

# 后盾网 人人做后盾

[www.houdunwang.com](http://www.houdunwang.com)

## Nginx服务器

后盾网 2011-2017

---

- Nginx是俄罗斯人编写的十分轻量级的HTTP服务器，Nginx是由俄罗斯人 Igor Sysoev为俄罗斯访问量第二的 Rambler.ru站点开发的。
- Nginx 超越 Apache 的高性能和稳定性，使得国内使用 Nginx 作为 Web 服务器的网站也越来越多，其中包括新浪博客、新浪播客、网易新闻、腾讯网、搜狐博客.....
- 在高并发连接的情况下，Nginx是Apache服务器不错的替代品。Nginx使用了最新的epoll网络I/O模型，Apache所采用的select网络I/O模型非常低效。下面用一个比喻来解析Apache采用的select模型和Nginx采用的epoll模型进行之间的区别：
- 假设你在后盾上课，住的宿舍楼有很多房间，你的朋友要来找你，select版宿管就会带着你的朋友挨个房间去找，直到找到你为止。而epoll版宿管会先记下每位同学的房间号，你的朋友来时只需告诉你的朋友你住在哪个房间即可，不用亲自带着你的朋友满大楼找人。如果来了10000个人，都要找自己住这栋楼的同学时，select版和epoll版宿管谁的效率更高，不言自明。

# Nginx HTTP Server

---

配置CentOS 第三方yum源（CentOS默认的标准源里没有nginx软件包）

- `wget http://www.atomicorp.com/installers/atomic`
- `sh atomic`

安装Nginx

- `yum install -y nginx`
- `service nginx start`
- `chkconfig --level 35 nginx on`

安装mysql

- `yum install -y mysql mysql-server mysql-devel`
- `service mysqld start`
- `chkconfig --level 35 mysqld on`

安装php和所需组件使PHP支持MySQL、FastCGI模式

- `yum install -y php lighttpd-fastcgi php-common php-devel php-fpm php-mysql`
- `service php-fpm start`
- `chkconfig --level 35 php-fpm on`

# 安装Lnmp环境

## 配置nginx支持php

- `mv /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak` 将配置文件改为备份文件
  - `cp /etc/nginx/nginx.conf.default /etc/nginx/nginx.conf` 由于原配置文件要自己去写 因此可以使用默认的配置文件作为配置文件
  - `vi /etc/nginx/nginx.conf` 修改nginx配置文件，添加fastcgi支持
- 
- `location /{`
  - `root /www;`
  - `index index.php index.html index.htm;`
  - `}`
  - `location ~ \.php$ {`
  - `root         /www;`
  - `fastcgi_pass  127.0.0.1:9000;`
  - `fastcgi_index index.php;`
  - `fastcgi_param SCRIPT_FILENAME /www$fastcgi_script_name;`
  - `include       fastcgi_params;`
  - `}`

# 安装Lnmp环境



配置nginx支持php

- vi /etc/php.ini
- 在文件末尾添加cgi.fix\_pathinfo = 1

重启nginx php-fpm

- service nginx restart
- service php-fpm restart

建立info.php文件

- vi /www/info.php
- <?php
- phpinfo();
- ?>

本地浏览器输入：192.168.1.100/info.php

- 显示php界面 环境搭建成功

# 安装Lnmp环境

---

开启

- `service nginx start`

关闭

- `service nginx stop`

重起

- `service nginx restart`

# Nginx启动与关闭

---

## 主配置文件

- /etc/nginx/nginx.conf

## 扩展配置文件

- /etc/nginx/conf.d/\*.conf, 需要主配置文件加载

```
http {  
    # Load config files from the /etc/nginx/conf.d directory  
    include /etc/nginx/conf.d/*.conf;  
}
```
- conf.d下面的conf文件才能生效

# Nginx配置文件说明

---

```
#http服务器
```

```
http{
```

```
#主机设置
```

```
server{
```

```
#默认访问请求
```

```
location / {
```

```
}
```

```
}
```

```
}
```

# Nginx配置文件架构

---



```
server {  
    #侦听80端口  
    listen    80;  
    #定义使用www.xx.com访问  
    server_name www.xx.com;  
    #设定本虚拟主机的访问日志  
    access_log logs/www.xx.com.access.log main;  
    #默认请求  
    location / {  
        root /www;    #定义服务器的默认网站根目录位置  
        index index.php index.html index.htm; #定义首页索引文件的名称  
    }  
}
```

# 主配置文件设置

---

#PHP 脚本请求全部转发到 FastCGI处理. 使用FastCGI默认配置.

```
location ~ \.php$ {  
  
    root /www;  
  
    fastcgi_pass 127.0.0.1:9000;  
  
    fastcgi_index index.php;  
  
    fastcgi_param SCRIPT_FILENAME /www$fastcgi_script_name;  
  
    include fastcgi_params;  
  
}
```

# 主配置文件设置

---

## • 开启目录列表

打开nginx.conf文件，在location server 或 http段中加入

```
http {
```

```
    autoindex on;//自动显示目录
```

```
    autoindex_exact_size off;//人性化方式显示文件大小否则以byte显示
```

```
    autoindex_localtime on;//按服务器时间显示，否则以gmt时间显示
```

```
}
```

# 显示目录

- 修改**PHP**配置文件

```
vi /etc/php.ini
```

```
display_errors = On
```

- 修改**php-fpm.conf**配置文件

```
vi /etc/php-fpm.d/www.conf
```

```
php_flag[display_errors] = on
```

- 重启**nginx**和**php-fpm**

```
service nginx restart
```

```
service php-fpm restart
```

# 开启PHP错误



- 在/www目录下面创建qq和hd两个目录，里面放入index.html
- 通过虚拟主机www.qq.com访问qq目录， www.hd.com访问hd目录
- nginx下，一个server标签就是一个虚拟主机。

# 虚拟主机

---

- nginx.conf配置文件中的http{}中新增如下代码：

```
server {  
    listen 80;  
    server_name www.qq.com;  
    index index.html;  
    root /www/qq;  
}  
  
server {  
    listen 80;  
    server_name www.hd.com;  
    index index.html;  
    root /www/hd;  
}
```

# 虚拟主机

- 重启nginx, service nginx restart
- 更改本地hosts文件指向测试主机
- 浏览器访问不同地址就会出现不同页面

# 虚拟主机

---

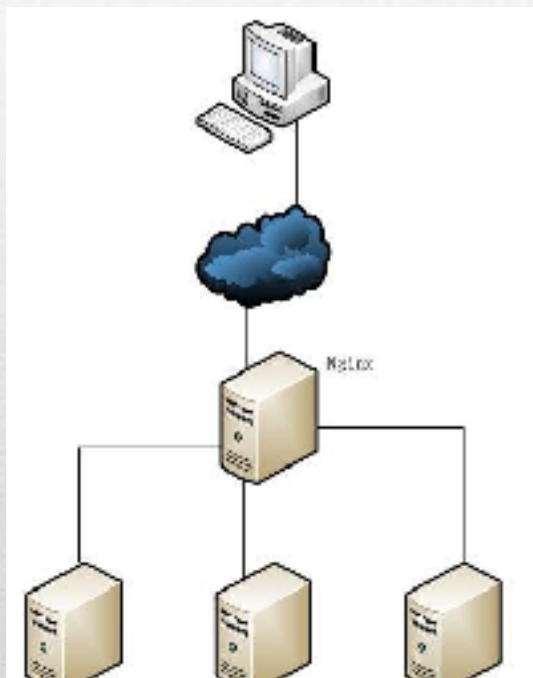
- 当一个代理服务器能够代理外部网络上的主机，访问内部网络时，这种代理服务的方式称为反向代理服务。此时代理服务器对外就表现为一个Web服务器，外部网络就可以简单把它当作一个标准的Web服务器而不需要特定的配置。不同之处在于，这个服务器没有保存任何网页的真实数据，所有的静态网页或者CGI程序，都保存在内部的Web服务器上。
- 例：下面的负载均衡

# Nginx反向代理

---



- 通过反向代理实现负载均衡，负载均衡是大流量网站要做的措施，单从字面上的意思来理解就可以解释N台服务器平均分担负载，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。那么负载均衡的前提就是要有多台服务器才能实现，也就是两台以上即可。



# 负载均衡

- 删除之前的虚拟主机配置server，在虚拟机里克隆三台CentOS，A服务器做为主服务器，域名直接解析到A服务器（192.168.1.10）上，由A服务器负载均衡到B服务器（192.168.1.100）与C服务器（192.168.1.200）上。这样就算B或者C服务器宕机也不影响网站访问，这样就不会担心在负载均衡模式下因为某台机器宕机而拖累整个站点了。
- 测试域名：hd.com
- A服务器IP：192.168.1.10（主）也称之为反向代理服务器
- B服务器IP：192.168.1.100
- C服务器IP：192.168.1.200
- 由于测试，域名就随便使用一个hd.com用作测试，所以解析在hosts文件设置  
打开：C:\Windows\System32\drivers\etc\hosts文件，在末尾添加192.168.1.10 hd.com

# 部署思路

---

- A服务器nginx.conf配置，设置完毕记得重启nginx

在http段加入以下代码，配置服务器集群server cluster

下面weight代表2/5访问192.168.1.100，3/5访问192.168.1.200

```
upstream hd.com {  
    server 192.168.31.31:80 weight=3;  
    server 192.168.31.32:80 weight=2;  
}  
server{  
    listen 80;  
    server_name hd.com;  
    location / {  
        #反向代理地址  
        proxy_pass      http://hd.com;  
        #设置主机头和客户端真实地址，以便服务器获取客户端真实IP  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    }  
}
```

# 主服务器部署(反向代理)

- B、C服务器nginx.conf设置，设置完毕记得重启nginx

在http段加入以下代码

```
server{  
listen 80;  
server_name hd.com;  
index index.html;  
root /www;  
}
```

- 当访问hd.com的时候，为了区分是转向哪台服务器处理所以分别在B、C服务器下写一个不同内容的/www/index.html文件以作区分。
- 打开浏览器访问hd.com结果，刷新会发现所有的请求均分别被主服务器（192.168.1.10）分配到B服务器（192.168.1.100）与C服务器（192.168.1.200）上，实现了负载均衡效果。

# 从服务器部署



- 实现nginx的虚拟主机与负载均衡

# 作业

---