，通过yum进行安装软件时，发现不能连接yum源服务器，所以需要修改yum的配置文件

CentOS 默认已经安装了yum，不需要另外安装，查看是否安装可用如下命令

# rpm -qa|grep yum

yum 的配置文件分为两部分：main 和repository

* main 部分定义了全局配置选项，整个yum 配置文件应该只有一个main。常位于/etc/yum.conf 中。
* repository 部分定义了每个源/服务器的具体配置，可以有一到多个。常位于/etc/yum.repo.d 目录下的各文件中。

yum.conf 文件一般位于/etc目录下，一般其中只包含main部分的配置选项。如下命令可以查看其内容

# cat /etc/yum.conf

Last login: Fri Nov 20 15:21:16 2015 from 183.63.97.40

[root@hadoopmaster ~]# cat /etc/yum.conf

[main]

cachedir=/var/cache/yum/$basearch/$releasever

keepcache=0

debuglevel=2

logfile=/var/log/yum.log

exactarch=1

obsoletes=1

gpgcheck=1

plugins=1

installonly\_limit=5

bugtracker\_url=http://bugs.centos.org/set\_project.php?project\_id=23&ref=http://bugs.centos.org/bug\_report\_page.php?category=yum

distroverpkg=centos-release

# This is the default, if you make this bigger yum won't see if the metadata

# is newer on the remote and so you'll "gain" the bandwidth of not having to

# download the new metadata and "pay" for it by yum not having correct

# information.

# It is esp. important, to have correct metadata, for distributions like

# Fedora which don't keep old packages around. If you don't like this checking

# interupting your command line usage, it's much better to have something

# manually check the metadata once an hour (yum-updatesd will do this).

# metadata\_expire=90m

# PUT YOUR REPOS HERE OR IN separate files named file.repo

# in /etc/yum.repos.d

里面有个设置：gpgcheck=1　　// 有1和0两个选择，分别代表是否是否进行gpg(GNU Private Guard) 校验，以确定rpm 包的来源是有效和安全的。这个选项如果设置在[main]部分，则对每个repository 都有效。默认值为0。

/etc/yum.repos.d 是存放yum源的目录，yum源被保存在以repos结尾的文件中。

**配置国内yum源**

系统默认的yum 源速度往往不尽人意，为了达到快速安装的目的，在这里修改yum源为国内源。

上海交通大学yum源

a. 修改/etc/yum.repos.d/\*\*\*.repo为：

# CentOS-Base.repo

#

# The mirror system uses the connecting IP address of the client and the

# update status of each mirror to pick mirrors that are updated to and

# geographically close to the client. You should use this for CentOS updates

# unless you are manually picking other mirrors.

#

# If the mirrorlist= does not work for you, as a fall back you can try the

# remarked out baseurl= line instead.

#

#

[base]

name=CentOS-$releasever - Base

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os

baseurl=http://ftp.sjtu.edu.cn/centos/$releasever/os/$basearch/

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

#released updates

[updates]

name=CentOS-$releasever - Updates

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates

baseurl=http://ftp.sjtu.edu.cn/centos/$releasever/updates/$basearch/

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

#additional packages that may be useful

[extras]

name=CentOS-$releasever - Extras

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras

baseurl=http://ftp.sjtu.edu.cn/centos/$releasever/extras/$basearch/

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

#additional packages that extend functionality of existing packages

[centosplus]

name=CentOS-$releasever - Plus

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=centosplus

baseurl=http://ftp.sjtu.edu.cn/centos/$releasever/centosplus/$basearch/

gpgcheck=1

enabled=0

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

#contrib - packages by Centos Users

[contrib]

name=CentOS-$releasever - Contrib

#mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=contrib

baseurl=http://ftp.sjtu.edu.cn/centos/$releasever/contrib/$basearch/

gpgcheck=1

enabled=0

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

关于变量

* $releasever：代表发行版的版本，从[main]部分的distroverpkg获取，如果没有，则根据redhat-release包进行判断。
* $arch：cpu体系，如i686,athlon等
* $basearch：cpu的基本体系组，如i686和athlon同属i386，alpha和alphaev6同属alpha。

详细见：

<http://www.cnblogs.com/mchina/archive/2013/01/04/2842275.html>

1，进入yum源配置目录  
cd /etc/yum.repos.d

下载163网易的yum源：  
wget http://mirrors.163.com/.help/CentOS6-Base-163.repo

3，更新玩yum源后，执行下边命令更新yum配置，使操作立即生效  
yum makecache

4，除了网易之外，国内还有其他不错的yum源，比如中科大和搜狐的，大家可以根据自己需求下载  
中科大的yum源：  
wget <http://centos.ustc.edu.cn/CentOS-Base.repo>  
sohu的yum源  
wget <http://mirrors.sohu.com/help/CentOS-Base-sohu.repo>

yum安装常用软件的命令  
#yum check-update   
#yum remove 软件包名   
#yum install 软件包名   
#yum update 软件包名

**yum命令常见使用方法**   
yum -y install 包名（支持\*） ：自动选择y，全自动  
yum install 包名（支持\*） ：手动选择y or n  
yum remove 包名（不支持\*）  
rpm -ivh 包名（支持\*）：安装rpm包  
rpm -e 包名（不支持\*）：卸载rpm包

**yum安装后的文件在哪里？**   
查看/etc/yum.conf  
有个cachedir设置，默认是在/var/cache/yum  
默认不保存下载的文件，安装完后就自动删除的。  
要保存的话，修改keepcache，将0改为1。  
即：  
cachedir=/var/cache/yum  
keepcache=1

yum -y install httpd 　 ← 在线安装httpd Apache服务器及相关组件  
yum -y install php 　 ← 在线安装PHP  
yum -y install mysql-server 　 ← 安装MySQL   
yum -y install php-mysql 　 ← 安装php-mysql

# MySQL安装

Centos7已经不再含有MySQL，因为MySQL要收费，不开源了。如果需要安装，自己下载MySQL,主要有三个文件

MySQL-server-advanced-5.6.22-1.el7.x86\_64.rpm

MySQL-client-advanced-5.6.22-1.el7.x86\_64.rpm

MySQL-devel-advanced-5.6.22-1.el7.x86\_64.rpm

在安装MySQL-server-advanced时提示需要安装Perl（Data：：Dumper）

先安装Perl，在安装过程中有众多的依赖项，所以通过yum安装

yum install perl\*

然后再重新安装MySQL-server成功。成功后，提示随机设置了root的密码，保存在一个文件中

A RANDOM PASSWORD HAS BEEN SET FOR THE MySQL root USER !

You will find that password in '/root/.mysql\_secret'.

**启动MYSQL**

[root@bogon 桌面]#service mysql start

查看MySQL运行状态：

[root@bogon 桌面]# service mysql status

**默认root用户登录MYSQL**

[root@bogon 桌面]# mysql -u root -p

Enter password:

# MySQL修改root密码的多种方法

方法1： 用SET PASSWORD命令

　　mysql -u root

　　mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpass');

方法2：用mysqladmin

　　mysqladmin -u root password "newpass"

　　如果root已经设置过密码，采用如下方法

　　mysqladmin -u root password oldpass "newpass"

方法3： 用UPDATE直接编辑user表

　　mysql -u root

　　mysql> use mysql;

　　mysql> UPDATE user SET Password = PASSWORD('newpass') WHERE user = 'root';

　　mysql> FLUSH PRIVILEGES;

在丢失root密码的时候，可以这样

　　mysqld\_safe --skip-grant-tables&

　　mysql -u root mysql

　　mysql> UPDATE user SET password=PASSWORD("new password") WHERE user='root';

　　mysql> FLUSH PRIVILEGES;

# mysql-connector安装

安装完MySQL还要安装mysql-connector， 这是MySQL的连接驱动，直接yum安装：

yum install mysql-connector-java -y

# Hive安装

到官网下载了Hive1.2.1，根据官网提示安装：

<https://cwiki.apache.org/confluence/display/Hive/GettingStarted#GettingStarted-InstallationandConfiguration>

Hive类似于一个数据库管理工具，

Installing Hive from a Stable Release

Start by downloading the most recent stable release of Hive from one of the Apache download mirrors (see Hive Releases).

Next you need to unpack the tarball. This will result in the creation of a subdirectory named hive-x.y.z (where x.y.z is the release number):

$ tar -xzvf hive-x.y.z.tar.gz

Set the environment variable HIVE\_HOME to point to the installation directory:

$ cd hive-x.y.z

$ export HIVE\_HOME={{pwd}} //此处是你解压Hive后的目录。

Finally, add $HIVE\_HOME/bin to your PATH:

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

**运行 Hive，首先确保已经安装了Hadoop，Hive就是为了方便Hadoop操作的管理工具。**

Hive uses Hadoop, so:

•you must have Hadoop in your path OR

•export HADOOP\_HOME=<hadoop-install-dir>

如果安装了Hadoop，环境变量一般都已经设置好。

In addition, you must create

/tmp

/user/hive/warehouse

(aka hive.metastore.warehouse.dir) and set them

chmod g+w in HDFS before you can create a table in Hive.

Commands to perform this setup:

$ $HADOOP\_HOME/bin/hadoop fs -mkdir /tmp

$ $HADOOP\_HOME/bin/hadoop fs -mkdir /user/hive/warehouse

$ $HADOOP\_HOME/bin/hadoop fs -chmod g+w /tmp

$ $HADOOP\_HOME/bin/hadoop fs -chmod g+w /user/hive/warehouse

You may find it useful, though it's not necessary, to set HIVE\_HOME:

$ export HIVE\_HOME=<hive-install-dir>

Running Hive CLI

To use the Hive command line interface (CLI) from the shell:

$ $HIVE\_HOME/bin/hive

注：网上说了很多关于hive的配置，例如：hive-site.xml，其中在里面配，经过实际测试，不需要。

只是在hive-config.sh里面最后添加了环境变量：

export HIVE\_HOME=/home/hadoop/hive-1.2.1

export JAVA\_HOME=/usr/java/jdk1.7.0\_55

export HADOOP\_HOME=/home/hadoop/hadoop-2.7.1

在hive-env.sh中添加了：

HADOOP\_HOME=/home/hadoop/hadoop-2.7.1

# Hive Configuration Directory can be controlled by:

export HIVE\_CONF\_DIR=/home/hadoop/hive-1.2.1/conf