|  |
| --- |
| Neusoft |
| CentOS & Weblogic & Oracle & Nginx |
| CentOS 6.4 Nginx 安装、负载均衡 |

|  |
| --- |
| 成阳阳  2013/11/27 |
|  |

目录

[CentOS 6.4 安装 Nginx 3](#_Toc387323851)

[1、RPM包管理,添加nginx rpm包自动下载配置文件 (http://nginx.org/en/linux\_packages.html#mainline ) 3](#_Toc387323852)

[2 CentOS中设置系统级代理 (根据实际需要) 3](#_Toc387323853)

[3、安装nginx 4](#_Toc387323854)

[4、nginx的几个默认目录 4](#_Toc387323855)

[5、常用命令 5](#_Toc387323856)

[6、手动安装 5](#_Toc387323857)

# CentOS 6.4 安装 Nginx

## 1、RPM包管理,添加nginx rpm包自动下载配置文件 (http://nginx.org/en/linux\_packages.html#mainline )

官网说明

|  |
| --- |
| Pre-Built Packages for Mainline version  To set up the yum repository for RHEL/CentOS, create the file named /etc/yum.repos.d/nginx.repo with the following contents:  [nginx]  name=nginx repo  baseurl=http://nginx.org/packages/mainline/OS/OSRELEASE/$basearch/  gpgcheck=0  enabled=1  Replace “OS” with “rhel” or “centos”, depending on the distribution used, and “OSRELEASE” with “5” or “6”, for 5.x or 6.x versions, respectively. |

## 2 [CentOS中设置系统级代理](http://www.cnblogs.com/cocowool/archive/2012/07/05/2578487.html) (根据实际需要)

**YUM代理设置**

编辑/etc/yum.conf，在最后加入

# Proxy

proxy=http://username:password@proxy\_ip:port/

也可以使用proxy\_username和proxy\_password来配置代理的用户名和密码

这样的配置完成后，所有的用户在使用yum时，都会使用代理，可以说是全局代理。

如果需要为单独的用户配置代理，可以在用户的~/.bash\_profile中设置。

**Wget的代理设置**

编辑/etc/wgetrc，在最后加入

# Proxy

http\_proxy=http://username:password@proxy\_ip:port/

ftp\_proxy=http://username:password@proxy\_ip:port/

**系统全局代理**

如果需要为某个用户设置一个系统级的代理，可以在~/.bash\_profile中设置：

http\_proxy="http://username:password@proxy\_ip:port"

export\_http\_proxy

上面的设置只对某个用户生效，如果要对所有系统用户生效，写在/etc/profile中就可以了。

## 3、安装nginx

默认安装 生成nginx服务

|  |
| --- |
| yum update 更新yum  wget http://nginx.org/packages/centos/6/noarch/RPMS/nginx-release-centos-6-0.el6.ngx.noarch.rpm  rpm –ivh nginx-release-centos-6-0.el6.ngx.noarch.rpm  yum install nginx |

会显示一大堆信息，问你ok不ok啊：Is this ok [y/N]:  
输入y，屏幕滚了一会之后就安装完毕，最后提示“Complete!”就是安完了。

## 4、nginx的几个默认目录

whereis nginx  
nginx: /usr/sbin/nginx /etc/nginx /usr/share/nginx  
  
1 配置所在目录：/etc/nginx/  
2 PID目录：/var/run/nginx.pid  
3 错误日志：/var/log/nginx/error.log  
4 访问日志：/var/log/nginx/access.log  
5 默认站点目录：/usr/share/nginx/html

## 5、常用命令

1 启动nginx：nginx

2 重启nginx：killall -HUP nginx

3 测试nginx配置：nginx -t

4 重新加载：nginx -s reload

5 停止： nginx -s stop

6 版本： nginx –v

7 卸载： yum remove nginx

注： 使用 yum install 安装后，末将nginx自动配置在系统服务中，所以也可以使用service命令查看状态或对nginx服务进行启动、停止、重启等操作。

## 6、手动安装

安装nginx   
    1、在nginx官方网站下载一个包，下载地址是：<http://nginx.org/en/download.html>   
    2、WinSCP(ftp上传工具).exe FTP 上传工具上传达到CentOS中   
    3、我是先建一个目录，把要用的到的安装包放到一起.   
    #tar zxf nginx-1.4.1.tar.gz   
    
    4、安装pcre开发包   
    #yum install -y pcre-devel   
    5、如果安装出现在下面的错误是缺少编译环境。安装编译源码所需的工具和库   
    ./configure: error: C compiler cc is not found   
    #yum install gcc gcc-c++ ncurses-devel perl   
    6、安装cmake，从http://www.cmake.org下载源码并编译安装   
    #yum -y install make gcc gcc-c++ ncurses-devel   
    #yum -y install zlib zlib-devel   
    7、如果需要ssl功能需要openssl库   
    #yum -y install openssl openssl--devel   
    8、安装nginx   
    #cd nginx-1.4.1   
    #./configure --prefix=/usr/local/nginx --with-http\_stub\_status\_module

|  |
| --- |
| Nginx的编译参数如下： ./configure --user=root --group=root --prefix=/usr/local/nginx/nginx-app --add-module=../ngx\_cache\_purge-2.3 --with-openssl=/usr/lib64/openssl --with-pcre --with-http\_stub\_status\_module --with-http\_gzip\_static\_module --with-http\_image\_filter\_module |
| ＃mac中安装  # http://www.pcre.org/  #下载安装pcre  ./configure --prefix=/Users/mattdamon/Application/lib/pcre --enable-utf8  ./configure --prefix=/Users/mattdamon/Application/nginx/nginx-proxy --with-pcre=/Users/mattdamon/Application/lib/pcre --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_gzip\_static\_module --with-http\_image\_filter\_module --add-module=/Users/mattdamon/Application/lib/ngx\_cache\_purge --with-cc-opt="-Wno-deprecated-declarations" --without-http\_fastcgi\_module  make  make install |

    #make   
    #make install   
    9、启动服务   
    #/usr/local/nginx/sbin/nginx  -c /usr/local/nginx/conf/nginx.conf

注： 启动服务前需要在 /usr/local/nginx 下创建logs文件夹  
    10、停止服务   
     #/usr/local/nginx/sbin/nginx -s stop   
    11、查看端口占用情况   
    #netstat -tunlp   
    12、如果其它机器无法访问，解决方法如下：   
    #/sbin/iptables -I INPUT -p tcp --dport 80 -j ACCEPT   
    然后保存：   
    #/etc/rc.d/init.d/iptables save   
    重启防火墙   
    #/etc/init.d/iptables restart

# 标准 HTTP 模块

这些模块是默认均被编译进去的，除非通过configure明确地禁用它们。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **版本** | **禁用选项** |
| [Core](http://wiki.nginx.org/HttpCoreModuleChs) | 控制端口、location、错误页、别名和其他必要配置。 |  | --without-http |
| [Access](http://wiki.nginx.org/HttpAccessModuleChs) | 基于客户端IP地址允许/拒绝Http请求。 |  | --without-http\_access\_module |
| [Auth Basic](http://wiki.nginx.org/HttpAuthBasicModuleChs) | Basic HTTP 认证。 |  | --without-http\_auth\_basic\_module |
| [Auto Index](http://wiki.nginx.org/HttpAutoindexModuleChs) | 自动为没有索引页面的目录生成文件列表。 |  | --without-http\_autoindex\_module |
| [Browser](http://wiki.nginx.org/index.php?title=HttpBrowserModuleChs&action=edit&redlink=1) | 翻译"User-Agent" 字符串。 | 0.4.3 | --without-http\_browser\_module |
| [Charset](http://wiki.nginx.org/index.php?title=HttpCharsetModuleChs&action=edit&redlink=1) | 对Web页面重新编码。 |  | --without-http\_charset\_module |
| [Empty GIF](http://wiki.nginx.org/HttpEmptyGifModuleChs) | 在内存中生成一个高1像素、宽1像素的GIF图片，并对外提供服务。 | 0.3.10 | --without-http\_empty\_gif\_module |
| [FastCGI](http://wiki.nginx.org/HttpFastcgiModuleChs) | 提供对FastCGI的支持。 |  | --without-http\_fastcgi\_module |
| [Geo](http://wiki.nginx.org/HttpGeoModuleChs) | 通过IP地址的键值对来设置配置参数。 | 0.1.17 | --without-http\_geo\_module |
| [Gzip](http://wiki.nginx.org/HttpGzipModuleChs) | 支持使用Gzip压缩响应内容。 |  | --without-http\_gzip\_module |
| [Headers](http://wiki.nginx.org/HttpHeadersModuleChs) | 可以设定任意的HTTP响应头。 |  |  |
| [Index](http://wiki.nginx.org/HttpIndexModuleChs) | 控制将哪些文件作为索引页面。 |  |  |
| [Limit Requests](http://wiki.nginx.org/index.php?title=HttpLimitReqModuleChs&action=edit&redlink=1) | 限定客户端创建连接的频率。 | 0.7.20 | --without-http\_limit\_req\_module |
| [Limit Zone](http://wiki.nginx.org/HttpLimitZoneModuleChs) | 限定客户端同时建立连接的最大数。从1.1.8起不再推荐使用，建议使用 Limit Conn。 | 0.5.6 | --without-http\_limit\_zone\_module |
| [Limit Conn](http://wiki.nginx.org/index.php?title=HttpLimitConnModuleChs&action=edit&redlink=1) | 通过配置参数限定客户端的最大并发连接数。 |  | --without-http\_limit\_conn\_module |
| [Log](http://wiki.nginx.org/HttpLogModuleChs) | 可以定制访问日志。 |  |  |
| [Map](http://wiki.nginx.org/index.php?title=HttpMapModuleChs&action=edit&redlink=1) | 以键值对的方式设定任意配置参数。 | 0.3.16 | --without-http\_map\_module |
| [Memcached](http://wiki.nginx.org/HttpMemcachedModuleChs) | 支持Memcached。 |  | --without-http\_memcached\_module |
| [Proxy](http://wiki.nginx.org/HttpProxyModuleChs) | 代理上级服务器。 |  | --without-http\_proxy\_module |
| [Referer](http://wiki.nginx.org/index.php?title=HttpRefererModuleChs&action=edit&redlink=1) | 基于RefererHTTP请求头对请求进行过滤。 |  | --without-http\_referer\_module |
| [Rewrite](http://wiki.nginx.org/HttpRewriteModuleChs) | 使用正则表达式对请求进行重写。 |  | --without-http\_rewrite\_module |
| [SCGI](http://wiki.nginx.org/index.php?title=HttpScgiModuleChs&action=edit&redlink=1) | 支持SCGI协议。 | 0.8.42 | --without-http\_scgi\_module |
| [Split Clients](http://wiki.nginx.org/index.php?title=HttpSplitClientsModuleChs&action=edit&redlink=1) | 基于某些条件将客户端分类。 | 0.8.37 | --without-http\_split\_clients\_module |
| [SSI](http://wiki.nginx.org/HttpSsiModuleChs) | 支持服务器端的include。 |  | --without-http\_ssi\_module |
| [Upstream](http://wiki.nginx.org/HttpUpstreamModuleChs) | 支持负载均衡。 |  | --without-http\_upstream\_ip\_hash\_module (ip\_hash directive only) |
| [User ID](http://wiki.nginx.org/index.php?title=HttpUseridModuleChs&action=edit&redlink=1) | 发行用户身份标识cookie。 |  | --without-http\_userid\_module |
| [uWSGI](http://wiki.nginx.org/index.php?title=HttpUwsgiModuleChs&action=edit&redlink=1) | 支持uWSGI协议。 | 0.8.40 | --without-http\_uwsgi\_module |
| [X-Accel](http://wiki.nginx.org/index.php?title=X-accelChs&action=edit&redlink=1) | 类X-Sendfile模块。 |  |  |

# 可选 HTTP 模块

下面的模块必须在编译时为 configure 明确地设定相应的启用选项。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **版本** | **启用选项** |
| [Addition](http://wiki.nginx.org/index.php?title=HttpAdditionModuleChs&action=edit&redlink=1) | 可以为页面追加文本。 |  | --with-http\_addition\_module |
| [Degradation](http://wiki.nginx.org/index.php?title=HttpDegradationModuleChs&action=edit&redlink=1) | 允许当剩余内存较低时对某些位置的请求返回 204 或 404 状态码。 | 0.8.25 | --with-http\_degradation\_module |
| [Embedded Perl](http://wiki.nginx.org/index.php?title=HttpPerlModuleChs&action=edit&redlink=1) | 允许在配置文件中使用 Perl。 | 0.3.21 | --with-http\_perl\_module |
| [FLV](http://wiki.nginx.org/index.php?title=HttpFlvModuleChs&action=edit&redlink=1) | 支持 Flash视频(Flash Streaming Video)。 | 0.4.7 | --with-http\_flv\_module |
| [GeoIP](http://wiki.nginx.org/index.php?title=HttpGeoipModuleChs&action=edit&redlink=1) | 根据来自 [MaxMind](http://www.maxmind.com/) 的 GeoIP 二进制文件中的信息创建变量。 | 0.8.6, 0.7.63 | --with-http\_geoip\_module |
| [Google Perftools](http://wiki.nginx.org/index.php?title=GooglePerftoolsModuleChs&action=edit&redlink=1) | 支持 Google 性能工具。 | 0.6.29 | --with-google\_perftools\_module |
| [Gzip Precompression](http://wiki.nginx.org/index.php?title=HttpGzipStaticModuleChs&action=edit&redlink=1) | 提供预压缩的静态文件。 | 0.6.23 | --with-http\_gzip\_static\_module |
| [Image Filter](http://wiki.nginx.org/index.php?title=HttpImageFilterModuleChs&action=edit&redlink=1) | 支持使用 Libgd 对图片进行变换。 | 0.7.54 | --with-http\_image\_filter\_module |
| [MP4](http://wiki.nginx.org/index.php?title=HttpMp4ModuleChs&action=edit&redlink=1) | 允许对 mp4 流进行寻址。 | 1.1.3, 1.0.7 | --with-http\_mp4\_module |
| [Random Index](http://wiki.nginx.org/index.php?title=HttpRandomIndexModuleChs&action=edit&redlink=1) | 允许使目录索引随机化。 | 0.7.15 | --with-http\_random\_index\_module |
| [Real IP](http://wiki.nginx.org/HttpRealipModuleChs) | 允许将 nginx 作为后端服务器。 | 0.3.8 | --with-http\_realip\_module |
| [Secure Link](http://wiki.nginx.org/HttpSecureLinkModuleChs) | 允许通过密钥保护指定的文件。 | 0.7.18 | --with-http\_secure\_link\_module |
| [SSL](http://wiki.nginx.org/index.php?title=HttpSslModuleChs&action=edit&redlink=1) | 支持 HTTPS/SSL。 |  | --with-http\_ssl\_module |
| [Stub Status](http://wiki.nginx.org/HttpStubStatusModuleChs) | 提供查看服务器统计信息的能力。 | 0.1.18 | --with-http\_stub\_status\_module |
| [Substitution](http://wiki.nginx.org/index.php?title=HttpSubModuleChs&action=edit&redlink=1) | 可以替换页面中的文本。 |  | --with-http\_sub\_module |
| [WebDAV](http://wiki.nginx.org/HttpDavModuleChs) | 支持 WebDAV 穿透。 | 0.3.38 | --with-http\_dav\_module |
| [XSLT](http://wiki.nginx.org/index.php?title=HttpXsltModuleChs&action=edit&redlink=1) | 允许对页面使用 XSLT 进行后处理。 | 0.7.8 | --with-http\_xslt\_module |

# 邮件模块

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **配置参数** |
| [Core](http://wiki.nginx.org/index.php?title=MailCoreModuleChs&action=edit&redlink=1) | 使 Nginx 能够处理/代理 IMAP、POP3、SMTP协议。 | --with-mail --without-mail\_pop3\_module --without-mail\_imap\_module --without-mail\_smtp\_module |
| [Auth](http://wiki.nginx.org/index.php?title=MailAuthModuleChs&action=edit&redlink=1) | 使用 Nginx 为邮件服务进行认证。 |  |
| [Proxy](http://wiki.nginx.org/index.php?title=MailProxyModuleChs&action=edit&redlink=1) | 使 Nginx 能够代理 IMAP、POP3、SMTP协议。 |  |
| [SSL](http://wiki.nginx.org/index.php?title=MailSslModuleChs&action=edit&redlink=1) | 该模块可以让 POP3/IMAP/SMTP 使用 SSL/TLS。 | --with-mail\_ssl\_module |