



链滴

Enable proxy to docker

作者: [expoli](#)

原文链接: <https://ld246.com/article/1581944823604>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



为 Docker 启用代理加速镜像拉取与构建

本文介绍如何为docker启用代理，最近在家中进行docker构建的时候，那是真的感受到了家中网络我的满满的恶意。

原来以为在终端中 `export http_proxy` 之后就万事大吉了，但是实际上速度并没有任何的变化，于是连续几次测试之后，我发现终端的这个代理设置对docker并没有任何的作用，无论是使用 `docker pull` 拉取镜像的时候，还是使用 `docker build` 进行镜像构建的时候都不走终端的代理。emmmmm 崩溃！

1. 配置 docker systemd 环境变量加速镜像拉取

容器内部的代理与 docker 服务所使用的代理配置不同、后续介绍如何为容器启用代理进行加速。

1. 首先，为Docker服务创建一个systemd插入目录：

```
mkdir /etc/systemd/system/docker.service.d
```

2. 然后新建一个 `/etc/systemd/system/docker.service.d/http-proxy.conf` 文件，并在此文件中下类似下面的代理配置内容，添加 `HTTP_PROXY` 环境变量的值。

```
[Service]
Environment="HTTP_PROXY=http://proxy.example.com:port"
```

3. 如果你有内部的Docker 仓库，需要在不使用代理的情况下连接，可以通过 `NO_PROXY` 环境变量行指定：

```
Environment="HTTP_PROXY=http://proxy.example.com:port"
Environment="NO_PROXY=localhost,127.0.0.0/8,docker-registry.somecorporation.com"
```

4. 刷新更改

```
$ sudo systemctl daemon-reload
```

5. 验证更改是否生效

```
$ sudo systemctl show --property Environment docker
Environment=HTTP_PROXY=http://proxy.example.com:port
```

6. 重启docker服务使更改生效

```
$ sudo systemctl restart docker
```

2 配置使容器使用代理

如果您的容器需要使用HTTP，HTTPS或FTP代理服务器，则可以通过下面的方式对其进行配置：

方式1：配置Docker 客户端文件

在 Docker 客户端上，在启动容器的用户主目录中创建或编辑文件 `~/.docker/config.json`。添加一个JSON文件，如下所示，必要时用 `httpsProxy` 或 `ftpProxy` 替换代理类型，并替换代理服务器的地址端口。您可以同时配置多个代理服务器。

通过将 `noProxy` 键设置为一个或多个逗号分隔的 IP 地址或主机，可以选择排除通过代理服务器的主或范围。支持使用 `*` 字符作为通配符，如本示例所示。

```
{
  "proxies":
  {
    "default":
    {
      "httpProxy": "http://127.0.0.1:3001",
      "httpsProxy": "http://127.0.0.1:3001",
      "noProxy": "/*.test.example.com,example2.com"
    }
  }
}
```

创建或启动新容器时，环境变量会自动读取至容器内。

方式2：使用环境变量

生成镜像或使用 `--env` 标志创建或运行容器时，可以将以下一个或多个变量设置为适当的值。此方法镜像的可移植性降低，因此，如果您有 Docker 17.07 或更高版本，则应改为配置 Docker 客户端。

Variable	Dockerfile example
<code>HTTP_PROXY</code> <code>-env HTTP_PROXY="http://127.0.0.1:3001"</code>	<code>ENV HTTP_PROXY "http://127.0.0.1:3001"</code>
<code>HTTPS_PROXY</code> <code>-env HTTPS_PROXY="https://127.0.0.1:3001"</code>	<code>ENV HTTPS_PROXY "https://127.0.0.1:3001"</code>
<code>FTP_PROXY</code> <code>-env FTP_PROXY="ftp://127.0.0.1:3001"</code>	<code>ENV FTP_PROXY "ftp://127.0.0.1:3001"</code>

```
NO_PROXY  
m"
```

```
ENV NO_PROXY "*.test.example.com,.example2.c  
--env NO_PROXY="*.test.example.com,.example2.com
```

参考资料

<https://stackoverflow.com/questions/23111631/cannot-download-docker-images-behind-a-proxy>

<https://docs.docker.com/network/proxy/>