Linux 配置 Nginx 服务完整详细版

前言

当你需要配置Nginx服务器来托管网站或应用程序时，以下是一些基本步骤和示例配置，以帮助你入门。请注意，Nginx的配置可以非常灵活，可以根据你的具体需求进行自定义。以下示例假设你已经在服务器上安装了Nginx。

1、打开终端并登录到你的服务器。

2、使用文本编辑器（比如**nano**或**vim**）打开Nginx配置文件。

配置文件通常位于 /etc/nginx/nginx.conf或 /etc/nginx/sites-available/default，具体位置可能因你的操作系统而异。以下是使用vim编辑器的示例：

```XML
sudo vim /etc/nginx/nginx.conf
```

配置Nginx监听端口和服务器块

在**nginx.conf**中，你可以找到一个名为**http**的块，其中包含Nginx的全局配置。你可以更改默认监听端口（默认为80）和添加服务器块。

XML

```
# HTTP模块配置段
http {

    # 防DDoS配置
    limit_req_zone $binary_remote_addr zone=ddos:10m rate=10r/s;

    # 日志配置
    access_log /var/log/nginx/access.log;

    # 设置服务器块
    server {
        listen 80; # 监听端口

        server_name example.com; # 域名

        location / {
            root /var/www/html; # 网站根目录
            index index.html; # 默认文件
        }

        location /static/ {
            alias /var/www/static/; # 静态文件目录
        }

        location /images/ {
            alias /var/www/images/; # 图像文件目录
        }

        # 自定义错误页面
        error_page 404 /404.html;
        location = /404.html {
        root /var/www/html;
        internal;
        }

        # 反向代理配置
        location /api/ {
            proxy_pass http://backend-server; # 后端服务器地址
        }

        # 配置SSL/TLS
        listen 443 ssl;
        server_name example.com;
        ssl_certificate /path/to/your/certificate.crt;
        ssl_certificate_key /path/to/your/private-key.key;
```

```
        # 配置SSL协议版本和密码套件
        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_prefer_server_ciphers off;
        ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384';

        # 配置SSL会话缓存
        ssl_session_cache shared:SSL:10m;
        ssl_session_timeout 10m;

        # 启用HSTS标头，告诉浏览器始终使用HTTPS
        add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload";

        # 防止点击劫持
        add_header X-Frame-Options SAMEORIGIN;

        # 安全头部配置
        add_header X-Content-Type-Options "nosniff";
        add_header X-XSS-Protection "1; mode=block";
        add_header X-Frame-Options "SAMEORIGIN";

    }
}
```

防DDoS配置

limit_req_zone: 这是一个Nginx指令，用于定义一个请求限制区域。这个区域用来记录每个客户端的请求频率信息。

$binary_remote_addr: 这是一个Nginx变量，表示客户端的IP地址。每个不同的IP地址都会被视为一个单独的客户端。

zone=ddos:10m: 这个部分定义了请求限制区域的名称为"ddos"，并分配了10兆字节的内存空间（10m）来存储相关数据。

rate=10r/s: 这部分规定了请求速率限制。它表示每个IP地址最多允许发送10个请求每秒（10r/s）。

日志配置

access_log 是 Nginx 配置指令，用于定义访问日志的设置。

/var/log/nginx/access.log 是日志文件的路径，它告诉 Nginx 将访问日志写入到名为 access.log 的文件中。通常，Nginx 日志文件会放在 /var/log/nginx/ 目录下。

设置服务器块

监听端口

监听端口是指在计算机网络中，一台计算机或网络设备通过指定一个特定的网络端口号来等待和接收传入的网络连接或数据流。每个网络服务或应用程序可以监听一个或多个端口，这些端口用于标识不同的网络服务或通信通道。

网站根目录

网站根目录（也称为网站根文件夹、网站根文件目录或网站根目录结构）是一个Web服务器上的主要文件夹，它包含了构成整个网站的文件和资源。这个目录通常是Web服务器用来提供网站内容的起点，也是访问网站时的默认基础路径。

默认文件

当你访问一个网站时，通常会看到网站的首页或默认页面。这个默认页面被称为索引文件，它是网站的第一个展示给访问者的页面。

静态文件目录

静态文件目录是一个包含网站的静态文件（不需要服务器端处理的文件）的文件夹或目录。这些静态文件可以包括HTML、CSS、JavaScript、图像、字体文件等，它们不需要在服务器端动态生成或处理，而是直接提供给客户端浏览器。

图像文件目录

图像文件目录是一个用于存储网站或应用程序中的图像文件的文件夹或目录。这些图像文件可以包括各种图像类型，例如JPEG、PNG、GIF、SVG等。图像文件目录通常用于组织和管理网站中的图像资源，使其能够在网页上展示或通过链接提供给用户。

自定义错误页面

这个配置告诉Nginx当发生404错误时，将用户重定向到/404.html页面。location块内的root指令定义了404页面所在的目录，这里是/var/www/html。internal指令用于限制该location仅在Nginx内部处理，不会向外部暴露这个页面的路径。

反向代理配置

location /api/ { ... }: 这是一个location块，指定了要处理以/api/开头的URL路径的请求。只有满足这个条件的请求会进入这个location块中进行处理。

proxy_pass http://backend-server;: 这是配置块中最重要的部分。它指定了Nginx应该将请求转发到的后端服务器的地址。

例如，如果你的后端服务器的地址是http://localhost:8000，那么这里应该写成 proxy_pass http://localhost:8000;。

配置SSL/TLS

1、获取SSL/TLS证书

首先，您需要获取SSL/TLS证书。您可以从权威的证书颁发机构（如Let's Encrypt、Comodo、DigiCert等）购买证书，或者使用自签名证书。自签名证书适用于测试和开发环境，但在生产环境中，建议使用受信任的证书颁发机构颁发的证书，以确保浏览器和客户端的兼容性。

2、安装证书

获得证书后，需要将其安装到服务器上。通常，证书文件包括一个公钥文件（通常以.crt或.pem为扩展名）和一个私钥文件（通常以.key为扩展名）。将这些文件存储在服务器上的安全位置。

3、配置SSL/TLS

在配置文件中，找到与SSL/TLS相关的部分，在Nginx中，通常是在server块内配置SSL。

server { ... }: 这是一个Nginx服务器块，用于定义服务器的配置。

listen 443 ssl;: 这一行指定服务器监听的端口是443，并启用SSL加密。所有传入的HTTPS请求都将在这个端口上被处理。

server_name example.com;: 这里定义了服务器的域名。

ssl_certificate /path/to/your/certificate.crt; 这行指定了SSL证书的路径，该证书用于加密传输的数据。

ssl_certificate_key /path/to/your/private-key.key;: 这行指定了SSL私钥文件的路径，用于解密传入的加密数据。

配置SSL协议版本和密码套件

配置SSL协议版本和密码套件通常不需要更改为自己的，因为这部分配置是针对服务器的安全性和性能进行优化的。

ssl_protocols 指定了支持的TLS版本，通常TLSv1.2和TLSv1.3是安全的选择，无需更改，除非你有特定的需求。

ssl_prefer_server_ciphers 设置为 **off** 以确保Nginx不会强制使用服务器端密码套件的顺序，通常无需更改。

ssl_ciphers 定义了支持的密码套件，使用ECDHE（椭圆曲线Diffie-Hellman Ephemeral）密钥交换和AES-GCM模式，通常无需更改。

配置SSL会话缓存

这两行配置是用于配置SSL会话缓存的设置，它们对于提高服务器的SSL/TLS性能非常重要。让我解释它们的含义：

ssl_session_cache shared:SSL:10m;：这行配置指定了SSL会话缓存的类型、名称和大小。

10m：这部分指定了会话缓存的大小。在示例中，缓存的大小被设置为10兆字节（MB）。这意味着服务器可以存储大约10兆字节的SSL会话数据。

ssl_session_timeout 10m;：这行配置指定了SSL会话在缓存中的超时时间。

10m：这部分指定了会话的超时时间，与上面的缓存大小相对应。在示例中，会话将在10分钟后过期并从缓存中删除。

启用HSTS标头，告诉浏览器始终使用HTTPS

max-age=31536000：指定了HSTS策略的持续时间，以秒为单位。在这里，**max-age** 被设置为31536000秒，等于一年的时间。这意味着一旦浏览器接收到这个HSTS标头，它将在一年内记住你的网站，并强制使用HTTPS连接访问。

防止点击劫持

这个配置的目的是增强网站的安全性，防止点击劫持攻击，其中攻击者将您的网页嵌套到他们的恶意网站中，以欺骗用户。通过设置X-Frame-Options为SAMEORIGIN，您告诉浏览器只允许您的网页在相同的源内被嵌套，从而提高了您的网站的安全性

安全头部配置

1、X-Content-Type-Options "nosniff"：

X-Content-Type-Options 头部用于控制浏览器是否应该执行MIME类型嗅探。

"nosniff" 指令告诉浏览器不要执行嗅探，即使服务器返回的响应中包含了不一致的MIME类型信息，浏览器也不会尝试猜测响应的内容类型。

这有助于防止MIME类型混淆攻击，其中攻击者可能会在响应中注入恶意内容，并依赖浏览器错误地解释响应的MIME类型。

2、X-XSS-Protection "1; mode=block"：

X-XSS-Protection 头部用于启用浏览器内置的跨站点脚本（XSS）过滤器。

"1; mode=block" 指令启用了XSS过滤器，并在检测到潜在XSS攻击时，将页面设置为阻止加载。

这有助于防止XSS攻击，其中攻击者尝试在网页中注入恶意脚本以执行恶意操作，如窃取用户信息或劫持用户会话。

3、X-Frame-Options "SAMEORIGIN"：

X-Frame-Options 头部用于控制是否允许将网页嵌入到 <iframe> 中。

"SAMEORIGIN" 指令表示只允许网页在与原始网页相同的域名下嵌套到 <iframe> 中。

这有助于防止点击劫持攻击，其中攻击者可能会尝试将您的网站嵌入到恶意站点中，以欺骗用户进行操作或窃取信息。