

## 07 打扮你的nginx：配置文件初识

更新时间：2019-12-12 09:31:11



“

读书给人以快乐、给人以光彩、给人以才干。——培根

”

### 前言

在之前的两篇文章中，我们详细的介绍了 `nginx` 安装的过程，以及常用的控制命令。这一篇文章我们重点讲解 `nginx` 配置文件。

### 配置文件在哪

由于我们在编译的时候设置了 `--prefix` 选项，指定前缀路径是 `/usr/local/nginx`，那么默认的配置文件就在 `/usr/local/nginx/conf/nginx.conf`。

如果找不到配置文件也没关系，我们有两种方式来看配置文件：

① 使用 `ps` 命令查看

```
[root@79d6ba1cc92a ~]#  
[root@79d6ba1cc92a ~]# ps -ef | grep nginx  
root      38      0  0 12:06 ?        00:00:00 nginx: master process /usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf  
nobody    39      38  0 12:06 ?        00:00:00 nginx: worker process  
root      41      14  0 12:06 pts/1    00:00:00 grep --color=auto nginx
```

配置文件

## ② 使用 `-t` 命令查看

```
[root@79d6ba1cc92a /]#  
[root@79d6ba1cc92a /]# ps -ef | grep nginx  
root      46      0  0 12:10 ?        00:00:00 nginx: master process /usr/local/nginx/sbin/nginx  
nobody    47      46  0 12:10 ?        00:00:00 nginx: worker process  
[root@79d6ba1cc92a /]#  
[root@79d6ba1cc92a /]#
```

没有 `-c` 参数

有时候我们会遇到上图的情况，这表明了 `nginx` 启动的时候使用了默认的配置文件的。

这种情况下我们可以通过 `-t` 命令来比较 `取巧` 的获取配置文件的路径：

```
[root@79d6ba1cc92a /]#  
[root@79d6ba1cc92a /]# /usr/local/nginx/sbin/nginx -t  
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok  
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful  
[root@79d6ba1cc92a /]#  
[root@79d6ba1cc92a /]#
```

## 配置文件结构

我们首先看一下默认的配置文件的：

```
#user nobody;  
worker_processes 1;  
  
#pid logs/nginx.pid;  
  
events {  
    worker_connections 1024;  
}  
  
http {  
    include mime.types;  
    default_type application/octet-stream;  
  
    sendfile on;  
    #tcp_nopush on;  
  
    #keepalive_timeout 0;  
    keepalive_timeout 65;  
  
    #gzip on;  
  
    server {  
        listen 80;  
        server_name localhost;  
  
        #charset koi8-r;  
  
        #access_log logs/host.access.log main;  
  
        location / {  
            root html;  
            index index.html index.htm;  
        }  
  
        error_page 500 502 503 504 /50x.html;  
        location = /50x.html {  
            root html;  
        }  
    }  
}
```

为了节省篇幅，我删除了一些注释的内容。

这是一个典型的 **最小配置** 文件，通过短短的配置文件，实现了一个 **http** 服务器。

## 配置文件一览

**nginx** 采用了简单的文本格式的配置文件，下图总结了 **nginx** 指令一些特性。



### 指令的特点

按照指令的特点，可以将指令分为四种形式，分别是注释，简单指令，块指令和包含指令。

注释：**nginx** 把以 **#** 开头的 **单行** 内容视为注释，在解析配置文件时候会忽略这部分内容。注意，注释必须是单行的，如果多行注释，那么每行都要以 **#** 开头。

简单指令：以分号结尾，比如下面的：

```
user nobody;
worker_processes 1;
```

块指令：将一组组合组合在一起，使用花括号 **{ }** 进行分组，如：

```
events {
    use epoll;
}
```

这里的 **events** 就是一个块指令。

包含指令：将一些指令放置到一个单独的文件中，然后使用 **include** 指令将这些文件包含到配置文件中，就好像在把这个文件的内容填充到了这个位置一样。可以参考 **c** 语言的 **include** 预处理以及 **php** 的 **include** 等指令。

```
include mime.types;
```

### 指令层次

如果按照层次划分，**nginx** 的指令分为了多个层级，如下：

```

user nobody; # main 级别配置, nginx核心功能

events {
    use epoll; # events级别配置, nginx事件机制
}

http { # http 级别 nginx的http核心模块

    server { # server 级别, 一个server代表一个虚拟主机

        location {
            # location级别, 一个location代表一类url
        }
    }
}

```

一个 `http` 模块可以包含多个 `server`，同样的，一个 `server` 可以包含多个 `location`。

有些指令可以同时存在多个不同的 `级别` 中，那么这里就存在了一个指令覆盖问题。

我们以 `root` 指令为例，下面是官方文档中关于 `root` 的解释：

```

Syntax: root path;
Default: root html;
Context: http, server, location, if in location
使用范围: http, server, location, location中的if

```

如果我们存在下面的配置：

```

http {
    server { # server1
        server_name my.first.server.com
        root /root/server1;

        location / { #location1

        }

        location /img { #location2
            root /root/img;
        }
    }
}

```

那么在 `location2` 中的 `root` 会覆盖 `server1` 中的 `root`，而 `location1` 中由于没有 `root`，所以会继承它的上一层级的 `root`。

在 `nginx` 中，下一层级如果没有配置指令的话，会继承上一层级的同一个配置的值。

如果大家感兴趣的话，可以看一下 `nginx` 的源码，我们可以自由的定义是否继承上一级的值。我看了 `nginx` 的部分源码，基本上都是继承上一层级的值。

## 总结

这一篇文章中我们从整体上介绍了 `nginx` 配置文件的结构，为后面的学习奠定一个良好的基础。

```

}

```

