# nginx

# 什么是nginx?

Nginx是一款高性能的http 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器。由俄罗斯的程序设计师Igor Sysoev所开发，官方测试nginx能够支支撑5万并发链接，并且cpu、内存等资源消耗却非常低，运行非常稳定。

# 应用场景

1. http服务器。Nginx是一个http服务可以独立提供http服务。可以做网页静态服务器。
2. 虚拟主机。可以实现在一台服务器虚拟出多个网站。例如个人网站使用的虚拟主机。
3. 反向代理，负载均衡。当网站的访问量达到一定程度后，单台服务器不能满足用户的请求时，需要用多台服务器集群可以使用nginx做反向代理。并且多台服务器可以平均分担负载，不会因为某台服务器负载高宕机而某台服务器闲置的情况。

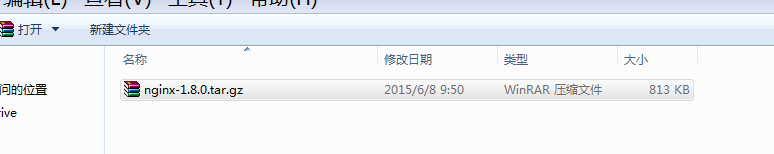
# nginx安装

下载nginx：

官方网站：

<http://nginx.org/>

使用的版本是1.8.0版本。



## 要求的安装环境

1. 需要安装gcc的环境。yum install gcc-c++
2. 第三方的开发包。

* PCRE

PCRE(Perl Compatible Regular Expressions)是一个Perl库，包括 perl 兼容的正则表达式库。nginx的http模块使用pcre来解析正则表达式，所以需要在linux上安装pcre库。

yum install -y pcre pcre-devel

注：pcre-devel是使用pcre开发的一个二次开发库。nginx也需要此库。

* zlib

zlib库提供了很多种压缩和解压缩的方式，nginx使用zlib对http包的内容进行gzip，所以需要在linux上安装zlib库。

* yum install -y zlib zlib-devel

OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL协议，并提供丰富的应用程序供测试或其它目的使用。

nginx不仅支持http协议，还支持https（即在ssl协议上传输http），所以需要在linux安装openssl库。

yum install -y openssl openssl-devel

如果linux上没有make命令请按装make

**yum -y install gcc automake autoconf libtool make**

## 安装步骤

第一步：把nginx的源码包上传到linux系统

第二步：解压缩

[root@localhost ~]# tar zxf nginx-1.8.0.tar.gz

第三步：使用configure命令创建一makeFile文件。

./configure \

--prefix=/usr/local/nginx \

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

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

--error-log-path=/var/log/nginx/error.log \

--http-log-path=/var/log/nginx/access.log \

--with-http\_gzip\_static\_module \

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

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

--http-fastcgi-temp-path=/var/temp/nginx/fastcgi \

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

--http-scgi-temp-path=/var/temp/nginx/scgi

**注意：启动nginx之前，上边将临时文件目录指定为/var/temp/nginx，需要在/var下创建temp及nginx目录**

[root@localhost sbin]# mkdir -p /var/temp/nginx/client

第四步：make

第五步：make install

## 启动nginx

进入sbin目录

[root@localhost sbin]# ./nginx

关闭nginx：

[root@localhost sbin]# ./nginx -s stop

推荐使用：

[root@localhost sbin]# ./nginx -s quit

重启nginx：

1. 先关闭后启动。
2. 刷新配置文件：

[root@localhost sbin]# ./nginx -s reload

## 访问nginx

默认是80端口。

注意：是否关闭防火墙。

<http://192.168.31.151>

# 配置虚拟主机

就是在一台服务器启动多个网站。

如何区分不同的网站：

1. 域名不同
2. 端口不同

## 通过端口区分不同虚拟机

Nginx的配置文件：

/usr/local/nginx/conf/nginx.conf

4.2通过域名区分虚拟主机

# 反向代理

## Nginx实现反向代理

第一步：安装两个tomcat，分别运行在8080和8081端口。

第二步：启动两个tomcat。

第三步：反向代理服务器的配置

|  |
| --- |
| upstream tomcat1 {  server 192.168.18.130:8080;  }  server {  listen 80;  server\_name www.text1.com;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  proxy\_pass http://tomcat1;  index index.html index.htm;  }  }  upstream tomcat2 {  server 192.168.18.130:8081;  }  server {  listen 80;  server\_name www.text2.com;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  proxy\_pass http://tomcat2;  index index.html index.htm;  }  } |

第四步：nginx重新加载配置文件

第五步：配置域名

在hosts文件中添加域名和ip的映射关系

192.168.18.130 www.text1.com

192.168.18.130 [www.text2.com](http://www.text2.com)

# 负载均衡

如果一个服务由多个服务器提供，需要把负载分配到不同的服务器处理，需要负载均衡。

upstream tomcat2 {

server 192.168.18.130:8081;

server 192.168.18.130:8082;

}

可以根据服务器的实际情况调整服务器权重。权重越高分配的请求越多，权重越低，请求越少。默认是都是1

|  |
| --- |
| upstream tomcat2 {  server 192.168.18.130:8081;  server 192.168.25.130:8082 weight=2  } |