

nginx 命令

Nginx

网络服务器web服务器反向代理HTTP代理

命令行

命令 nginx.exe 命令 nginx

```
nginx -s signal
```

信号命令

- stop 停止
- quit 强制停止
- reload 重新加载配置
- reopen 重新打开日志

nginx 命令

```
nginx -v
```

显示版本号 nginx 版本号

```
ps aux | grep nginx
```

显示进程数

```
nginx -t
```

测试配置

配置文件

- 配置文件 Nginx 配置文件
 - user 用户名
 - worker_processes 工作进程数
 - error_log 错误日志文件
 - access_log 访问日志文件

```
user www-data www-data;
worker_processes auto;
worker_rlimit_nofile 65535;
pid /run/nginx.pid;

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

- events Nginx 事件处理器
 - worker_connections 工作连接数

- epoll 用于 epoll 监控
- multi_accept 多线程多连接

```
events {
    worker_connections 2048;
    use epoll;
    multi_accept on;
}
```

- **http** 服务器配置
 - include 包含文件
 - default_type 默认类型
 - 为 Nginx 提供静态文件类型的映射
 - keepalive_timeout 保持连接超时
 - gzip 压缩
 - 通过压缩响应体来减少带宽消耗

```
http {
    include      /etc/nginx/mime.types;
    default_type application/octet-stream;
    keepalive_timeout  65;
    gzip  on;

    # 服务器配置
}
```

- **server** 服务配置
 - listen 80 监听80端口
 - server_name 服务器名称
 - root 静态文件目录
 - index 首页文件
 - error 错误文件

```
server {
    listen      80;
    server_name example.com;
    root        /var/www/html;
    index       index.html;

    # 位置配置
}
```

- **location** 路径URI配置
 - 位置
 - location /path/ 路径匹配
 - location ~ regex 正则表达式匹配
 - location = /path 固定字符串匹配

- 亂子块
 - root 亂子块
 - proxy_pass 亂子块
 - try_files 亂子块
 - expires 亂子块
 - s 亂子块
 - m 亂子块
 - h 亂子块
 - d 亂子块
 - w 亂子块
 - m 亂子块
 - y 亂子块

```
location / {
    try_files $uri $uri/ /index.php?$query_string;
}

location ~ \.php$ {
    proxy_pass     http://php_backend;
}

location ~* \.(jpg|png)$ {
    root /data/media;
    expires 7d;
}
```

亂子块

- \$ 亂子块 Nginx 亂子块
 - \$uri 亂子块 uri 亂子块
 - \$request_uri 亂子块 uri
 - \$args 亂子块
 - \$host 亂子块
 - \$scheme 亂子块

配置文件

```
# 亂子块main context
user nobody; # 亂子块Nginx 亂子块'no body'

events {
    # 亂子块
}

http {
    # HTTP 亂子块
    server {
        # 亂子块HTTP 亂子块
        location /one {
```

10 of 10

```
main[][]
├── events[]
├── http[]HTTP[]
│   ├── server[][]
│   │   ├── location /one[]
│   │   └── location /two[]
│   └── server[][]
└── stream[]TCP/UDP[]
    └── server[]TCP[]1
```

1

□ □ □ □ □ □ □ □

1

- □ http □□□ upstream backend servers

```
http {  
    upstream backend_servers {  
        # 服务器列表  
        server 192.168.1.101:8080;  
        server 192.168.1.102:8080;  
        server 192.168.1.103:8080;  
    }  
}
```

```
    }
}
```

- `server` 服务器 upstream

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

    location / {
        proxy_pass http://backend_servers; # 代理到上游
        proxy_set_header Host $host;
    }
}
```

返回

`return` 用于 rewrite 重写**`server` 位置 **返回

`return` 0

- `code` HTTP状态
- `URL` 重定向

```
return code [URL/text];
```

`rewrite` 0

- `regex` URL正则表达式
- `replacement` 替换URL
- `flag` 标志
 - `last` 最后 `rewrite` URI位置 `location`
 - `break` 中断 `rewrite` 位置 `location` 重新开始
 - `redirect` 302重定向
 - `permanent` 301重定向

```
rewrite regex replacement [flag];
```

返回

重定向

- `client_max_body_size` Javascript 客户端最大API
- `client_body_timeout` ip客户端连接超时

返回

- `add_header 'Access-Control-Allow-Origin' 值`
- `add_header 'Access-Control-Allow-Methods' 值`HTTP方法
- `add_header 'Access-Control-Allow-Headers' 值`
- `add_header 'Access-Control-Allow-Credentials' 值`Cookie

```

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

    location / {
        # CORS 配置
        add_header 'Access-Control-Allow-Origin' '*';

        # HTTP 头部
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';

        # 头部
        add_header 'Access-Control-Allow-Headers' 'Content-Type, Authorization, X-
Requested-With';

        # Cookie 头部
        add_header 'Access-Control-Allow-Credentials' 'true';

        # OPTIONS 头部
        add_header 'Access-Control-Max-Age' 1728000;

        # 代理
        proxy_pass http://backend_server;
    }
}

```

HTTP 状态码

1xx 状态码

- 100 服务器响应
- 101 交替连接

2xx 状态码

- 200 成功
- 201 Post 成功
- 204 无内容

3xx 状态码

- 301 永久重定向
- 302/307 临时重定向
- 304 未修改

4xx 错误状态

- 400 错误请求
- 401 未授权
- 403 禁止访问
- 404 未找到
- 405 方法错误
- 429 超过速率限制

5xx ດ້ວຍເຫັນ

- 500 ດ້ວຍເຫັນ
- 502 ດ້ວຍເຫັນ
- 503 ດ້ວຍເຫັນ
- 504 ດ້ວຍເຫັນ