# Redhat 6.5软件安装文档

该文档所有操作除了特殊需要以外都是以root用户操作的。

## 首先检查我们需要的目录：

/app/apps 和 /app/appdata 无则创建。

mkdir /app/apps

mkdir /app/appdata

## 安装jdk(这里我们安装的是解压版的jdk-7u79-linux-x64.tar.gz)：

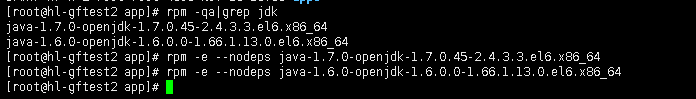
先检查系统自带的java: **rpm -qa | grep jdk**



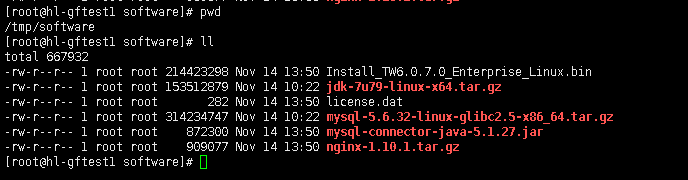
删除掉以前的jdk

**rpm -e --nodeps java-1.7.0-openjdk-1.7.0.45-2.4.3.3.el6.x86\_64**

**rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-1.66.1.13.0.el6.x86\_64**



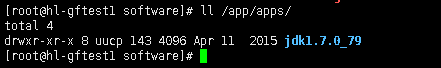
将软件安装包都上传到服务器临时目录/tmp/software(此位置自己选择)。



跳转到临时目录解压jdk到指定目录/app/apps下:

**cd /tmp/software**

**tar zxvf jdk-7u79-linux-x64.tar.gz -C /app/apps （此命令有时粘贴不行需要手敲）**



编辑环境变量:

**vim /etc/profile**

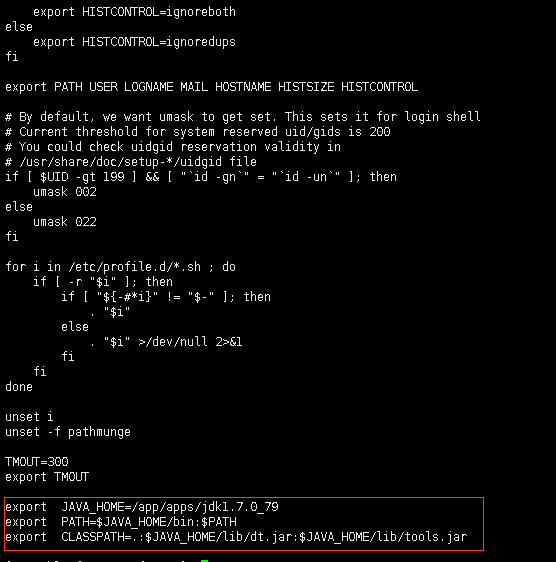


在最后添加:

**export JAVA\_HOME=/app/apps/jdk1.7.0\_79**

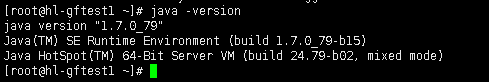
**export PATH=$JAVA\_HOME/bin:$PATH**

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



重新连接服务器，验证java是否安装成功：

**java -version** 或者 **which java**





至此java安装成功。

参考在线安装文档：

<http://www.cnblogs.com/qinersky902/p/5247729.html>

参考者可根据实际安装环境和自己的安装经验变通使用。

## 安装MySQL(这里我们安装的是mysql-5.6.32-linux-glibc2.5-x86\_64.tar.gz)

先检查系统自带的mysql:

**rpm –qa|grep –i mysql**



卸载掉：

**rpm -e --nodeps mysql-libs-5.1.71-1.el6.x86\_64**



解压mysql到指定安装目录下/app/apps:

**cd /tmp/software**

**tar zxvf mysql-5.6.32-linux-glibc2.5-x86\_64.tar.gz -C /app/apps/**



设置mysql的配置文件：

cp /app/apps/mysql-5.6.32-linux-glibc2.5-x86\_64/support-files/my-default.cnf /etc/my.cnf



编辑配置my.cnf

**vim /etc/my.cnf**

内容如下(红字为需要添加的内容):

[root@hl-gftest1 etc]# vi /etc/my.cnf

# For advice on how to change settings please see

# http://dev.mysql.com/doc/refman/5.6/en/server-configuration-defaults.html

# \*\*\* DO NOT EDIT THIS FILE. It's a template which will be copied to the

# \*\*\* default location during install, and will be replaced if you

# \*\*\* upgrade to a newer version of MySQL.

[mysqld]

# Remove leading # and set to the amount of RAM for the most important data

# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.

# innodb\_buffer\_pool\_size = 128M

# Remove leading # to turn on a very important data integrity option: logging

# changes to the binary log between backups.

# log\_bin

# These are commonly set, remove the # and set as required.

basedir =/app/apps/mysql-5.6.32-linux-glibc2.5-x86\_64

datadir =/app/apps/mysql-5.6.32-linux-glibc2.5-x86\_64/data

max\_allowed\_packet=200M

# port = .....

# server\_id = .....

# socket = .....

character\_set\_server = utf8

init\_connect = 'SET NAMES utf8'

# Remove leading # to set options mainly useful for reporting servers.

# The server defaults are faster for transactions and fast SELECTs.

# Adjust sizes as needed, experiment to find the optimal values.

# join\_buffer\_size = 128M

# sort\_buffer\_size = 2M

# read\_rnd\_buffer\_size = 2M

sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES

lower\_case\_table\_names=1

[mysql]

default-character-set = utf8

[mysql.server]

default-character-set = utf8

[mysqld\_safe]

default-character-set = utf8

[client]

default-character-set = utf8

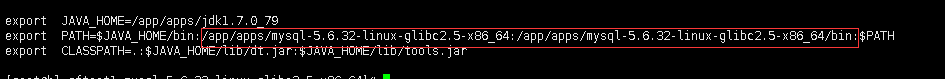
将mysql配置到环境变量:

**vim /etc/profile**

在PATH中添加mysql的安装目录和bin目录：

**/app/apps/mysql-5.6.32-linux-glibc2.5-x86\_64:/app/apps/mysql-5.6.32-linux-glibc2.5-x86\_64/bin:**

添加位置如下：



重新连接服务器使得配置文件生效。添加mysql到系统服务开机启动：

**cp -ar /app/apps/mysql-5.6.32-linux-glibc2.5-x86\_64/support-files/mysql.server /etc/init.d/mysqld**



添加用户mysql:依次执行下面语句：注意最后两行后面是有点的

**groupadd mysql**

**useradd -r -g mysql mysql**

**cd /app/apps/mysql-5.6.32-linux-glibc2.5-x86\_64/**

**chown -R mysql .**

**chgrp -R mysql .**



用mysql用户初始化数据库。

**cd /app/apps/mysql-5.6.32-linux-glibc2.5-x86\_64/**

**scripts/mysql\_install\_db --user=mysql --defaults-file=/etc/my.cnf**

启动mysql服务:

**service mysqld start**



**每次执行到这里我都得重连一下，不然后面的命令都不行。不知道为什么。。。。。**

**cd /app/apps/mysql-5.6.32-linux-glibc2.5-x86\_64/bin/**

修改root密码:

**mysqladmin -uroot password '123qwe!@#'**

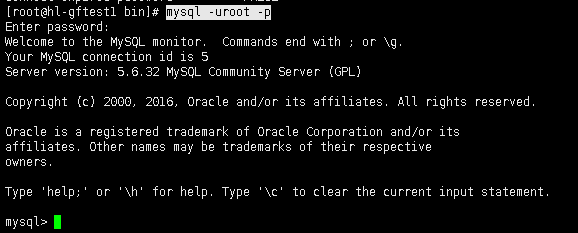


有时候粘贴命令不行的话就手敲，手敲不行就重新连一下再手敲。

登陆查看字符集：

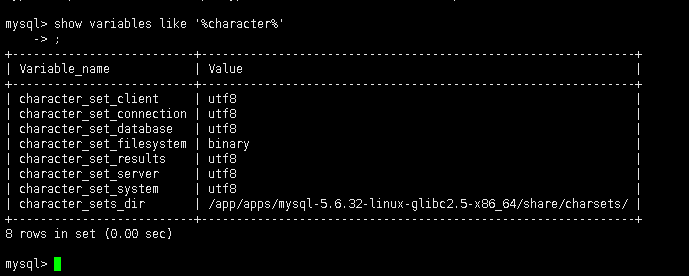
**mysql -uroot –p**

然后输入密码回车进入数据库：



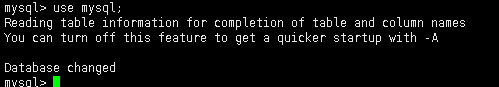
查看字符集设置是否正确：

**show variables like '%character%';**



开启远程连接：选择要使用的mysql数据库，修改远程连接的基本信息。

**use mysql**



**更改远程连接设置：**

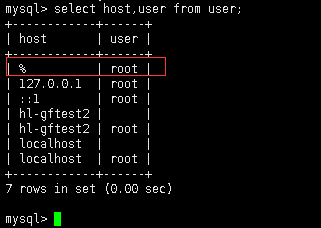
**GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY '123qwe!@#' WITH GRANT OPTION;**

刷新刚才修改的权限，使其生效

**flush privileges;**

查看修改是否成功：

**select host,user from user;**



有红框里的东西就成功,退出重启mysql。

退出mysql：mysql>quit

**service mysqld restart**

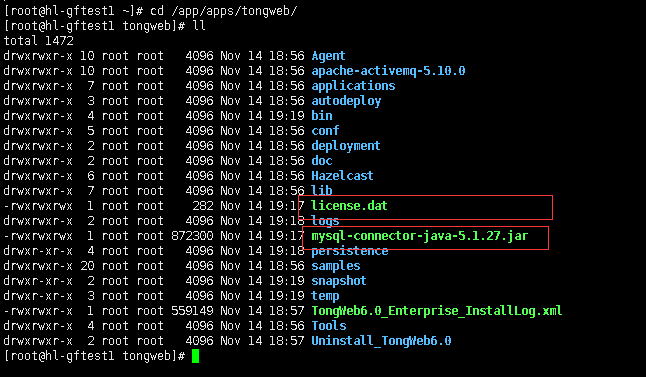


Mysql至此安装完毕，安装mysql比较容易出错，我们只要根据mysql的data目录下生成的xxxx.err日志文件信息来调整就行了。

## TongWeb安装：参见TongWeb6.0用户使用手册.pdf

注意事项:

将lincese.dat 和 mysql-connector-java-5.1.27.jar 放在tongweb的安装目录下，并赋予执行权限；



执行

**cd /app/apps/tongweb/bin**

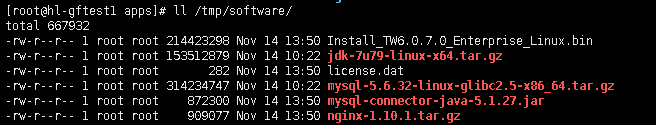
**./startservernohup.sh**

启动**。**

访问[**http://192.166.39.9:9060/console**](http://192.166.39.9:9060/console) **账号**twnt/twnt123.com

## Nginx安装(这里安装的是nginx-1.10.1.tar.gz):

查看安装包目录：



将安装包解压到指定目录(/app/apps/)下：

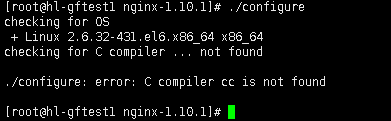
**tar zxvf /tmp/software/nginx-1.10.1.tar.gz -C /app/apps/.**



转到安装目录执行

**cd /app/apps/nginx-1.10.1**

**./configure**



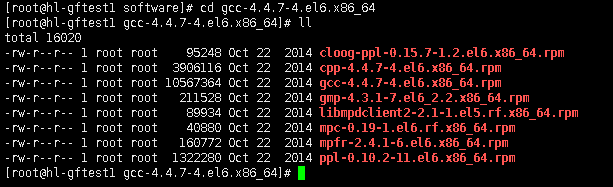
显然报错了。

Gcc安装（gcc-6.2.0.tar.bz2）

**cd /tmp/software**

**unzip gcc-4.4.7-4.el6.x86\_64.zip**





**cd /tmp/software/gcc-4.4.7-4.el6.x86\_64**

依次执行：

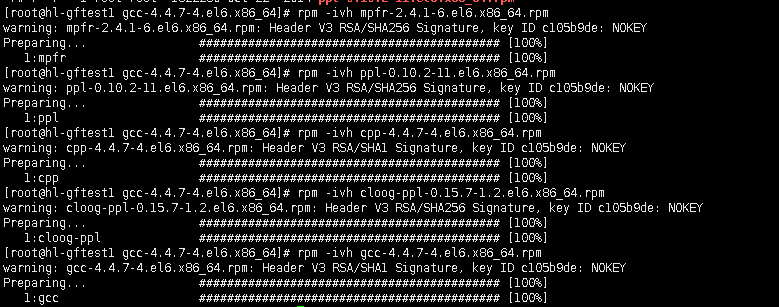
**rpm -ivh mpfr-2.4.1-6.el6.x86\_64.rpm**

**rpm -ivh ppl-0.10.2-11.el6.x86\_64.rpm**

**rpm -ivh cpp-4.4.7-4.el6.x86\_64.rpm**

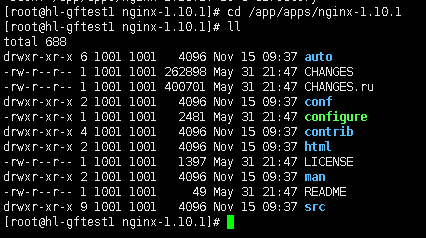
**rpm -ivh cloog-ppl-0.15.7-1.2.el6.x86\_64.rpm**

**rpm -ivh gcc-4.4.7-4.el6.x86\_64.rpm**

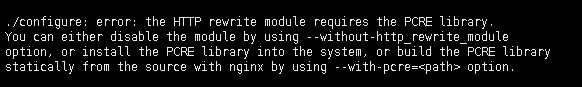


**安装完gcc接着安装nginx**

**cd /app/apps/nginx-1.10.1**



**执行./** **configure**



**又安装PCRE库**

**cd /tmp/software**

**unzip pcre-8.39.zip**

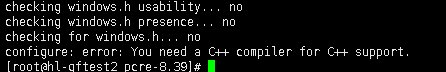
**为防止pcre-8.39安装文件夹被删除掉，将gcc文件夹挪到/app/apps下。**

**mv /tmp/software/pcre-8.39 /app/apps**

**cd /app/apps/pcre-8.39**

**./configure**

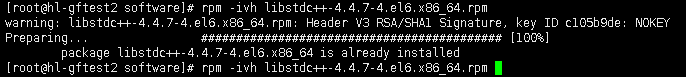
**报错，要安装gcc-c++**



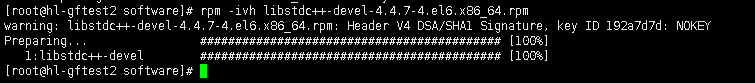
**//rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release**

**cd /tmp/software**

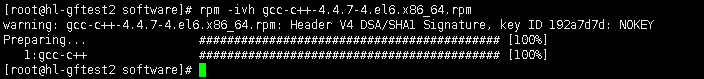
**rpm -ivh libstdc++-4.4.7-4.el6.x86\_64.rpm**

****

**rpm -ivh libstdc++-devel-4.4.7-4.el6.x86\_64.rpm**



**rpm -ivh gcc-c++-4.4.7-4.el6.x86\_64.rpm**



**好，gcc-c++安装好了，接着执行configure编译安装gcc**

**cd /app/apps/pcre-8.39/**

**./configure**

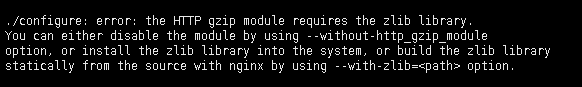
**make**

**make install**

**好，pcre库安装成功。接着返回去安装nginx；**

**cd /app/apps/nginx-1.10.1**

**./configure**



**还得装zlib**

**cd /tmp/software**

**解压zlib到/app/apps下**

**tar zxvf /tmp/software/zlib-1.2.8.tar.gz -C /app/apps/**

**cd /app/apps/zlib-1.2.8/**

**./configure**

**make**

**make install**

**安装好zlib接着返回去装nginx;**

**cd /app/apps/nginx-1.10.1**

**./configure**

**没报错；**

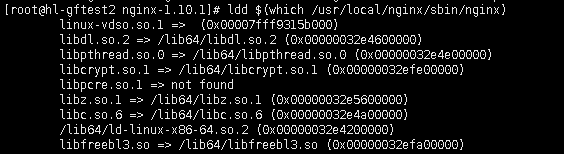
**make**

**make install**

启动Nginx出现的问题：  
**/usr/local/nginx/sbin/nginx**

  
1、/usr/local/nginx/sbin/nginx: error while loading shared libraries: libpcre.so.1: cannot open shared object file: No such file or directory

从错误看出是缺少lib文件导致，进一步查看下  
**ldd $(which /usr/local/nginx/sbin/nginx)**

  
linux-gate.so.1 => (0x0071b000)  
libpthread.so.0 => /lib/libpthread.so.0 (0×00498000)  
libcrypt.so.1 => /lib/libcrypt.so.1 (0×00986000)  
libpcre.so.1 => not found  
libcrypto.so.6 => /lib/libcrypto.so.6 (0×00196000)  
libz.so.1 => /lib/libz.so.1 (0×00610000)  
libc.so.6 => /lib/libc.so.6 (0x002d7000)  
/lib/ld-linux.so.2 (0x006a8000)  
libdl.so.2 => /lib/libdl.so.2 (0x008c3000)  
可以看出 libpcre.so.1 => not found 并没有找到，进入/lib64目录中手动链接下

**cd /lib64  
ln -s libpcre.so.0.0.1 libpcre.so.1**

**/usr/local/nginx/sbin/nginx**

然后在启动nginx ok 了

哎呀我曹。好了。

Mysql 启动命令: **service mysqld start**

Tongweb后台启动命令:

**cd /app/apps/tongweb/bin**

**startservernohup.sh**

**nginx启动命令:**

**/usr/local/nginx/sbin/nginx**

**修改配置文件的时候修改**/usr/local/nginx/conf/nginx.conf

**/usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf**