

nginx安装配置说明

一、nginx简介

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。

二、nginx安装

1、下载nginx相关组件

```
wget http://nginx.org/download/nginx-1.10.2.tar.gz
```

```
wget http://www.openssl.org/source/openssl-fips-2.0.10.tar.gz
```

```
wget http://zlib.net/zlib-1.2.11.tar.gz
```

```
wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.40.tar.gz
```

2、安装c++编译环境

```
yum install gcc-c++
```

3、安装nginx及相关组件

```
openssl安装  
tar zxvf openssl-fips-2.0.10.tar.gz  
cd pcre-8.40  
./configure && make && make install
```

```
pcre安装
tar zxvf pcre-8.40.tar.gz
cd pcre-8.40
./configure && make && make install
```

```
zlib安装
tar zxvf zlib-1.2.11.tar.gz
cd zlib-1.2.11
./configure && make && make install
```

```
nginx安装
tar zxvf nginx-1.10.2.tar.gz
cd nginx-1.10.2
./configure && make && make install
```

三、nginx启动

1、查看nginx安装位置

```
whereis nginx
nginx:/usr/local/nginx
```

2、启动nginx

```
cd /usr/local/nginx
#在nginx目录下输入启动命令
./sbin/nginx
#然后在浏览器输入localhost页面如下
```



3、nginx的基本操作

启动

```
[root@localhost ~]# /usr/local/nginx/sbin/nginx
```

停止/重启

```
[root@localhost ~]# /usr/local/nginx/sbin/nginx -s stop(quit、reload)
```

命令帮助

```
[root@localhost ~]# /usr/local/nginx/sbin/nginx -h
```

验证配置文件

```
[root@localhost ~]# /usr/local/nginx/sbin/nginx -t
```

配置文件

```
[root@localhost ~]# vim /usr/local/nginx/conf/nginx.conf
```

四、nginx配置

1、conf配置

打开nginx配置文件位于nginx目录下的conf文件夹下，配置相关内容，其内容如下：

```
http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #              '$status $body_bytes_sent "$http_referer" '
    #              '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile      on;
    #tcp_nopush   on;

    #keepalive_timeout 0;
    keepalive_timeout 65;

    #gzip on;

    server {
        listen      3001;
        server_name  isstest;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location / {
            root      html/issweb/build;
            try_files $uri /index.html;
            index      index.html index.htm;
        }

        #error_page  404              /404.html;

        # redirect server error pages to the static page /50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root      html;
        }

        # proxy the PHP scripts to Apache listening on 127.0.0.1:80
        #
        #location ~ /\.php$ {
        #    proxy_pass http://127.0.0.1;
        #}

        # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
        #
        #location ~ /\.php$ {
        #    root           html;
        #    fastcgi_pass   127.0.0.1:9000;
        #    fastcgi_index  index.php;
        #    fastcgi_param  SCRIPT_FILENAME /scripts$fastcgi_script_name;
    }
}
```

端口和服务名称

项目代码目录

把项目代码发到nginx的 html/issweb/build 下,从启动nginx服务
sbin/nginx -s reload

2、开启外网访问

在Linux系统中默认有防火墙iptables管理者所有的端口，只启用默认远程连接22端口其他都关闭，咱们上面设置的80等等也是关闭的，所以我们需要先把应用的端口开启

方法一直接关闭防火墙，这样性能较好，但安全性较差，如果有前置防火墙可以采取这种方式

```
关闭防火墙
[root@localhost ~]# service iptables stop
关闭开机自启动防火墙
[root@localhost ~]# chkconfig iptables off
[root@localhost ~]# chkconfig --list|grep ipt
```

方法二将开启的端口加入防火墙白名单中，这种方式较安全但性能也相对较差

```
编辑防火墙白名单
[root@localhost ~]# vim /etc/sysconfig/iptables
增加下面一行代码
-A INPUT -p tcp -m state -- state NEW -m tcp --dport 80 -j ACCEPT
保存退出，重启防火墙
[root@localhost ~]# service iptables restart
```

五、参考文献：

<http://www.nginx.cn/doc/>

<https://www.cnblogs.com/taiyonghai/p/6728707.html>