



链滴

# nginx 安装

作者: [boobam](#)

原文链接: <https://ld246.com/article/1495617319411>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 一、下载安装软件

官网: <https://ld246.com/forward?goto=http%3A%2F%2Fnginx.org%2F> <http://nginx.org/>

## 二、安装前准备

### 2.1.创建 nginx 账号:

```
groupadd -g 600 nginx
```

```
useradd -u 600 -g 600 -M -s /sbin/nologin nginx
```

```
mkdir /var/tmp/nginx/
```

### 2.2 安装依赖包

```
yum install pcre-devel pcre zlib zlib-devel openssl openssl-devel
```

## 三、编译安装 nginx

```
./configure --prefix=/usr/local/nginx/ \
```

```
--conf-path=/usr/local/nginx/conf/nginx.conf \
```

```
--error-log-path=/usr/local/nginx/logs/error.log \
```

```
--http-log-path=/usr/local/nginx/logs/access.log \
```

```
--pid-path=/var/run/nginx/nginx.pid \
```

```
--lock-path=/var/lock/nginx.lock \
```

```
--user=nginx \
```

```
--group=nginx \
```

```
--with-http_ssl_module \
```

```
--with-http_flv_module \
```

```
--with-http_stub_status_module \
```

```
--with-http_gzip_static_module \
```

```
--http-client-body-temp-path=/var/tmp/nginx/client/ \
```

```
--http-proxy-temp-path=/var/tmp/nginx/proxy/ \
```

```
--http-fastcgi-temp-path=/var/tmp/nginx/fcgi/ \
```

```
--http-uwsgi-temp-path=/var/tmp/nginx/uwsgi \
```

```
--http-scgi-temp-path=/var/tmp/nginx/scgi \
```

```
--with-pcre
```

```
make
```

```
make install
```

## 附录: nginx 编译参数:

--prefix= 指向安装目录

--sbin-path= 指向 (执行) 程序文件 (nginx)

--conf-path= 指向配置文件 (nginx.conf)

--error-log-path= 指向错误日志目录

--pid-path= 指向 pid 文件 (nginx.pid)

--lock-path= 指向 lock 文件 (nginx.lock) (安装文件锁定, 防止安装文件被别人利用, 或自误操作。)

--user= 指定程序运行时的非特权用户

--group= 指定程序运行时的非特权用户组

--builddir= 指向编译目录

--with-rtsig\_module 启用 rtSIG 模块支持 (实时信号)

--with-select\_module 启用 select 模块支持 (一种轮询模式, 不推荐在高载环境下使用) 禁用: --without-select\_module

--with-poll\_module 启用 poll 模块支持 (功能与 select 相同, 与 select 特性相同, 为一种轮询模式, 不推荐在高载环境下使用)

--with-file-aio 启用 file aio 支持 (一种 APL 文件传输格式)

--with-ipv6 启用 ipv6 支持

--with-http\_ssl\_module 启用 ngx\_http\_ssl\_module 支持 (使支持 https 请求, 需已安装 openssl)

--with-http\_realip\_module 启用 ngx\_http\_realip\_module 支持 (这个模块允许从请求标头更客户端的 IP 地址值, 默认为关)

--with-http\_addition\_module 启用 ngx\_http\_addition\_module 支持 (作为一个输出过滤器

支持不完全缓冲，分部分响应请求) </p>

<p>--with-http\_xslt\_module 启用 ngx\_http\_xslt\_module 支持 (过滤转换 XML 请求) </p>

<p>--with-http\_image\_filter\_module 启用 ngx\_http\_image\_filter\_module 支持 (传输 JPEG/GIF PNG 图片的一个过滤器) (默认为不启用。gd 库要用到) </p>

<p>--with-http\_geoip\_module 启用 ngx\_http\_geoip\_module 支持 (该模块创建基于与 MaxMind GeoIP 二进制文件相配的客户端 IP 地址的 ngx\_http\_geoip\_module 变量) </p>

<p>--with-http\_sub\_module 启用 ngx\_http\_sub\_module 支持 (允许用一些其他文本替换 nginx 应中的一些文本) </p>

<p>--with-http\_dav\_module 启用 ngx\_http\_dav\_module 支持 (增加 PUT,DELETE,MKCOL: 创集合,COPY 和 MOVE 方法) 默认情况下为关闭, 需编译开启</p>

<p>--with-http\_flv\_module 启用 ngx\_http\_flv\_module 支持 (提供寻求内存使用基于时间的偏移文件) </p>

<p>--with-http\_gzip\_static\_module 启用 ngx\_http\_gzip\_static\_module 支持 (在线实时压缩输数据流) </p>

<p>--with-http\_random\_index\_module 启用 ngx\_http\_random\_index\_module 支持 (从目录中机挑选一个目录索引) </p>

<p>--with-http\_secure\_link\_module 启用 ngx\_http\_secure\_link\_module 支持 (计算和检查要求需的安全链接网址) </p>

<p>--with-http\_degradation\_module 启用 ngx\_http\_degradation\_module 支持 (允许在内存足的情况下返回 204 或 444 码) </p>

<p>--with-http\_stub\_status\_module 启用 ngx\_http\_stub\_status\_module 支持 (获取 nginx 自次启动以来的工作状态) </p>

<p>--without-http\_charset\_module 禁用 ngx\_http\_charset\_module 支持 (重新编码 web 页面但只能是一个方向--服务器端到客户端, 并且只有一个字节的编码可以被重新编码) </p>

<p>--without-http\_gzip\_module 禁用 ngx\_http\_gzip\_module 支持 (该模块同-with-http\_gzip\_satic\_module 功能一样) </p>

<p>--without-http\_ssi\_module 禁用 ngx\_http\_ssi\_module 支持 (该模块提供了一个在输入端处处理服务器包含文件 (SSI) 的过滤器, 目前支持 SSI 命令的列表是不完整的) </p>

<p>--without-http\_userid\_module 禁用 ngx\_http\_userid\_module 支持 (该模块用来处理用来确客户端后续请求的 cookies) </p>

<p>--without-http\_access\_module 禁用 ngx\_http\_access\_module 支持 (该模块提供了一个简的基于主机的访问控制。允许/拒绝基于 ip 地址) </p>

<p>--without-http\_auth\_basic\_module 禁用 ngx\_http\_auth\_basic\_module (该模块是可以使用户名和密码基于 http 基本认证方法来保护你的站点或其部分内容) </p>

<p>--without-http\_autoindex\_module 禁用 disable ngx\_http\_autoindex\_module 支持 (该模用于自动生成目录列表, 只在 ngx\_http\_index\_module 模块未找到索引文件时发出请求。) </p>

<p>--without-http\_geo\_module 禁用 ngx\_http\_geo\_module 支持 (创建一些变量, 其值依赖于户端的 IP 地址) </p>

<p>--without-http\_map\_module 禁用 ngx\_http\_map\_module 支持 (使用任意的键/值对设置配变量) </p>

<p>--without-http\_split\_clients\_module 禁用 ngx\_http\_split\_clients\_module 支持 (该模块用来于某些条件划分用户。条件如: ip 地址、报头、cookies 等等) </p>

<p>--without-http\_referer\_module 禁用 disable ngx\_http\_referer\_module 支持 (该模块用来过请求, 拒绝报头中 Referer 值不正确的请求) </p>

<p>--without-http\_rewrite\_module 禁用 ngx\_http\_rewrite\_module 支持 (该模块允许使用正则达式改变 URI, 并且根据变量来转向以及选择配置。如果在 server 级别设置该选项, 那么他们将在 location 之前生效。如果在 location 还有更进一步的重写规则, location 部分的规则依然会被执行。果这个 URI 重写是因为 location 部分的规则造成的, 那么 location 部分会再次被执行作为新的 URI 这个循环会执行 10 次, 然后 Nginx 会返回一个 500 错误。) </p>

<p>--without-http\_proxy\_module 禁用 ngx\_http\_proxy\_module 支持 (有关代理服务器) </p>

<p>--without-http\_fastcgi\_module 禁用 ngx\_http\_fastcgi\_module 支持 (该模块允许 Nginx 与 astCGI 进程交互, 并通过传递参数来控制 FastCGI 进程工作。) FastCGI 一个常驻型的公网网关接。 </p>

<p>--without-http\_uwsgi\_module 禁用 ngx\_http\_uwsgi\_module 支持 (该模块用来医用 uwsgi

议, uWSGI 服务器相关) </p>

<p>--without-http\_scgi\_module 禁用 ngx\_http\_scgi\_module 支持 (该模块用来启用 SCGI 协议支持, SCGI 协议是 CGI 协议的替代。它是一种应用程序与 HTTP 服务接口标准。它有些像 FastCGI 但的设计 更容易实现。) </p>

<p>--without-http\_<a href="https://ld246.com/forward?goto=http%3A%2F%2Fwww.ttlsa.com%2Fnosql%2Fmemcache%2F" target="\_blank" rel="nofollow ugc">memcache</a>\_module 禁用 ngx\_http\_memcached\_module 支持 (该模块用来提供简单的缓存, 以提高系统效率) </p>

<p>--without-http\_limit\_zone\_module 禁用 ngx\_http\_limit\_zone\_module 支持 (该模块可以针条件, 进行会话的并发连接数控制) </p>

<p>--without-http\_limit\_req\_module 禁用 ngx\_http\_limit\_req\_module 支持 (该模块允许你对一个地址进行请求数量的限制用一个给定的 session 或一个特定的事件) </p>

<p>--without-http\_empty\_gif\_module 禁用 ngx\_http\_empty\_gif\_module 支持 (该模块在内存常驻了一个 1\*1 的透明 GIF 图像, 可以被非常快速的调用) </p>

<p>--without-http\_browser\_module 禁用 ngx\_http\_browser\_module 支持 (该模块用来创建依赖于请求报头的值。如果浏览器为 modern, 则<span class="language-math">modern\_browser</span>于modern\_browser\_value指令分配的值; 如果浏览器为old, 则ancient\_browser等于ancient\_browser\_value指令分配的值; 如果浏览器为 MSIE 中的任意版本, 则 \$msie 等于 1) </p>

<p>--without-http\_upstream\_ip\_hash\_module 禁用 ngx\_http\_upstream\_ip\_hash\_module 支持 (该模块用于简单的负载均衡) </p>

<p>--with-http\_<a href="https://ld246.com/forward?goto=http%3A%2F%2Fwww.ttlsa.com%2Fperl%2F" target="\_blank" rel="nofollow ugc">perl</a>\_module 启用 ngx\_http\_perl\_module 支持 (该模块使 nginx 可以直接使用 perl 或通过 ssi 调用 perl) </p>

<p>--with-perl\_modules\_path= 设定 perl 模块路径</p>

<p>--with-perl= 设定 perl 库文件路径</p>

<p>--http-log-path= 设定 access log 路径</p>

<p>--http-client-body-temp-path= 设定 http 客户端请求临时文件路径</p>

<p>--http-proxy-temp-path= 设定 http 代理临时文件路径</p>

<p>--http-fastcgi-temp-path= 设定 http fastcgi 临时文件路径</p>

<p>--http-uwsgi-temp-path= 设定 http uwsgi 临时文件路径</p>

<p>--http-scgi-temp-path= 设定 http scgi 临时文件路径</p>

<p>-without-http 禁用 http server 功能</p>

<p>--without-http-cache 禁用 http cache 功能</p>

<p>--with-mail 启用 POP3/IMAP4/SMTP 代理模块支持</p>

<p>--with-mail\_ssl\_module 启用 ngx\_mail\_ssl\_module 支持</p>

<p>--without-mail\_pop3\_module 禁用 pop3 协议 (POP3 即邮局协议的第 3 个版本,它是规定个计算机如何连接到互联网上的邮件服务器进行收发邮件的协议。是因特网电子邮件的第一个离线协议标准,POP3 协议允许用户从服务器上把邮件存储到本地主机上,同时根据客户端的操作删除或保存在邮服务器上的邮件。POP3 协议是 TCP/IP 协议族中的一员, 主要用于支持使用客户端远程管理在服务器上的电子邮件) </p>

<p>--without-mail\_imap\_module 禁用 imap 协议 (一种邮件获取协议。它的主要作用是邮件客户可以通过这种协议从邮件服务器上获取邮件的信息, 下载邮件等。IMAP 协议运行在 TCP/IP 协议之, 使用的端口是 143。它与 POP3 协议的主要区别是用户可以不用把所有的邮件全部下载, 可以通过客户端直接对服务器上的邮件进行操作。) </p>

<p>--without-mail\_smtp\_module 禁用 smtp 协议 (SMTP 即简单邮件传输协议,它是一组用于由地址到目的地址传送邮件的规则, 由它来控制信件的中转方式。SMTP 协议属于 TCP/IP 协议族, 它助每台计算机在发送或中转信件时找到下一个目的地。) </p>

<p>--with-google\_perftools\_module 启用 ngx\_google\_perftools\_module 支持 (调试用, 剖析序性能瓶颈) </p>

<p>--with-cpp\_test\_module 启用 ngx\_cpp\_test\_module 支持</p>

<p>--add-module= 启用外部模块支持</p>

<p>--with-cc= 指向 C 编译器路径</p>

<p>--with-cpp= 指向 C 预处理路径</p>

<p>--with-cc-opt= 设置 C 编译器参数 (PCRE 库, 需要指定--with-cc-opt=" -I /usr/local/includ", 如果使用 select()函数则需要同时增加文件描述符数量, 可以通过--with-cc- opt=" -D FD\_SETSI

E=2048" 指定。) </p>  
<p>--with-ld-opt= 设置连接文件参数。(PCRE 库, 需要指定--with-ld-opt=" -L /usr/local/lib"  
) </p>  
<p>--with-cpu-opt= 指定编译的 CPU, 可用的值为: pentium, pentiumpro, pentium3, pentium4,  
athlon, opteron, amd64, sparc32, sparc64, ppc64</p>  
<p>--without-pcre 禁用 pcre 库</p>  
<p>--with-pcre 启用 pcre 库</p>  
<p>--with-pcre= 指向 pcre 库文件目录</p>  
<p>--with-pcre-opt= 在编译时为 pcre 库设置附加参数</p>  
<p>--with-md5= 指向 md5 库文件目录(消息摘要算法第五版, 用以提供消息的完整性保护) </p>  
<p>--with-md5-opt= 在编译时为 md5 库设置附加参数</p>  
<p>--with-md5-asm 使用 md5 汇编源</p>  
<p>--with-sha1= 指向 sha1 库目录(数字签名算法, 主要用于数字签名) </p>  
<p>--with-sha1-opt= 在编译时为 sha1 库设置附加参数</p>  
<p>--with-sha1-asm 使用 sha1 汇编源</p>  
<p>--with-zlib= 指向 zlib 库目录</p>  
<p>--with-zlib-opt= 在编译时为 zlib 设置附加参数</p>  
<p>--with-zlib-asm= 为指定的 CPU 使用 zlib 汇编源进行优化, CPU 类型为 pentium, pentiumpr  
</p>  
<p>--with-libatomic 为原子内存的更新操作的实现提供一个架构</p>  
<p>--with-libatomic= 指向 libatomic ops 安装目录</p>  
<p>--with-openssl= 指向 openssl 安装目录</p>  
<p>--with-openssl-opt 在编译时为 openssl 设置附加参数</p>  
<p>--with-debug 启用 debug 日志</p>