第三章 使用nginx搭建静态资源服务器

传统的web项目，一般都将静态资源(包括图片、JS、CSS、HTML等)存放在 webroot的目录下，这样做很方便获取静态资源，但是如果说web项目很大，用户很多，静态资源也很多时，服务器的性能 或许就会很低下了。这种情况下一般都会需要一个静态资源的服务器。

一、一个简单静态资源服务器的配置

搭建nginx服务器首先得安装nginx服务，直接介绍静态服务器的配置   
进入nginx安装目录的conf目录下（etc/nginx/default.conf，修改default.conf文件，在一个server{}中添加 一个location 部分配置代码如下:

server {

listen 80;

#server\_name localhost;

server\_name bochytest.com;

location /image/ {

root /usr/local/myImage/;

autoindex on;

}

}

从上面的配置可以看出来 端口为80，server\_name为bochytest.com（写ip地址也可以）.

这个配置表示输入localhost:80/image/ 时会访问本机的/usr/local/myImage/image/ 目录。所以要新建/usr/local/myImage/image/ 目录，同时还要在nginx安装目录的html目录中新建一个 与 location中 image同名的image目录，安装配置好nginx服务器后默认目录是/usr/share/nginx/html,

虽然该目录里面什么也没有，在/usr/local/myImage/image/ 中我们放一张图片1.jpg上去，重启nginx服务，就可以通过 localhost:80/image/1.jpg访问了。

其中autoindex on;表示可以访问服务端的目录。Nginx目录浏览功能(autoindex)，Nginx默认是不允许列出整个目录的。如需此功能，打开default.conf文件，在location server 或 http段中加入

autoindex on;

验证时，最会需要加上/,

<http://localhost/image/>

参数含义：

listen：表示当前的代理服务器监听的端口，默认的是监听80端口。注意，如果我们配置了多个server，这个listen要配置不一样，不然就不能确定转到哪里去了。

server\_name：表示监听到之后需要转到哪里去，这时我们直接转到本地，这时是直接到nginx文件夹内。

location：表示匹配的路径，这时配置了/表示所有请求都被匹配到这里

root：里面配置了root这时表示当匹配这个请求的路径时，将会在这个文件夹内寻找相应的文件，这里对我们之后的静态文件伺服很有用。

index：当没有指定主页时，默认会选择这个指定的文件，它可以有多个，并按顺序来加载，如果第一个不存在，则找第二个，依此类推。

配置一个简单的静态资源服务器

1、在default.conf中添加一个location

location /images/ {

root /usr/local/myImages/;

}

2、创建/usr/local/myImages/images

3、把图片放入/usr/local/myImages/images

4、录是/usr/share/nginx/html添加一个空目录images

5、测试

**2 配置支持静态资源的服务器**

可以参考 expires 指令手册来设置 HTTP 头部过期时间，这个标记可以放在 "http {}"、"server {}"、"location {}" 等语句块或者 "location {}" 语句块中的条件语句中。一般会在 "location" 语句块中用 "expires" 指令控制你的静态文件，就像下面一样：

location ~\* \.(jpg|jpeg|png|gif|ico|css|js)$ {

expires 365d;

}

在上面的例子中，所有后缀名是 .jpg、 .jpeg、 .png、 .gif、 .ico、 .css 和 .js 的文件会在浏览器访问该文件之后的 365 天后过期。因此你要确保 location {} 语句块仅仅包含能被浏览器缓存的静态文件。

* ●"ms": 毫秒
* ●"s": 秒
* ●"m": 分钟
* ●"h": 小时
* ●"d": 天
* ●"w": 星期
* ●"M": 月 (30 天)
* ●"y": 年 (365 天)

例如："1h30m" 表示一小时三十分钟，"1y6M" 表示一年六个月。

200 OK (from cache)  是浏览器没有跟服务器确认，直接用了浏览器缓存；而 304 Not Modified 是浏览器和服务器多确认了一次缓存有效性，再用的缓存。200(from cache) 是速度最快的,因为不需要访问远程服务器,直接使用本地缓存.304 的过程是, 先请求服务器, 然后服务器告诉我们这个资源没变, 浏览器再使用本地缓存.

**开发实例：**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title></title>

<script src="http://bochytest.com/jquery-2.1.0.js"></script>

<link rel="stylesheet" type="text/css" href="http://bochytest.com/bootstrap.css" />

</head>

<body>

<h1>从Nginx中获取静态图片资源</h1>

<img src="http://bochytest.com/image/3.jpg" />

</body>

</html>

三、配置京东的静态页面



在配置文件配置

location /images/ {

root /usr/share/nginx/html/;

#index default.jpg

#autoindex on;

}

location ~\*\.(jpg|jpeg|png|gif|ico|css|js|html)$ {

root /usr/share/nginx/html/;

index index.html;

#expires 365d;

}

然后把JD.css和JD.html和images文件夹放到/usr/share/nginx/html/文件夹下

启动nginx,在window下输入<http://bochytest.com/JD.html>



1. “~” for case sensitive matching:区分大小写
2. “~\*” for case insensitive matching:不区分大小写