# 一、Nginx基本概念

## 1.2、为什么选择Nginx

Nginx 是一个高性能的 Web 和反向代理服务器, 它具有有很多非常优越的特性:

作为 Web 服务器：相比 Apache，Nginx 使用更少的资源，支持更多的并发连接，体现更高的效率，这点使 Nginx 尤其受到虚拟主机提供商的欢迎。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.

作为负载均衡服务器：Nginx 既可以在内部直接支持 Rails 和 PHP，也可以支持作为 HTTP代理服务器 对外进行服务。Nginx 用 C 编写, 不论是系统资源开销还是 CPU 使用效率都比 Perlbal 要好的多。

作为邮件代理服务器: Nginx 同时也是一个非常优秀的邮件代理服务器（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 安装非常的简单，配置文件 非常简洁（还能够支持perl语法），Bugs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。你还能够在不间断服务的情况下进行软件版本的升级。

**Nginx因稳定性、丰富的功能集、示例配置文件和低系统资源的消耗而闻名了**

# 二、Nginx安装

## 2.1、Windows下Nginx的安装与配置

（1）、首先去官网下载 nginx1.0.11的Windows版本，官网下载：http://nginx.org/download/nginx-1.0.11.zip

下载到软件包后，解压 nginx-nginx1.0.11.zip 包到你喜欢的根目录，并将目录名改为nginx。

然后，执行下列操作：

cd nginx

start nginx

这样，nginx 服务就启动了。打开任务管理器，查看 nginx.exe 进程，有二个进程会显示，占用系统资源，那是相当的少。

常用命令：

nginx -s stop // 停止nginx

nginx -s reload // 重新加载配置文件

nginx -s quit // 退出nginx

## 2.2、Linux下Nginx+Tomcat整合的安装与配置

### 2.2.1、安装Tomcat和JDK

(1)、上传apache-tomcat-6.0.18.tar.gz和jdk-6u12-linux-i586.bin至/usr/local

(2)、执行如下命令安装tomcat：

#cd /usr/local

#tar zxvf apache-tomcat-6.0.18.tar.gz

解压完成后将apache-tomcat-6.0.18重命名为tomcat

(3)、执行如下命令安装JDK:

#./jdk-6u12-linux-i586.bin

(4)、配置环境变量：

编辑/etc下的profile文件，加上如下内容：

JAVA\_HOME="/usr/local/jdk1.6.0\_12"

CLASS\_PATH="$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib"

PATH=".:$PATH:$JAVA\_HOME/bin "

CATALINA\_HOME="/usr/local/tomcat"

export JAVA\_HOME CATALINA\_HOME

(5)、启动tomcat并输入http://localhost:8080，如果看到猫的页面即tomcat和jdk安装成功

(6)、新建文件目录/home/www为网站存放目录，设置server.xml文件，在Host name=”localhost”处将appBase=的指向路径改为/home/www/web

(7)、创建index.jsp至/home/www/web/ROOT，内容为：“My web!”

### 2.2.2、安装Nginx

#### (1)、上传nginx-0.7.63.tar.gz至/usr/local

#### (2)、执行如下命令解压nginx：

#cd /usr/local

#tar zxvf nginx-0.7.63.tar.gz

#### (3)、编译安装nginx

#cd nginx-0.7.63

#./configure --with-http\_stub\_status\_module --with-http\_ssl\_module #启动server状态页和https模块

执行完后会提示一个错误，说缺少PCRE library 这个是HTTP Rewrite 模块，也即是url静态化的包

可上传pcre-7.9.tar.gz，输入如下命令安装：

#tar zxvf pcre-7.9.tar.gz

#cd pcre-7.9

#./configure

#make

#make install

安装pcre成功后，继续安装nginx

#cd nginx-0.7.63

#./configure

#make

#make install

#### (4)、nginx安装成功后的安装目录为/usr/local/nginx

在conf文件夹中新建proxy.conf，用于配置一些代理参数，内容如下：

#!nginx (-)

# proxy.conf

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr; #获取真实ip

#proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for; #获取代理者的真实ip

client\_max\_body\_size 10m;

client\_body\_buffer\_size 128k;

proxy\_connect\_timeout 90;

proxy\_send\_timeout 90;

proxy\_read\_timeout 90;

proxy\_buffer\_size 4k;

proxy\_buffers 4 32k;

proxy\_busy\_buffers\_size 64k;

proxy\_temp\_file\_write\_size 64k;

编辑安装目录下conf文件夹中的nginx.conf，输入如下内容

#运行nginx所在的用户名和用户组

#user www www;

#启动进程数

worker\_processes 8;

#全局错误日志及PID文件

error\_log /usr/local/nginx/logs/nginx\_error.log crit;

pid /usr/local/nginx/nginx.pid;

#Specifies the value for maximum file descriptors that can be opened by this process.

worker\_rlimit\_nofile 65535;

#工作模式及连接数上限

events

{

use epoll;

worker\_connections 65535;

}

#设定http服务器，利用它的反向代理功能提供负载均衡支持

http

{

#设定mime类型

include mime.types;

default\_type application/octet-stream;

include /usr/local/nginx/conf/proxy.conf;

#charset gb2312;

#设定请求缓冲

server\_names\_hash\_bucket\_size 128;

client\_header\_buffer\_size 32k;

large\_client\_header\_buffers 4 32k;

client\_max\_body\_size 8m;

sendfile on;

tcp\_nopush on;

keepalive\_timeout 60;

tcp\_nodelay on;

# fastcgi\_connect\_timeout 300;

# fastcgi\_send\_timeout 300;

# fastcgi\_read\_timeout 300;

# fastcgi\_buffer\_size 64k;

# fastcgi\_buffers 4 64k;

# fastcgi\_busy\_buffers\_size 128k;

# fastcgi\_temp\_file\_write\_size 128k;

# gzip on;

# gzip\_min\_length 1k;

# gzip\_buffers 4 16k;

# gzip\_http\_version 1.0;

# gzip\_comp\_level 2;

# gzip\_types text/plain application/x-javascript text/css application/xml;

# gzip\_vary on;

#limit\_zone crawler $binary\_remote\_addr 10m;

###禁止通过ip访问站点

server{

server\_name \_;

return 404;

}

server

{

listen 80;

server\_name localhost;

index index.html index.htm index.jsp;#设定访问的默认首页地址

root /home/www/web/ROOT;#设定网站的资源存放路径

#limit\_conn crawler 20;

location ~ .\*.jsp$ #所有jsp的页面均交由tomcat处理

{

index index.jsp;

proxy\_pass http://localhost:8080;#转向tomcat处理

}

location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$ #设定访问静态文件直接读取不经过tomcat

{

expires 30d;

}

location ~ .\*\.(js|css)?$

{

expires 1h;

}

#定义访问日志的写入格式

log\_format access '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" $http\_x\_forwarded\_for';

access\_log /usr/local/nginx/logs/localhost.log access;#设定访问日志的存放路径

}

}

#### (5)、修改/usr/local/nginx/conf/nginx.conf配置文件后，请执行以下命令检查配置文件是否正确：

#/usr/local/nginx/sbin/nginx -t

如果屏幕显示以下两行信息，说明配置文件正确：

the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok

　　the configuration file /usr/local/nginx/conf/nginx.conf was tested successfully

如果提示unknown host，则可在服务器上执行：ping www.baidu.com如果也是同样提示unknown host则有两种可能：

a、服务器没有设置DNS服务器地址，查看/etc/resolv.conf下是否设置，若无则加上

b、防火墙拦截

#### (6)、启动nginx的命令

#/usr/local/nginx/sbin/nginx

这时，输入以下命令查看Nginx主进程号：

ps -ef | grep "nginx: master process" | grep -v "grep" | awk -F ' ' '{print $2}'

#### (7)、停止nginx的命令

#/usr/local/nginx/sbin/nginx -s stop

#### (8)、在不停止Nginx服务的情况下平滑变更Nginx配置

a、修改/usr/local/nginx/conf/nginx.conf配置文件后，请执行以下命令检查配置文件是否正确：

/usr/local/nginx/sbin/nginx -t

　　如果屏幕显示以下两行信息，说明配置文件正确：

　　the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok

　　the configuration file /usr/local/nginx/conf/nginx.conf was tested successfully

b、这时，输入以下命令查看Nginx主进程号：

ps -ef | grep "nginx: master process" | grep -v "grep" | awk -F ' ' '{print $2}'

屏幕显示的即为Nginx主进程号，例如：

　　6302

　　这时，执行以下命令即可使修改过的Nginx配置文件生效：

kill -HUP 6302

或者无需这么麻烦，找到Nginx的Pid文件：

kill -HUP `cat /usr/local/nginx/nginx.pid`

#### (9)、nginx启动好后启动tomcat，此时输入http://主机ip地址即可看到“My web!”

### 2.2.3、stub\_status查看nginx状态

#### （1）、加载http\_stub\_status模块

[root@10.10.90.97 ~]# ./configure --prefix=/usr/local/nginx --with-http\_stub\_status\_module

[root@10.10.90.97 ~]# make && make install

#### （2）、修改nginx配置文件

在server中，添加如下代码：

location /NginxStatus {

stub\_status on;

access\_log on;

auth\_basic "NginxStatus";

auth\_basic\_user\_file htpasswd;

}

[root@10.10.90.97 ~]# htpasswd -c /usr/local/nginx/conf/htpasswd nginx\_focus #连续输入两次密码

New password:

Re-type new password:

Adding password for user nginx\_focus

重新加载nginx.conf:

[root@10.10.90.97 ~]# kill -USR1 `cat /usr/local/nginx/logs/nginx.pid`

#### （3）、打开nginx的stub\_status可以通过页面链接看到如下信息:

http:/…./stub\_status

语法: stub\_status on

默认值: None

作用域: location

创建一个 location 区域启用 stub\_status

“stub status” 模块返回的状态信息跟 mathopd’s 的状态信息很相似. 返回的状态信息如下：

Active connections: 291

server accepts handled requests

16630948 16630948 31070465

Reading: 6 Writing: 179 Waiting: 106

active connections — 对后端发起的活动连接数

server accepts handled requests — nginx 总共处理了 16630948 个连接, 成功创建 16630948 次握手 (证明中间没有失败的), 总共处理了 31070465 个请求 (平均每次握手处理了 1.8个数据请求)

reading — nginx 读取到客户端的Header信息数

writing — nginx 返回给客户端的Header信息数

waiting — 开启 keep-alive 的情况下，这个值等于 active – (reading + writing)，意思就是Nginx说已经处理完正在等候下一次请求指令的驻留连接

# 中文文档地址

<http://www.nginx.cn/doc/general/overview.html>

http://www.cnblogs.com/huangjingzhou/category/318465.html