

## 一、从0构建Nginx WEB平台

Nginx是一款开源的，免费的WEB服务器软件，2019年3月12日被F5硬件负载均衡厂家以6.7亿美金收购，Nginx主要使用于发布网站代码（开发人员编写的）提供网页信息服务的，用户通过浏览器可以实现页面的访问：

Nginx web软件默认只能处理静态网页，不能直接处理动态网页，动态网页交于第三方的程序去解析，Nginx官方宣称其处理静态网页的并发能力可以达到5w/s其相当于Apache WEB整体性能的5~10倍：并发更高，性能更好（相对于Apache）

Apache：是传统web服务器，开发与Nginx之前。现在Nginx正在一步步取代Apache；也是用来处理静态网页的；不能处理动态网页。

Tomcat：是Apache基金会旗下的一款产品：是处理动态网页的。

静态网页：一般是跟后端数据库不发生交互的网页，网页内容很少更新或者几乎不更新。网页文件后缀命名通常是以htm、html、xml等结尾的。

动态网页：跟后端数据发生交互，网页内容经常更新，或者随后端数据库内容变化而更新；网页文件后缀以asp、php、jsp等结尾。

## 二、基于Centos Linux 操作系统，从0开始构建一套Nginx WEB服务平台，主要有两种方式、

YUM二进制的方式：部署方式相对比较简单，快捷高效，可以自动校验软件的安全性可以自动校验软件包的正确性，可以自动解决软件包之间的依赖关系。可以自动安装软件服务并设置为系统服务。不能自定义软件服务特定的功能和模块。默认安装完毕之后文件和目录相对比较分散，不变与后期的维护和管理。

MAKE源码编译的方式：相对较复杂，不能校验，不能自动安装系统的依赖，自动配置自定义软件的模块和功能，安装目录比较统一，便于后期的维护和管理。

## 三、基于MAKE源码编译方式：

1、构建Nginx WEB平台，首先从Nginx官网下载Nginx软件包稳定的版本，nginx-1.16.0.tar.gz

主机IP地址：192.168.1.81

```
1 [root@localhost ~]# wget -c http://nginx.org/download/nginx-1.16.0.tar.gz -P /tmp/
2                                     // -c 断点续传， -P 指定特定目录    下载nginx源码包。
3 [root@cdeba90ec46e ~]# tar -zxvf nginx-1.16.0.tar.gz // -x解压、 -z gzip压缩、 -v详细、 -f文件
```

因为Nginx是基于C语言开发的源代码软件程序，默认不能被linux操作系统直接使用，需要借助C编译器将源代码编译器将源代码文件编译生成二进制文件，所以需执行源代码软件部署三步骤：

```
1 // 预编译：主要是检测linux系统安装该软件所需的依赖环境、库文件、检测linux系统是否存在GCC编译环境
2 // (C编译器)，指定软件服务部署的路径，自定义软件服务特定的模块和功能，最终会产生Makefile文件。
3 [root@localhost nginx-1.16.0]# ./configure --prefix=/usr/local/nginx --user=www --group=www
4           // 设置我们的用户和组www、便于后期管理我们的nginx。
5           // 需要我们手动创建。--with-http_stub_status_module 开启状态监控模块
6 // 编译：主要通过make编译工具，读取Makefile文件，调用linux操作系统下GCC编译环境(C编译器)将软件包中的
7 源代码文件编译生成二进制文件，Makefile文件用途，主要告知make编译工具，在编译源代码文件时，
```

```

8 在哪个源码开始编译至哪个源码结束编译。以及记录编译时的依赖关系
9 [root@localhost nginx-1.16.0]# make
10 cc -c // 只编译，不产生二进制文件
11 cc -o // 产生二进制文件
12 cc -o objs/nginx // 把这个文件拷贝到我们的安装目录之下。
13 [root@cdeba90ec46e nginx-1.16.0]# md5sum objs/nginx
14 ec4cf649ae95588e9906eaa71baa354f objs/nginx // 查看得objs/nginx文件的md5位354f
15 [root@cdeba90ec46e nginx-1.16.0]# md5sum /usr/local/nginx/sbin/nginx
16 ec4cf649ae95588e9906eaa71baa354f /usr/local/nginx/sbin/nginx // nginx文件md5值是354f,
17 是因为make的时候把文件拷贝到安装目录之下
18 // 安装：主要是将第二步make编译产生的二进制文件，拷贝或者安装至linux操作系统指定的安装目录
19 --prefix=/usr/local/nginx
20 [root@localhost nginx-1.16.0]# make install
21 [root@cdeba90ec46e nginx-1.16.0]# ls -l /usr/local/nginx/
22 total 0
23 drwxr-xr-x 2 root root 333 Nov 2 08:54 conf // 配置文件所在路径
24 drwxr-xr-x 2 root root 40 Nov 2 08:54 html // 发布目录，开发写的代码存放位置，
25 drwxr-xr-x 2 root root 6 Nov // 日志文件
26 drwxr-xr-x 2 root root 19 Nov 2 08:54 sbin // 启动程序

```

配置后检查状态：

```

1 [root@cdeba90ec46e ~]# /usr/local/nginx/sbin/nginx // 启动服务
2 [root@cdeba90ec46e ~]# ps -ef | grep nginx
3 root      3131      0 00:00:00 nginx: master process /usr/local/nginx/sbin/nginx
4 www       3132    3131 00:00:00 nginx: worker process
5 root      3134      161 00:00:00 grep --color=auto nginx
6 [root@cdeba90ec46e ~]# netstat -tunlp | grep -w 80 --color // -w 打印出来，查看80端口
7 tcp      0      0 0.0.0.0:80          0.0.0.0:* LISTEN      3131/nginx: master
8 [root@cdeba90ec46e ~]# netstat -tunlp | grep -v 80           // -v 取反。不看80端口
9 Active Internet connections (only servers)
10 Proto Recv-Q Send-Q Local Address      Foreign Address      State      PID/Program name
11 tcp      0      0 0.0.0.0:22          0.0.0.0:* LISTEN      110/sshd
12 tcp6     0      0 :::22             ::::* LISTEN      110/sshd
13 [root@cdeba90ec46e ~]# firewall-cmd --add-port=80/tcp --permanent
14                                         // 开启防火墙的80 端口 // centos7
15 [root@cdeba90ec46e ~]# iptables -t filter -A INPUT -m tcp -p --dport 80 -j ACCEPT
16                                         // 开启80 端口 // centos6

```

默认门户网站：

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

2、Nginx降级实战：把nginx的版本从1.16.0降级为1.15.0（版本跨度太大可能编译不成功）

```
1 //1、查看服务的版本
2 [root@cdeba90ec46e ~]# /usr/local/nginx/sbin/nginx -v
3 nginx version: nginx/1.16.0
4 //2、发现我们的nginx版本比较高，降级
5 //我们降级需先知道降级的版本；降级到1.10.3版本，且不影响我们现有的服务。
6 //保留之前新版本的编译参数
7 //现有的nginx编译参数怎么查看
8 [root@cdeba90ec46e ~]# /usr/local/nginx/sbin/nginx -V // -V大写
9 nginx version: nginx/1.16.0
10 built by gcc 8.3.1 20191121 (Red Hat 8.3.1-5) (GCC)
11 configure arguments: --prefix=/usr/local/nginx --user=www --group=www
12     --with-http_stub_status_module //复制之前的参数预编译使用
13 //3、需先将降级之后的版本解压。
14 [root@cdeba90ec46e nginx-1.10.3]# ./configure --prefix=/usr/local/nginx --user=www
15     --group=www --with-http_stub_status_module // -V查找出的参数
16 //降级不执行make install，只执行make即可，只有安装的时候需要makeinstall
17
18 //二进制文件我们备份为.old文件：我们的服务没有收到任何影响nginx调用文件的时候是调用文件的描述符
19 //新的objs/nginx文件需让它生效，不生效还是使用之前的文件。
20 [root@cdeba90ec46e nginx-1.15.0]# kill USR2 cat /usr/local/nginx/logs/nginx.pid
21 // USR2陆续的把nginx.pid文件的程序一步步停掉。
```

### 3、Nginx平滑升级：

- 1 和nginx降级的方案一样，升级降级不必执行make install；
- 2 我们只要nginx二进制文件即可，无需安装。

### 四、Nginx WEB虚拟主机企业实战：

1、在企业生产环境中，一台nginx服务器通常不仅仅只发布一个网站，而是基于单台nginx服务器，发布+部署多套网站，可以最大化的利用Nginx WEB服务器操作系统资源，避免资源的浪费。

2、将单台Nginx WEB服务器，发布多套网站的方式，称为虚拟主机，nginx发布虚拟主机主要有三种方式。

基于同一个80端口，不同的IP地址：（多IP）不推荐，

基于同一个IP地址，不同的PORT端口：（多端口）用户直接访问都是用80端口，不推荐

基于同一个80端口，不同的访问域名：（多域名）

3、基于同一个80端口，不同的访问域名（v1.cvc.net、v2.cvc.net）；来配置Nginx多网站虚拟主机，操作方案：

```
1 //1、切换到nginx安装目录下进入conf目录，找到主配置文件：
2 [root@cdeba90ec46e conf]# vim /ur/local/nginx/conf/nginx.conf
3 //shell三剑客awk、sed、grep去除配置文件中#号和空行；
4 [root@cdeba90ec46e conf]# awk '/#/!' nginx.conf
5 //第一个斜杠开始查找，第二个斜杠查找结束，查找两个斜杠里面的内容
6 [root@cdeba90ec46e conf]# awk '!/#/' nginx.conf
```

```
7 //加上! 号表示取反，表示我们不需要显示空行。
8 [root@cdeba90ec46e conf]# awk '!/#/' nginx.conf|awk '!/^$/'
9 // ^ 表示空行; /^/以空开头以空结尾; /^$/以空开头, 不以空结尾。
10 [root@cdeba90ec46e conf]# sed '/#/d' nginx.conf|sed '/^$/d' //删除#号的空行的内容
11 [root@cdeba90ec46e conf]# grep -v "#" nginx.conf|grep -v "^$"
```

#### 4、最终nginx.conf的代码是：

```
1 [root@cdeba90ec46e conf]# grep -v "#" nginx.conf|grep -v "^$"
2 worker_processes 1; //worker_processes表示nginx的服务进程，负责接收和处理用户的请求。-9表强制kill
3 events { //events 时间模块
4     worker_connections 1024;
5         //worker_connections表示一个工作进程处理的用户请求为1024,
6         //整个nginx可以处理最大的用户并发为1024*4=4096个人用户请求
7 }
8 http {
9     include mime.types;
10    default_type application/octet-stream;
11    sendfile on;
12    keepalive_timeout 65;
13    server { //server里面的内容是我们的虚拟主机，也就是网站；默认nginx只有一套网站。
14        //若想发布多套网站，我们复制server再粘贴就可以了。
15        listen 80; //监听端口
16        server_name localhost; //server_name监听域名；localhost表示本机域名；
17        location / { //location表示我们的发布目录；
18            root html; // root:表示发布目录，html:表示相对路径;
19            //绝对路径是：/usr/local/nginx/html
20            index index.html index.htm;
21        }
22        error_page 500 502 503 504 /50x.html;
23        location = /50x.html {
24            root html;
25        }
26    }
27 }
28 }
```

```
1 [root@cdeba90ec46e conf]# ps -ef |grep nginx
2 root      3131      0 nginx: master process /usr/local/nginx/sbin/nginx
3                                         //master不负责接收用户请求，只负责管理进程。
4 www      3132  3131 nginx: worker process //worker进程
5 可以看到一个master和一个worker进程：（配置文件里面只有一个进程）
6 更改配置文件4个进程：
7 [root@cdeba90ec46e conf]# /usr/local/nginx/sbin/nginx -s reload //重启nginx
8
```

#### 5、基于同一个80端口，不同的访问域名（v1.cvc.net、v2.cvc.net）配置

```

1 [root@cdeba90ec46e conf]# vim /usr/local/nginx/conf/nginx.conf
2 worker_processes 4;
3 events {
4     worker_connections 1024;
5 }
6 http {
7     include mime.types;
8     default_type application/octet-stream;
9     sendfile on;
10    keepalive_timeout 65;
11 =====
12    server {                                //第一个虚拟主机
13        listen 80;
14        server_name v1.cvc.net;           //域名
15        location / {
16            root /usr/local/nginx/html/v1.cvc.net; //发布目录
17            index index.html index.htm;
18        }
19        error_page 500 502 503 504 /50x.html;
20        location = /50x.html {
21            root html;
22        }
23    }
24    server {                                //第二个虚拟主机
25        listen 80;
26        server_name v2.cvc.net;           //域名
27        location / {
28            root /usr/local/nginx/html/v2.cvc.net; //发布目录
29            index index.html index.htm;
30        }
31        error_page 500 502 503 504 /50x.html;
32        location = /50x.html {
33            root html;
34        }
35    }
36 }

```

## 6、手工方式创建mkdir两个发布目录;

```

1 [root@cdeba90ec46e conf]# mkdir -p /usr/local/nginx/html/v{1,2}.cvc.net //大括号表示范围
2 [root@cdeba90ec46e html]# cat >/usr/local/nginx/html/v1.cvc.net/index.html<<EOF
3 <html>
4 <h1>v1.cvc.net Test Pages 域名:www.cvc.com</h1>
5 <hr color=red>
6 </html>
7 EOF
8 [root@cdeba90ec46e html]# cat >/usr/local/nginx/html/v2.cvc.net/index.html<<EOF
9 <html>
10 <h1>v2.cvc.net Test Pages 域名:www.cvc.com</h1>
11 <hr color=red>
12 </html>

```

13 EOF  
14

```
1 [root@cdeba90ec46e /]# /usr/local/nginx/sbin/nginx -t
2 nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
3 nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
4 [root@cdeba90ec46e conf]# /usr/local/nginx/sbin/nginx -s reload //重启nginx
```

## 7、内网环境下我们需要配置本地hosts文件，域名解析；

```
1 1、看浏览器缓存，没有
2 2、看电脑本地的hosts文件，DNS服务器会将这个域名的请求交给我们本地DNS服务器；
3 3、本地DNS有域名的配置，会直接发给浏览器；
4 4、若本地的DNS服务器也没有，会发往根，
5 本地DNS配置，hosts文件配置：
6 本地电脑C:\Windows\System32\drivers\etc
7 192.168.1.81 v1.cvc.net
8 192.168.1.81 v2.cvc.net
9 然后再通过浏览器去访问；
10 [root@cdeba90ec46e v2.cvc.net]# pkill nginx    停止nginx
```

此课程有nginx虚拟主机配置脚本PS: 52:49

<https://ke.qq.com/webcourse/2025906/102127284#taid=8522396233296306&vid=5285890801428670469>

---

Walter Savage Landor: strove with none, for none was worth my strife. Nature I loved and, next to Nature, Art: I  
warm'd both hands before the fire of life. It sinks, and I am ready to depart

——W.S.Landor

---