<https://baike.baidu.com/item/nginx/3817705?fr=aladdin>

*Nginx* (engine x) 是一个高性能的[HTTP](https://baike.baidu.com/item/HTTP" \t "_blank)和[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86" \t "_blank)服务器，也是一个IMAP/POP3/SMTP[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)。Nginx是由伊戈尔·赛索耶夫为[俄罗斯](https://baike.baidu.com/item/%E4%BF%84%E7%BD%97%E6%96%AF/125568)访问量第二的Rambler.ru站点（俄文：Рамблер）开发的，第一个公开版本0.1.0发布于2004年10月4日。

其将[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81)以类BSD许可证的形式发布，因它的稳定性、丰富的功能集、示例配置文件和低系统资源的消耗而[闻名](https://baike.baidu.com/item/%E9%97%BB%E5%90%8D" \t "_blank)。2011年6月1日，nginx 1.0.4发布。

Nginx是一款[轻量级](https://baike.baidu.com/item/%E8%BD%BB%E9%87%8F%E7%BA%A7/10002835)的[Web](https://baike.baidu.com/item/Web/150564) 服务器/[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86)服务器及[电子邮件](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E9%82%AE%E4%BB%B6)（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。其特点是占有内存少，[并发](https://baike.baidu.com/item/%E5%B9%B6%E5%8F%91" \t "_blank)能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、[京东](https://baike.baidu.com/item/%E4%BA%AC%E4%B8%9C" \t "_blank)、[新浪](https://baike.baidu.com/item/%E6%96%B0%E6%B5%AA)、[网易](https://baike.baidu.com/item/%E7%BD%91%E6%98%93)、[腾讯](https://baike.baidu.com/item/%E8%85%BE%E8%AE%AF)、[淘宝](https://baike.baidu.com/item/%E6%B7%98%E5%AE%9D)等。

## 优点

[编辑](javascript:;)

Nginx 可以在大多数 [Unix](https://baike.baidu.com/item/Unix" \t "_blank)Linux OS 上编译运行，并有 [Windows](https://baike.baidu.com/item/Windows) 移植版。 Nginx 的1.4.0稳定版已经于2013年4月24日发布，一般情况下，对于新建站点，建议使用最新稳定版作为生产版本，已有站点的升级急迫性不高。

[](https://baike.baidu.com/pic/nginx/3817705/0/2fdda3cc7cd98d10d7efa38e2b3fb80e7bec9052?fr=lemma&ct=single)创始人伊戈尔·赛索耶夫

Nginx 的[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81" \t "_blank)使用 2-clause BSD-like license。

Nginx 是一个很强大的高性能[Web](https://baike.baidu.com/item/Web/150564)和[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86" \t "_blank)服务器，它具有很多非常优越的特性：

在连接高并发的情况下，Nginx是[Apache](https://baike.baidu.com/item/Apache/6265" \t "_blank)服务器不错的替代品：Nginx在美国是做虚拟主机生意的老板们经常选择的软件平台之一。能够支持高达 50,000 个并发连接数的响应，感谢Nginx为我们选择了 epoll and kqueue作为开发模型。

### 服务器

Nginx作为[负载均衡服务器](https://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "_blank)：Nginx 既可以在内部直接支持 Rails 和 PHP 程序对外进行服务，也可以支持作为 HTTP[代理服务器](https://baike.baidu.com/item/%E4%BB%A3%E7%90%86%E6%9C%8D%E5%8A%A1%E5%99%A8)对外进行服务。Nginx采用C进行编写，不论是系统资源开销还是CPU使用效率都比 Perlbal 要好很多。

处理静态文件，索引文件以及自动索引;打开文件描述符缓冲。

无缓存的反向代理加速，简单的负载均衡和容错。

FastCGI，简单的负载均衡和容错。

模块化的结构。包括 gzipping, byte ranges, chunked responses,以及 SSI-filter 等 filter。如果由 FastCG或其它代理服务器处理单页中存在的多个 SSI，则这项处理可以并行运行，而不需要相互等待。

支持 SSL 和 TLSSNI。[1]

### 代码

Nginx代码完全用[C语言](https://baike.baidu.com/item/C%E8%AF%AD%E8%A8%80" \t "_blank)从头写成，已经移植到许多体系结构和[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F/192)，包括：[Linux](https://baike.baidu.com/item/Linux)、[FreeBSD](https://baike.baidu.com/item/FreeBSD" \t "_blank)、[Solaris](https://baike.baidu.com/item/Solaris/3517" \t "_blank)、[Mac OS X](https://baike.baidu.com/item/Mac%20OS%20X" \t "_blank)、[AIX](https://baike.baidu.com/item/AIX" \t "_blank)以及[Microsoft Windows](https://baike.baidu.com/item/Microsoft%20Windows" \t "_blank)。Nginx有自己的函数库，并且除了[zlib](https://baike.baidu.com/item/zlib" \t "_blank)、[PCRE](https://baike.baidu.com/item/PCRE" \t "_blank)和[OpenSSL](https://baike.baidu.com/item/OpenSSL" \t "_blank)之外，标准模块只使用系统C[库函数](https://baike.baidu.com/item/%E5%BA%93%E5%87%BD%E6%95%B0)。而且，如果不需要或者考虑到潜在的授权冲突，可以不使用这些第三方库。

### 代理服务器

作为邮件[代理服务器](https://baike.baidu.com/item/%E4%BB%A3%E7%90%86%E6%9C%8D%E5%8A%A1%E5%99%A8)：Nginx 同时也是一个非常优秀的邮件[代理服务器](https://baike.baidu.com/item/%E4%BB%A3%E7%90%86%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "_blank)（最早开发这个产品的目的之一也是作为邮件代理服务器），Last.fm 描述了成功并且美妙的使用经验。

Nginx 是一个安装非常的简单、配置文件非常简洁（还能够支持perl语法）、[Bug](https://baike.baidu.com/item/Bug" \t "_blank)非常少的服务器。Nginx 启动特别容易，并且几乎可以做到7\*24不间断运行，即使运行数个月也不需要重新启动。你还能够不间断服务的情况下进行软件版本的[升级](https://baike.baidu.com/item/%E5%8D%87%E7%BA%A7/8358284)。

## 功能

[编辑](javascript:;)

支持操作系统

* FreeBSD 3— 10 / i386; FreeBSD 5— 10 / amd64;
* Linux 2.2— 4 / i386; Linux 2.6— 4 / amd64; Linux 3— 4 / armv6l, armv7l, aarch64;
* Solaris 9 / i386, sun4u; Solaris 10 / i386, amd64, sun4v;
* AIX 7.1 / powerpc;
* HP-UX 11.31 / ia64;
* Mac OS X / ppc, i386;
* Windows XP, Windows Server 2003.

结构与扩展

一个主进程和多个工作进程。工作进程是[单线程](https://baike.baidu.com/item/%E5%8D%95%E7%BA%BF%E7%A8%8B)的，且不需要特殊授权即可运行；

kqueue (FreeBSD 4.1+),epoll (Linux 2.6+),rt signals (Linux 2.2.19+),/dev/poll (Solaris 7 11/99+),select，以及 poll 支持；

kqueue支持的不同功能包括 EV\_CLEAR,EV\_DISABLE （临时禁止事件）， NOTE\_LOWAT,EV\_EOF，有效数据的数目，[错误代码](https://baike.baidu.com/item/%E9%94%99%E8%AF%AF%E4%BB%A3%E7%A0%81" \t "_blank)；

sendfile (FreeBSD 3.1+),sendfile (Linux 2.2+),sendfile64 (Linux 2.4.21+），和 sendfilev (Solaris 8 7/01+) 支持；

输入过滤 (FreeBSD 4.1+) 以及 TCP\_DEFER\_ACCEPT (Linux 2.4+) 支持；

10,000 非活动的 HTTP keep-alive 连接仅需要 2.5M内存。

最小化的数据拷贝操作；

其他HTTP功能；

基于IP 和名称的[虚拟主机](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E4%B8%BB%E6%9C%BA" \t "_blank)服务；

Memcached 的 GET 接口；

支持 keep-alive 和管道连接；

灵活简单的配置；

重新配置和在线升级而无须中断客户的工作进程；

可定制的访问日志，日志写入缓存，以及快捷的日志回卷；

4xx-5xx[错误代码](https://baike.baidu.com/item/%E9%94%99%E8%AF%AF%E4%BB%A3%E7%A0%81)重定向；

基于 PCRE 的 rewrite 重写[模块](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97" \t "_blank)；

基于客户端IP 地址和 HTTP 基本认证的访问控制；

PUT,DELETE，和 MKCOL 方法；

支持 FLV （Flash 视频）；

带宽限制。

实验特性

内嵌的 [perl](https://baike.baidu.com/item/perl" \t "_blank)；

通过 aio\_read()/aio\_write() 的[套接字](https://baike.baidu.com/item/%E5%A5%97%E6%8E%A5%E5%AD%97" \t "_blank)工作的实验[模块](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97)，仅在 FreeBSD 下；

对[线程](https://baike.baidu.com/item/%E7%BA%BF%E7%A8%8B)的实验化支持，FreeBSD 4.x 的实现基于 rfork()；

Nginx 主要的英语站点是 http://sysoev. ru/en/；

英语文档草稿由 Aleksandar Lazic 完成。

HTTP基础功能

处理静态文件，[索引文件](https://baike.baidu.com/item/%E7%B4%A2%E5%BC%95%E6%96%87%E4%BB%B6)以及自动索引；

[反向代理](https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BB%A3%E7%90%86)加速（无[缓存](https://baike.baidu.com/item/%E7%BC%93%E5%AD%98" \t "_blank)），简单的负载均衡和[容错](https://baike.baidu.com/item/%E5%AE%B9%E9%94%99" \t "_blank)；

FastCGI，简单的负载均衡和容错；

[模块](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97)化的结构。过滤器包括gzipping,byte ranges,chunked responses，以及 SSI-filter。在SSI过滤器中，到同一个 proxy 或者 FastCGI 的多个子请求并发处理；

SSL 和 TLS SNI 支持；

IMAP/POP3代理服务功能：

使用外部 [HTTP](https://baike.baidu.com/item/HTTP) 认证服务器重定向用户到 IMAP/POP3 后端；

使用外部 HTTP 认证服务器认证用户后连接重定向到内部的 SMTP 后端；

其他HTTP功能

基于名称和基于IP的虚拟服务器；

Keep-alive and pipelined connections support；保持活动和支持管线连接；

Flexible configuration；灵活的配置；

Reconfiguration and online upgrade without interruption of the client processing；重载配置，无间断程序升级；

Access log formats,bufferred log writing,and quick log rotation；访问日志格式，bufferred日志写，快速登录旋转；

3xx-5xx error codes redirection; 3xx的- 5xx[错误代码](https://baike.baidu.com/item/%E9%94%99%E8%AF%AF%E4%BB%A3%E7%A0%81)重定向；

The rewrite module；重写[模块](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97" \t "_blank);

Access control based on client IP address and HTTP Basic authentication；基于客户端IP地址访问控制和HTTP基本认证；

The PUT,DELETE,MKCOL,COPY and MOVE methods; 提交，删除，MKCOL，复制和移动方法；

FLV streaming;FLV[视频流](https://baike.baidu.com/item/%E8%A7%86%E9%A2%91%E6%B5%81" \t "_blank);

Speed limitation；速度限制；

Limitation of simultaneous connections or requests from one address.限制同个IP地址请求数量。

Embedded perl.嵌入式的[Perl](https://baike.baidu.com/item/Perl" \t "_blank)。

**邮件代理服务器功能**

用户重定向到IMAP/POP3后端使用外部HTTP认证服务器；

User authentication using an external HTTP authentication server and connection redirection to internal SMTP backend；用户[身份验证](https://baike.baidu.com/item/%E8%BA%AB%E4%BB%BD%E9%AA%8C%E8%AF%81" \t "_blank)使用外部HTTP认证服务器和连接重定向到内部的SMTP后端；

Authentication methods：验证方法：

POP3: USER/PASS,APOP,AUTH LOGIN/PLAIN/CRAM-MD5；的POP3：用户名/密码，的APOP，AUTH的LOGIN/PLAIN/CRAM-MD5;

IMAP: LOGIN,AUTH LOGIN/PLAIN/CRAM-MD5; IMAP的：登录，AUTH的LOGIN/PLAIN/CRAM-MD5;

SMTP: AUTH LOGIN/PLAIN/CRAM-MD5；的SMTP：AUTH的LOGIN/PLAIN/CRAM-MD5;

SSL support; SSL支持；

STARTTLS and STLS support. STARTTLS的和补充的支持。

**认证方法**

POP3: POP3 USER/PASS,APOP,AUTH LOGIN PLAIN CRAM-MD5;

IMAP: IMAP LOGIN;

SMTP: AUTH LOGIN PLAIN CRAM-MD5;

SSL 支持；

在 IMAP 和 POP3 模式下的 STARTTLS 和 STLS 支持。

## 安装

[编辑](javascript:;)

### 模块依赖性

gzip[模块](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97" \t "_blank)需要 [zlib](https://baike.baidu.com/item/zlib" \t "_blank) 库

rewrite模块需要 [pcre](https://baike.baidu.com/item/pcre" \t "_blank) 库

ssl 功能需要[openssl](https://baike.baidu.com/item/openssl" \t "_blank)库

### 编译安装包

Nginx在一些Linux发行版和BSD的各个变种版本的安装包仓库中都会有，通过各个系统自带的软件包管理方法即可安装。需要注意的是，很多预先编译好的安装包都比较陈旧，大多数情况下还是推荐直接从[源码](https://baike.baidu.com/item/%E6%BA%90%E7%A0%81" \t "_blank)编译。

### 源代码下载

特定平台的安装和记录

Nginx 在 Slackware 上的编译安装脚本

Nginx 在ubuntu和debian上的安装及脚本,debian针对nginx包的官方下载站点

使用[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81)进行构建

Nginx 使用 Unix 下常用的 './configure && make && make install' 过程来编译安装。

configure脚本确定系统所具有一些特性，特别是 nginx 用来处理连接的方法。然后，它创建 Makefile 文件。

configure 支持下面的选项：

--prefix=<path> - Nginx安装路径。如果没有指定，默认为 /usr/local/nginx。

--sbin-path=<path> - Nginx[可执行文件](https://baike.baidu.com/item/%E5%8F%AF%E6%89%A7%E8%A1%8C%E6%96%87%E4%BB%B6" \t "_blank)安装路径。只能安装时指定，如果没有指定，默认为<prefix>/sbin/nginx。

--conf-path=<path> - 在没有给定-c选项下默认的nginx.conf的路径。如果没有指定，默认为<prefix>/conf/nginx.conf。

--pid-path=<path> - 在nginx.conf中没有指定pid指令的情况下，默认的nginx.pid的路径。如果没有指定，默认为 <prefix>/logs/nginx.pid。

--lock-path=<path> - nginx.lock文件的路径。

--error-log-path=<path> - 在nginx.conf中没有指定error\_log指令的情况下，默认的[错误日志](https://baike.baidu.com/item/%E9%94%99%E8%AF%AF%E6%97%A5%E5%BF%97" \t "_blank)的路径。如果没有指定，默认为 <prefix>/logs/error.log。

--http-log-path=<path> - 在nginx.conf中没有指定access\_log指令的情况下，默认的访问日志的路径。如果没有指定，默认为 <prefix>/logs/access.log。

--user=<user> - 在nginx.conf中没有指定user指令的情况下，默认的nginx使用的用户。如果没有指定，默认为 nobody。

--group=<group> - 在nginx.conf中没有指定group指令的情况下，默认的nginx使用的组。如果没有指定，默认为 nobody。

--builddir=DIR - 指定编译的目录

--with-rtsig\_module - 启用 rtsig[模块](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97" \t "_blank)

--with-select\_module --without-select\_module - Whether or not to enable the select module. This module is enabled by default if a more suitable method such as kqueue,epoll,rtsig or /dev/poll is not discovered by configure.

//允许或不允许开启SELECT模式，如果 configure 没有找到更合适的模式，比如：kqueue(sun os),epoll (linux kenel 2.6+),rtsig（实时信号）或者/dev/poll（一种类似select的模式，底层实现与SELECT基本相 同，都是采用轮训方法） SELECT模式将是默认安装模式

--with-poll\_module --without-poll\_module - Whether or not to enable the poll module. This module is enabled by default if a more suitable method such as kqueue,epoll,rtsig or /dev/poll is not discovered by configure.

--with-http\_ssl\_module - Enable ngx\_http\_ssl\_module. Enables SSL support and the ability to handle HTTPS requests. Requires OpenSSL. On Debian,this is libssl-dev.

//开启HTTP SSL[模块](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97)，使NGINX可以支持HTTPS请求。这个模块需要已经安装了OPENSSL，在DEBIAN上是libssl

--with-http\_realip\_module - 启用 ngx\_http\_realip\_module

--with-http\_addition\_module - 启用 ngx\_http\_addition\_module

--with-http\_sub\_module - 启用 ngx\_http\_sub\_module

--with-http\_dav\_module - 启用 ngx\_http\_dav\_module

--with-http\_flv\_module - 启用 ngx\_http\_flv\_module

--with-http\_stub\_status\_module - 启用 "server status" 页

--without-http\_charset\_module - 禁用 ngx\_http\_charset\_module

--without-http\_gzip\_module - 禁用 ngx\_http\_gzip\_module. 如果启用，需要 zlib。

--without-http\_ssi\_module - 禁用 ngx\_http\_ssi\_module

--without-http\_userid\_module - 禁用 ngx\_http\_userid\_module

--without-http\_access\_module - 禁用 ngx\_http\_access\_module

--without-http\_auth\_basic\_module - 禁用 ngx\_http\_auth\_basic\_module

--without-http\_autoindex\_module - 禁用 ngx\_http\_autoindex\_module

--without-http\_geo\_module - 禁用 ngx\_http\_geo\_module

--without-http\_map\_module - 禁用 ngx\_http\_map\_module

--without-http\_referer\_module - 禁用 ngx\_http\_referer\_module

--without-http\_rewrite\_module - 禁用 ngx\_http\_rewrite\_module. 如果启用需要 PCRE。

--without-http\_proxy\_module - 禁用 ngx\_http\_proxy\_module

--without-http\_fastcgi\_module - 禁用 ngx\_http\_fastcgi\_module

--without-http\_memcached\_module - 禁用 ngx\_http\_memcached\_module

--without-http\_limit\_zone\_module - 禁用 ngx\_http\_limit\_zone\_module

--without-http\_empty\_gif\_module - 禁用 ngx\_http\_empty\_gif\_module

--without-http\_browser\_module - 禁用 ngx\_http\_browser\_module

--without-http\_upstream\_ip\_hash\_module - 禁用 ngx\_http\_upstream\_ip\_hash\_module

--with-http\_perl\_module - 启用 ngx\_http\_perl\_module

--with-perl\_modules\_path=PATH - 指定 perl[模块](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97" \t "_blank)的路径

--with-perl=PATH - 指定 perl 执行文件的路径

--http-log-path=PATH - Set path to the http access log

--http-client-body-temp-path=PATH - Set path to the http client request body temporary files

--http-proxy-temp-path=PATH - Set path to the http proxy temporary files

--http-fastcgi-temp-path=PATH - Set path to the http fastcgi temporary files

--without-http - 禁用 HTTP server

--with-mail - 启用 IMAP4/POP3/SMTP 代理[模块](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97" \t "_blank)

--with-mail\_ssl\_module - 启用 ngx\_mail\_ssl\_module

--with-cc=PATH - 指定 C[编译器](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E5%99%A8)的路径

--with-cpp=PATH - 指定 C[预处理器](https://baike.baidu.com/item/%E9%A2%84%E5%A4%84%E7%90%86%E5%99%A8)的路径

--with-cc-opt=OPTIONS - Additional parameters which will be added to the variable CFLAGS. With the use of the system library PCRE in FreeBSD,it is necessary to indicate --with-cc-opt="-I /usr/local/include". If we are using select() and it is necessary to increase the number of file descriptors,then this also can be assigned here: --with-cc-opt="-D FD\_SETSIZE=2048".

--with-ld-opt=OPTIONS - Additional parameters passed to the linker. With the use of the system library PCRE in FreeBSD,it is necessary to indicate --with-ld-opt="-L /usr/local/lib".

--with-cpu-opt=CPU - 为特定的 CPU 编译，有效的值包括：pentium,pentiumpro,pentium3,pentium4,athlon,opteron,amd64,sparc32,sparc64,ppc64

--without-pcre - 禁止 PCRE 库的使用。同时也会禁止 HTTP rewrite 模块。在 "location" 配置指令中的[正则表达式](https://baike.baidu.com/item/%E6%AD%A3%E5%88%99%E8%A1%A8%E8%BE%BE%E5%BC%8F" \t "_blank)也需要 PCRE。

--with-pcre=DIR - 指定 PCRE 库的[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81" \t "_blank)的路径。

--with-pcre-opt=OPTIONS - Set additional options for PCRE building.

--with-md5=DIR - Set path to md5 library sources.

--with-md5-opt=OPTIONS - Set additional options for md5 building.

--with-md5-asm - Use md5 assembler sources.

--with-sha1=DIR - Set path to sha1 library sources.

--with-sha1-opt=OPTIONS - Set additional options for sha1 building.

--with-sha1-asm - Use sha1 assembler sources.

--with-zlib=DIR - Set path to zlib library sources.

--with-zlib-opt=OPTIONS - Set additional options for zlib building.

--with-zlib-asm=CPU - Use zlib assembler sources optimized for specified CPU,valid values are: pentium,pentiumpro

--with-openssl=DIR - Set path to OpenSSL library sources

--with-openssl-opt=OPTIONS - Set additional options for OpenSSL building

--with-debug - 启用调试日志

--add-module=PATH - Add in a third-party module found in directory PATH

在不同版本间，选项可能会有些许变化，请总是使用 ./configure --help 命令来检查一下当前的选项列表。

## 使用技巧

[编辑](javascript:;)

**对于chroot的支持是否在计划之中**

在什么情况下使用Nginx比使用squid要好? 反之亦然。

大体上来说nginx主要用于反向加速代理而不是像squid那样作为常规[代理服务器](https://baike.baidu.com/item/%E4%BB%A3%E7%90%86%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "_blank)。Nginx的最大优势在于高负载情况下内存和CPU的低消耗。我不认为squid能给你带来比nginx更好的性能。

依照 [NginxImapProxyExample] 开始你的配置. 关于不同配置参数的具体信息，请查看 [NginxMailCoreModule] 页。

示例1: 用运行于apache上的php脚本做后端验证

示例2: 使用运行于同一个服务器的 nginx-embedded-perl[模块](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97" \t "_blank)作为 imap/pop代理和认证后端

**某些功能不工作**

（URL重写，代理，路径，...)

例如：如[URL](https://baike.baidu.com/item/URL" \t "_blank)重写（rewrite）不工作了或者是unix的路径（/$PATH）的问题云云...

请仔细阅读 [NginxDebugging] 并且 逐行 查看[错误日志](https://baike.baidu.com/item/%E9%94%99%E8%AF%AF%E6%97%A5%E5%BF%97" \t "_blank)。

如果你没找到错误 打起精神 试着到IRC或邮件列表里说明一下你碰到的问题。

有没有其它类似的Web服务器

Cherokee

Lighttpd (Lighty)

thttpd

关于各自的优缺点请使用自己喜欢的搜索引擎查找

**让Nginx成为以postfix做为后端的SMTP代理**

Nginx使用什么算法来实现[负载均衡](https://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1" \t "_blank)它能实现基于连接数的负载均衡吗?

Nginx使用简单的轮巡算法，所以无法做基本链接计数的负载均衡。这个可能会在将来的版本中有所改变。

我能关闭从[代理服务器](https://baike.baidu.com/item/%E4%BB%A3%E7%90%86%E6%9C%8D%E5%8A%A1%E5%99%A8)到后端服务器的[缓存](https://baike.baidu.com/item/%E7%BC%93%E5%AD%98)吗或者使用上传进度特性?

## 反向代理实践

[编辑](javascript:;)

nginx + substitutions 安装  
　　nginx 自带一个Substitution模块，但该模块只能写一行，所以我们改用 substitutions  
　　下面是安装一些预备软件

1. yum -y --noplugins install wget zip
2. yum -y --noplugins install unzip
3. yum -y --noplugins install gcc
4. yum -y --noplugins install make
5. yum -y --noplugins install pcre-devel
6. yum -y --noplugins install openssl-devel

**编译软件**

1. tar zxf nginx-1.0.8.tar.gz
2. cd nginx-1.0.8
3. ./configure ./configure --add-module=path/substitutions4nginx-read-only //注意这里的path是相对应的真实路径
4. make
5. make install
6. 配置 nginx.conf  
   　　此时，nginx应该安装在于 /usr/local/nginx 下面
7. server\_name servername;
8. location / {
9. subs\_filter ca-pub-\*\*\*\*\*\*\*\*\*\* ca-pub-\*\*\*\*\*\*\*\*\*\*; //把google ad 的用户号 ca-pub-9805743306566114 改成你自己的，比如 ca-pub-\*\*\*\*\*\*\*\*\*\*
10. subs\_filter \*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*; //把google ad 的广告号 \*\*\*\*\*\*\*\*\*\*改成你自己的，比如 \*\*\*\*\*\*\*\*\*\*，你懂的！：D
11. proxy\_pass 这里是填写需要代理的网址; //反向代理站。
12. index index.html index.htm;
13. }

记住subs\_filter 命令的格式即可随意发挥想象力替换你想替换的广告、超链接，等等。

**基于太多人询问下面的问题：**

问：我能为了得到上传进度而关闭代理的缓存吗？

问：使用nginx我怎么才能给用户显示上传进度？

答：到目前为止 （2007-Apr-26） 还没有办法关闭掉后端服务器的缓存。

## 其他信息

[编辑](javascript:;)

1.Nginx Rewrite 基本标记(flags)复制内容到剪贴板代码:last – 基本上都用这个Flag[2]  。

break – 中止Rewrite，不在继续匹配

redirect – 返回临时重定向的HTTP状态302

permanent – 返回永久重定向的HTTP状态301

2. 正则表达式匹配，其中：代码:

CODE:

~ 为区分大小写匹配

~\* 为不区分大小写匹配

!~和!~\*分别为区分大小写不匹配及不区分大小写不匹配

3. 文件及目录匹配，其中：代码:

CODE:

-f和!-f用来判断是否存在文件

-d和!-d用来判断是否存在目录

-e和!-e用来判断是否存在文件或目录

-x和!-x用来判断文件是否可执行

4.Nginx 的一些可用的全局变量，可用做条件判断：

代码:

CODE:

$args

$content\_length

$content\_type

$document\_root

$document\_uri

$host

$http\_user\_agent

$http\_cookie

$limit\_rate

$request\_body\_file

$request\_method

$remote\_addr

$remote\_port

$remote\_user

$request\_filename

$request\_uri

$query\_string

$scheme

$server\_protocol

$server\_addr

$server\_name

$server\_port

$uri

## 配置

[编辑](javascript:;)

cls

@ECHO OFF

SET NGINX\_PATH=D:

SET NGINX\_DIR=D:\nginx-1.3.6\

color 0a

TITLE Nginx 管理程序 Power By ChenJianxiang

GOTO MENU

:MENU

CLS

ECHO.

ECHO. \* \* \* \* Nginx 管理程序 Power By ChenJianxiang \* \* \*

ECHO. \* \*

ECHO. \* 1 启动Nginx \*

ECHO. \* \*

ECHO. \* 2 关闭Nginx \*

ECHO. \* \*

ECHO. \* 3 重启Nginx \*

ECHO. \* \*

ECHO. \* 4 退 出 \*

ECHO. \* \*

ECHO. \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

ECHO.

ECHO.请输入选择项目的序号：

set /p ID=

IF "%id%"=="1" GOTO cmd1

IF "%id%"=="2" GOTO cmd2

IF "%id%"=="3" GOTO cmd3

IF "%id%"=="4" EXIT

PAUSE

:cmd1

ECHO.

ECHO.启动Nginx......

IF NOT EXIST %NGINX\_DIR%nginx.exe ECHO %NGINX\_DIR%nginx.exe不存在

%NGINX\_PATH%

cd %NGINX\_DIR%

IF EXIST %NGINX\_DIR%nginx.exe start %NGINX\_DIR%nginx.exe

ECHO.OK

PAUSE

GOTO MENU

:cmd2

ECHO.

ECHO.关闭Nginx......

taskkill /F /IM nginx.exe > nul

ECHO.OK

PAUSE

GOTO MENU

:cmd3

ECHO.

ECHO.关闭Nginx......

taskkill /F /IM nginx.exe > nul

ECHO.OK

GOTO cmd1

GOTO MENU

## 版本发布

[编辑](javascript:;)

2012年08月22日，Nginx 1.3.5 开发版发布。

2012年09月12日，Nginx 1.3.6 开发版发布。

2012年12月11日，Nginx 1.2.6 稳定版发布

2013年02月12日，Nginx 1.2.7 稳定版发布

2013年02月19日，Nginx 1.3.13 开发版发布

2013年04月03日，Nginx 1.2.8 稳定版发布。

2013年04月16日，Nginx 1.3.16 开发者版本发布。

2013年04月24日，Nginx 1.4.0 稳定版发布。

2013年05月07日，Nginx 1.4.1 稳定版发布。

2013年05月07日，Nginx 1.5.0 开发版本发布。

2013年06月04日，Nginx 1.5.1 主版本发布。

2013年07月02日，Nginx 1.5.2 开发版发布。

2013年07月17日，Nginx 1.4.2 稳定版发布。

2013年07月30日，Nginx 1.5.3 开发版本发布。

2013年08月27日，Nginx 1.5.4 开发版本发布。

2013年09月17日，Nginx 1.5.5 开发版本发布。[2-3]

2015年10月27日，Nginx 1.9.6 主线版本发布。

2016年02月09日，Nginx 1.9.11 开发版本发布。[4]

2016年07月26日，Nginx1.11.3 主线版本。

2016年10月11日，Nginx1.11.5 主线版本发布。

2016年10月18日，Nginx1.10.2 稳定版本发布。

2016年11月15日，Nginx1.11.6 主线版本发布。

2016年12月13日，Nginx1.11.7主线版本发布。

2016年12月27日，Nginx1.11.8主线版本发布。

2017年01月24日，Nginx1.11.9主线版本已经发布。

2017年06月27日，nginx-1.13.2主线版本发布

2017年07月11日，nginx-1.13.3主线版本发布[5]

2017年08月08日，nginx-1.13.4主线版本发布[6]

2017年09月05日，nginx-1.13.5主线版本发布[7]

2017年10月10日，nginx-1.13.6主线版本发布[8]

2017年11月21日，nginx-1.13.7主线版本发布[9]

2017年12月26日，nginx-1.13.8主线版本发布[9]

2018年02月20日，nginx-1.13.9主线版本发布[9]

待更新。。。。。。