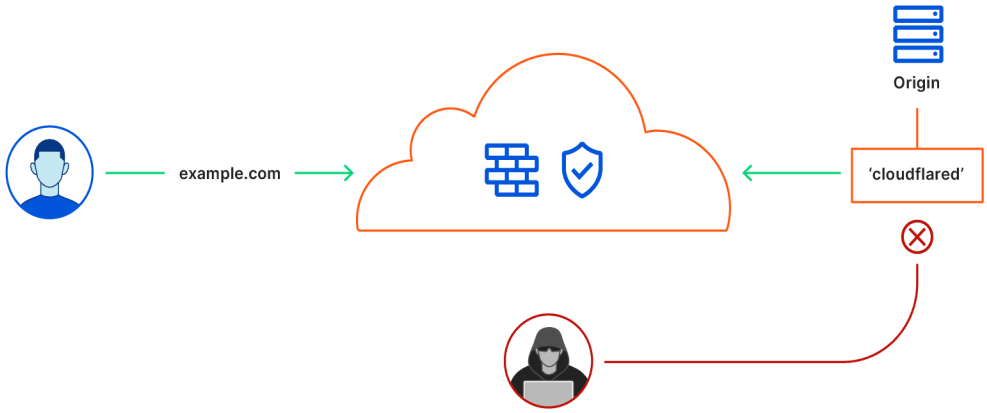
emengweb 's Blog (/)

Figma YES!

探索Cloudflared Tunnel (原Argo Tunnel) 实现中间层网络加速

2022-02-15 14:16:48 622 0 0



如何让外网能访问到没有公网IP的局域网设备或服务一直是一个头疼的问题，但确实也有一些解决方案，例如商业的花生壳，开源的frp、ngrok等。今天推荐一款cloudflare旗下的产品——Cloudflare Tunnel (https://www.cloudflare.com/zh-cn/products/tunnel/)，之前叫Argo Tunnel。以下是官网博客介绍和文档。
相较于其他内网穿透服务，Cloudflare Tunnel有其他服务不能比拟的一些优势。

1. 不需要自购服务器和搭建中转服务
2. 属于Cloudflare服务之一，自动接入了CDN
3. 跟域名直接绑定，且是https

安装

loudflare Tunnel支持Mac、Windows、Linux和Docker。在开始的时候先总结个小经验：任何一端安装好的配置文件，都可以直接复制到其他段做到开箱即用。

根据系统选择对应的安装包，进行安装：

https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/install-and-setup/installation (https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/install-and-setup/installation)

验证登录

安装好后执行Cloudflared账号的登录授权，此操作前请确保已经将准备好的域名已经添加到Cloudflare账号。

```
1. cloudflared tunnel login
```

执行后会打开浏览器，登录账号并选择要授权绑定的域名。成功后会在 `~/.cloudflared` 目录下生成一个 `cert.pem` 文件。此文件是配置文件之一，将它复制到任何其他平台相同配置目录下，可以直接用，而不需要再次验证登录。

新增配置

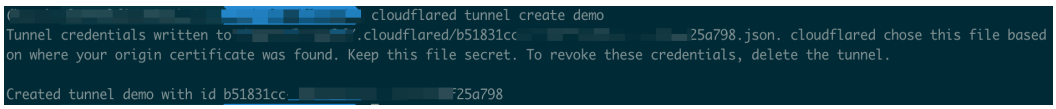
然后开始创建一个tunnel，命名随便，例如“demo”

emengweb 's Blog (/)

Figma YES!

```
cloudflared tunnel create demo
```

创建成功后会得到一个id，这个复制好，下一步要用。



查看已有的tunnel，他会列出该Cloudflare账号下所有的tunnel，同一个设备只能授权访问一个域名，list会列出所有的设备所有域名下的tunnel。

```
cloudflared tunnel list
```

接下来配置要暴露的服务和端口，在 `~/.cloudflared` 目录新建配置文件 `config.yml` ，写入如下配置

```
tunnel: 8132d959-xxxxxx-06f4a82c147a
credentials-file: ./8132d959-xxxxxx-06f4a82c147a.json
ingress:
  - hostname: dsm.abc.com
        service: http://192.168.1.6:5000
  - hostname: lychee.abc.com
        service: http://192.168.1.6:13800
  - service: http_status:404
```

需要注意的是：

1. 一个tunnel对应一个id，但可以对应多个服务多个子域名，如上
2. credentials-file最好配置绝对路径

下面给出一个**高级配置**的用法

emengweb 's Blog (/)

Figma YES!

```
tunnel: 12345678-1ae9-44e7-bf47-12345678
credentials-file: /root/.cloudflared/12345678-1ae9-44e7-bf47-12345678.json
origincert: /root/.cloudflared/cert.pem
protocol: quic #使用基于udp协议的quic方式传输数据，速度最优
originRequest:
  connectTimeout: 30s

#metrics: localhost:5432
#tag: cmm=test
pidfile: /var/run/cf-test-argo.pid
autoupdate-freq: 24h
loglevel: info #debug
logfile: /root/.cloudflared/cloudflared.log

ingress:
  - hostname: site.my.com
        service: https://localhost:443
        originRequest:
            connectTimeout: 10s
            originServerName: hostname.my.com #指定后端服务的解析名，用于在ssl协议下区分不同的域名
        noTLSVerify: true #忽略ssl证书有效性检测
  - hostname: demo.my.com
    service: hello_world
  - service: http_status:404
```

额外参数

`protocol` 传输协议目前支持 http2和quic（推荐）

UDP Receive Buffer Size

quic协议基于UDP，性能比http2要强一大截，如果使用quic协议，建议调整系统默认UDP缓冲区的大小，提高数据传输性能。

As of quic-go v0.19.x, you might see warnings about the receive buffer size.

It is recommended to increase the maximum buffer size by running:

```
sysctl -w net.core.rmem_max=2500000
```

This command would increase the maximum receive buffer size to roughly 2.5 MB.

除了支持暴露这些服务以外，官网文档介绍还支持其他的协议。

https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/configuration/ingress (https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/configuration/ingress)

Service	Description	Example `service` value
HTTP/S	Incoming HTTP requests are proxied directly to your local service	`https://localhost:8000`
HTTP/S over unix socket	Just like HTTP/S, but using a unix socket instead	`unix:/home/production/echo.sock`
TCP	TCP connections are proxied to your local service.	`tcp://localhost:2222`
SSH	SSH connections are proxied to your local service. Learn more.	`ssh://localhost:22`
RDP	RDP connections are proxied to your local service. Learn more.	`rdp://localhost:3389`
kubectl bastion mode	`cloudflared` will act like a jumphost, allowing access to any local address.	`bastion`
Hello World	Test server for validating your Cloudflare Tunnel setup	`hello_world`
HTTP status	Responds to all requests with the given HTTP status	`http_status:404`

配置DNS记录

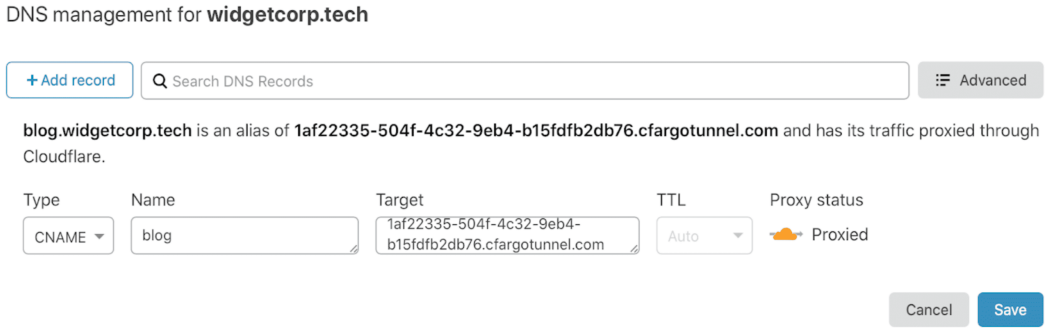
配置域名的dns记录有两种方式：

使用命令行

```
1. # cloudflared tunnel route dns <UUID or NAME> www.app.com
2. cloudflared tunnel route dns dsm dsm.abc.com
3. cloudflared tunnel route dns dsm lychee.abc.com
```

使用后台界面操作

在对应域名下添加CNAME记录，指向上一步中的 `<UUID>.cfargotunnel.com` 即可。如果一个tunnel下对应多个服务，那就创建多图，例如我就要分别创建dsm和lychee子域名的CNAME记录分别指向同一目标地址。



运行配置

配置好CNAME记录后，开启Tunnel

```
1. cloudflared tunnel --config ~/.cloudflared/config.yml run
```

运行成功会输出一串日志，此时打开浏览器访问对应域名查看是否正常打开。

当然为了使得cloudflared tunnel服务在后台一直运行，在macos、linux等平台可以采取screen等方式进行后台保持。

注册服务

也可以通过注册服务的形式进行自启动，运行以下命令：

```
sudo cloudflared service install
```

如果您已经登录并有一个配置文件 ~/.cloudflared/ ，这些将被复制到 /etc/cloudflared/ .

如果您没有配置文件，则需要创建一个包含上面列出的字段的 config.yml 文件。

指定特定配置文件

如需指定特定的配置文件，您可以通过运行来传递自定义文件：

```
cloudflared --config CONFIG-FILE service install
```

然后，使用以下命令启动系统服务：

```
sudo systemctl start cloudflared
```

或者在启动时使用：

```
sudo systemctl enable cloudflared
```

基于Docker的环境

将此服务使用Docker运行，将~/.cloudflared目录的两个文件cert.pem和config.yml复制到Docker要挂载的目录下，比如/volume1/cloudflared，然后在运行docker命令，即可。

```
docker run -d -v /volume1/cloudflared:/etc/cloudflared cloudflare/cloudflared:2022
```

总结

难得发现这样一个不错的内网穿透服务，不过当前只是尝试了暴露了网页服务，还有ssh等服务还没尝试过，参考官方文档后续尝试下。

https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/install-and-setup (https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/install-and-setup)

参考资料

https://lxx.im/cloudflare-tunnel (https://lxx.im/cloudflare-tunnel)

https://blog.186526.xyz/post/argo-tunnel-for-free/ (https://blog.186526.xyz/post/argo-tunnel-for-free/)

https://lxx.im/cloudflare-tunnel (https://lxx.im/cloudflare-tunnel)

https://www.fwqzdm.com/mgvps/243520.html (https://www.fwqzdm.com/mgvps/243520.html)

https://1024.ee/index.php/2021/06/04/%E6%96%B0%E6%9C%8D%E5%8A%A1cloudflare-argo-tunnel%E7%94%A8%E4%BA%8E%E5%BB%BA%E7%AB%8B%E6%98%AF%E6%9C%89%E9%9A%A7%E9%81%93/ (https://1024.ee/index.php/2021/06/04/%E6%96%B0%E6%9C%8D%E5%8A%A1cloudflare-

emengweb 's Blog (/)

Figma YES!

argo-tunnel%E7%94%A8%E4%BA%8E%E5%BB%BA%E7%AB%8B%E6%98%AF%E6%9C%89%E9%9A%A7%E9%81%93/)

https://hub.docker.com/r/cloudflare/cloudflared (https://hub.docker.com/r/cloudflare/cloudflared)

https://blog.centminmod.com/2021/02/09/2250/how-to-setup-cloudflare-argo-tunnel-on-centos-7/#step-3-creating-cloudflared-yaml-config-file (https://blog.centminmod.com/2021/02/09/2250/how-to-setup-cloudflare-argo-tunnel-on-centos-7/#step-3-creating-cloudflared-yaml-config-file)

https://blog.cloudflare.com/zh-cn/network-onramp-partnerships-zh-cn/ (https://blog.cloudflare.com/zh-cn/network-onramp-partnerships-zh-cn/)

https://www.dll3.cn/534.html (https://www.dll3.cn/534.html)

https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/install-and-setup/tunnel-guide (https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/install-and-setup/tunnel-guide)

上一篇: Linux下查看实时网络流量的方法 (/p/Linux%E4%B8%8B%E6%9F%A5%E7%9C%8B%E5%AE%9E%E6%97%B6%E7%BD%91%E7%BB%9C%E6%B5%81%E9%87%8F%E7%

下一篇: linux查看磁盘空间df，内存空间free，文件大小du (/p/linux%E6%9F%A5%E7%9C%8B%E7%A3%81%E7%9B%98%E7%A9%BA%E9%97%B4df%EF%BC%8C%E5%86%85%E5%AD%98%E

导航

主页 (/)
Debian (/c/life)
CAT玉米 (/c/CAT%E7%8E%89%E7%B1%B3)
VI设计 (/c/VI%E8%AE%BE%E8%AE%A1)
NAS (/c/NAS)
OctoberCMS (/c/OctoberCMS)
UIKit3 (/c/UIKit3)
宝塔面板 (/c/%E5%AE%9D%E5%A1%94%E9%9D%A2%E6%9D%BF)
Python (/c/Python)
SVG相关 (/c/SVG%E7%9B%B8%E5%85%B3)
Figma (/c/Figma)
VPS独服务记录 (/c/VPS%E8%B8%A9%E5%9D%91%E8%AE%B0%E5%BD%95)
About Me (/single/About-Me)
归档 (/archives)
标签 (/tags)

最近发表

将kimsufi的机器从raid1改为raid0 (/p/%E5%B0%86kimsufi%E7%9A%84%E6%9C%BA%E5%99%A8%E4%BB%8Eraid1%E6%94%B9%E4%B8%BAraid0)
Debian11手动修改官方镜像源 (/p/Debian11%E6%89%8B%E5%8A%A8%E4%BF%AE%E6%94%B9%E5%AE%98%E6%96%B9%E9%95%9C%E5%83%8F%E6%BA
Debian快速配置网络IPv4&IPv6地址教程 (/p/Debian%E5%BF%AB%E9%80%9F%E9%85%8D%E7%BD%AE%E7%BD%91%E7%BB%9CIPv4-IPv6%E5%9C%B0%E5%9D%80%E6%95%99%E7%A8%8B)
彻底修复BT宝塔面板后台中文乱码，手动配置区域语言 (/p/%E5%BD%BB%E5%BA%95%E4%BF%AE%E5%A4%8DBT%E5%AE%9D%E5%A1%94%E9%9D%A2%E6%9D%BF%E5%90%8E%E
在Proxmox VE 中直接备份至Google Drive (vzdump + rclone) (/p/%E5%9C%A8Proxmox-VE-%E4%B8%AD%E7%9B%B4%E6%8E%A5%E5%A4%87%E4%BB%BD%E8%87%B3Google-Drive-vzdump-rclone)

友情链接

UI Day (http://uiday.com)

emengweb 's Blog (/)

Figma YES!