

一. 初识Nginx和环境准备

"Nginx是□款轻量级的HTTP服务器，采□事件驱动的异步□阻塞处理□式框架，这让其具有

极好的IO性能，时常□于服务端的反向代理和负载均衡。"

1,Nginx的优点

□持海量□并发：采□IO多路复□epoll。官□测试Nginx能够□持5万并发链接，实际□产环境中可以□撑2-4万并发连接数。

内存消耗少：在主流的服务器中Nginx□前是内存消耗最□的了，□如我们□Nginx+PHP，在3万并发链接下，开启10个Nginx进程消耗150M内存。

免费使□可以商业化：Nginx为开源软件，采□的是2-clause BSD-like协议，可以免费使□，并且可以□于商业。

配置□件简单：□络和程序配置通俗易懂，即使□专业运维也能看懂。当然它的有点还有很多，□如反向代理功能，负载均衡功能 现在Nginx□常□：我们可以通过<https://w3techs.com/> 这个□站看到，Nginx在服务器中排在第□位，但是是上升最快的□站，占有率达到39.7%。

2,学习环境

操作系统 CentOS 。

二. Nginx的快速搭建

1，Nginx版本说明

Mainline version：开发版,主要是给□□Nginx爱好者，测试、研究和学习的，但是不建议使□于□产环境。

Stable version：稳定版,也就是我们说的□期更新版本。这种版本□般□较成熟，经过□时间的更新测试，所以这种版本也是主流版本。

legacy version：历史版本，如果你需要以前的版本，Nginx也是有提供的。

2，基于Yum的□式安装Nginx

查看□下yum是否已经存在，命令如下：

```
1 yum list | grep nginx
```

如果出现类似下□的内容，说明yum源是存在的。

```
[root@instance-6vczgke8 ~]# yum list | grep nginx
nginx.x86_64 1:1.12.2-3.el7
nginx-all-modules.noarch 1:1.12.2-3.el7
nginx-filesystem.noarch 1:1.12.2-3.el7
nginx-mod-http-geoip.x86_64 1:1.12.2-3.el7
nginx-mod-http-image-filter.x86_64 1:1.12.2-3.el7
nginx-mod-http-perl.x86_64 1:1.12.2-3.el7
nginx-mod-http-xslt-filter.x86_64 1:1.12.2-3.el7
nginx-mod-mail.x86_64 1:1.12.2-3.el7
nginx-mod-stream.x86_64 1:1.12.2-3.el7
collectd-nginx.x86_64 5.8.1-1.el7
munin-nginx.noarch 2.0.49-3.el7
nextcloud-nginx.noarch 10.0.4-2.el7
owncloud-nginx.noarch 9.1.5-1.el7
pcp-pmda-nginx.x86_64 4.3.2-2.el7
python2-certbot-nginx.noarch 0.38.0-1.el7
sympa-nginx.x86_64 6.2.44-3.el7
```

如果都已经准备好了，那就可以开始安装了，安装的命令非常简单：

```
1 yum install nginx
```

安装完成后可以使⌘命令，来检测Nginx的版本。

```
1 nginx -v
```

如果出现下⌘图⌘的内容，说明Nginx就安装成功了。

```
[root@instance-6vczgke8 ~]# nginx -v
nginx version: nginx/1.12.2
[root@instance-6vczgke8 ~]#
```

到这⌘你就把最新版本的Nginx安装到了Linux系统中。

3. Nginx基本配置⌘件详讲

查看Nginx的安装⌘录 在使⌘yum安装完Nginx后，需要知道系统中多了那些⌘件，它们都安装到了那⌘。可以使⌘下⌘的命令进⌘查看：

```
1 rpm -ql nginx
```

rpm 是linux的rpm包管理⌘具，-q 代表询问模式，-l 代表返回列表，这样我们就可以找到nginx的所有安装位置了。列表列出的内容还是⌘较多的，我们尽量给⌘家进⌘讲解，我们这节先来看看⌘要

的文件。

4,nginx.conf文件解读

nginx.conf 文件是Nginx总配置文件，在我们搭建服务器时经常调整的文件。进入 **etc/nginx** 目录下，然后 **vim** 进入打开

```
1 cd /etc/nginx
```

```
1 vim nginx.conf
```

下面是文件的详细注释，根据你的需要进行配置。

```
1 #运行用户，默认即是nginx，可以不进行设置
2 user nginx;
3 #Nginx进程，一般设置为和CPU核数一样
4 worker_processes 1;
5 #错误日志存放目录
6 error_log /var/log/nginx/error.log warn;
7 #进程pid存放位置
8 pid /var/run/nginx.pid;
9 events {
10     worker_connections 1024; # 单个后台进程的最大并发数
11 }
12 http {
13     include /etc/nginx/mime.types; #文件扩展名与类型映射表
14     default_type application/octet-stream; #默认文件类型
15     #设置日志模式
16     log_format main '$remote_addr - $remote_user [$time_local]
17 "$request" '
18 '$status $body_bytes_sent "$http_referer" '
19 '"$http_user_agent" "$http_x_forwarded_for"';
20     access_log /var/log/nginx/access.log main; #nginx访问日志存放位置
21     sendfile on; #开启高效传输模式
22     #tcp_nopush on; #减少网络报段的数量
23     keepalive_timeout 65; #保持连接的时间，也叫超时时间
24     #gzip on; #开启gzip压缩
25     include /etc/nginx/conf.d/*.conf; #包含的子配置项位置和文件
```

```
1 server {
2     listen 80; #配置监听端口
3     server_name localhost; //配置域名
4     #charset koi8-r;
5     #access_log /var/log/nginx/host.access.log main;
```

```

6  location / {
7  root /usr/share/nginx/html; #服务默认启动目录
8  index index.html index.htm; #默认访问文件
9  }
10 #error_page 404 /404.html; # 配置404
11 # redirect server error pages to the static page /50x.html
12 #
13 error_page 500 502 503 504 /50x.html; #错误状态码的显示，配置后
14 需要重启
15 location = /50x.html html {
16 root /usr/share/nginx/html;
17 }

```

明白了这些配置项，知道我们的服务目录放在了/usr/share/nginx/html下，可以使用命令进入看下目录下的文件。

```
1 cd /usr/share/nginx/html
```

```
1 ls
```

可以看到目录下有两个文件，50x.html 和 index.html。我们可以使用vim进入编辑。现在nginx服务器已经可以为html提供服务器了。打开浏览器，访问ip地址试试。

5，退出文件夹编辑

```

1 ESC
2
3 :quit!

```

三、阿里云的安全组配置

如果你使用的是阿里云，记得到ECS实例下打开端口。

步骤如下：

- 进入阿里云控制台，并找到ECS实例。
- 点击实例后边的“更多”
- 点击“网络和安全组”，再点击“安全组配置”
- 右上角添加“安全组配置”
- 进入80端口的设置，具体设置如图就好。
- 到这我们就可以浏览到程序的页面了。

四：Nginx服务启动、停□、重启

启动Nginx服务。默认的情况下，Nginx是不会□动启动的，需要我们□动进□启动，当然启动Nginx

的□法也不是单□的。

1，nginx直接启动

在CentOS7.4版本□（低版本是不□的），是可以直接直接使□nginx启动服务的。

```
1 nginx
```

2，使□systemctl命令启动

还可以使□个Linux的命令进□启动：

```
1 systemctl start nginx.service
```

输□命令后，没有任何提示，那我们如何知道Nginx服务已经启动了哪？可以使□Linux的组合命令，进□查询服务的运□状况。

```
1 ps aux | grep nginx
```

如果启动成功会出现如下图□中类似的结果。

```
[root@instance-6vczgke8 ~]# ps aux | grep nginx
root      31968  0.0  0.2 125012  2124 ?        Ss   19:14   0:00 nginx: master p
rocess nginx
nginx     31969  0.0  0.3 127484  3580 ?        S    19:14   0:00 nginx: worker p
rocess
root      31971  0.0  0.0 112724   996 pts/1    R+   19:14   0:00 grep --color=au
to nginx
```

有这三条记录，说明我们Nginx被正常开启了。

3,停□Nginx服务的四种□法

停□Nginx □法有很多种，可以根据需求采□不□样的□法，我们□个□个说明。

- □即停□服务

```
1 nginx -s stop
```

这种□法□较强硬，□论进程是否在□作，都直接停□进程。

- 从容停□服务

```
1 nginx -ss quit
```

这种□法较stop相□就□较温和□些了，需要进程完成当前□作后再停□。

- killall □法杀死进程

这种□法也是□较野蛮的，我们直接杀死进程，但是在上

□使□没有效果时，我们□这种□法还是□较好的。

```
1 killall nginx
```

- systemctl 停□

```
1 systemctl stop nginx.service
```

4，重启Nginx服务

有时候我们需要重启Nginx服务，这时候可以使⌘下⌘的命令。

```
1 systemctl restart nginx.service
```

5，重新载⌘配置⌘件

在更新编写或者修改Nginx的配置⌘件后，都需要作⌘下更新载⌘，这时候可以⌘Nginx给的命令。

```
1 nginx -s reload
```

6，查看端⌘号

在默认情况下，Nginx启动后会监听80端⌘，从⌘提供HTTP访问，如果80端⌘已经被占⌘则会启动失败。我么可以使⌘

```
1 netstat -tlnp
```

命令查看端⌘号的占⌘情况。