# LINUX 安装 apache

前言：

各软件包下载地址：

Apache下载地址：

<http://httpd.apache.org/download.cgi>

http://archive.apache.org/dist/httpd/

Mysql下载地址：

<https://dev.mysql.com/downloads/mysql/>

http://mirror.bit.edu.cn/mysql/Downloads/

Php下载地址：

<http://php.net/downloads.php>

<http://php.net/releases/>

Php扩展下载：

<http://pecl.php.net/>

<http://windows.php.net/downloads/pecl/releases/>

<http://downloads.php.net/pierre/>

依赖包下载地址：<http://download.chinaunix.net/>

Rpm包下载地址：

<http://www.rpmfind.net/linux/>

[https://pkgs.org/download/](https://pkgs.org/download/pcre)

开源软件镜像服务下载地址：<http://mirror.bit.edu.cn/web/>

Linux系统下载地址：

<https://www.linux.org/pages/download/>

<http://archive.kernel.org/centos-vault/>

Libxml2下载地址：

<http://xmlsoft.org/sources/>

## 一，源码安装

测试安装环境： CentOS6.8 64位。

安装位置： /usr/local/

软件包存放位置：/home/

系统支持：gcc，gcc-c++，make编译环境

如无gcc，gcc-c++，make编译环境，则按以下步骤下载安装编译工具包（已安装则忽略）

1. 使用rpm方式安装编译工具包

下载以下软件包（操作系统不同，软件包也不一样）:

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

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

cpp-4.4.7-17.el6.x86\_64.rpm

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

gcc-4.4.7-17.el6.x86\_64.rpm

gcc-c++-4.4.7-17.el6.x86\_64.rpm

gcc-java-4.4.7-17.el6.x86\_64.rpm

libstdc++-devel-4.4.7-17.el6.x86\_64.rpm

安装顺序：

1，rpm -ivh ppl-0.10.2-11.el6.x86\_64.rpm

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

3，rpm -ivh mpfr-2.4.1-6.el6.x86\_64.rpm

4，rpm -ivh cpp-4.4.7-17.el6.x86\_64.rpm

5，rpm -ivh gcc-4.4.7-17.el6.x86\_64.rpm

6，rpm -ivh gcc-c++-4.4.7-17.el6.x86\_64.rpm

7，rpm -ivh libstdc++-devel-4.4.7-17.el6.x86\_64.rpm

8，rpm -ivh gcc-c++-4.4.7-17.el6.x86\_64.rpm

9，rpm -ivh gcc-java-4.4.7-17.el6.x86\_64.rpm(编译java需要，可选)

安装成功后，写一个c语言（hello.c）文件：

内容如下: #include<stdio.h>

main(){

printf("hello word\n");

}

保存退出。

执行gcc hello.c，在查看目录，若出现a.out文件，则运行./a.out，可运行表示安装成功。

B) 使用yum方式安装编译工具包

前提：电脑必须可以联网。

执行命令: yum -y install gcc gcc-c++ make

### 安装前准备软件包

安装apache必要软件包如下：

zlib-1.2.11.tar.gz 代码优化压缩包

apr-1.6.3.tar.gz 【2.3以上版本必备】

apr-util-1.6.1.tar.gz 【2.3以上版本必备】

expat-2.0.1-13.el6\_8.x86\_64.rpm

expat-devel-2.0.1-13.el6\_8.x86\_64.rpm

pcre-7.8-7.el6.x86\_64.rpm

pcre-devel-7.8-7.el6.x86\_64.rpm

httpd-2.4.20.tar.gz

httpd-2.4.20-deps.tar.gz 【可选，相当于apr和apr-util】

注：若安装的apache版本为2.3及以上，则需要apr和apr-util包 或者 deps包。否则不需要。

建议：若有deps包，则使用deps包。

### 安装zlib(已安装而忽略)

Shell># cd /home/

Shell># tar -zxvf zlib-1.2.11.tar.gz

Shell># cd zlib-1.2.11.tar.gz

Shell># ./configure

Shell># make && make install

注意：

1，若要指定zlib安装路径的话，则在/configure 之后追加 --prefix=安装路径，例如：./configure --prefix=/usr/local/zlib

2，如果系统是64位，需要使用64位元的方法编译安装zlib:

CFLAGS=”-03 -fPIC” ./configusr --prefix=/usr/local/zlib

3，若想知道./configure 有什么参数可以配置? 可以使用：

./configure --help【适用所有源码编译的软件】

4，若要指定安装路径的话，记得先卸载系统自带的zlib库，并且卸载时间必须在安装zlib时已经使用make，但未使用make install 的时候卸载。

5，指定安装好zlib后，在编辑/etc/ld.so.conf 文件，在下面加入一行/usr/local/zlib/lib，保存退出。再使用ldconfig更新缓存即可。

### 安装expat

1. 使用rpm方式安装

Shell># cd /home/

Shell># rpm -ivh expat-2.0.1-13.el6\_8.x86\_64.rpm

Shell># rpm -ivh expat-devel-2.0.1-13.el6\_8.x86\_64.rpm

B) 使用yum方式安装

前提：必须可以上网

Shell># yum -y install expat expat-devel

### 安装pcre

1. 使用rpm方式安装

Shell># rpm cd /home

Shell># rpm -ivh pcre-7.8-7.el6.x86\_64.rpm

Shell># rpm -ivh pcre-devel-7.8-7.el6.x86\_64.rpm

B) 使用yum方式安装

前提：必须可以上网

Shell># yum -y install pcre pcre-devel

### 安装apache

Shell># cd /home/

Shell># tar -zxvf httpd-2.4.20.tar.gz

Shell># tar -zxvf httpd-2.4.20-deps.tar.gz

Shell># cd ./httpd-2.4.20-deps/srclib

Shell># cp apr /home/httpd-2.4.20/srclib

Shell># cp apr-util /home/httpd-2.4.20/srclib

Shell># cd /home/httpd-2.4.20/

Shell># ./configure --prefix=/usr/local/httpd24 \

> --enable-modules=all \

> --enable-mods-shared=all \

> --enable-so

Shell># make && make install

注：

\ 表示在还有命令未写完，换行在写。（\前面必须有空格）

--prefix apache安装路径

--enable-modules 指定开启的功能模块。（all所有）

--enable-mods-shared 把全部的功能模块代码内容都编译到“apache软件本身”内部【注：apache本身体积会变大，但是在调用相关模块的时候速度会很快】,相对应类: --enable-static静态类型，apache内部不要编译模块代码，其他模块代码都独立存在，需要什么模块就立即include引入。【注：apache本身会非常小巧，其运行速度会很快】

--enable-so 可以识别so后缀的模块文件

--with-apr=$DIR 若apr扩展的位置不是在apache的srclib目录里，则需要指定路径

--with-apr-util=$DIR 同--with-apr

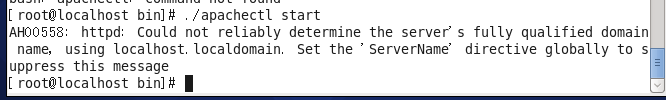
--with-pcre=$DIR 若pcre的安装方式不是rpm或者yum的话，需要指明安装路径

若检测环境失败会有提示，请根据提示安装需要的依赖包，重新检测编译即可。

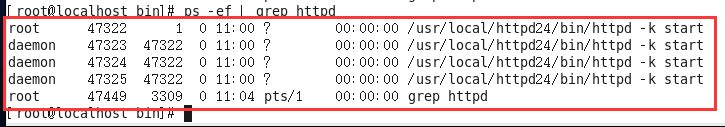
### 运行apache，检测安装是否成功

Shell># cd /usr/local/httpd24/bin

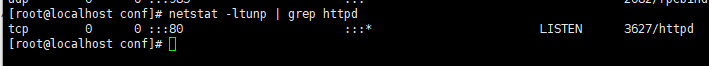
Shell># apachestl start

 注：提示ServerName错误，则需要修改/usr/local/httpd24/conf/httpd.conf，去掉ServerName前面的”#”号即可。

Shell>#ps ef | grep httpd



Shell># netstat -ltunp | grep httpd



打开网址输入：127.0.0.1 或者 localhost 是否访问成功。



### **卸载源码安装的apache**

步骤：

1，进入apache的解压目录

2，执行make uninstall（若提示无该命令的话，则查看Makefile文件，主要看install部分，从其中找出tar格式的文件被复制到了什么路径，然后进入相应的目录进行删除即可）

3，删除./configure指定apache的安装路径。如:/usr/local/httpd24

4，删除apache解压目录

Shell># cd /home/

Shell># cd httpd2.4.20

Shell># make uninstall

Shell># rm -rf /usr/local/httpd24

Shell># cd /home

Shell># rm -rf httpd-2.4.20

### 源码软件安装错误解决

1. 已执行configure操作

》根据正确的参数重新configure即可

2，已经执行configure、make操作

》删除解压后的目录，重新解压、configure、make

3，已经执行configure、make、make install

》删除安装后的文件（有指定安装目录情况/usr/local/httpd24）

》删除解压后的目录

》重新解压，重新configure，重新make，重新make install

参考：更详细的源码安装apache教程

<http://www.jb51.net/article/59474.htm>

## 二，rpm安装

### rpm安装apache

1. 准备安装包

apr-util-1.3.9-3.el6\_0.1.x86\_64.rpm

apr-util-ldap-1.3.9-3.el6\_0.1.x86\_64.rpm

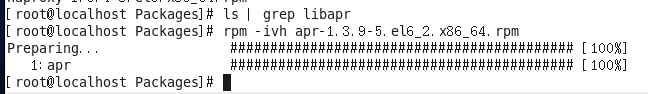
apr-1.3.9-5.el6\_2.x86\_64.rpm

httpd-2.2.15-53.el6.centos.x86\_64.rpm

httpd-tools-2.2.15-53.el6.centos.x86\_64.rpm

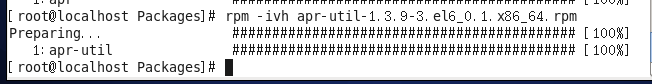
B) 安装apr

Shell># rpm -ivh apr-1.3.9-5.el6\_2.x86\_64.rpm



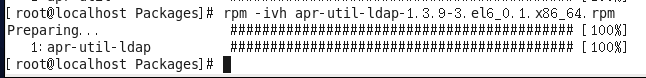
C) 安装apr-util

Shell># rpm -ivh apr-util-1.3.9-3.el6\_0.1.x86\_64.rpm



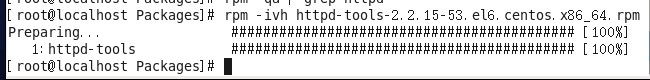
D) 安装apr-util-ldap

Shell># rpm -ivh apr-util-ldap-1.3.9-3.el6\_0.1.x86\_64.rpm



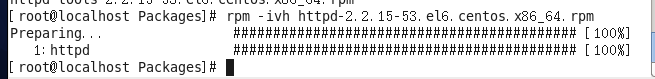
E) 安装 httpd-tools

Shell># rpm -ivh httpd-tools-2.2.15-53.el6.centos.x86\_64.rpm



F) 安装httpd

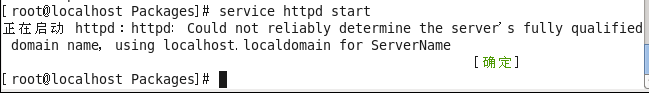
Shell># rpm -ivh httpd-2.2.15-53.el6.centos.x86\_64.rpm



注意：若在安装过程中提示相关依赖问题，则安装指定的依赖软件包即可。

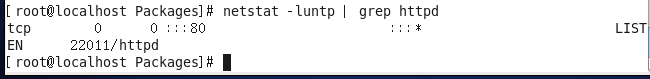
G) 启动apache服务

Shell># service httpd start

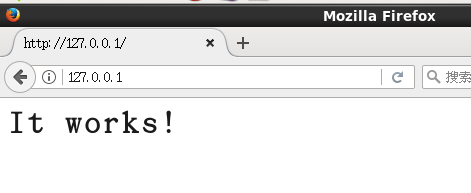


H) 查看启动进程

Shell># netstat -ltunp | grep httpd



I) 打开网址，输入127.0.0.1，测试网址是否可以成功打开。

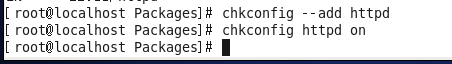


测试OK！！！

J) 设置开机自启动

Shell># chkconfig --add httpd

Shell># chkconfig httpd on



K) 配置apache的配置文件

Shell># vim /etc/httpd/conf/httpd.conf

.....以下自行配置

RPM默认安装路径：

/etc 一些设置文件放置的目录如/etc/crontab

/usr/bin 一些可执行文件

/usr/lib 一些程序使用的动态函数库

/usr/share/doc 一些基本的软件使用手册与帮助文档

/usr/share/man 一些man page文件

例如apache安装位置：

网站根目录 /var/www/html/   
 apache主目录 /etc/httpd/   
 apache主配置文件 /etc/httpd/conf/httpd.conf   
 apache日志 /etc/httpd/logs/

### rpm卸载apache

Rpm包的卸载顺序有讲究的：

例如：安装包的顺序是 A > B > C > D

卸载包的顺序是 D > C > B > A

若想不按顺序卸载的话，可以使用 --nodeps 参数。

rpm包卸载指令：

rpm -e 软件包名称【不包含.rpm后缀】[--nodeps]

解释：-e 表示删除软件包

--nodeps 表示强制删除软件包，忽略依赖关系（不建议使用）

根据上面的情况卸载apache:

Shell># rpm -e httpd-2.2.15-53.el6.centos.x86\_64

Shell># rpm -e httpd-tools-2.2.15-53.el6.centos.x86\_64

Shell># rpm -e apr-util-ldap-1.3.9-3.el6\_0.1.x86\_64 --nodeps

Shell># rpm -e apr-util-1.3.9-3.el6\_0.1.x86\_64 --nodeps

Shell># rpm -e apr-1.3.9-5.el6\_2.x86\_64 --nodeps

### rpm常用命令（~代表软件包名称）

1，rpm包的名称格式

例如：apache-1.3.23-11.i386.rpm

"apache" 软件名称

"1.3.23-11" 软件的版本号，主版本和次版本

"i386" 是软件所运行的硬件平台

"rpm" 文件扩展名，代表RPM包

2，查询已安装的所有rpm软件包

Shell># rpm -qa

3，查看某个软件包是否已安装

Shell># rpm -q ~

4，查找包含关键字的已安装软件包名

Shell># rpm -qa | grep ~

5，查看软件包信息

Shell># rpm -qi ~

6，导入软件包的校验key

Shell># rpm --import KEY

7，校验软件包

Shell># rpm -K ~

8，列出软件包中的文件

Shell># rpm -ql ~

9，查询文件所属的软件包

Shell># rpm -qf 文件全路径名

10，查看软件包的介绍信息

Shell># rpm -qp ~

11，安装软件包

Shell># rpm -ivh ~

参数解释：

i = install 安装

v = verbose 提示

h = hash 进度条

12，卸载软件包

Shell># rpm -e ~

13，强制卸载软件包

Shell># rpm -e ~ -f --nodeps

注：如果其他软件依赖于您要卸载的软件包，卸载时则会产生错误信息。若想让RPM忽略这个错误继续卸载（不推荐这么做），请使用--nodeps命令行选项。

-f 是指强制

14，升级软件包

Shell># rpm -U ~

## 三，yum安装

### yum安装apache

Yum方式安装之前请确保可以联网！

执行：

Shell># yum -y install httpd



yum方式安装会自动解决软件依赖问题。

apache目录：

网站根目录 /var/www/html/   
 apache主目录 /etc/httpd/   
 apache主配置文件 /etc/httpd/conf/httpd.conf   
 apache日志 /etc/httpd/logs/

注：yum方式安装只是Red Hat 和CentOS独有的安装方式。

不同linux系统之间的安装命令都不一样。

例如：Ubuntu

Shell>#sudo apt-get -y install apache2

apache目录：

网站根目录 /var/www

apache主目录 /etc/apache2/

apache主配置文件 /etc/apache2/apache2.conf

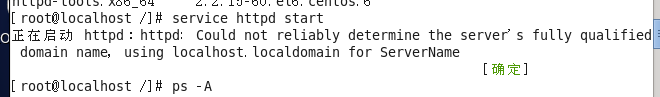
apache日志 /var/log/apache2/

测试apache是否可以正常运行：

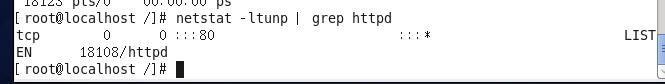
启动： Shell># service httpd start

重新启动： Shell># service httpd restart

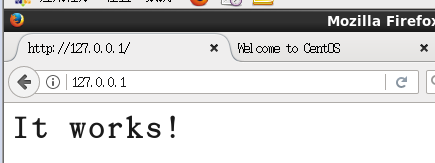
停止： Shell># service httpd stop



查看端口：shell># netstat -ltunp | grep httpd



输入网址：127.0.0.1



启动成功！！！

### yum方式卸载apache

查看已安装的yum源：

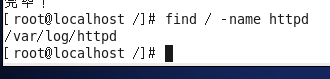
Shell># yum list installed | grep httpd

卸载apache:

Shell># service httpd stop

Shell># yum -y remove httpd





移除成功！！！

### 更新yum源

建议：

加快yum的速度，可以使用yum的扩展插件yum-fastestmirror

Yum -y install yum-fastestmirror

更新步骤：

1，进入yum源配置目录

Shell># cd /etc/yum.repos.d

2，备份源系统的yum源文件

Shell># mv /etc/yum.repos.d/CentOS-Base.repo \

> /etc/yum/yum.repos.d/CentOS-Base.repo.bak

3，使用wget下载对应系统的yum源

国内yum源地址：

1. [http://mirrors.163.com/](http://mirrors.163.com/.help/CentOS6-Base-163.repo)
2. <http://centos.ustc.edu.cn/>
3. <http://mirrors.sohu.com/>
4. 北京理工大学：  
    [http://mirror.bit.edu.cn](http://mirror.bit.edu.cn/) (IPv4 only)  
    [http://mirror.bit6.edu.cn](http://mirror.bit6.edu.cn/) (IPv6 only)
5. 北京交通大学：  
    [http://mirror.bjtu.edu.cn](http://mirror.bjtu.edu.cn/) (IPv4 only)  
    [http://mirror6.bjtu.edu.cn](http://mirror6.bjtu.edu.cn/) (IPv6 only)  
    [http://debian.bjtu.edu.cn](http://debian.bjtu.edu.cn/) (IPv4+IPv6)
6. 兰州大学：<http://mirror.lzu.edu.cn/>
7. 厦门大学：<http://mirrors.xmu.edu.cn/>
8. 清华大学：  
    <http://mirrors.tuna.tsinghua.edu.cn/> (IPv4+IPv6)  
    <http://mirrors.6.tuna.tsinghua.edu.cn/> (IPv6 only)  
    <http://mirrors.4.tuna.tsinghua.edu.cn/> (IPv4 only)
9. 天津大学：<http://mirror.tju.edu.cn/>
10. 中国科学技术大学：  
     <http://mirrors.ustc.edu.cn/> (IPv4+IPv6)  
     <http://mirrors4.ustc.edu.cn/>  
     <http://mirrors6.ustc.edu.cn/>
11. 东北大学：  
     <http://mirror.neu.edu.cn/> (IPv4 only)  
     <http://mirror.neu6.edu.cn/> (IPv6 only)
12. 电子科技大学：<http://ubuntu.uestc.edu.cn/>

下载.repo结尾的文件，放到 /etc/yum.repos.d/ 目录里，重命名。

Shell># cd /etc/yum.repos.d/

Shell># wget <http://mirrors.163.com/.help/CentOS6-Base-163.repo>

Shell># mv CentOS6-Base-163.repo CentOS-Base.repo

4，重新生成yum缓存

Shell># yum clean all

Shell># yum makecache

### yum配置文件说明

**A) yum资源库文件配置**

**配置文件：/etc/yum.repos.d/CentOS-Base.repo，也可以在此目录下建别的以.repo后缀的文件就行。**

**[base]：代表容器的名字！中刮号一定要存在，里面的名称则可以随意取。但是不能有两个相同的容器名称， 否则 yum 会不晓得该到哪里去找容器相关软体清单档案。**

**基本参数：**

**name：只是说明一下这个容器的意义而已，重要性不高！**

**mirrorlist=：列出这个容器可以使用的映射站台，如果不想使用，可以注解到这行；**

**baseurl=：这个最重要，因为后面接的就是容器的实际网址！ mirrorlist 是由 yum 程式自行去捉映射站台， baseurl 则是指定固定的一个容器网址！我们刚刚找到的网址放到这里来啦！**

**baseurl 是服务器设置中最重要的部分，只有设置正确，才能从上面获取软件。它的格式是：**

**baseurl=url://server1/path/to/repository/**

**url://server2/path/to/repository/**

**url://server3/path/to/repository/**

**其中url 支持的协议有 http:// ftp:// file:// 三种。baseurl 后可以跟多个url，你可以自己改为速度比较快的镜像站，但baseurl 只能有一个，也就是说不能像如下格式：**

**baseurl=url://server1/path/to/repository/**

**baseurl=url://server2/path/to/repository/**

**baseurl=url://server3/path/to/repository/**

**其中url 指向的目录必须是这个repository header 目录的上一级，它也支持$releasever $basearch 这样的变量。**

**url 之后可以加上多个选项，如gpgcheck、exclude、failovermethod 等，比如：**

**baseurl=http://download.atrpms.net/mirrors/fedoracore/updates/$releasever/$basearch**

**http://redhat.linux.ee/pub/fedora/linux/core/updates/$releasever/$basearch**

**http://fr2.rpmfind.net/linux/fedora/core/updates/$releasever/$basearch**

**gpgcheck=1**

**exclude=gaim**

**failovermethod=priority**

**enable=1：就是让这个容器被启动。如果不想启动可以使用 enable=0 喔！**

**gpgcheck=1：还记得 RPM 的数位签章吗？这就是指定是否需要查阅 RPM 档案内的数位签章！0表示不检测。**

**gpgkey=：就是数位签章的公钥档所在位置！使用预设值即可**

**failovermethode 有两个选项roundrobin 和priority，意思分别是有多个url可供选择时，yum 选择的次序，roundrobin 是随机选择，如果连接失败则使用下一个，依次循环，priority 则根据url 的次序从第一个开始。如果不指明，默认是roundrobin。**

**常规设置：**

**[root@www ~]# vi /etc/yum.repos.d/CentOS-Base.repo**

**[base]**

**name=CentOS-$releasever - Base**

**baseurl=http://ftp.twaren.net/Linux/CentOS/5/os/i386/**

**gpgcheck=1**

**gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5**

**查看容器配置是否生效：yum repolist all**

**B) yum.conf文件配置**

**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。**

**Shell># cat /etc/yum.conf**

**[main]**

**cachedir=/var/cache/yum**

**//yum 缓存的目录，yum 在此存储下载的rpm 包和数据库，默认设置为/var/cache/yum**

**keepcache=0**

**//安装完成后是否保留软件包，0为不保留（默认为0），1为保留**

**debuglevel=2**

**//Debug 信息输出等级，范围为0-10，缺省为2**

**logfile=/var/log/yum.log**

**//yum 日志文件位置。用户可以到/var/log/yum.log 文件去查询过去所做的更新。**

**pkgpolicy=newest**

**//包的策略。一共有两个选项，newest 和last，这个作用是如果你设置了多个repository，而同一软件在不同的repository 中同时存在，yum 应该安装哪一个，如果是newest，则yum 会安装最新的那个版本。如果是last，则yum 会将服务器id 以字母表排序，并选择最后的那个服务器上的软件安装。一般都是选newest。**

**distroverpkg=redhat-release**

**//指定一个软件包，yum 会根据这个包判断你的发行版本，默认是redhat-release，也可以是安装的任何针对自己发行版的rpm 包。**

**tolerant=1**

**//有1和0两个选项，表示yum 是否容忍命令行发生与软件包有关的错误，比如你要安装1,2,3三个包，而其中3此前已经安装了，如果你设为1,则yum 不会出现错误信息。默认是0。**

**exactarch=1**

**//有1和0两个选项，设置为1，则yum 只会安装和系统架构匹配的软件包，例如，yum 不会将i686的软件包安装在适合i386的系统中。默认为1。**

**retries=6**

**//网络连接发生错误后的重试次数，如果设为0，则会无限重试。默认值为6.**

**obsoletes=1**

**//这是一个update 的参数，具体请参阅yum(8)，简单的说就是相当于upgrade，允许更新陈旧的RPM包。**

**plugins=1**

**//是否启用插件，默认1为允许，0表示不允许。我们一般会用yum-fastestmirror这个插件。**

**bugtracker\_url=http://bugs.centos.org/set\_project.php?project\_id=16&ref=http://bugs.centos.org/bug\_report\_page.php?category=yum**

**# Note: yum-RHN-plugin doesn't honor this.**

**metadata\_expire=1h**

**installonly\_limit = 5**

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

**# in /etc/yum.repos.d**

**除了上述之外，还有一些可以添加的选项，如：**

**exclude=selinux\*　　// 排除某些软件在升级名单之外，可以用通配符，列表中各个项目要用空格隔开，这个对于安装了诸如美化包，中文补丁的朋友特别有用。**

**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。**

注：

CentOS-Base.repo 是yum 网络源的配置文件

CentOS-Media.repo 是yum 本地源的配置文件

### yum常用命令（~代表软件名称）

1，yum安装软件包

Shell># yum [-y] install ~

2，yum删除软件包

Shell># yum [-y] remove ~

3，yum 查询软件包

Shell># yum search ~

4，yum 查看软件包用途

Shell># yum info ~

5，列出所有可安装的软件包

Shell># yum list

6，列出所有可更新的软件包

Shell># yum list update

7，列出所有已安装的软件包

Shell># yum list install

8，列出所有已安装但不在Yum Repository内的软件包

Shell># yum list extras

9，列出所指定的软件包

Yum list ~

10，列出所有软件包信息

Shell># yum info

11，列出所有已安装的软件包信息

Shell># yum info installed

12，列出所有可更新的软件包信息

Shell># yum info updates

13，列出所有已安装但不在Yum Repository内的软件包信息

Shell># yum info extras

14，列出软件包提供哪些文件

Shell># yum provides ~

15，更新所有可更新的软件包

Shell># yum update

16，清除缓存目录（/var/cache/yum）下的软件包

Shell># yum clean packages

17，清除缓存目录（/var/cache/yum）下的headers

Shell># yum clean headers

18，清除缓存目录（/var/cache/yum）下的旧headers

Shell># yum clean old headers

19，清除所有缓存

Shell># yum clean

20，查看系统中已经安装的和可用的软件组，可用的可以安装

Shell># yum group list

21，安装上一个命令显示的可用的软件组中的一个

Shell># yum grooup install ~

22，更新指定软件组的软件包

Shell># yum grooup update ~

23，卸载指定软件组中的软件包

Shell># yum grooup remove ~

24，查询指定软件包的依赖关系

Shell># yum deplist ~

25，列出所有以yum开头的软件包

Shell># yum list yum\\*

26，从硬盘安装rpm包并使用yum解决依赖

Shell># yum local install ~

27，升级内核

yum install kernel-headers kernel-devel

28，查看yum文档

Shell># man yum

29，重新生成yum缓存

Shell># yum makecache

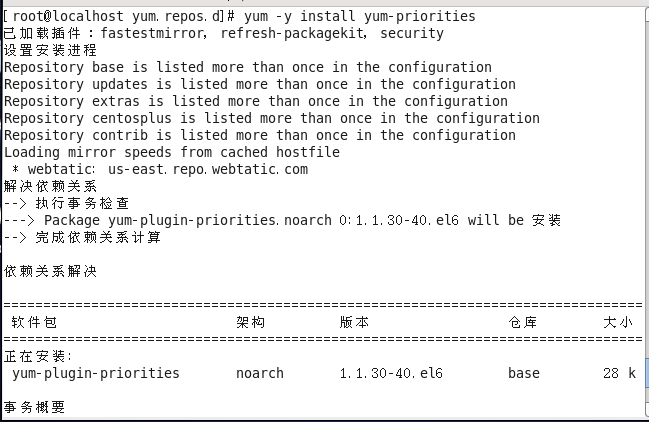
### 使用第三方软件库

Centos/RHEL默认的yum软件仓库非常有限，仅仅限于发行版本那几张盘里面的常规包和一些软件包的更新，利用RpmForge，可以增加非常多的第三方rpm软件包。RpmForge库现在已经拥有超过10000种的CentOS的软件包，被CentOS社区认为是最安全也是最稳定的一个第三方软件库。

1、安装yum-priorities插件

这个插件是用来设置yum在调用软件源时的顺序的。因为官方提供的软件源，都是比较稳定和被推荐使用的。因此，官方源的顺序要高于第三方源的顺序。如何保证这个顺序，就需要安装yum-priorities这插件了。

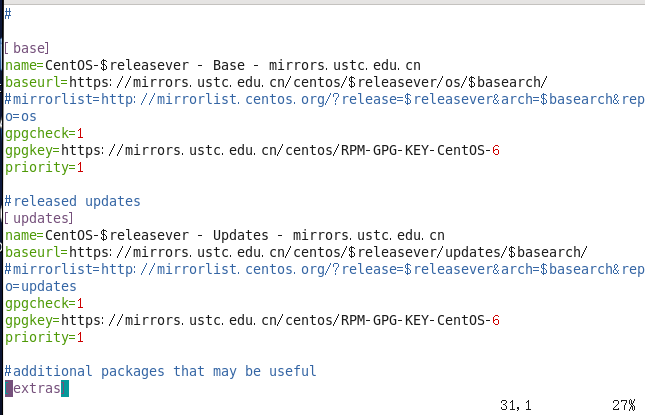
Shell># yum -y install yum-priorities



2、安装完yum-priorities插件后需要设置/etc/yum.repos.d/ 目录下的.repo相关文件（如CentOS-Base.repo），在这些文件中插入顺序指令：priority=N （N为1到99的正整数，数值越小越优先）

一般配置[base], [addons], [updates], [extras] 的priority=1，[CentOSplus], [contrib] 的priority=2，其他第三的软件源为：priority=N （推荐N>10）

以CentOS-Base.repo 为例：



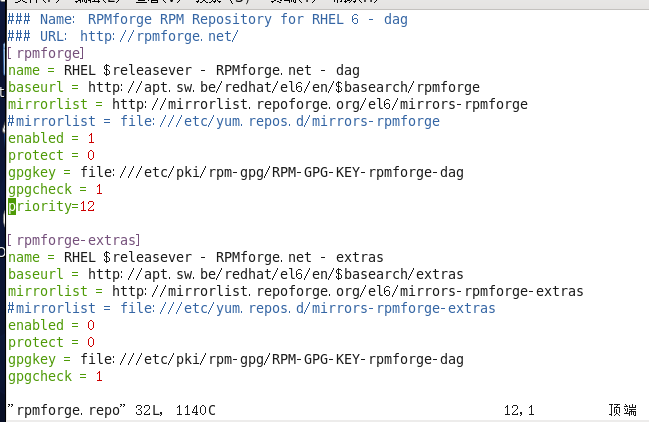
1. 下载与安装相应rpmforge的rpm文件包

Shell># wget <http://repository.it4i.cz/mirrors/repoforge/redhat/el6/en/x86_64/rpmforge/RPMS/rpmforge-release-0.5.3-1.el6.rf.x86_64.rpm>

4、安装rpmforge的rpm文件包

Shell># rpm -i rpmforge-release-0.5.2-2.el5.rf.i386.rpm

5、设置/etc/yum.repos.d/rpmforge.repo文件中源的级别



8、测试安装

Shell># yum install htop

参考 <http://wiki.centos.org/AdditionalResources/Repositories/RPMForge#head-5aabf02717d5b6b12d47edbc5811404998926a1b>

其他第三方软件库如 EPEL(Extra Packages for Enterprise Linux ) 和 RPMFusion 的安装与使用和RPMForge类似，可自行查找资料安装。

## 四，apache配置文件详解

### httpd.conf配置

配置文件在：/usr/local/httpd24/conf/httpd.conf



指定apache的运行目录

****

**当前域名。若未开启（前面#号未去掉），则apache开启会提示错误。**

**可以修改成localhost 或者 自己的网址域名**



管理员的邮箱

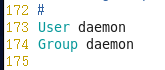


监听的端口号。可以自己指定，默认80



若开启mod\_rewrite.so模块的话，则表示网址可以重写。比如：隐藏index.php文件等..

某些框架需要开启该模块。



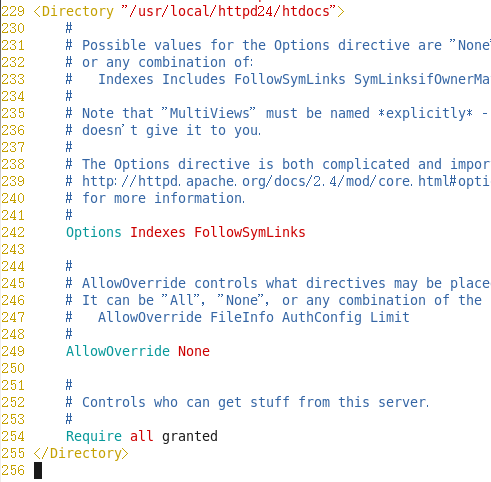
启动服务后转换的身份，在启动服务时通常以root身份，然后转换身份，这样增加系统安全



对根目录的一个权限的设置



网页文件存放的目录



对网页文件存放目录的一个权限设置。

options中Indexes表示当网页不存在的时候允许索引显示目录中的文件，FollowSymLinks是否允许访问符号链接文件。有的选项有ExecCGI表是否使用CGI，如Options Includes ExecCGI FollowSymLinks表示允许服务器执行CGI及SSI，禁止列出目录。

SymLinksOwnerMatch表示当符号链接的文件和目标文件为同一用户拥有时才允许访问。

AllowOverride None表示不允许这个目录下的访问控制文件来改变这里的配置，这也意味着不用查看这个目录下的访问控制文件，修改为：AllowOverride All 表示允许.htaccess。

Order对页面的访问控制顺序后面的一项是默认选项，如allow，deny则默认是deny，

Allowfromall表示允许所有的用户，通过和上一项结合可以控制对网站的访问控制

Require　访问验证方式

Require all granted 允许所有请求访问资源

Require all denied 拒绝所有请求访问资源

Require env env-var [env-var] ...   当指定环境变量设置时允许访问

Require method http-method [http-method] ...允许指定的http请求方法访问资源

Require expr expression 当expression返回true时允许访问资源

Require user userid [userid] ... 允许指定的用户id访问资源

Require group group-name [group-name] ... 允许指定的组内的用户访问资源

Require valid-user 所有有效的用户可访问资源

Require ip 10 172.20 192.168.2 允许指定IP的客户端可访问资源

Require not group selectselect 组内的用户不可访问资源

通常的访问设置如下：

Order Deny,Allow

Allow from All

解释：

注意“Deny,Allow”中间只有一个逗号，也只能有一个逗号，有空格都会出错；单词的大小写不限。上面设定的含义是先设定“先检查禁止设定，没有禁止的全部允许”，而第二句没有Deny，也就是没有禁止访问的设定，直接就是允许所有访问了。这个主要是用来确保或者覆盖上级目录的设置，开放所有内容的访问权。

Order Allow,Deny

Deny from All

解释：无条件禁止访问

Order Deny,Allow

Deny from ip1 ip2

或者

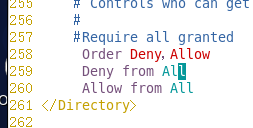
Order Allow,Deny

Allow from all

Deny from ip1 ip2

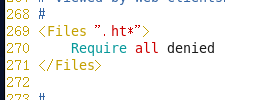
解释：禁止部分内访问，其他的全部开放

注：apache会按照order决定最后使用哪一条规则，比如上面的第二种方式，虽然第二句allow允许了访问，但由于在order中allow不是最后规则，因此还需要看有没有deny规则，于是到了第三句，符合ip1和ip2的访问就被禁止了。注意，order决定的“最后”规则非常重要

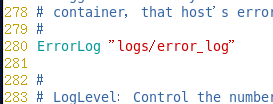
例如：（不存的话在自己加上）



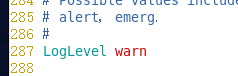
指定所要访问的主页的默认主页名字，默认首页文件名为index.html



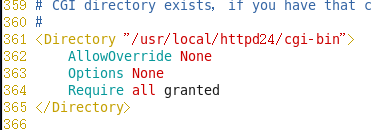
控制不让web用户来查看.ht开头的所有文件



错误日志存放位置

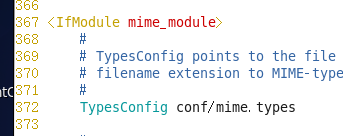


Apache日志的级别。选项：debug(调试) ,info ,notice(提示) ,warn(警告) ,error(错误) ,crit

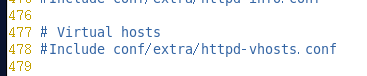


对/usr/local/httpd4/cgi-bin/文件访问权限。

修改为 Options MultiViews FollowSymLinks表示不在浏览器上显示树状目录结构



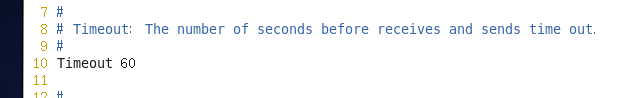
用于设置保存不同MIME类型数据的文件名

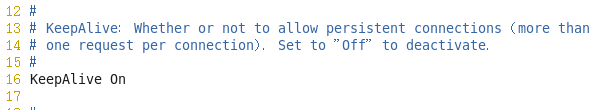


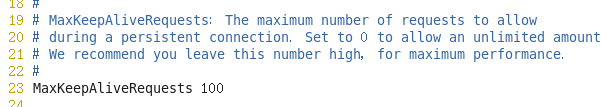
开启虚拟机配置。配置文件在conf/extra/httpd-vhosts.conf

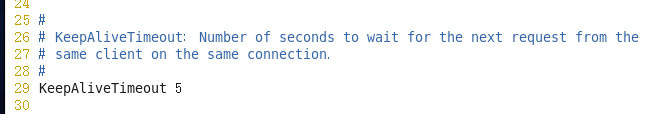
### httpd-default.conf配置

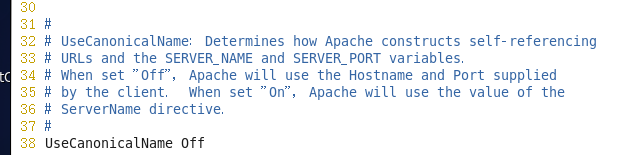
配置文件在：/usr/local/httpd24/conf/extra/httpd-default.conf

 服务器与客户端断开的时间

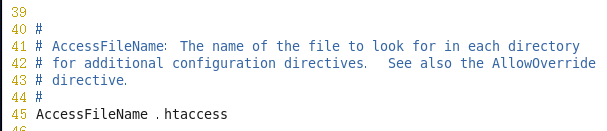
 是否持续连接（因为每次连接都得三次握手，如果是访问量不大，建议打开此项，如果网站访问量比较大关闭此项比较好），修改为：KeepAlive On 表示允许程序性联机

 表示一个连接的最大请求数

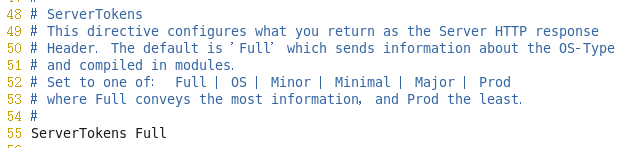
 断开连接前的时间



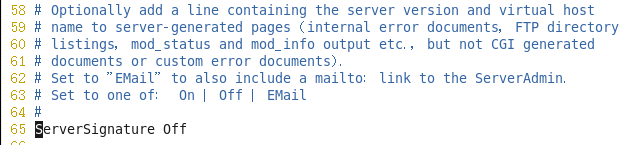
如果客户端提供了主机名和端口，Apache将会使用客户端提供的这些信息来构建自引用URL。这些值与用于实现基于域名的虚拟主机的值相同，并且对于同样的客户端可用。CGI变量SERVER\_NAME和SERVER\_PORT也会由客户端提供的值来构建



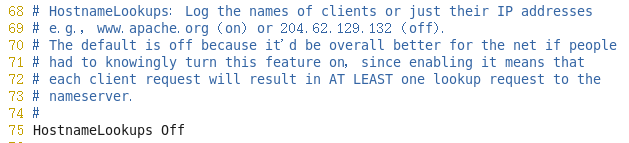
定义每个目录下的访问控制文件名，缺省为.htaccess



在出现错误页的时候是否显示服务器操作系统的名称，ServerTokens Prod为不显示



定义当客户请求的网页不存在，或者错误的时候是否提示apache的版本的一些信息



当打开此项功能时，在记录日志的时候同时记录主机名，这需要服务器来反向解析域名，增加了服务器的负载，通常不建议开启

建议：更多配置详情请参考：<http://www.linuxidc.com/Linux/2015-02/113921.htm>

或者 <http://httpd.apache.org/docs/2.4/configuring.html>

注意事项:

1. 修改配置后记得要重新启动apache服务
2. 在修改配置前记得备份

## 五，注意事项

### 若需要局域网访问，则需要配置防火墙

方案一：关闭防火墙

service iptables stop 关闭防火墙

chkconfig iptables off 禁止开机启动

方案二：开放80端口

vim /etc/sysconfig/iptables 编辑防火墙文件

添加一条规则：

-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT

service iptables restart

### 配置apache快捷启动

进入apache的bin目录：

Shell># cd /usr/local/httpd24/bin/

复制apache可执行文件到系统服务目录：

Shell># cp apachectl /etc/init.d/apache

增加执行权限：

Shell># chmod +x /etc/init.d/apache

编辑环境变量：

Shell># vim /etc/profile

加入环境变量：export PATH=/usr/local/httpd24/bin:$PATH

刷新环境变量：

Shell># source /etc/profile

### 配置apache开机启动

方案一：chkconfig

Shell># chkconfig --add apache 添加apache到系统服务

Shell># chkconfig apache on 设置开启启动

注：需要先设置快捷启动方式。

方案二：编辑自启动配置文件 /etc/rc.d/rc.local

Shell># vim /etc/rc.d/rc.local

加入启动apache的命令：

/usr/local/httpd24/bin/apachectl start

保存退出

Shell># reboot

重新启动即可。

方案三：编辑用户自启动配置文件 /用户目录/.bashrc

例如：当前是admin用户

Shell># vim /home/admin/.bashrc

加入启动 apche命令：

/usr/local/httpd24/bin/apachectl start

保存退出

Shell># reboot

重新启动即可。

注：该方案之后admin用户登陆系统才会自启动apache，其他用户登陆系统是不会自启动apache服务的。