

nginx

Web 服务器

web 服务器 http 服务器 http 服务器 / 静态 (html/css/图片) 文件
支持 http 协议 apache(httpd) Nginx IIS Lighttpd 等
(Java\PHP\Python\go), 数据库 MySQL Oracle Opengauss
Tomcat\Weblogic\Jboss

Nginx 简介

nginx 是一个 web 服务器，也可以作为反向代理，支持 http 协议
1. 静态文件 2. 动态文件 CGI FastCGI Perl PHP
支持 html/静态文件 / 动态文件 CGI FastCGI Perl PHP
支持 java 等语言

特点

支持 http/https 协议 tcp/udp 协议
支持 静态文件 http 代理 反向代理
支持 静态文件 http 代理 反向代理
nginx 是一个 master 进程，nginx.conf 文件 worker 进程 worker
进程 CPU 进程 worker 进程

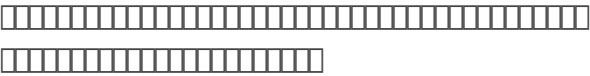
Master → 1. listenfd → fork() → Worker1, Worker2, Worker3... 1. worker
2. worker
tcp 连接

Nginx 配置

worker 进程 master 进程 lua 进程 master+worker
worker 进程 master 进程 5 进程 2.5 进程
2-3 进程 timeout 进程 \ 进程



[<https://nginx.org/en/download.html>]



mainline version [\[link\]](#) stable version [\[link\]](#)
[\[link\]](#) bug [\[link\]](#)
[\[link\]](#)

nginx plus [\[link\]](#)

tengeine [\[link\]](#) openresty nginx [\[link\]](#) nginx+lua [\[link\]](#) Web [\[link\]](#) lua [\[link\]](#)
apisix [\[link\]](#)

[[Nginx]]

[\[link\]](#) yum [\[link\]](#) gcc [\[link\]](#)
autoconf [\[link\]](#) makefile [\[link\]](#) pcre [\[link\]](#) zlib [\[link\]](#) openssl
[\[link\]](#) ssl [\[link\]](#)



1. [[yum]]

```
sed -e 's|^mirrorlist=|#mirrorlist=|g' \  
-e \  
's|^#baseurl=http://dl.rockylinux.org/$contentdir|baseurl=https://mirrors.aliyun.com/rockylinu \  
x|g' \  
-i.bak \  
/etc/yum.repos.d/Rocky-*.repo  
  
dnf makecache
```

2. [[[[link]]]]

```
dnf install -y gcc gcc-c++ make libtool wget pcre pcre-devel zlib zlib-devel openssl openssl-devel
```

3. 创建nginx用户

```
useradd -s /sbin/nologin nginx -M
```

4. 下载nginx源码

```
mkdir -p /opt/soft && cd /opt/soft
wget https://nginx.org/download/nginx-1.24.0.tar.gz
tar -zxvf nginx-1.24.0.tar.gz
cd nginx-1.24.0
```

5. 编译安装

```
./configure \
--prefix=/usr/local/nginx-1.24.0 \
--user=nginx \
--group=nginx \
--with-http_ssl_module \
--with-http_v2_module \
--with-http_realip_module \
--with-http_stub_status_module \
--with-http_gzip_static_module \
--with-pcre \
--with-stream \
--with-stream_ssl_module \
--with-stream_realip_module
```

```
./configure \
--prefix=/usr/local/nginx-1.24.0 # 安装路径
```

```
--user=nginx # nginx用户 --group=nginx # nginx组 --with-http_ssl_module # 支持ssl --with-http_v2_module # 支持http2 --with-http_realip_module # 支持realip --with-http_stub_status_module # 支持stub status --with-http_gzip_static_module # 支持gzip --with-pcre # 支持pcre --with-stream # 支持stream --with-stream_ssl_module # 支持stream ssl --with-stream_realip_module # 支持stream realip
```

6. 编译

```
echo $(nproc) # 查看CPU数量
make -j$(nproc) #并行编译
make install
```

7. 创建软链接

```
ln -s /usr/local/nginx-1.24.0 /usr/local/nginx
ln -s /usr/local/nginx-1.24.0/sbin/nginx /usr/local/bin/nginx
```

8. 启动

```
/usr/local/nginx-1.24.0/sbin/nginx -V
/usr/local/bin/nginx -V
nginx -V
nginx -t
/usr/local/nginx/sbin/nginx -t
/usr/local/nginx/sbin/nginx

ss -ant
ps aux | grep nginx
curl -I http://localhost/
```

9. 配置 systemd

```
cat>/etc/systemd/system/nginx.service<<'EOF'
[Unit]
Description=nginx
After=network.target

[Service]
Type=forking
PIDFile=/usr/local/nginx-1.24.0/logs/nginx.pid
ExecStartPre=/usr/local/nginx-1.24.0/sbin/nginx -t -c /usr/local/nginx-1.24.0/conf/nginx.conf
ExecStart=/usr/local/nginx-1.24.0/sbin/nginx -c /usr/local/nginx-1.24.0/conf/nginx.conf
ExecReload=/usr/local/nginx-1.24.0/sbin/nginx -s reload
ExecStop=/usr/local/nginx-1.24.0/sbin/nginx -s stop
```

```
PrivateTmp=true

[Install]
WantedBy=multi-user.target
EOF

systemctl daemon-reload
```

10. nginx systemd 启动

```
/usr/local/nginx/sbin/nginx -s stop
systemctl start nginx
systemctl status nginx --no-pager
systemctl stop nginx
systemctl restart nginx
systemctl enable nginx
```

11. nginx systemd 日志

```
systemctl status nginx -l
journalctl -u nginx -xe

### yum 安装
dnf install nginx # 或者 dnf install nginx8 dnf yum install nginx8 yum dnf
```

dnf 安装 nginx, 启动

启动

```
cat>/etc/yum.repos.d/nginx.repo<<'EOF'
[nginx-stable]
name=nginxstablerepo
baseurl=http://nginx.org/packages/centos/8/$basearch/
gpgcheck=0
enabled=1
gpgkey=https://nginx.org/packages/keys/nginx_signing.key module_hotfixes=true
EOF

dnf install -y nginx # 安装 nginx

dnf list| grep nginx # 查看 nginx 版本
```

```
dnf module list nginx #nginx
dnf module reset nginx #nginx
dnf module enable nginx #nginx
```

docker

[<https://developer.aliyun.com/mirror/>] [<https://download.docker.com>]

1. yum-utils & 安装

```
dnf install -y yum-utils
```

2. docker-ce

```
sudo yum-config-manager --add-repo https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
sed -i 's+download.docker.com+repo.huaweicloud.com/docker-ce+g' /etc/yum.repos.d/docker-ce.repo
```

3. docker-ce --showduplicates

```
dnf list docker-ce --showduplicates | sort -r
```

4. docker-ce-24.0.9-1.el8 docker-ce-cli-24.0.9-1.el8 containerd.io

```
dnf clean all
dnf makecache
sudo yum install -y docker-ce-24.0.9-1.el8 docker-ce-cli-24.0.9-1.el8 containerd.io
```

5. 卸载 docker-ce

```
yum remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-engine
```

6. 安装

```
docker version
docker info
```

7. 配置 docker 镜像源 (国内)

```
cat > /etc/docker/daemon.json <<EOF
{
    "registry-mirrors": [
        "https://docker.m.daocloud.io",
        "https://dockerproxy.com",
        "https://docker.mirrors.ustc.edu.cn",
        "https://docker.nju.edu.cn",
        "https://ijyu9kaj2.mirror.aliyuncs.com",
        "https://hub-mirror.c.163.com"
    ]
}
EOF
```

8. 配置 docker 守护进程

```
systemctl daemon-reload # 重载systemd配置
systemctl start docker
systemctl stop docker
systemctl enable --now docker
systemctl status docker --no-pager
systemctl restart docker
```

9. 配置 docker 容器

```
docker run hello-world
docker images
```



nginx

1. 安装 nginx

```
docker pull nginx:1.24.0
```

2. 启动 nginx:1.24.0

```
docker run -d --name my_nginx1.24.0 -p 8010:80 nginx:1.24.0
```

3. 验证是否启动成功

```
curl http://localhost:8010/  
ss -ant | grep 8010 #查看端口是否监听  
docker ps #查看容器  
docker images #查看镜像
```

4. 启动 nginx:1.28.0

```
docker pull nginx:1.28.0
```

5. 删除 nginx:1.24.0 容器和镜像

```
#查看容器  
docker stop my_nginx1.24.0  
#查看容器ID  
docker rm my_nginx1.24.0  
docker rm CONTAINER ID/NAME  
#查看镜像  
docker rmi REPOSITORY/IMAGE ID
```

6. 创建 nginx-docker 目录

```
mkdir -p /data/nginx-docker/{html,conf,logs,ssl}  
mkdir -p /data/nginx-docker/conf/conf.d
```

7. 复制 nginx.conf 文件

```

cat > /data/nginx-docker/conf/nginx.conf <<'EOF'
user  nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log  notice;
pid        /var/run/nginx.pid;
events {
    worker_connections  1024;
}
http {
    include      /etc/nginx/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  /var/log/nginx/access.log  main;
    sendfile on;
    #tcp_nopush on;

    keepalive_timeout  65;

    #gzip on;
    include /etc/nginx/conf.d/*.conf;
}
EOF

```

8.

```

cat > /data/nginx-docker/conf/conf.d/default.conf <<'EOF'
server {
    listen 80;
    listen [::]:80;
    server_name  localhost;

    location / {
        root    /usr/share/nginx/html;
        index  index.html index.htm;
    }
}
EOF

```

```
}  
}  
EOF
```

9. 创建 index.html

```
cat > /data/nginx-docker/html/index.html << EOF  
<meta charset="UTF-8">  
<style>  
    body {  
        background-color: #000000;  
        color: #9900cc;  
    }  
</style>  
<h1>nginx-docker</h1>  
EOF
```

10. 安装 nginx 1.28

```
ss -ant | grep 80 # 检查端口  
  
docker run -d \  
--name my-nginx1.28 \  
-p 80:80 \  
-p 443:443 \  
-v /data/nginx-docker/conf/nginx.conf:/etc/nginx/nginx.conf \  
-v /data/nginx-docker/conf/conf.d:/etc/nginx/conf.d \  
-v /data/nginx-docker/logs:/var/log/nginx \  
-v /data/nginx-docker/html:/usr/share/nginx/html \  
-v /data/nginx-docker/conf/ssl:/etc/nginx/ssl \  
nginx:1.28.1
```

11. 清理环境

```
docker ps -a  
docker images  
docker rm my-nginx1.28
```

```
ps -ef | grep nginx
curl http://localhost:80/
ss -ant | grep 80
```

编译 nginx

Nginx 编译选项示意图：
 1. 编译选项 2. 编译选项 3. 编译选项 4. 编译选项

```
./configure \
--add-module=/path/to/third-party/module \
--add-dynamic-module=/path/to/dynamic/module
```

编译选项 1. 编译选项 2. 编译选项 3. 编译选项
 --add-dynamic-module --without-xxx-module --with-xxx-module

编译选项

```
./configure --prefix=/usr/local/nginx-1.24.0-2 --user=nginx --group=nginx \
--with-http_ssl_module --with-http_v2_module --with-http_realip_module \
--with-http_stub_status_module --with-http_gzip_static_module \
--with-pcre --with-stream --with-stream_ssl_module \
--with-stream_realip_module --with-compat \
--add-dynamic-module=/opt/soft/nginx-rtmp-module-master
```

```
make -j 4
make install
```

```
vi nginx-1.24.0-2/conf/nginx.conf
load_module modules/nginx_rtmp_module.so;
```

```
/usr/local/nginx-1.24.0-2/sbin/nginx
```

```
dnf install -y lsof
lsof -p $(cat /usr/local/nginx-1.24.0-2/logs/nginx.pid) | grep \\.so
```



nginx



Ctrl + C TERM/INT stop - Nginx
QUIT quit - HUP reload
- USR1 reopen - USR2 upgrade
- WINCH winch USR2

1. nginx

```
/usr/local/nginx-1.24.0-2/sbin/nginx -s stop  
kill -TERM $(cat /usr/local/nginx-1.24.0-2/logs/nginx.pid)
```

2.

```
/usr/local/nginx-1.24.0-2/sbin/nginx -s quit  
kill -QUIT $(cat /usr/local/nginx-1.24.0-2/logs/nginx.pid)
```

 --> --> --> -->

3.

```
/usr/local/nginx-1.24.0-2/sbin/nginx -t  
/usr/local/nginx-1.24.0-2/sbin/nginx
```

4.

```
/usr/local/nginx-1.24.0-2/sbin/nginx -s reload  
kill -HUP $(cat /usr/local/nginx-1.24.0-2/logs/nginx.pid)
```

5.

```
kill -USR1 $(cat /usr/local/nginx-1.24.0-2/logs/nginx.pid)
```

```
### nginx
```

```
/usr/local/nginx/sbin/nginx -t  
/usr/local/nginx/sbin/nginx
```

1. 检查

```
/usr/local/nginx/sbin/nginx -v
```

2. 检查

```
/usr/local/nginx/sbin/nginx -V
```

3. 备份

```
cp -aR /usr/local/nginx /usr/local/nginx.bak
```

4. 安装 nginx

```
wget https://nginx.org/download/nginx-1.25.0.tar.gz
```

5. 编译 nginx

```
tar -zxvf nginx-1.25.0.tar.gz
cd nginx-1.25.0

./configure \
--prefix=/usr/local/nginx-1.24.0 \
--user=nginx \
--group=nginx \
--with-http_ssl_module \
--with-http_v2_module \
--with-http_realip_module \
--with-http_stub_status_module \
--with-http_gzip_static_module \
--with-pcre \
--with-stream \
--with-stream_ssl_module \
--with-stream_realip_module
make -j 4

cp -f objs/nginx /usr/local/nginx/sbin/nginx

/usr/local/nginx/sbin/nginx -V
```

6. 安装 nginx

```
ss -ant | grep :80
ps aux | grep nginx | grep -v grep
pid
kill -USR2 $(cat /usr/local/nginx/logs/nginx.pid)
ps aux | grep nginx | grep -v grep
kill -WINCH
kill -QUIT
ps aux | grep nginx | grep -v grep

ps -ef | grep nginx | grep -v grep

### 安装nginx
dnf install tree -y
```

1. nginx 配置 fastcgi_params # CGI 配置 mime.types # 配置 html # 配置 logs # 配置 pid # 配置 ID 配置 sbin # 配置 conf # 配置 modules # 配置 obj # 配置 nginx.conf # 配置 配置 events # 配置 http # HTTP 配置 server # 配置 location # 配置
2. location 配置 (配置)

```
location [配置] url { 配置 }
https://www.linuxnote.com:443/path/to/page?name=value
```

= 配置 location = /path/to/page ^~ 配置 location ^~ /path/to/page

- 配置 (配置) location ~ /path/to/page ~* 配置 (配置) location ~* /path/to/page 配置 配置 location /path/to/page / 配置 location /

配置 nginx 配置

配置 ip

配置 ip 配置

```
ip a
nmcli con mod ens160 +ipv4.address 192.168.20.90/24
nmcli con up ens160
```

```
mkdir -p /usr/local/nginx-1.24.0/conf/vhost
```

```
mkdir -p /data/www/{151,90}
```

```
cat > /usr/local/nginx-1.24.0/conf/vhost/151.conf <<EOF
```

```
server {  
    listen 80;  
    server_name 192.168.20.151;  
    location / {  
        root /data/www/151;  
        index index.html;  
    }  
}  
EOF
```

```
cat > /usr/local/nginx-1.24.0/conf/vhost/90.conf <<EOF
```

```
server {  
    listen 80;  
    server_name 192.168.20.90;  
    location / {  
        root /data/www/90;  
        index index.html;  
    }  
}  
EOF
```

```
sed -i '$i include /usr/local/nginx-1.24.0/conf/vhost/*.conf;' /usr/local/nginx-  
1.24.0/conf/nginx.conf
```

```
cat > /data/www/151/index.html <<EOF
```

```
<meta charset="utf-8">
```

```
<h1>151</h1>
```

```
EOF
```

```
cat > /data/www/90/index.html <<EOF
```

```
<meta charset="utf-8">
```

```
<h1>90</h1>
```

```
EOF
```

```
nginx -t
```

```
nginx -s reload
```

□□□□

```
http://192.168.20.151
```

```
http://192.168.20.90
```

□□□

```
curl 192.168.20.151
```

```
curl 192.168.20.90
```

□□□□□

□□ ip □□□ □□□

```
mkdir -p /data/www/{2000,3000}
```

```
cat > /usr/local/nginx-1.24.0/conf/vhost/2000.conf <<EOF
```

```
server {  
    listen 2000;  
    server_name 192.168.20.151;  
    location / {  
        root /data/www/2000;  
        index index.html;  
    }  
}  
EOF
```

```
cat > /usr/local/nginx-1.24.0/conf/vhost/3000.conf <<EOF
```

```
server {  
    listen 3000;  
    server_name 192.168.20.151;  
    location / {  
        root /data/www/3000;  
        index index.html;  
    }  
}  
EOF
```

```
cat > /data/www/2000/index.html <<EOF
<meta charset="utf-8">
<h1>2000</h1>
EOF
```

```
cat > /data/www/3000/index.html <<EOF
<meta charset="utf-8">
<h1>3000</h1>
EOF
```

```
nginx -t
nginx -s reload
```

2000

```
http://192.168.20.151:2000
http://192.168.20.151:3000
```

3000

```
curl 192.168.20.151:2000
curl 192.168.20.151:3000
```

2000

ip 2000

```
mkdir -p /data/www/{web1,web2}

cat > /usr/local/nginx-1.24.0/conf/vhost/web1.conf <<EOF
server {
    listen 80;
    server_name www.linuxnote1.com;
    location / {
        root /data/www/web1;
        index index.html;
    }
}
```

```

}
EOF

cat > /usr/local/nginx-1.24.0/conf/vhost/web2.conf <<EOF
server {
    listen 80;
    server_name www.linuxnote2.com;
    location / {
        root /data/www/web2;
        index index.html;
    }
}
EOF

cat > /data/www/web1/index.html <<EOF
<meta charset="utf-8">
<h1>🐧web1</h1>
EOF

cat > /data/www/web2/index.html <<EOF
<meta charset="utf-8">
<h1>🐧web2</h1>
EOF

nginx -t
nginx -s reload

echo 192.168.20.151    www.linuxnote1.com >> /etc/hosts
echo 192.168.20.151    www.linuxnote2.com >> /etc/hosts

```

🐧

```

curl www.linuxnote1.com
curl www.linuxnote2.com

```

window C🐧

```
C:\Windows\System32\drivers\etc\hosts
192.168.20.151 www.linuxnote1.com
192.168.20.151 www.linuxnote2.com
```

□□□□

```
http://www.linuxnote1.com
http://www.linuxnote2.com
```

□□□ Server Name

□□□□ :□ " *.□□ " □□□□□□□□

```
mkdir -p /data/www/{www_linuxnote3.com,all_linuxnote3.com}

cat > /usr/local/nginx-1.24.0/conf/vhost/www_linuxnote3.com.conf <<EOF
server {
    listen 80;
    server_name www.linuxnote3.com;
    location / {
        root /data/www/www_linuxnote3.com;
        index index.html;
    }
}
EOF

cat > /usr/local/nginx-1.24.0/conf/vhost/all_linuxnote3.com.conf <<EOF
server {
    listen 80;
    server_name *.linuxnote3.com;
    location / {
        root /data/www/all_linuxnote3.com;
        index index.html;
    }
}
EOF

cat > /data/www/www_linuxnote3.com/index.html <<EOF
```

```
<meta charset="utf-8">
<h1>www_linuxnote3.com</h1>
EOF
```

```
cat > /data/www/all_linuxnote3.com/index.html <<EOF
<meta charset="utf-8">
<h1>all_linuxnote3.com</h1>
EOF
```

```
nginx -t
nginx -s reload
```

```
echo 192.168.20.151 www_linuxnote3.com >> /etc/hosts
echo 192.168.20.151 aaa_linuxnote3.com >> /etc/hosts
echo 192.168.20.151 all_linuxnote3.com >> /etc/hosts
```

```
curl www_linuxnote3.com
curl aaa_linuxnote3.com
curl all_linuxnote3.com
```

window C

```
C:\Windows\System32\drivers\etc\hosts
192.168.20.151 www_linuxnote3.com
192.168.20.151 aaa_linuxnote3.com
192.168.20.151 all_linuxnote3.com
```

```
http://www_linuxnote3.com
http://aaa_linuxnote3.com
http://all_linuxnote3.com
```

nginx

1.

```
nginx -V
```

2.

```
/opt/soft/nginx-1.24.0/auto/options
```

--with

--without

```
studtax: stub_status on
```

```
default: -
```

```
context: server.location
```

```
cat > /usr/local/nginx-1.24.0/conf/vhost/studtax.conf << EOF
```

```
server {
```

```
    listen 80 ;
```

```
    server_name 192.168.20.151;
```

```
    location / {
```

```
        root /data/www/all_linuxnote3.com;
```

```
        index index.html;
```

```
    }
```

```
    location /qianshan {
```

```
        stub_status;
```

```
        access_log off;
```

```
    }
```

```
}
```

```
EOF
```

```
cat > /data/www/all_linuxnote3.com/index.html << EOF
```

```
<meta charset="UTF-8">
```

```
<h1>all.linuxnote3.com</h1>
```

```
EOF
```

```
nginx -t
```

```
nginx -s reload
```

```
192.168.20.151
192.168.20.151/qianshan
curl http://192.168.20.151
curl http://192.168.20.151/qianshan
```

```
Active connections: 1
server accepts handled requests
 46 46 44
Reading: 0 Writing: 1 Waiting: 0
```

accepts #	Nginx	handled #	Nginx
		accepts	Reading #
		, ,	, Writing #
		, ,	Waiting #

web

```
syntax auth_basic [ text|off ]
default: auth_basic off
auth_basic_user_file file_path
context: http, server, location, limit_except

cat > /usr/local/nginx-1.24.0/conf/vhost/studtax.conf << EOF
server {
    listen 80 ;
    server_name 192.168.20.151;
    location / {
        root /data/www/all_linuxnote3.com;
        index index.html;
    }
    location /qianshan {
        stub_status;
        auth_basic "secret";
        auth_basic_user_file /usr/local/nginx-1.24.0/conf/vhost/nginx-passwd.db;
        access_log off;
```

```

    }
}
EOF

dnf install -y httpd-tools
htpasswd -c /usr/local/nginx-1.24.0/conf/vhost/nginx-passwd.db aaa
chmod 400 /usr/local/nginx-1.24.0/conf/vhost/nginx-passwd.db
chown nginx.nginx /usr/local/nginx-1.24.0/conf/vhost/nginx-passwd.db

nginx -t
nginx -s reload

http://192.168.20.151/qianshan

```



```

HttpAccess[] :[] IP[] []
[] deny IP/IP [] IP [] IP [] allow IP/IP [] IP [] IP
[] all[] IP[] x.x.x.x/24

```

```

cat > /usr/local/nginx-1.24.0/conf/vhost/studtax.conf << EOF
server {
    listen 80 ;
    server_name 192.168.20.151;
    location / {
        root /data/www/all_linuxnote3.com;
        index index.html;
    }
    location /qianshan {
        stub_status;
        auth_basic "secret[]";
        auth_basic_user_file /usr/local/nginx-1.24.0/conf/vhost/nginx-passwd.db;
        allow 192.168.20.0/24;
        allow 124.221.251.28;
        deny all;
        access_log off;
    }
}

```

```
E0F

http://192.168.20.151/qianshan
curl http://192.168.20.151/qianshan -u aaa:123456
curl -L http://aaa:123456@192.168.20.151/qianshan
curl ifconfig.me
```

EOF

```
nginx -t
nginx -s reload

http://192.168.20.151/qianshan

#### 配置nginx
ngx_http_proxy_connect_module-master.zip
unzip ngx_http_proxy_connect_module-master.zip

cd /opt/soft/nginx-1.24.0
make clean

dnf install -y patch
patch -p1 < /opt/ngx_http_proxy_connect_module-master/patch/proxy_connect_rewrite_102101.patch

./configure \
--prefix=/usr/local/nginx-1.24.0 \
--user=nginx \
--group=nginx \
--with-http_ssl_module \
--with-http_v2_module \
--with-http_realip_module \
--with-http_stub_status_module \
--with-http_gzip_static_module \
--with-pcre \
--with-stream \
--with-stream_ssl_module \
--with-stream_realip_module \
--add-module=/opt/ngx_http_proxy_connect_module-master

make -j 4
make install

kill -QUIT $(pid)
systemctl restart nginx
ps -ef | grep nginx
nginx -v
```

```

cat >/usr/local/nginx-1.24.0/conf/vhost/zproxy.conf<<'EOB'
server {
listen 10000;
resolver 114.114.114.114;#DNS
server_name localhost;
proxy_connect; # proxy_connect CONNECT
proxy_connect_allow 443 80; #httphttps
proxy_connect_connect_timeout 10s;
proxy_connect_read_timeout 10s;
proxy_connect_send_timeout 10s;
location / {
proxy_pass $scheme://$host$request_uri; #
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_buffers 256 4k;
proxy_max_temp_file_size 0k;
proxy_connect_timeout 30;
proxy_send_timeout 60;
proxy_read_timeout 60;
proxy_next_upstream error timeout invalid_header http_502;
}
access_log logs/proxy_access.log;
error_log logs/proxy_error.log;
}
EOB

nginx -t
nginx -s reload
ss -ant

ping www.baidu.com
ip route del default
ip route add default via 192.168.20.10 dev ens160

```

```
curl --proxy 192.168.20.151:10000 https://www.qq.com
curl --proxy 192.168.20.151:10000 https://www.163.com
curl --proxy 192.168.20.151:10000 https://www.baidu.com
```

```
dnf install -y telnet
telnet 192.168.20.151 10000
```



1. 负载均衡 Load Balancing

负载均衡的概述

2. Nginx 负载均衡 支持 多种 负载均衡 策略 HTTP/HTTPS 负载均衡 TCP/UDP 负载均衡 IP 负载均衡
- Web 应用 API 应用 SSH MySQL

3. 负载均衡 Round Robin

负载均衡的概述

负载均衡的部署

```
upstream backend_servers {
server 172.22.4.203:8080;
server 172.22.4.204:8080;
server 172.22.4.205:8080;
}
```

4. 负载均衡 Weight 负载均衡的部署 负载均衡的测试
- least_conn ip_hash 负载均衡的部署

```
upstream backend_servers {
```

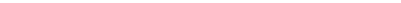
```
server 172.22.4.203:8080 weight=50; # 负载均衡的部署
# 负载均衡的部署 server 172.22.4.205:8080 weight=2; # 负载均衡的部署
server 172.22.4.204:8080 weight=3;
```

```
}
```

5. IP 负载均衡 ip_hash 负载均衡的部署 IP 负载均衡的部署 负载均衡的测试
- Session 负载均衡 backup 负载均衡的部署 down

```
upstream backend_servers {
    ip_hash;
    server 172.22.4.203:8080;
    server 172.22.4.204:8080;
```

```
server 172.22.4.205:8080 down; # ☐☐☐☐☐
```

6. 

```
upstream backend_servers {
    least_conn;
    server 172.22.4.203:8080;
    server 172.22.4.204:8080;
    server 172.22.4.205:8080;
}
```

```
7. [ ] [ ] [ ] weight [ ] [ ] 1 weight=5 max_fails  
[ ] [ ] [ ] 1 max_fails=3 fail_timeout [ ] [ ] [ ] 10s fail_timeout=30s  
backup [ ] [ ] backup down [ ] [ ] down max_conns [ ] [ ] [ ]  
max_conns=1000
```

```
upstream backend_servers {
```

```
server 192.168.1.10:8080 max_fails=3 fail_timeout=30s;  
server 192.168.1.11:8080 max_fails=3 fail_timeout=30s;
```

```
server 192.168.1.12:8080 backup;
```

```
server 192.168.1.13:8080 down;
}
```



HTTP/HTTPS

1.

| Nginx |

172.22.4.200

Web01

Web02

.201

.202

145 151 152 IP

```
ps -ef | grep nginx
ll /etc/nginx/conf.d/

[145]
cat > /etc/nginx/conf.d/webshare.conf << 'EOB'
upstream webshare {
    server 192.168.20.151;
    server 192.168.20.152;
}
EOB
cat > /etc/nginx/conf.d/linuxnote.conf << 'EOB'
server {
    listen 80;
    server_name www.linuxnote.asia;
```

```

    location / {
        proxy_pass http://webshare;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}

```

EOB

[151][152]

```
cat > /usr/local/nginx-1.24.0/conf/vhost/linuxnote.conf << 'EOB'
```

```
server {
    listen 80 default_server;
    root /usr/share/nginx/html;
    index index.html ;

```

```

    location / {
    }
}

```

}

EOB

```
nginx -s reload
```

```
cat > /etc/nginx/conf.d/linuxnote.conf << 'EOB'
```

```
server {
    listen 80 ;
    root /usr/share/nginx/html;
    index index.html ;

```

```

    location / {
    }
}

```

}

EOB

```
mkdir -p /usr/share/nginx/html
```

```
cat > /usr/share/nginx/html/index.html << 'EOB'
```

```
<meta charset="UTF-8">
```

```
<h1>151</h1>
```

EOB

```
cat > /usr/share/nginx/html/index.html << 'EOB'
```

```
<meta charset="UTF-8">
```

```
<h1>152</h1>
```

EOB

```
for i in {1..60};do curl http://192.168.20.145;done > file.txt
curl http://192.168.20.145
```



□□□□□□□□

TCP/UDP□ □□□□

stream □□

```
./configure \
--prefix=/usr/local/nginx \
--with-stream \
--with-stream_ssl_module \
--with-stream_realip_module

vim /etc/nginx/nginx.conf
include conf.d/proxy.4.layer;

cat > /etc/nginx/conf.d/proxy.4.layer << 'EOB'
stream {
    □□□□□□
    log_format proxy '$remote_addr $remote_port - [$time_local] $status $protocol '
    '$$upstream_addr' "$upstream_bytes_sent" "$upstream_connect_time";
    □□□□
    access_log logs/stream_access.log proxy;
    □□□□□□□□ssh 22□□□□
    upstream backend_ssh {
        server 192.168.20.151:22 weight=3;
    }
    MySQL □□
    server {
        listen 2222;
        proxy_pass backend_ssh;
        proxy_connect_timeout 5s;
        proxy_timeout 10s;
    }
}

EOB
```

```
nginx -s reload
```

```
ss -ant
```

```
telnet 192.168.20.145 2222
```

```
dnf install -y telnet
```

```
### 测试测试
```

```
mkdir -p /web/static
```

```
cat > /etc/nginx/conf.d/linuxnote1.conf << 'EOB'
```

```
server {
```

```
    listen 80 ;
```

```
    server_name www.linuxnote1.asia;
```

```
    root /web;
```

```
    index index.html ;
```

```
    location / {
```

```
        try_files $uri $uri/ /index.html;
```

```
    }
```

```
    location /static {
```

```
    }
```

```
}
```

```
EOB
```

```
systemctl restart nginx
```

```
cat > /web/index.html << 'EOB'
```

```
<meta charset="UTF-8">
```

```
<h1>测试145测试测试</h1>
```

```
EOB
```

```
curl http://192.168.20.145
```

```
curl http://192.168.20.151
```

```
curl http://192.168.20.152
```

```
cat > /web/static/index.html << 'EOB'
```

```
<meta charset="UTF-8">
```

```
<h1>测试145测试测试</h1>
```

```
EOB
```

```
cd /web/static && ll
```

```
http://192.168.20.151/static/aaa.png
```

```
cat >index.html<<'EOB'
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>
Hello! This is 145 Server <br/> <br/> <br/>
The following photo source 151 server: <br/>

<br/>
</body>
</html>
EOB
```

```
192.168.20.145 www.linuxnote1.asia
192.168.20.151 www.linuxnote2.asia
192.168.20.152 www.linuxnote3.asia
```

```
echo "192.168.20.145 www.linuxnote1.asia" >> /etc/hosts
echo "192.168.20.151 www.linuxnote2.asia" >> /etc/hosts
echo "192.168.20.152 www.linuxnote3.asia" >> /etc/hosts
```

```
windows
C:\Windows\System32\drivers\etc\hosts
```

```
http://www.linuxnote1.asia/static/index.html
http://www.linuxnote2.asia/static/aaa.png
```

```
cat >/etc/nginx/conf.d/linuxnote2.conf<<'EOB'
server {
    listen 80 default_server;
    server_name www.linuxnote2.asia;
    root /web;
    index index.html ;

    location ~* \.(js|css|jpg|jpeg|png|gif|webp|bmp|ico)$ {
```

```

valid_referers none blocked server_names
192.168.20.151
*.linuxnote2.asia
linuxnote2.asia;
if ($invalid_referer) {
    return 403;
}
}
}
E0B
nginx -t
systemctl restart nginx

```

www.linuxnote2.asia/static/index.html



1. limit_req /
limit_conn limit_rate
2. limit_req - 201

| |



| | ←

| |

| | | |

| |



- 000 00000000
- 00 0000000
- 000 00000000
- 00 0000000000

```

http {
    limit_req_zone $binary_remote_addr zone=api_limit:10m rate=5r/s;

    upstream backend {
        server 192.168.20.151:80;
    }
}
grep limit_req_zone /etc/nginx/nginx.conf
cat >/etc/nginx/conf.d/linuxnote2_1.conf<<'E0B'
server {
    listen 80 ;
    server_name www.linuxnote2.asia;
    access_log /var/log/nginx/linuxnote2_access.log main;
    error_log /var/log/nginx/linuxnote2_error.log;

    location /api/ {
        limit_req zone=api_limit burst=10 nodelay;
        limit_req_status 503; # 00000000503

        proxy_pass http://backend;
    }
    error_page 503 = @rate_limit;
    location @rate_limit {
        default_type application/json;
        return 503 '{"code":503,"message":"000000000000"}';
    }
}
E0B

systemctl reload nginx

- ` $binary_remote_addr ` 00000IP00

```

```
- `zone=mylimit:10m`mylimit10M
- `rate=5r/s`10

dnf -y install httpd-tools
ab -c 10 -n 100 http://www.linuxnote2.asia:8020/api/
www.linuxnote2.asia/api/
```



http

```
limit_conn_zone $binary_remote_addr zone=perip:10m;
limit_conn_zone $server_name zone=perserver:10m;

server {
    listen 80;
    server_name example.com;

    limit_conn perip 6;
    limit_conn perserver 6000;
    limit_conn_status 503;
    location / {
        root /web;
    }
}
```

IP 10-20 IP 3-5 API IP 2-5



```
server {
    listen 80;
    server_name example.com;

    location / {
```

```

        limit_rate 1m;
    }

    location /download/ {
        limit_rate 500k;
        limit_rate_after 10m;
    }
}

122.152.231.125-zy
http://k1.zhynet.net/down.html

```

📦📦📦📦 https📦📦

http📦📦 https=ssl📦📦 +http

<https://euxs8.xetslk.com/sl/4lr9yg>

📦📦 SSL📦📦📦📦📦📦📦📦

📦📦

```

mkdir -p /etc/nginx/ssl
cd /etc/nginx/ssl

openssl req -x509 -nodes -days 365 \
-newkey rsa:2048 \
-keyout 2_https.key \
-out 2_https.crt \
-subj "/C=CN/ST=Shanghai/L=Shanghai/O=MengKe/OU=aaaa/CN=www.linuxnote2.asia"

cat > /etc/nginx/conf.d/linunote2_https.conf<<'E0B'
server {
    listen 443 ssl;
    server_name www.linuxnote2.asia;
    ssl_certificate /etc/nginx/ssl/2_https.crt;
    ssl_certificate_key /etc/nginx/ssl/2_https.key;
    ssl_protocols TLSv1.2 TLSv1.3;
}

```

```
ssl_ciphers HIGH:!aNULL:!MD5;
```

```
location / {  
    root /web;  
    index index.html;  
}
```

```
}
```

```
EOB
```

```
systemctl reload nginx
```

```
https://www.linuxnote2.asia
```



Nginx Rewrite

Rewrite ☐ ● URL ☐ URL ☐ ● ☐ URL ☐ SEO ☐ URL ☐
☐ ● ☐ ngx_http_rewrite_module

☐ ☐ ☐ rewrite server, location, if URL ☐ return server, location, if ☐
set server, location, if ☐

```
if server, location ☐
```

break server, location, if ☐ rewrite ☐

Rewrite ☐ ☐



☐ ☐ ☐ ^ ☐ ^/api/

```
$ ☐.html$
```

☐ ☐ ☐ . ☐ ^/user/.+\$

- ☐ 0 ☐ .*
- ☐ 1 ☐ .+ ? ☐ 0 ☐ 1 ☐ index.html?

```
() ☐ ^/([0-9]+)/
```

```

### 
cat > /web/index.php<<'EOB'
<?php
echo "<h1>Demo</h1>";
echo "<p><a href='/aaa/123.html'>123</a></p>";
echo "<p><a href='/bbb/bbc.html'></a></p>";
echo "<p><a href='/ccc/123.html'>apple</a></p>";
?>
EOB

mkdir -p /web/aaa /web/bbb /web/ccc
cat > /web/aaa.php<<'EOB'
<?php
$id = $_GET['id'] ?? '';
echo "<h1></h1>";
echo "<p>ID: $id</p>";
echo "<p>URL: " . $_SERVER['REQUEST_URI'] . "</p>";
echo "<p>: /aaa/123.php</p>";
?>
EOB

cat > /web/bbb.php<<'EOB'
<?php
$name = $_GET['name'] ?? '';
$id = $_GET['id'] ?? '';
echo "<h1>Get</h1>";
echo "<p>URL: " . $_SERVER['REQUEST_URI'] . "</p>";
echo "<p>id: $id</p>";

?>
EOB

cat > /web/ccc.php<<'EOB'
<?php
// ===== 1. =====
$id = $_GET['id'] ?? '';
$name = $_GET['name'] ?? '';
$page = $_GET['page'] ?? 1;

// ===== 2. =====

```

```
$request_uri = $_SERVER['REQUEST_URI'];      // 请求 URL
$client_ip = $_SERVER['REMOTE_ADDR'];        // 客户端 IP
$host = $_SERVER['HTTP_HOST'];                // 主机名
```

```
// ===== 3. 输出数据 =====
```

```
if (isset($_GET['id'])) {
    echo "<div id='id'>";
}
echo "<p>id: $id</p> \n<br>";
echo "<p>name: $name</p> \n<br>";
echo "<p>page: $page</p> \n<br>";
echo "<p>host: $host</p> \n<br>";
echo "<p>client_ip: $client_ip</p> \n<br>";
echo "<p>request_uri: $request_uri</p> \n<br>";
```

```
?>
```

EOB

```
cat > /etc/nginx/conf.d/rewrite.conf<<'EOB'
```

```
server {
    listen 8080;
    server_name www.linuxnote1.asia;
    root /web;
    index index.php index.html;

    rewrite ^/aaa/([0-9]+)\.html$ /aaa.php?id=$1 last;
    rewrite ^/bbb/([a-z]+)\.html$ /bbb.php?id=$1 last;
    rewrite ^/ccc/([0-9]+)\.html$ /ccc.php?id=$1&name=$1&page=1 last;

    location / {
        rewrite ^ /index.php last;
    }

    location ~ \.php$ {
        fastcgi_pass unix:/run/php-fpm/www.sock;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
}
```

```
rewrite_log on;
error_log /var/log/nginx/rewrite.log notice;
}
```

EOB

```
systemctl reload nginx
```

```
http://www.linuxnote1.asia:8080/aaa/123.html
```

```
curl http://www.linuxnote1.asia:8080/aaa/123.html
```

```
curl http://www.linuxnote1.asia:8080/bbb/bbc.html
```

```
curl http://www.linuxnote1.asia:8080/ccc/123.html
```

```
tail -n 20 /var/log/nginx/rewrite.log
```

```
$id = $_GET['id']
```

```
$name = $_GET['name']
```

```
$client_ip = $_SERVER['REMOTE_ADDR']
```

```
$host = $_SERVER['HTTP_HOST']
```

```
$request_uri = $_SERVER['REQUEST_URI']
```

```
### httphttps
```

```
server {
```

```
• listen 80;
```

```
• server_name qianshan.cc;
```

```
• return 301 https://$server_name$request_uri;
```

```
}
```

```
###
```

```
log_format main '$remote_addr - $remote_user [$time_local] "$request" '
```

```
'$status $body_bytes_sent "$http_referer" '
```

```
'"$http_user_agent" "$http_x_forwarded_for"';
```

```
$remote_addr #IP
```

```
$remote_user #
```

```
$time_local
#$$$time_iso8601ISO8601
```

```
$request #http
$status #()
$body_bytes_sent #
$http_referer #
$http_user_agent #
$http_x_forwarded_for #IP

### 
log_format log_json '{"@timestamp": "$time_local", '
                        '"remote_addr": "$remote_addr", '
                        '"referer": "$http_referer", '
                        '"request": "$request", '
                        '"status": $status, '
                        '"bytes": $body_bytes_sent, '
                        '"agent": "$http_user_agent", '
                        '"x_forwarded": "$http_x_forwarded_for", '
                        '"up_addr": "$upstream_addr", '
                        '"up_host": "$upstream_http_host", '
                        '"up_resp_time": "$upstream_response_time", '
                        '"request_time": "$request_time" '
                        ' }';
```

```
access_log /var/log/nginx/access.log log_json;

more /usr/local/nginx-1.24.0/conf/vhosts/https.conf
server {
    listen 80;
    server_name qianshan.cc www.qianshan.cc;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl;
    server_name qianshan.cc www.qianshan.cc;
```

```

ssl_certificate /etc/nginx/ssl/qianshan.crt;
ssl_certificate_key /etc/nginx/ssl/qianshan.key;
ssl_protocols TLSv1.2 TLSv1.3;
ssl_ciphers HIGH:!aNULL:!MD5;

access_log logs/qianshan.cc.log log_json;
location / {
    root /data/www/;
    index index.html;
}
}

### 配置日志轮转
cat >/etc/logrotate.d/nginx<<'EOB'
/usr/local/nginx-1.24.0/logs/*.log
{
    daily
    rotate 15
    missingok
    compress
    delaycompress
    notifempty
    postrotate
        if [ -f /usr/local/nginx-1.24.0/logs/nginx.pid ]; then
            kill -USR1 `cat /usr/local/nginx-1.24.0/logs/nginx.pid`
        fi
endscript
}
EOB

/etc/logrotate.d/nginx,配置日志轮转

logrotate -vf /etc/logrotate.d/nginx

#crontab -e
59 23 * * * /usr/sbin/logrotate -vf /etc/logrotate.d/nginx #配置23点59分轮转

```



- 1.  nginx.conf 
- 2.  .conf  conf.d/ 

```
worker_rlimit_nofile 65535;

events {
    worker_connections 4096;
    use epoll;
    multi_accept on;
}

http {
    # 配置
    client_body_buffer_size 10K;
    client_header_buffer_size 1k;
    client_max_body_size 8m;
    large_client_header_buffers 4 8k;
    # 配置
    client_body_timeout 12;
    client_header_timeout 12;
    send_timeout 10;
    # 配置
    gzip on;
    gzip_vary on;
    gzip_min_length 1024;
    gzip_proxied any;
    gzip_types text/plain text/css application/json application/javascript;
}
```



```
server_tokens off;
```



```
add_header X-Frame-Options "SAMEORIGIN" always;
add_header X-Content-Type-Options "nosniff" always;
```

```
add_header X-XSS-Protection "1; mode=block" always;
```

□□□□□

```
if ($request_method !~ ^(GET|HEAD|POST)$ ) {  
    return 405;  
}
```

□□□□□□□

```
location ~ /\. (ht|git|svn) {  
    deny all;  
    return 404;  
}
```

CPU□□□ □□□□□ CPU□□

```
worker_processes auto;  
worker_cpu_affinity auto;
```

□□□□□ □□□□□□

```
open_file_cache max=10000 inactive=60s;  
open_file_cache_valid 60s;  
open_file_cache_min_uses 2;  
open_file_cache_errors on;
```

□□ sendfile□□□□□□□

```
sendfile on;
```

tcp_nopush on; # □□ sendfile□□□□□□□□□□ tcp_nodelay on; # □□ Nagle□□ □□□□

```
keepalive_timeout 65;
```

keepalive_requests 1000; # □□ keepalive□□□□□□□ reset_timedout_connection on; #
□□□□□□□□

#####

```
location ~* \.(jpg|jpeg|png|gif|ico|css|js)$ {
```

expires 30d; # #####

```
add_header Cache-Control "public, immutable";
```

```
### #####
```

```
```bash
```

```
cat >nginx.conf<<'EOB'
```

```
####
```

```
user nginx;
```

```
#####CPU####
```

```
worker_processes auto;
```

```
CPU#####
```

```
worker_cpu_affinity auto;
```

```
#####
```

```
worker_rlimit_nofile 65535;
```

```
####
```

```
error_log /var/log/nginx/error.log warn;
```

```
##ID
```

```
pid /run/nginx.pid;
```

```
Events##
```

```
events {
```

```
 worker_connections 4096;
```

```
 use epoll;
```

```
 multi_accept on;
```

```
}
```

```
HTTP##
```

```
http {
```

```
####
```

```
 include /etc/nginx/mime.types;
```

```
 default_type application/octet-stream;
```

```
 # #####
```

```
 server_tokens off;
```

```
 # ####
```

```
 log_format main '$remote_addr - $remote_user [$time_local] "$request" '
```

```
 '$status $body_bytes_sent "$http_referer" '
```

```
 '$http_user_agent' '$http_x_forwarded_for';
默认
access_log /var/log/nginx/access.log main;
默认
sendfile on;
tcp_nopush on;
tcp_nodelay on;
默认
keepalive_timeout 65;
keepalive_requests 1000;
reset_timedout_connection on;
默认
client_body_buffer_size 10K;
client_header_buffer_size 1k;
client_max_body_size 8m;
large_client_header_buffers 4 8k;
默认
client_body_timeout 12;
client_header_timeout 12;
send_timeout 10;
默认
gzip on;
gzip_vary on;
gzip_min_length 1024;
gzip_proxied any;
gzip_types text/plain text/css application/json application/javascript;
默认
open_file_cache max=10000 inactive=60s;
open_file_cache_valid 60s;
open_file_cache_min_uses 2;
open_file_cache_errors on;
默认
add_header X-Frame-Options "SAMEORIGIN" always;
add_header X-Content-Type-Options "nosniff" always;
add_header X-XSS-Protection "1; mode=block" always;
默认
server {
 listen 80;
 server_name localhost;
 # 默认
```

```
 root /usr/share/nginx/html;
 index index.html index.htm;

 # 禁止访问
 if ($request_method !~ ^(GET|HEAD|POST)$) {
 return 405;
 }

 # 禁止目录访问
 location ~ /\.(ht|git|svn) {
 deny all;
 return 404;
 }

 # 静态文件
 location ~* \.(jpg|jpeg|png|gif|ico|css|js)$ {
 expires 30d;
 add_header Cache-Control "public, immutable";
 }

 # 500 502 503 504 错误页面
 error_page 500 502 503 504 /50x.html;
 location = /50x.html {
 root /usr/share/nginx/html;
 }
}

包含其他配置文件
include /etc/nginx/conf.d/*.conf;
}

EOB
```



# nginx

```
dnf module list nginx
dnf module reset nginx -y
dnf module enable nginx:1.24
dnf -y install nginx
```



```
proxy_pass http://192.168.20.146:8080;
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```



```
upstream aaa_servers {server 111;222;333}
server {proxy_pass aaa_servers}
```

 https

```
listen 443 ssl;
ssl_certificate /etc/ssl/certs/example.com.crt;
ssl_certificate_key /etc/ssl/private/example.com.key;
```



```
/etc/hosts
```



```
/static/{alias /aaa; autoindex on;}
```

 linux

```
wget https://gitee.com/Discuz/DiscuzX/attach_files/2714621/download -O Discuz_X3.4_SC_UTF8.zip

dnf install php-mysqlnd php-xml php-json -y
```

```
grep -E '^listen|^user|^group' /etc/php-fpm.d/www.conf
```

```
server {
 listen 80 default_server ;
 server_name www.bbs.com.cn ;
 root /web/upload ;
 index index.php index.html ;

 location / {
 index index.php ;
 }

 location ~ \.php$ {
 fastcgi_pass unix:/run/php-fpm/www.sock;
 fastcgi_index index.php;
 fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
 include fastcgi_params;
 }

}
```



## tcpdump

```
dnf install -y tcpdump
```

```
dnf install -y telnet
```

```
tcpdump -i ens160 port 80
```

```
telnet 192.168.20.146 80
```



```
ping 192.168.20.145
telnet 192.168.20.145 80
iptables -A INPUT -s 192.168.20.135 -j DROP
service iptables save
iptables -nL
iptables -nL --line-numbers
```



## iptables

```
iptables -D INPUT 1 && iptables -nL INPUT --line-number

iptables -nL INPUT --line-number && awk '{print $1}' /var/log/nginx/access.log | sort | uniq -c

yum install httpd-tools -y
ab -c 100 -n 1000 -t 30 http://192.168.20.145/

dnf install git make gcc -y
git clone https://github.com/wg/wrk.git
cd wrk
make
cp wrk /usr/local/bin/
wrk --version
wrk -t4 -c100 -d30 http://192.168.20.145/
```



```
watch -n 1 "awk '{print \$1}' /var/log/nginx/access.log | sort | uniq -c"
```

```
worker_cpu_affinity auto; []
worker_connections 65535; []
use epoll;[]
worker_rlimit_nofile 65535;[]
```

```
location ~* \.(jpg|png|css) { expires 30d; }
```

```
gzip on;
gzip_comp_level 5;
gzip_types text/plain text/css application/json;
sendfile on;
tcp_nopush on;
server_tokens off;
```

---

Revision #2

Created 2026-08-20 14:10:19 CST by Admin

Updated 2026-08-20 14:44:57 CST by Admin