Nginx

HTTP服务器，Nginx可以作为独立的Web服务器来托管网站和应用程序。

反向代理服务器：它可以将客户端请求转发给后端服务器，并将响应返回给客户端，从而实现负载均衡和提高系统的可用性。

负载均衡：通过在多个后端服务器之间分配流量，确保系统能够处理大量并发请求。

静态内容服务，Nginx擅长快速提供静态文件（如HTML页面、图片，CSS和JavaScript文件）。

缓存：支持对动态和静态内容进行缓存，以减少后端服务器的负载并加快响应速度。

SSL/TLS终止：可以处理SSL/TLS加密和解密，减轻后端服务器的负担，并提高安全性。

邮件代理：虽然较少使用，但Nginx也可以配置为邮件代理服务器。

原理

Nginx采用了事件驱动，event-driven，和异步非阻塞I/O模型，这使得它能够在处理大量并发连接时保持高效性能。以下是其核心机制的一些关键点：

事件驱动架构，Nginx使用一个主进程管理多个工作进程。每个工作进程负责监听网络事件，并在事件发生时处理请求。

异步非阻塞I/O，Nginx不会为每个连接创建新的进程或线程，而是使用异步I/O操作，这意味着它可以同时处理成千上万的并发连接。

模块化设计，Nginx的核心功能相对简单，大部分功能是通过模块实现的。这些模块可以根据需要加载或卸载，提供了高度的灵活性。

部署

#安装Nginx服务器

```
#卸载apache2
apt purge apache2 apache2-bin apache2-data apache2-utilsi
#安装Nginx环境依赖
apt -y install curl gnupg2 ca-certificates lsb-release debian-archive-
keyring
#安装
apt -y install nginx
#启动
systemctl start nginx
systemctl enable nginx

#使用浏览器访问localhost或ip访问成功
http://123.0.0.1/

apt -y install php php-fpm
grep '^listen =' /etc/php/*/fpm/pool.d/www.conf
```

```
#如果返回sock文件地址说明默认监听sock文件
listen = /run/php/php*-fpm.sock
#如果返回IP:port 说明默认监听本地IP端口
listen = 127.0.0.1:9000
# nginx的ffastcgi_pass `<listen>` 编写规则□sock必须加unix:
fastcgi_pass  unix:/run/php/php8.4-fpm.sock;
fastcgi_pass  127.0.0.1:9000;


#移除默认站点配置
sudo rm -f /etc/nginx/sites-enabled/*
#写入配置文件 http协议80端口转https协议443端口
sudo tee /etc/nginx/conf.d/default.conf <<-'EOF'
server {
    listen [::]:80; # IPv6
    listen 80;
    server_name localhost;
    # Redirect all HTTP requests to HTTPS with a 301 Moved Permanently
response.
    location / {
        return 301 https://$host$request_uri;
    }
}
EOF

sudo vi /etc/nginx/conf.d/default.conf
server {
    listen  [::]:433; # IPv6
    listen       433;
    server_name  localhost;
    root /var/www/html;
    location / {
            index index.php index.html index.htm;
    }
    location ~ .php$ {
            root /var/www/html;
            fastcgi_pass <listen>;
            fastcgi_index index.php;
            fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
            include fastcgi_params;
    }
    # error_page  404              /404.html;
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}

vi /etc/nginx/nginx.conf
#把include /etc/nginx/sites-enabled/* 注释
```

```
# include /etc/nginx/sites-enabled/*
按`esc`键，输入`:wq`退出

在终端执行命令
#配置是否正确
nginx -t
#查看有没有ok和successful
#nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
#nginx: configuration file /etc/nginx/nginx.conf test is successful
#重启
sudo systemctl restart nginx




#无法转跳网页
sudo vi /etc/nginx/nginx.conf
#在location /大括号内，添加如下代码
if (-f $request_filename/index.html){
rewrite (.*) $1/index.html break;
}
if (-f $request_filename/index.php){
rewrite (.*) $1/index.php;
}
if (!-f $request_filename){
rewrite (.*) /index.php;
}
#重启
sudo systemctl restart nginx
#默认网站根目录：
Nginx: /usr/share/nginx/html
Apache: /var/www/html

Nginx的日志文件默认存放在/var/log/nginx/
Nginx的主配置文件默认/etc/nginx/nginx.conf
Nginx默认读取/etc/nginx/conf.d目录下所有以.conf为后缀的附加配置文件
```

SSL

自生成测试

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout localhost.key \
    -out localhost.pem \
    -subj "/C=CN/ST=Beijing/L=Beijing/O=MyOrg/CN=localhost"
```

-x509生成自签名证书（不是 CSR

-nodes不加密私钥Nginx 启动时不需要输密码）

-days 365□有效期 365 天（可改）

-newkey rsa:2048□生成 2048 位 RSA 私钥

-keyout localhost.key□私钥输出文件

-out localhost.pem□证书输出文件□PEM 格式）

-subj "…"□自动填写证书信息（避免交互）

-subj 字段含义（按需修改）：

字段 含义 示例

C 国家（2字母□ CN

ST 省份 Beijing

L 城市 Dongcheng

O 组织/公司 MyCompany

CN 通用名称（必须匹配访问的域名或 IP□ localhost 或 127.0.0.1

```
chmod 600 localhost.key
chmod 644 localhost.pem
mkdir /etc/nginx/conf.d/cert
mv localhost.* /etc/nginx/conf.d/cert/
```

生产环境

登录数字证书管理服务控制台

在左侧导航栏，选择证书管理 > SSL证书管理

在个人测试证书（原免费证书）页签，单击立即购买

在立即购买面板，保持默认选项，仔细阅读并勾选服务协议，单击立即购买并完成支付

阿里云→数字证书管理服务→SSL证书管理→个人测试证书（原免费证书）→更多→下载

在终端执行命令

#查看nginx的配置文件路径，记录nginx.conf路径

nginx -t

#创建证书目录，命名为cert

mkdir /etc/nginx/conf.d/cert

#将证书文件和私钥文件上传到Nginx服务器的证书目录

mv localhost_nginx.zip /etc/nginx/conf.d/cert/

#解压

unzip /etc/nginx/conf.d/cert/localhost_nginx.zip

#配置

vi /etc/nginx/conf.d/default.conf

修改default.conf文件

```
listen 443 ssl;

#填写证书文件绝对路径
ssl_certificate /etc/nginx/conf.d/cert/localhost.pem;
#填写证书私钥文件绝对路径
ssl_certificate_key /etc/nginx/conf.d/cert/localhost.key;
ssl_session_cache shared:SSL:1m;
ssl_session_timeout 5m;
#自定义设置使用的TLS协议的类型以及加密套件（以下为配置示例，请您自行评估是否需要配置）
#TLS协议版本越高HTTPS通信的安全性越高，但是相较于低版本TLS协议，高版本TLS协议对浏览器的兼容性较差。
ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3;
#表示优先使用服务端加密套件。默认开启
ssl_prefer_server_ciphers on;
```

按`esc`键，输入`:wq`退出