

[nginx 安装](#)

Nginx

nginx 是一个 web 服务器，它支持 **Http** 协议

nginx 的启动

nginx 的 **nginx.exe** 文件在 **nginx** 目录下

```
nginx -s signal
```

nginx 的 **signal** 选项

- **stop** 关闭 nginx
- **quit** 关闭 nginx
- **reload** 重新加载配置文件
- **reopen** 重新打开日志文件

nginx 的 **nginx** 选项

```
nginx -v
```

nginx 的 **nginx** 选项

```
ps aux | grep nginx
```

nginx 的 **nginx** 选项

```
nginx -t
```

nginx 的 **nginx** 选项

nginx 的 **nginx** 选项

- nginx 的 **Nginx** 选项
 - **user** 指定运行 nginx 的用户
 - **worker_processes** 指定 nginx 的工作进程数
 - **error_log** 指定 nginx 的错误日志文件
 - **access_log** 指定 nginx 的访问日志文件

```
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** 选项
 - **worker_connections** 指定 nginx 的工作进程可以打开的最大连接数

- `epoll` 是 `epoll` 的别名
- `multi_accept` 是否同时接受多个连接

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

- **http** 配置块
 - `include` 包含其他配置文件
 - `default_type` 默认 MIME 类型
 - 是 Nginx 默认 MIME 类型
 - `keepalive_timeout` 保持连接超时时间
 - `gzip` 是否开启 GZIP 压缩
 - 默认是开启的

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

    # 配置 server 块
}
```

- **server** 配置块
 - `listen` 80 监听 80 端口
 - `server_name` 域名
 - `root` 根目录
 - `index` 默认索引文件
 - `error` 错误页面

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

    # 配置 location 块
}
```

- **location** 配置块
 - 匹配 URI
 - `location /path/` 匹配以 `/path/` 开头的 URI
 - `location ~ regex` 匹配符合正则表达式 `regex` 的 URI
 - `location = /path` 精确匹配 `/path`

- 配置
 - root 配置
 - proxy_pass 配置
 - try_files 配置
 - expires 配置
 - s 秒
 - m 分
 - h 小时
 - d 天
 - w 周
 - M 月
 - Y 年

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

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

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

配置

- \$ 是 Nginx 配置
- \$uri 配置 uri
- \$request_uri 配置 uri
- \$args 配置
- \$host 配置
- \$scheme 配置

配置

```
# 配置main context
user nobody; # 配置Nginx配置'nobody'

events {
    # 配置
}

http {
    # HTTP配置

    server {
        # 配置HTTP配置
        location /one {
```

```

        # 访问 '/one' 的URI
    }
    location /two {
        # 访问 '/two' 的URI
    }
}

server {
    # HTTP
}

stream {
    # TCP/UDP
    server {
        # TCP
    }
}

```

配置

```

main
├─ events
├─ http
│  └─ server1
│     │   └─ location /one1
│     │   └─ location /two2
│     └─ server2
└─ stream
   └─ serverTCP1

```

配置

配置

- 配置
- 配置
- 配置
- IP
- 配置

配置

- http upstream backend_servers

```

http {
    upstream backend_servers {
        #
        server 192.168.1.101:8080;
        server 192.168.1.102:8080;
        server 192.168.1.103:8080;
    }
}

```

```
}  
}
```

- `server` 指定 upstream

```
server {  
    listen 80;  
    server_name example.com;  
  
    location / {  
        proxy_pass http://backend_servers; # 指定 upstream  
        proxy_set_header Host $host;  
    }  
}
```

返回

`return` 使用 `rewrite` 指定 `server` 的 `location`

`return` 指定

- `code` HTTP 状态码
- `URL` 返回的 URL

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

`rewrite` 指定

- `regex` 正则表达式
- `replacement` 替换后的 URL
- `flag` 标志
 - `last` 使用 `rewrite` 指定的 `URI` 的 `location`
 - `break` 使用 `rewrite` 指定的 `location`
 - `redirect` 302 重定向
 - `permanent` 301 永久重定向

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

其他

其他配置

- 使用 `JavaScript` 的 `API`
- 使用 `ip` 地址

其他

- `add_header` 'Access-Control-Allow-Origin' 指定
- `add_header` 'Access-Control-Allow-Methods' 指定 HTTP 方法
- `add_header` 'Access-Control-Allow-Headers' 指定头部
- `add_header` 'Access-Control-Allow-Credentials' 指定 Cookie

```

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

    location / {
        # 跨域请求
        add_header 'Access-Control-Allow-Origin' '*';

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

        # 支持的请求头
        add_header 'Access-Control-Allow-Headers' 'Content-Type, Authorization, X-
Requested-With';

        # 允许 Cookie 跨域 *
        add_header 'Access-Control-Allow-Credentials' 'true';

        # 支持 OPTIONS 请求
        add_header 'Access-Control-Max-Age' 1728000;

        # 反向代理
        proxy_pass http://backend_server;
    }
}

```

状态码

1xx

- 100 继续
- 101 切换协议

2xx

- 200 成功
- 201 创建成功
- 204 无内容

3xx

- 301 永久重定向
- 302/307 临时重定向
- 304 缓存命中

4xx

- 400 错误请求
- 401 未授权
- 403 禁止
- 404 资源不存在
- 405 方法不允许
- 429 请求过多

5xx□□□□□

- 500 □□□□□□□
- 502 □□□□
- 503 □□□□□
- 504 □□□□