

## 09 必要的全局配置选项

更新时间: 2019-12-24 09:40:58



不想当将军的士兵，不是好士兵。——拿破仑

### 前言

前面我们说了关于 **Nginx** 很多的东西，比如 **Nginx** 作用，事件机制，如何选择事件机制等等，我们在工作中，绝大多数都是把 **Nginx** 作为一个 **HTTP** 服务器或者一个反向代理服务器(**PS**: 我还没见过有人用 **Nginx** 做其他的用途，比如 **Nginx** 可以作为邮件服务器，**but**, 我还从未见过~)。

这一节内容我们用一个配置文件来体验一下一个可用的 **Nginx** 配置文件长什么样子。

### 配置文件初体验

俗话说"人靠衣裳马靠鞍"，**Nginx** 也要进行配置才能够使用，我们看一下怎样才能把 **Nginx** 打扮的 **漂漂亮亮** 的吧。

```

user nobody;
worker_processes auto;

error_log  "/worker/log/error.log" warn;
pid        "/worker/pid/nginx.pid";

events {
    use epoll;
    worker_connections 8192;
}

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 "/worker/log/access.log" main;

    server_names_hash_bucket_size 128;
    client_header_buffer_size 4k;
    large_client_header_buffers 4 32k;
    client_max_body_size 20m;
    client_body_buffer_size 4m;

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;

    gzip on;
    gzip_min_length 1k;
    gzip_buffers 4 16k;
    gzip_http_version 1.0;
    gzip_comp_level 2;
    gzip_types text/plain application/x-javascript text/css application/xml;
    gzip_vary on;

    server {
        listen 80;
        server_name www.test.com;
        location ~ \.php$ {
            root      /project/src;
            fastcgi_pass 127.0.0.1:9000;
            fastcgi_index index.php;
            fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
            include   fastcgi.conf;
        }
        location / {
            root      /static;
        }
    }
}

```

上面是一个真实服务器中 **Nginx** 配置的简略版，“麻雀虽小五脏俱全”，它配置了一个 **HTTP** 服务器，可以处理静态文件以及 **php** 动态请求。

我们先使用这个配置文件启动一下 **Nginx**，来体验一下.....

```
[root@a691d858ef77 conf]#  
[root@a691d858ef77 conf]# /usr/local/nginx/sbin/nginx  
[root@a691d858ef77 conf]# ps -ef | grep nginx | grep -v grep  
root      348      0  00:39 ?          00:00:00 nginx: master process /usr/local/nginx/sbin/nginx  
nobody    349    348  0  00:39 ?          00:00:00 nginx: worker process  
nobody    350    348  0  00:39 ?          00:00:00 nginx: worker process  
[root@a691d858ef77 conf]# curl http://www.test.com/hello.html  
hello Nginx  
[root@a691d858ef77 conf]# cat /static/hello.html  
hello Nginx  
[root@a691d858ef77 conf]#
```

## 全局配置

### **user**指令

我们知道，操作系统中的所有资源都具有所属者以及其他各种访问权限(强大的 **Linux** 也不例外~)。这种权限控制可以让操作系统变得很安全，防止 **黑客** 破坏我们的系统。

**Nginx** 提供了 **user** 全局配置指令来设置 **worker** 进程的所属者，这样就可以让 **Nginx** 访问特定权限的资源，保护我们服务器的稳定性。

```
[root@a691d858ef77 conf]# ps -ef | grep nginx | grep -v grep  
root      348      0  00:39 ?          00:00:00 nginx: master process /usr/local/nginx/sbin/nginx  
nobody    349    348  0  00:39 ?          00:00:00 nginx: worker process  
nobody    350    348  0  00:39 ?          00:00:00 nginx: worker process
```

我们可以通过 **ps** 命令查看 **worker** 进程，可以看到这个进程的所属者是 **nobody** 用户。

我们下面体验一下这个 **user** 的作用。

在上面体验 **Nginx** 的时候，我们成功的访问到了服务器上面的 **hello.html** 文件，并且返回了 **hello Nginx** 字符串。

下面我们修改 **hello.html** 文件的权限，删除 **nobody** 的可读权限，重新访问这个文件，这个时候 **Nginx** 就会返回 **403** 错误。

```
[root@a691d858ef77 conf]# ls -lah /static/hello.html  
-rw-r--r-- 1 root root 12 Dec 23 00:30 /static/hello.html  
[root@a691d858ef77 conf]# chmod o-r /static/hello.html  
[root@a691d858ef77 conf]# ls -lah /static/hello.html  
-rw-r----- 1 root root 12 Dec 23 00:30 /static/hello.html  
[root@a691d858ef77 conf]# curl http://www.test.com/hello.html  
<html>  
<head><title>403 Forbidden</title></head>  
<body>  
<center><h1>403 Forbidden</h1></center>  
<hr><center>nginx/1.16.1</center>  
</body>  
</html>  
[root@a691d858ef77 conf]#
```

这里就是 **user** 指令的作用，大家是否熟悉了呢？

### **worker\_processes** 指令

我们前面说过，`Nginx`是一个`Master-Worker`的工作模式。`Nginx`在启动之后会生成一个`master`进程，`master`不处理请求，而是生成和管理`worker`进程，`worker`进程主要负责干活（处理请求）。

那么大家可能要问了？这个值应该设置为多大呢？是不是这个值越大越好？当然不是了，我们看一下大神们给出的建议：

```
nginx doesn't benefit from more than one worker per CPU.  
If Nginx is doing CPU-intensive work such as SSL or gzipping and you have 2 or more CPUs/cores, then you may set worker_processes to be equal to the number of CPUs or cores.
```

所以`worker_processes`的值并不是越多越好，最好是和我们服务器`CPU`核数量一样。`Nginx`提供了一个默认的`auto`值就是这样的原理。

## event配置

`event`是配置`Nginx`使用的事件处理机制的。这一部分是非常重要的。我们在上一节中介绍了关于如何选择事件处理机制以及为什么要这么选，大家可以复习一下~

## http配置

`http`配置是一个非常重要的配置，这部分也是我们最关心的部分，这是让`Nginx`提供`HTTP`服务必不可少的，我们在后面的内容中会详细的介绍这部分。

## 总结

我们在这一节内容中简单的介绍了两个几个必要的全局配置项，包括了`user`，`worker_processes`，`event`，和`http`，让大家对整体的配置有一个概念性的认识，后面遇到的时候我们还会详细的介绍。

}

← 08 如何选择事件模块？

10 条条大路通罗马，形形色色的虚拟主机

→