

IBMYes

本项目包括3部分

1. IBM Cloud Fonudray搭建V2Ray ws
2. 利用Github的Actions 每周重启 IBM Cloud Fonudray
3. Cloudflare 高速节点中转

使用IBM Cloud Fonudray搭建V2Ray

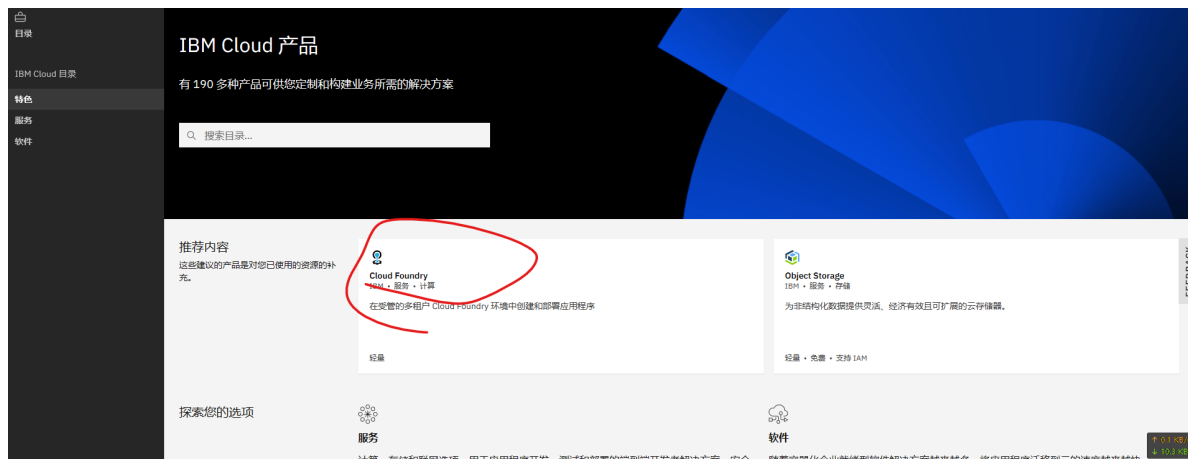
首先注册<https://cloud.ibm.com/>

注册步骤略过

登录后点击右侧 创建资源



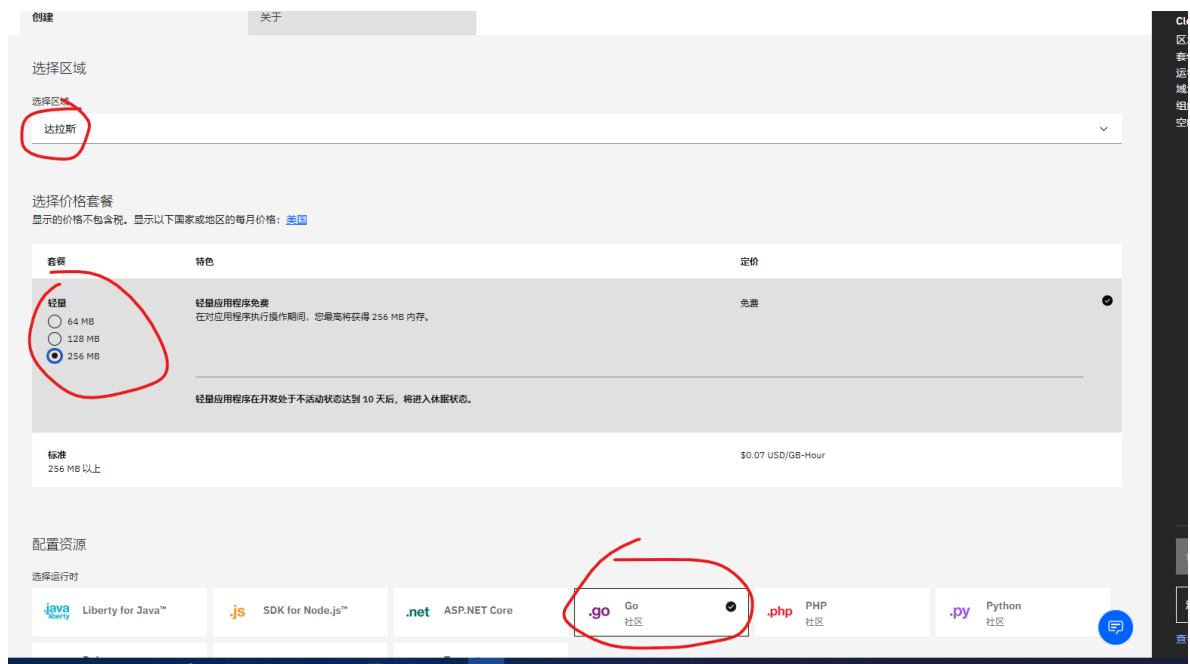
可以找到Cloud Foundry



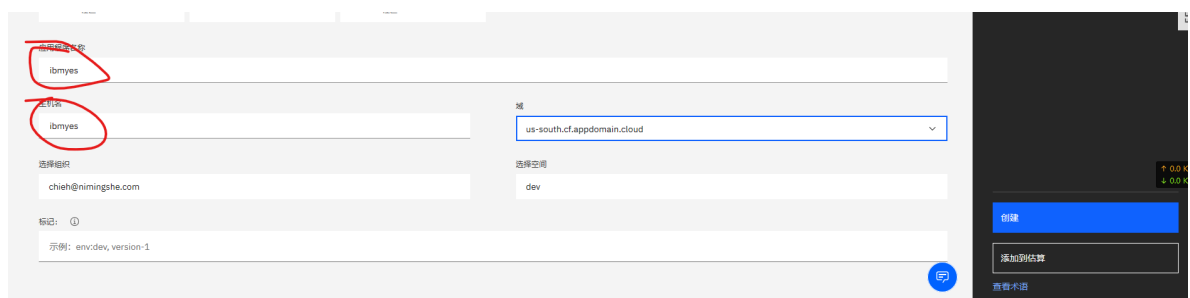
创建公共应用程序



填写相关信息



区域必须达拉斯，只有那里有免费的。



填写应用名称

接着进入右上角命令行



打开命令行粘贴一键安装脚本：

```
1 wget --no-check-certificate -O install.sh https://raw.githubusercontent.com/CCChieh/IBMYes/master/install.sh && chmod +x install.sh && ./install.sh
```

```
chieh@cloudshell:~$ wget --no-check-certificate -O install.sh https://raw.githubusercontent.com/CCChieh/IBMYes/master/install.sh && chmod +x install.sh && ./install.sh
--2020-06-15 13:04:08-- https://raw.githubusercontent.com/CCChieh/IBMYes/master/install.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 199.232.8.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|199.232.8.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1144 (1.1K) [text/plain]
Saving to: 'install.sh'

install.sh                               100%[=====>] 1.1
2020-06-15 13:04:09 (34.7 MB/s) - 'install.sh' saved [1144/1144]

进行初始化。。。

```

在配置的时候需要输入应用名称（这里就是我创建应用的时候输入应用名称我输入的是ibmyes，你需要改成你自己的名称）和应用内存大小（我们刚刚选择的是256）

```
进行配置。。。
请输入你的应用名称：ibmyes
应用名称：ibmyes
请输入你的应用内存大小(默认256)：
内存大小：256
配置完成。
进行安装。。。

```

配置好，等待几分钟，便可自动完成安装。

```
Waiting for app to start...

name:          ibmyes
requested state: started
routes:        ibmyes.us-south.cf.appdomain.cloud
last uploaded: Mon 15 Jun 13:06:17 UTC 2020
stack:         cflinuxfs3
buildpacks:    go

type:          web
instances:     1/1
memory usage:  256M
start command: ./v2ray/v2ray

   state   since                cpu    memory    disk    details
#0  running  2020-06-15T13:06:36Z  0.0%   0 of 256M  0 of 1G

安装完成。

```

然后访问我们刚刚的应用的域名，如果不记得可以返回我们刚才的资源，点击访问应用程序

IBM Cloud 搜索资源和产品...

资源列表 / **ibmmyes** 正在运行 访问应用程序... 添加标记

入门

概述

运行时

连接

日志

API Management

自动缩放

Availability Monitoring

实例 编辑

运行状况

100%

1 个实例 (共 1 个) 正在运行

每个实例 MB 内存

0 256 256

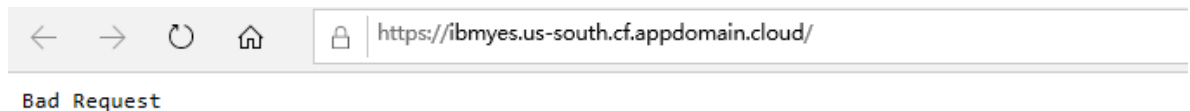
运行时开销

当前和估算开销不包括已连接的服务。

US\$ 0.00

连接 (0)

看到 Bad Request 便成功了



这里请记住你的域名

默认使用的配置是：

```
1 vmess://ew0KICAidiI6ICIyIiwNCiAgInBzIjogImlibXllcyIsDQogICJhZGQiOiAiaWJteWVZLnVzLXNvdXROlMmNmLmFwcGRvbWpbi5jbG91ZCIsDQogICJwb3J0IjogIjQ0MyIsDQogICJpZCI6ICI4YzM1YmVmMy1kNTFmLTQxYWItYWM4Ny03YjA1MzQxMDQ5NWIlLA0KICAiYWlkIjogIjY0IiwNCiAgIm5ldCI6ICJ3cyIsDQogICJ0eXB1IjogIm5vbmUiLA0KICAiaG9zdCI6ICIiLA0KICAicGF0aCI6ICIiLA0KICAidGxzIjogInRscyINCn0=
```

把上面的配置复制到你的v2rayN或v2rayNg中，修改地址为你的应用的域名（前面我们 Bad Request 那个网页的域名）。

导入配置文件

服务器

地址(address)

ibmyes.us-south.cf.appdomain.cloud

端口(port)

443

用户ID(id)

8c35bef3-d51f-41ab-ac87-7b053410495b

生成(G)

额外ID(alterId)

64

加密方式(security)

auto

*随便选, 建议(auto)

传输协议(network)

ws

*默认tcp, 选错会无法连接

别名(remarks)

ibmyes

*手填, 方便识别管理

不清楚则保持默认值

伪装类型(type)

none

*tcp或kcp或QUIC伪装类型, 默认none

伪装域名(host)

1)http host中间逗号(,)隔开
2)ws host
3)h2 host中间逗号(,)隔开
4)QUIC 加密方式

路径(path)

1)ws path
2)h2 path
3)QUIC 加密密钥

底层传输安全

tls

allowInsecure

默认true

确定(O)

取消(C)

至此我们已经有一个可用的v2ray了, 但是他每10天会重启一次, 而且网速延迟很差, 所以接下来会解决这个问题。

利用Github的Actions 每周重启 IBM Cloud Fonudray

IBM Cloud 10天不操作就会关机, 所以我们需要 十天内对其重启一次, 避免关机。

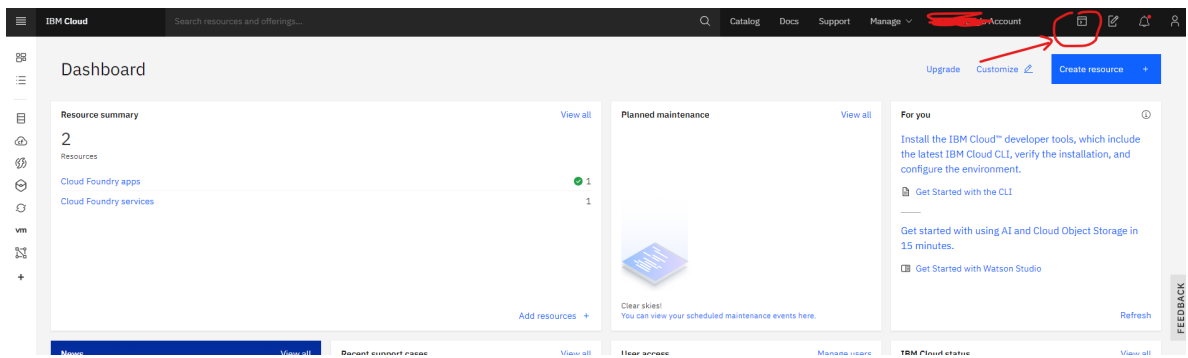
首先登录IBM Cloud

点击又上角的命令行

在这一步我们主要是记录4个值

```
1 IBM_ACCOUNT // IBM Cloud的登录邮箱和密码
2 IBM_APP_NAME // 应用的名称
3 REGION_NUM // 区域编码
4 RESOURCE_ID // 资源组ID
```

具体后面会一步一步完成

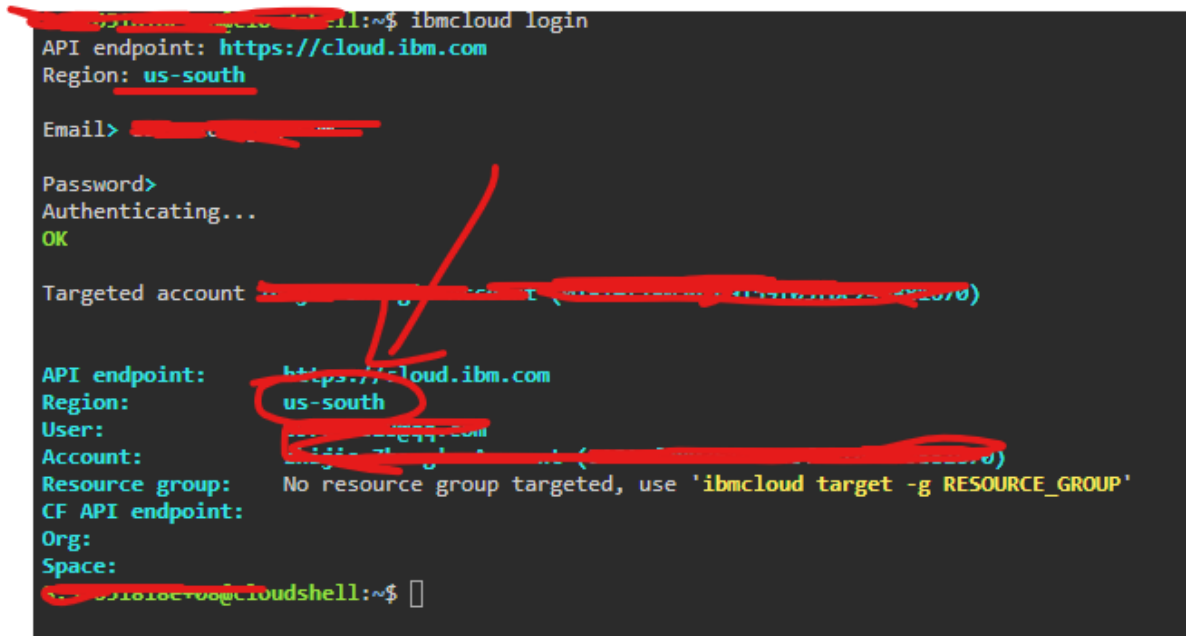


进入命令行先执行

```
1 | ibmcloud login
```

输入邮箱和密码。

之后记录下区域(Region)

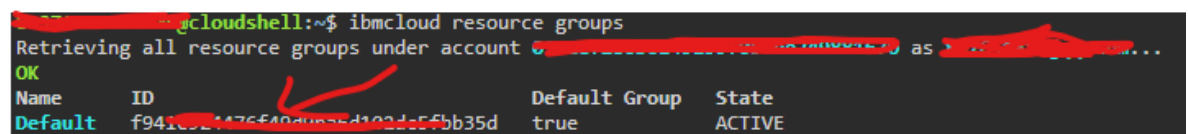


- 1 #1. au-syd
- 2 #2. in-che
- 3 #3. jp-tok
- 4 #4. kr-seo
- 5 #5. eu-de
- 6 #6. eu-gb
- 7 #7. us-south
- 8 #8. us-east

这里需要记下和区域对应的编号也就是 `REGION_NUM` , 比如我这里是us-south,那么我的区域编号是 7

接下来获取资源组id `RESOURCE_ID`

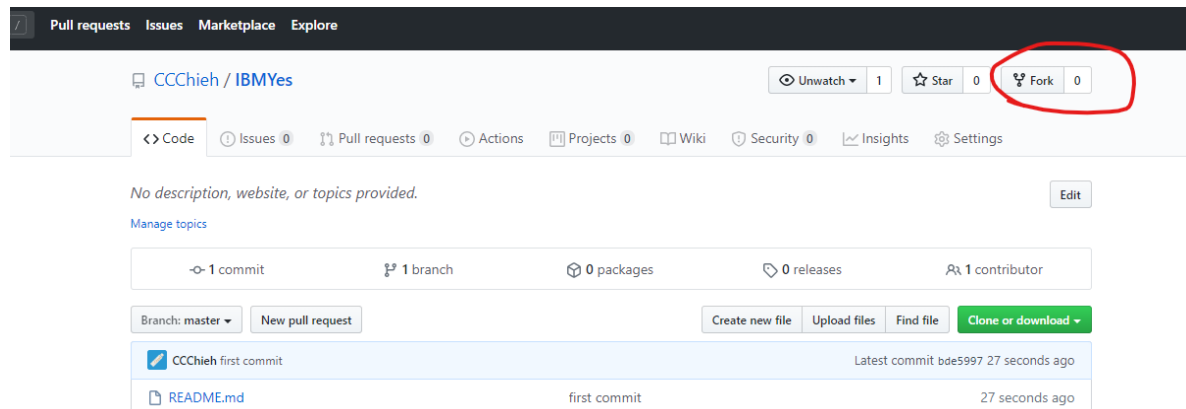
```
1 | ibmcloud resource groups
```



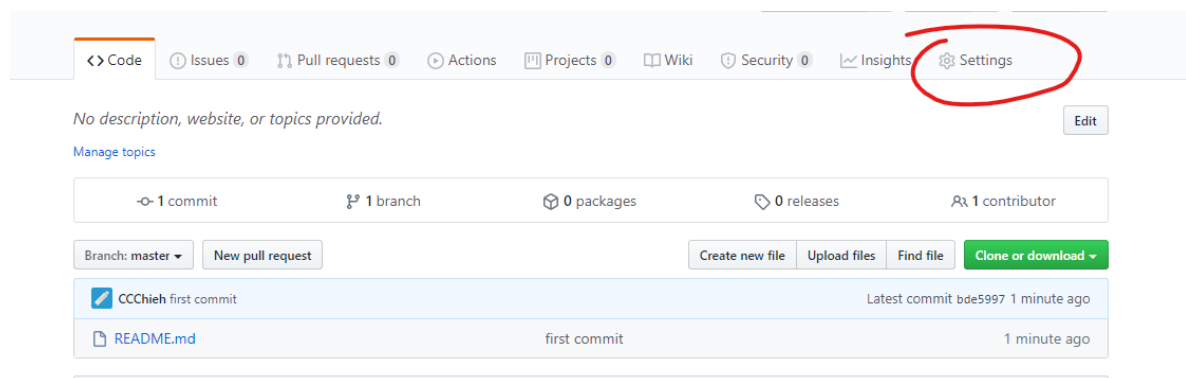
图中所指向便是 RESOURCE_ID

现在返回github，到本项目

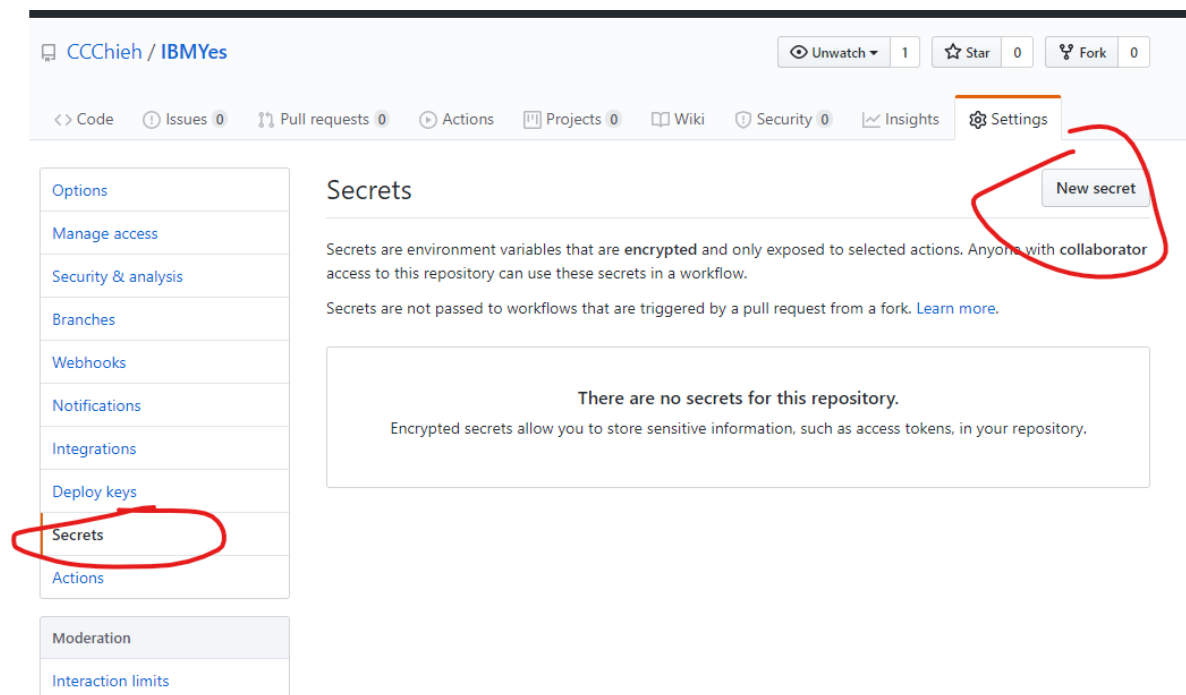
1 | <https://github.com/CCChieh/IBMYes>



右上角fork到自己的github下，然后进入setting



选择Secrets



New secret

分别建立四个secret

- 1 IBM_ACCOUNT // IBM Cloud的登录邮箱和密码
- 2 IBM_APP_NAME // 应用的名称
- 3 REGION_NUM // 区域编码
- 4 RESOURCE_ID // 资源组ID

以 IBM_ACCOUNT 为例

Secrets / New secret

Name

IBM_ACCOUNT

Value

your@email.com
yourpassword

Add secret

第一行为邮箱，第二行为密码。

这里需要邮箱和密码所以中间换行，其他的不需要换行。

把四个secret补充完成

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Secrets

Actions

Moderation

Interaction limits

Secrets

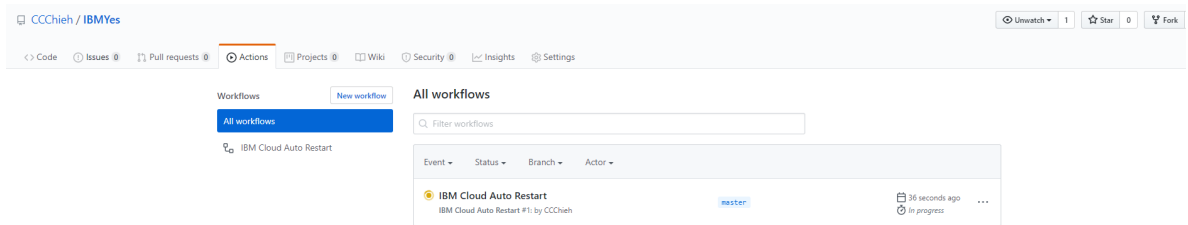
New secret

Secrets are environment variables that are **encrypted** and only exposed to selected actions. Anyone with **collaborator** access to this repository can use these secrets in a workflow.

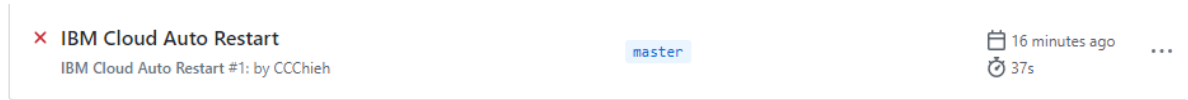
Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more](#).

IBM_ACCOUNT	Updated 2 minutes ago	Update	Remove
IBM_APP_NAME	Updated now	Update	Remove
REGION_NUM	Updated 42 seconds ago	Update	Remove
RESOURCE_ID	Updated 13 seconds ago	Update	Remove

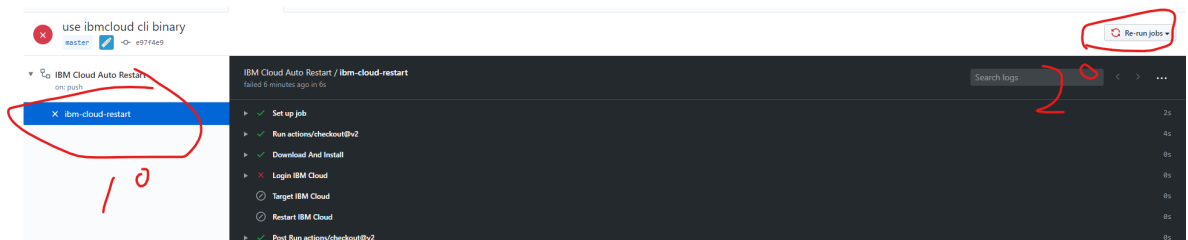
之后点击上方Actions，在这里你就会看到有个IBM Cloud Auto Restart在执行



第一次可能因为secret没添加导致workflow执行失败，只需要点下



进去后按照下图



找到 Re-run jobs 重新执行一次即可，至此自动重启已经ok了。

感谢药油@[My Flavor](#)，原本打算弄bash在自己服务器定期执行脚本，现在看了他的帖子，发现用 Actions是一个更好的选择。

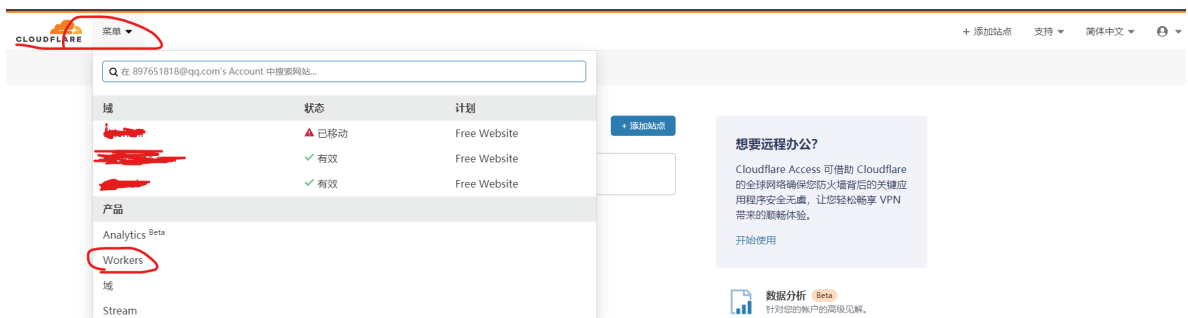
Cloudflare 高速节点中转

此部分贡献来自药油@[Joyace](#)、@[老婆](#)以及@[小俊博客](#)

cloudflare官网: <https://www.cloudflare.com/>

注册，登录这里不再赘述。

登录后左上角点击菜单找到workers



创建Worker

Workers KV 计划

Workers

通过 Workers 在边缘运行 JavaScript、Rust、C 和 C++ 等。

搜索

排序方式

创建 Worker

12:00凌晨 星期一 (UTC) - 1:41下午 星期一 (UTC)
今天的请求 424 / 100,000 Free

您的子域 **chieh.workers.dev** 更改

打开和复制脚本

```
1 addEventListener(  
2   "fetch", event => {  
3     let url=new URL(event.request.url);  
4     url.hostname="ibmyes.us-south.cf.appdomain.cloud";  
5     let request=new Request(url,event.request);  
6     event. respondWith(  
7       fetch(request)  
8     )  
9   }  
10  )
```

修改第四行为你的应用的域名

脚本 ●

```
1 addEventListener(  
2   "fetch",event => {  
3     let url=new URL(event.request.url);  
4     url.hostname="ibmyes.us-south.cf.appdomain.cloud";  
5     let request=new Request(url,event.request);  
6     event. respondWith(  
7       fetch(request)  
8     )  
9   }  
10  )
```

点击发送，测试是否出现 **Bad Request**，出现则成功，点击保存并部署。

< steep-shape-1a72 .chieh.workers.dev

脚本

```
1 addEventListener(  
2   "fetch",event => {  
3     let url=new URL(event.request.url);  
4     url.hostname="ibmyes.us-south.cf.appdomain.cloud";  
5     let request=new Request(url,event.request);  
6     event. respondWith(  
7       fetch(request)  
8     )  
9   }  
10  )
```

与 HTTP 预览

GET **发送**

标头 未设置标头 **+ 添加标头**

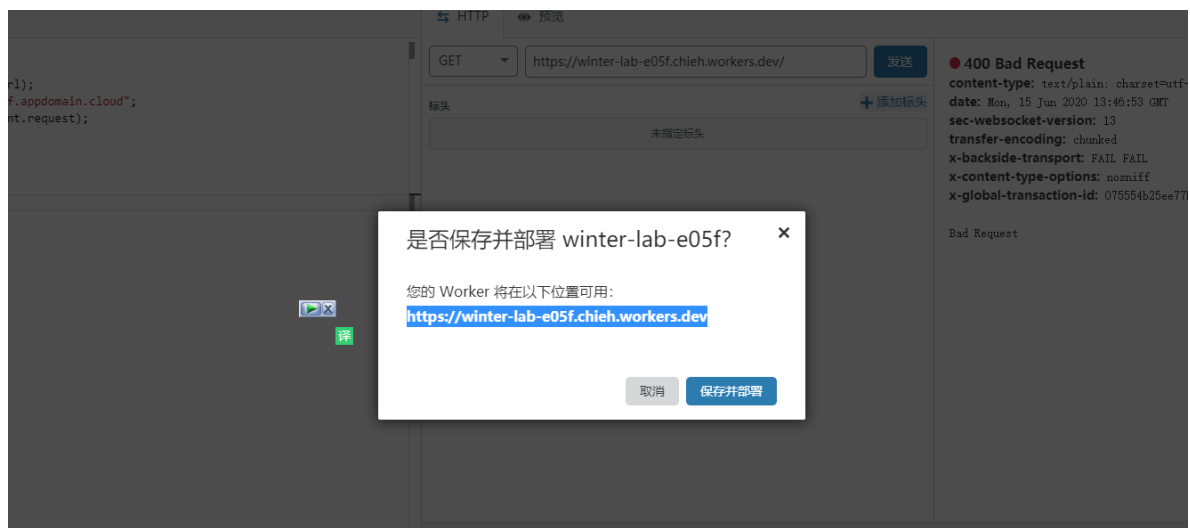
400 Bad Request
content-type: text/plain; charset=utf-8
date: Mon, 15 Jun 2020 13:44:47 GMT
sec-websocket-version: 13
transfer-encoding: chunked
x-backside-transport: FAIL FAIL
x-content-type-options: nosniff
x-global-transaction-id: 0428ff5e5ea77b4f0b622bc7

Bad Request

Console Sources Network

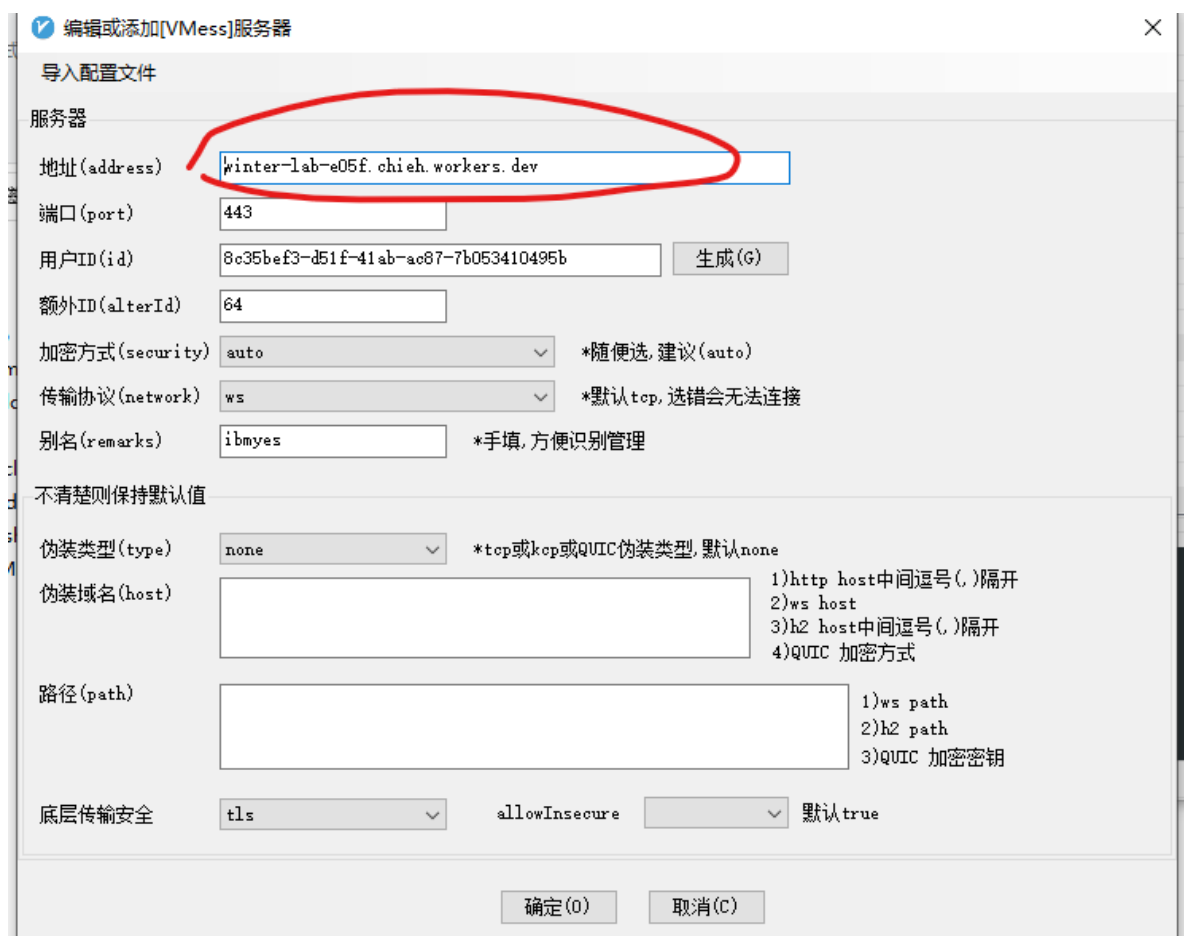
Worker Filter Default levels

上次保存时间: 6月 6日 1:42:00 下午 **保存并部署**



这里会给一个网址, *.workers.dev,这就是你的cloudflare中转后的域名。

然后我们去v2的客户端中修改地址



现在已经使用了cloudflare的代理。

下面我们将筛选cloudflare的高速节点。

克隆本项目到你的电脑上。

打开项目下的 fping-msys2.0 目录运行 自动查找最优CF节点-懒人专用.bat

名称	修改日期	类型	大小
1-自动查找100个丢包最少的IP.bat	2020/4/2 10:32	Windows 批处理...	243 KB
2-对100个丢包最少的IP测速.bat	2020/4/1 0:47	Windows 批处理...	1 KB
3-单IP测速.bat	2020/3/29 10:03	Windows 批处理...	1 KB
fping.exe	2020/4/2 13:01	应用程序	257 KB
msys-2.0.dll	2019/7/11 19:08	应用程序扩展	3,032 KB
使用说明.txt	2020/4/1 0:40	TXT 文件	2 KB
自动查找最优CF节点-懒人专用.bat	2020/4/2 10:33	Windows 批处理...	244 KB

类型: Windows 批处理文件
大小: 243 KB
修改日期: 2020/4/2 10:33

这里假设我获取的最优ip是 104.17.188.91

在客户端把地址换成ip，伪装域名换成我们cloudflare的workers的域名即可

编辑或添加[VMess]服务器

导入配置文件

服务器

地址(address)

端口(port)

用户ID(id) 生成(G)

额外ID(altId)

加密方式(security) *随便选, 建议(auto)

传输协议(network) *默认tcp, 选错会无法连接

别名(remarks) *手填, 方便识别管理

不清楚则保持默认值

伪装类型(type) *tcp或kcp或QUIC伪装类型, 默认none

伪装域名(host)

- 1)http host中间逗号(,)隔开
- 2)ws host
- 3)h2 host中间逗号(,)隔开
- 4)QUIC 加密方式

路径(path)

- 1)ws path
- 2)h2 path
- 3)QUIC 加密密钥

底层传输安全 allowInsecure 默认true

如果不方便用电脑优选ip也可以把地址设为 cloudflare.com 或 icook.tw,这两个一个cloudflare官网，自然也是使用自家cdn，另外一家是台湾省的一个网站，域名指向的ip一般也是比较好的线路。

编辑或添加[VMess]服务器

×

导入配置文件

服务器

地址(address)

cloudflare.com

端口(port)

443

用户ID(id)

8c35bef3-d51f-41ab-ac87-7b053410495b

生成(G)

额外ID(alterId)

64

加密方式(security)

auto

*随便选, 建议(auto)

传输协议(network)

ws

*默认tcp, 选错会无法连接

别名(remarks)

ibmyes

*手填, 方便识别管理

不清楚则保持默认值

伪装类型(type)

none

*tcp或kcp或QUIC伪装类型, 默认none

伪装域名(host)

winter-lab-e05f.chieh.workers.dev

1)http host中间逗号(,)隔开
2)ws host
3)h2 host中间逗号(,)隔开
4)QUIC 加密方式

路径(path)

1)ws path
2)h2 path
3)QUIC 加密密钥

底层传输安全

tls

allowInsecure

默认true

确定(O)

取消(C)

这里稍微提下原理吧，主要涉及CDN和请求头部，CDN识别流量是访问哪个网站的是根据请求头的Host来识别，所以这里要么host用我们的域名，要么我们伪装成我们的域名，这样都可以达到回源我们网站的请求。如果自己有域名也可以换自己的域名，域名也可以从第三方接入商cname，有兴趣的同学可以自己研究下。