

# 路径（十六）：在本地 CentOS 7 上搭建 Web 开发环境 — nginx 的基本配置

 [ninghao.net/blog/2087](http://ninghao.net/blog/2087)

nginx 是由一些模块组成的，不同的模块定义了各自的一些指令（Directives），指令控制了模块的行为，在 nginx 的配置文件里可以去配置这些指令。主要的配置文件是 `nginx.conf`，在这个配置文件里，会用到 `include` 指令，把其它地方的配置文件包含到这个主要的配置文件里，用这种方法可以让配置文件更有条理，也更容易维护。

## 找到 `nginx.conf`

在不同的系统下 `nginx.conf` 的位置可能不太一样，使用搜索可以很容易找到它：

```
find / -name  
nginx.conf
```

意思是从根目录开始搜索 `nginx.conf` 文件的位置。在 CentOS 7 里，这个文件是在：

```
/etc/nginx/nginx.conf
```

打开 `/etc/nginx`，在这个目录下面，会包含一些其它的文件，`.default` 结尾的文件应该是原文件的备份，比如 `nginx.conf` 是真正用的配置文件，`nginx.conf.default` 是这个文件的一个备份，如果 `nginx.conf` 出了问题，你可以把 `nginx.conf.default` 重命名为 `nginx.conf` 代替原来的文件。

## 理解配置文件

打开 `nginx.conf`，你会看到 nginx 的配置文件的样式，里面有些说明，这些内容前面都有 `#` 号，表示这是注释内容，nginx 不会理会这些用 `#` 开头的内容。真正生效的配置，是不带 `#` 号开头的。这些配置一般就是一个指令的名字，后面一个空格，再加上这个指令的参数值，结尾用一个分号。

### main

```
user  
nginx;
```

`user` 是指令的名字，这个指令可以设置系统运行 nginx 时候用的用户名，这里设置成了 `nginx` 这个用户。

```
worker_processes  
1;
```

`worker_processes` 指令设置了 nginx 同时运行的进程数，或者叫 nginx 的实例。nginx 有一个 master 进程，还有一些 worker 进程。master 进程的主要工作是读取和鉴定配置，维护 worker 进程。真正提供服务的是 worker 进程，nginx 用了一种有效的方式，把请求分布到不同的 worker 进程上去处理。`worker_processes` 指令设置的就是这个 woker 进程的数量，这个数量可以根据服务器的 CPU 核心数来设定，8 核的 CPU 就设置成 8 个 worker 进程。

```
error_log /var/log/nginx/error.log;
```

`error_log` 指令设置了错误日志存放的位置。

```
pid
    /run/nginx.pid;
```

`pid` 指令设置了 nginx 的 master 进程 ID (PID) 写入的位置，操作系统会用到这个 PID 跟踪还有发送信号给 nginx 的进程。

上面这些配置直接放到了配置文件的下面，nginx 的配置还可以放到一些上下文里面，比如 `events`，`http`，`server`，`location`，等等，直接放到 nginx 配置文件的下面的上下文是 `main`，这些配置会影响到整个服务器。

## events

```
events {
    worker_connections
1024;
}
```

上面这块配置用到了一组大括号，上下文是 `events`，里面用了一个 `worker_connections` 指令，它可以设置每个 worker 进程同时能为多少个连接提供服务。它的值设置成多少，需要多在服务器上实践，一般你可以用 CPU 核心数 \* 1024，得到的结果设置成 `worker_connections` 的参数值。

## http

```
http {

...
}
```

接着是一个 `http` 上下文的配置区块，一般我们对服务器的设置都放到这个区块里面。`http` 配置区块里面会包含 `server` 配置区块，我们可以定义多个 `server` 区块，去配置不同的服务器，就是虚拟主机。`server` 配置区块下面又会包含 `location` 区块，这些区块可以设置匹配不同的请求，根据请求的地址，提供不同的服务，有些请求直接给它们静态文件，有些请求可能要交给其它的服务器处理，比如 FastCGI 服务器。

再看几个 `http` 配置区块里的配置。

```
include
    /etc/nginx/mime.types;
```

这里用了一个 `include` 指令，把 `/etc/nginx/mime.types` 这个文件的内容加载进来，在这个文件里定义了 MIME type，MIME type 告诉浏览器，怎么样去处理不同类型的文件。

```
access_log    /var/log/nginx/access.log
    main;
```

`access_log` 指令设置了访问的日志存储的位置，在 `server` 和 `location` 区块里也可以使用这个指令。

```
index    index.html
    index.htm;
```

index 指令设置了当请求的地址里不包含特定的文件的时候，默认打开的文件。这里设置成了 index.html index.htm，如果目录下面有 index.html 就打开它，如果没有就去找 index.htm，还没有就返回 404 错误。

```
include /etc/nginx/conf.d/*.conf;
```

include 指令可以把其它的文件包含进来，这样可以保持配置文件的整洁。这里包含的是 /etc/nginx/conf.d/\*.conf，\*.conf 表示所有的带 .conf 后缀的文件。也就是我们可以把自己的配置放到 /etc/nginx/conf.d 这个目录的下面，只要文件的后缀是 .conf，这些配置文件都会起作用。

## server

```
server
{
    ...
}
```

这又是一个配置区块，上下文是 server，在这种类型的配置区块里可以配置不同的服务器。就是每个 server 区块都可以定义一台虚拟主机，如果你想在一台服务器上运行多个网站的话，你应该会用到这种配置区块。一般每个 server 区块的配置都可以单独放到一个文件里。

```
listen      80
default_server;
```

listen 指令可以设置服务器监听的端口号，还有 IP 地址或者主机名。这里监听了 80 端口，这是 http 协议默认的端口号，default\_server 的意思是，在 80 端口的请求，如果不匹配在其它地方配置的虚拟主机，就会默认使用这个服务器 (default\_server)。在监听的端口前面可以加上 IP 地址或许本地的主机名，像这样：127.0.0.1:80，localhost:80，42.120.40.68:80 ...

```
server_name
    localhost;
```

server\_name，这个指令可以创建基于主机名的虚拟主机，比如我的域名是 ninghao.net，我又为这个域名添加了一些主机名，[www.ninghao.net](http://www.ninghao.net)，blog.ninghao.net，talk.ninghao.net ... 我想让用户在访问这些主机名的时候，打开不同的网站。这就可以去创建多个 server 配置区块，每个区块里用 server\_name 去指令这个虚拟主机的主机名，用户在访问这个主机名的时候，nginx 会根据请求的头部上的信息来决定用哪个虚拟主机为用户提供服务。下面是一些参考例子：

```
server_name
    ninghao.net;
```

nginx 会处理用户对 ninghao.net 的请求。

```
server_name  ninghao.net
www.ninghao.net;
```

nginx 会处理对 ninghao.net 还有 [www.ninghao.net](http://www.ninghao.net) 的请求。

```
server_name  *.ninghao.net;
```

nginx 会处理所有的对 ninghao.net 子域名的请求。继续再看一下 nginx.conf 里的 server 配置区块。

```
root
/usr/share/nginx/html;
```

root 指令配置了这个虚拟主机的根目录。之前安装好 nginx，在浏览器里打开服务器的 IP 地址，看到的测试页面，就在这个目录的下面。

## location

```
location / {
}
```

location 配置区块会定义在 server 配置区块里边儿。它可以配置 nginx 怎么样响应请求的资源，server\_name 指令告诉 nginx 怎么处理对域名的请求，location 指令设置的是对特定的文件还有目录的请求。上面设置的是 /，表示如果请求的是根目录，在这个配置区块里没有其它的东西，这样如果请求根目录，就会用在 server 区块里，root 指令设置的根目录，为用户提供文件。每个 server 区块里面可以定义多个 location 区块，分别去配置对不同目录或者文件的请求应该怎么样响应。下面再看几个 location 的配置例子：

```
location ~ \.(gif|jpg|png)$ { ...
}
```

上面的 location 后面是一个 ~ 号，表示它的后面是一个正则表示式，这里的意思是，请求的是服务器里的 .gif，.jpg，或者 .png 格式的文件。具体怎么处理，可以放到它后面的大括号里。注意这个匹配是区分大小写的，如果请求的是 .GIF，这个请求就不匹配这个 location 的配置。如果想不区分大小写，在 ~ 后面，加上一个 \* 号：

```
location ~* \.(gif|jpg|png)$ { ...
}
```

location 定义匹配，更具体的那个会胜出，比如一个是 /，另一个是 /blog，这样就会用 /blog 这个 location。你可以使用 ^~，让 nginx 停止查找更具体的匹配，意思就是，如果有请求匹配这个 location，就直接用它里面的配置，不要再继续查找别的 location 设置的匹配了。

```
location ^~ /blog/ { ...
}
```

精确的匹配，可以用一个 = 号。

```
location = / { ...
}
```

路径 CentOS nginx