

# Docker 应用部署-Nginx

## 1. 搜索nginx镜像

```
1 docker search nginx
```

## 2. 拉取nginx镜像

```
1 docker pull nginx
```

## 3. 创建容器

设置端口映射、目录映射

```
1 # 下面是在/root目录下创建nginx目录用于存储nginx数据信息
2 mkdir ~/nginx
3 cd ~/nginx
4 mkdir conf logs conf/conf.d
```

在~/nginx/conf/下创建nginx.conf文件,粘贴下面内容, vi conf/nginx.conf

```
1 user nginx;
2 worker_processes 1;
3 error_log /var/log/nginx/error.log warn;
4 pid /var/run/nginx.pid;
5 events {
6     worker_connections 1024;
7 }
8 http {
9     include /etc/nginx/mime.types;
10    default_type application/octet-stream;
11    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
12                  '$status $body_bytes_sent "$http_referer" '
13                  '"$http_user_agent" "$http_x_forwarded_for"';
14    access_log /var/log/nginx/access.log main;
15    sendfile on;
16    #tcp_nopush on;
17    keepalive_timeout 65;
18    #gzip on;
19    include /etc/nginx/conf.d/*.conf;
20 }
```

在~/nginx/conf/conf.d下创建80.conf文件,粘贴下面内容, vi conf/conf.d/80.conf

```

1 server {
2     listen 80;
3     root /usr/share/nginx/html;
4     index index.php index.html index.htm;
5     error_page 500 502 503 504 /50x.html;
6     location / {}
7 }

```

创建容器并启动

```

1 docker run -id --name nginx \
2 -p 80:80 \
3 -v $PWD/conf/nginx.conf:/etc/nginx/nginx.conf \
4 -v $PWD/conf/conf.d:/etc/nginx/conf.d \
5 -v $PWD/logs:/var/log/nginx \
6 -v $PWD/html:/usr/share/nginx/html \
7 nginx

```

参数说明：

- **-p 80:80**：将容器的 80端口映射到宿主机的 80 端口。
- **-v \$PWD/conf/nginx.conf:/etc/nginx/nginx.conf**：将主机当前目录下的 /conf/nginx.conf 挂载到容器的 /etc/nginx/nginx.conf。配置目录
- **-v \$PWD/logs:/var/log/nginx**：将主机当前目录下的 logs 目录挂载到容器的/var/log/nginx。日志目录

## 4. 开放80访问端口

```

1 #查看是否已经开放80端口
2 firewall-cmd --list-port
3 #没有开放使用下面命令开放
4 firewall-cmd --add-port 80/tcp --permanent
5 #重新加载防火墙
6 firewall-cmd --reload

```

## 5. 访问nginx

访问你的Nginx服务器，访问地址：http://你服务器的ip地址

## 6. 扩展

### 6.1 Vue刷新页面404

如果你在浏览器上方法Vue部署项目的时候，F5刷新页面的时候可能会出404 Not Found错误。

因为Vue的路由路径是虚拟的所以刷新会存在404的问题，通过修改配置解决，修改conf/conf.d/80.conf配置文件

```

1 server {
2     listen 80;
3     root /usr/share/nginx/html;

```

```

4     index index.php index.html index.htm;
5     error_page 500 502 503 504 /50x.html;
6     location / {
7         try_files $uri $uri/ @router;
8         index index.html;
9     }
10    # vue router
11    location @router {
12        rewrite ^.*$ /index.html last;
13    }
14 }

```

## 6.2 配置后端代理

如果需要在Nginx反向代理网关服务器，可以通过配置代理完成，同样修改conf/conf.d/80.conf配置文件

```

1  server {
2      listen 80;
3      root /usr/share/nginx/html;
4      index index.php index.html index.htm;
5      error_page 500 502 503 504 /50x.html;
6      location / {
7          try_files $uri $uri/ @router;
8          index index.html;
9      }
10     # vue router
11     location @router {
12         rewrite ^.*$ /index.html last;
13     }
14     # proxy gateway
15     location ^~/api/ {
16         proxy_set_header Host $host;
17         proxy_set_header X-Real-IP      $remote_addr;
18         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
19         proxy_set_header X-NginX-Proxy true;
20         # trim prefix
21         rewrite ^/api/(.*) /$1 break;
22         proxy_pass http://ip:port;
23     }
24 }

```

或者也可以这样

```

1  server {
2      listen 80;
3      root /usr/share/nginx/html;
4      index index.php index.html index.htm;
5      error_page 500 502 503 504 /50x.html;
6      location / {
7          try_files $uri $uri/ @router;
8          index index.html;
9      }
10     # vue router
11     location @router {

```

```

12     rewrite ^.*$ /index.html last;
13 }
14 # proxy gateway
15 location ^~/api/ {
16     proxy_set_header Host $host;
17     proxy_set_header X-Real-IP      $remote_addr;
18     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
19     proxy_set_header X-NginX-Proxy true;
20     proxy_pass http://ip:port/;
21 }
22 }

```

## 6.3 Nginx代理管理系统

nginx-proxy-manager 是一个Nginx可视化反向代理管理系统，它基于 NGINX，具有漂亮干净的 Web UI。还可以获得受信任的 SSL 证书，并通过单独的配置、自定义和入侵保护来管理多个代理。

软件开源地址：<https://github.com/NginxProxyManager/nginx-proxy-manager>

### 6.3.1 准备目录

创建并进入安装目录

```

1 mkdir nginx-pm
2 cd nginx-pm

```

### 6.3.2 服务编排文件

在nginx-pm目录下面创建docker-compose.yml文件，并写入以下内容

```

1 version: '3'
2 services:
3   app:
4     image: 'jc21/nginx-proxy-manager:latest'
5     restart: unless-stopped
6     ports:
7       - '80:80'
8       - '81:81'
9       - '443:443'
10    volumes:
11      - ./data:/data
12      - ./letsencrypt:/etc/letsencrypt

```

### 6.3.4 启动服务

执行docker-compose up -d命令启动服务

首次启动服务需要下载镜像等待即可，启动执行完成结果如下图所示。

```

# 42d509ced030 Pull complete
[+] Running 2/2
# Network nginx-pm_default Created
# Container nginx-pm-app-1 Started
[root@localhost nginx-pm]#

```

### 6.3.5 查看进程

输入 `docker ps | grep nginx` 命令查看，执行结果如下图所示

```
[root@localhost nginx-pm]# docker ps | grep nginx-  
lafaa042623d   jc21/nginx-proxy-manager:latest   "/init"  
nginx-pm-app-1  
[root@localhost nginx-pm]#
```

如果启动失败，那么有可能是你的80、81、443端口被占用，关闭调用占用端口的服务进程，重新执行6.3.4步骤，重启即可。

### 6.3.6 开放端口

开放防火墙端口，注意：如果你是云服务器，那么在云控制台去配置入方向规则，而不是使用下面的命令行开放端口。


```
1 firewall-cmd --add-port 81/tcp --add-port 443/tcp --permanent  
2 firewall-cmd --reload
```

```
[root@localhost nginx-pm]# firewall-cmd --add-port 81/tcp --add-port 443/tcp --permanent  
success  
[root@localhost nginx-pm]# firewall-cmd --reload  
success  
[root@localhost nginx-pm]#
```

### 6.3.7 初始化系统

系统安装好了后，需要对系统进行初始化操作

首先访问系统：<http://ip:81>



**NGINX**  
PROXY MANAGER  
v2.9.22

Login to your account

**Email address**

**Password**

**Sign in**

输入账号密码登录进行登录，系统默认用户如下：

```
1 Email: admin@example.com  
2 Password: changeme
```

登录成功后需要修改用户名，根据你自己的情况修改即可

Edit User

×

Full Name \*

Administrator

Nickname \*

Admin

Email \*

admin@example.com

Cancel

Save

用户信息修改完成后就是修改密码

Change Password

×

Current Password

.....

New Password

.....

Confirm Password

.....|


Cancel

Save

修改成功后





## Users

 [Add User](#)

NAME	EMAIL	ROLES
 zero-awei Created: 24th March 2023	2069682479@qq.com	Administrator

到此服务器初始化完成，后续你管理代理就可以在这个UI管理面板完成了

Hi awei

 0 Proxy Hosts	 0 Redirection Hosts	 0 Streams	 0 404 Hosts
---	---	---	---

更多使用介绍请参考：<https://nginxproxymanager.com/>