补充说明，需要拉取github文件，需要有 **霍格沃茨的门票**，否则拉《jqlang/jq》项目的时候特别特别慢

—------------------

这份脚本逻辑大体是正确的，能够实现 IPv4 和 IPv6 双栈更新。

但是在**全新**的群晖（尤其是 DSM 7.x）上直接运行这段代码，会有 **2 个致命缺失**（缺依赖、缺配置注册）。

以下是完整的**检查与补漏**流程，请按顺序执行：

### 1. 补漏：安装必要的 jq 工具

全新的群晖系统**默认没有**安装 jq（JSON 解析工具），而你的脚本中大量使用了 jq。如果没有它，脚本会直接报错退出。

请先执行以下命令下载并安装一个静态编译的 jq 到系统路径：

Bash

# 下载 jq 二进制文件 (适配群晖通常的 x86\_64 架构)  
sudo curl -L -o /bin/jq https://github.com/jqlang/jq/releases/download/jq-1.6/jq-linux64  
  
# 赋予执行权限  
sudo chmod +x /bin/jq  
  
# 验证安装 (如果显示版本号则成功)  
jq --version

### 2. 检查并写入脚本 (优化版)

你的脚本在提取根域名时使用 awk -F. '{print $(NF-1)"."$NF}'。

* **注意**：这适用于 .com, .net, .xyz 等标准域名。
* **隐患**：如果你使用的是 .co.uk 或 .com.cn 这种二级后缀，脚本会错误提取为 co.uk。
* *如果你的域名是标准的（如 baidu.com），则无需修改，直接运行你的命令即可：*

Bash

sudo cat > /sbin/cloudflareddns.sh << 'EOF'  
#!/bin/bash  
  
# --- 核心配置 ---  
proxy="false" # 强制关闭 CDN 代理 (如需开启改为 true)  
ipv6="true" # 开启 IPv6 支持  
  
# 接收群晖传入参数  
username="$1" # 邮箱 (DSM 必填但脚本不使用)  
password="$2" # API Token  
hostname="$3" # 完整域名 (例如 nas.example.com)  
ipAddr="$4" # IPv4 地址  
  
# --- 环境检查 ---  
PATH=/bin:/sbin:/usr/bin:/usr/local/bin:/usr/local/sbin:$PATH  
  
# 自动获取公网 IPv6 地址 (增加备用源)  
if [ "$ipv6" = "true" ]; then  
 ip6Addr=$(curl -s -6 https://api6.ipify.org || curl -s -6 https://v6.ident.me || curl -s -6 https://ipv6.icanhazip.com)  
fi  
  
# --- 提取根域名 (Zone Name) ---  
# 简单的两段式提取 (不适用 .co.uk / .com.cn)  
zoneName=$(echo "$hostname" | awk -F. '{print $(NF-1)"."$NF}')  
  
# 1. 获取 Zone ID  
zoneId=$(curl -s -X GET "https://api.cloudflare.com/client/v4/zones?name=$zoneName" \  
 -H "Authorization: Bearer $password" \  
 -H "Content-Type: application/json" | jq -r '.result[0].id')  
  
if [ "$zoneId" = "null" ] || [ -z "$zoneId" ]; then  
 echo "badauth"  
 exit 1  
fi  
  
# 2. 获取 DNS 记录 ID (IPv4)  
recordId=$(curl -s -X GET "https://api.cloudflare.com/client/v4/zones/$zoneId/dns\_records?type=A&name=$hostname" \  
 -H "Authorization: Bearer $password" | jq -r '.result[0].id')  
  
# 获取 DNS 记录 ID (IPv6)  
if [ "$ipv6" = "true" ]; then  
 recordIdv6=$(curl -s -X GET "https://api.cloudflare.com/client/v4/zones/$zoneId/dns\_records?type=AAAA&name=$hostname" \  
 -H "Authorization: Bearer $password" | jq -r '.result[0].id')  
fi  
  
# API 定义  
createDnsApi="https://api.cloudflare.com/client/v4/zones/${zoneId}/dns\_records"  
updateDnsApi="https://api.cloudflare.com/client/v4/zones/${zoneId}/dns\_records/${recordId}"  
update6DnsApi="https://api.cloudflare.com/client/v4/zones/${zoneId}/dns\_records/${recordIdv6}"  
  
# 3. 更新 IPv4  
if [[ $recordId = "null" ]] || [[ -z $recordId ]]; then  
 res=$(curl -s -X POST "$createDnsApi" -H "Authorization: Bearer $password" -H "Content-Type:application/json" \  
 --data "{\"type\":\"A\",\"name\":\"$hostname\",\"content\":\"$ipAddr\",\"proxied\":$proxy}")  
else  
 res=$(curl -s -X PUT "$updateDnsApi" -H "Authorization: Bearer $password" -H "Content-Type:application/json" \  
 --data "{\"type\":\"A\",\"name\":\"$hostname\",\"content\":\"$ipAddr\",\"proxied\":$proxy}")  
fi  
resSuccess=$(echo "$res" | jq -r ".success")  
  
# 4. 更新 IPv6  
res6Success="false"  
if [[ "$ipv6" = "true" && -n "$ip6Addr" ]] ; then  
 if [[ $recordIdv6 = "null" ]] || [[ -z $recordIdv6 ]]; then  
 res6=$(curl -s -X POST "$createDnsApi" -H "Authorization: Bearer $password" -H "Content-Type:application/json" \  
 --data "{\"type\":\"AAAA\",\"name\":\"$hostname\",\"content\":\"$ip6Addr\",\"proxied\":$proxy}")  
 else  
 res6=$(curl -s -X PUT "$update6DnsApi" -H "Authorization: Bearer $password" -H "Content-Type:application/json" \  
 --data "{\"type\":\"AAAA\",\"name\":\"$hostname\",\"content\":\"$ip6Addr\",\"proxied\":$proxy}")  
 fi  
 res6Success=$(echo "$res6" | jq -r ".success")  
fi  
  
# 最终结果  
if [[ $resSuccess = "true" ]] || [[ $res6Success = "true" ]]; then  
 echo "good"  
else  
 echo "badauth"  
fi  
EOF  
  
# 赋予脚本执行权限  
sudo chmod +x /sbin/cloudflareddns.sh

### 3. 补漏：注册到群晖系统

光有脚本，群晖的 Web 界面是看不到这个服务商的。你必须手动将配置追加到 ddns\_provider.conf 文件中。

复制以下命令直接执行：

Bash

sudo sh -c 'cat >> /etc.defaults/ddns\_provider.conf <<EOF  
  
[Cloudflare\_Custom]  
 modulepath=/sbin/cloudflareddns.sh  
 queryurl=https://www.cloudflare.com/  
EOF'

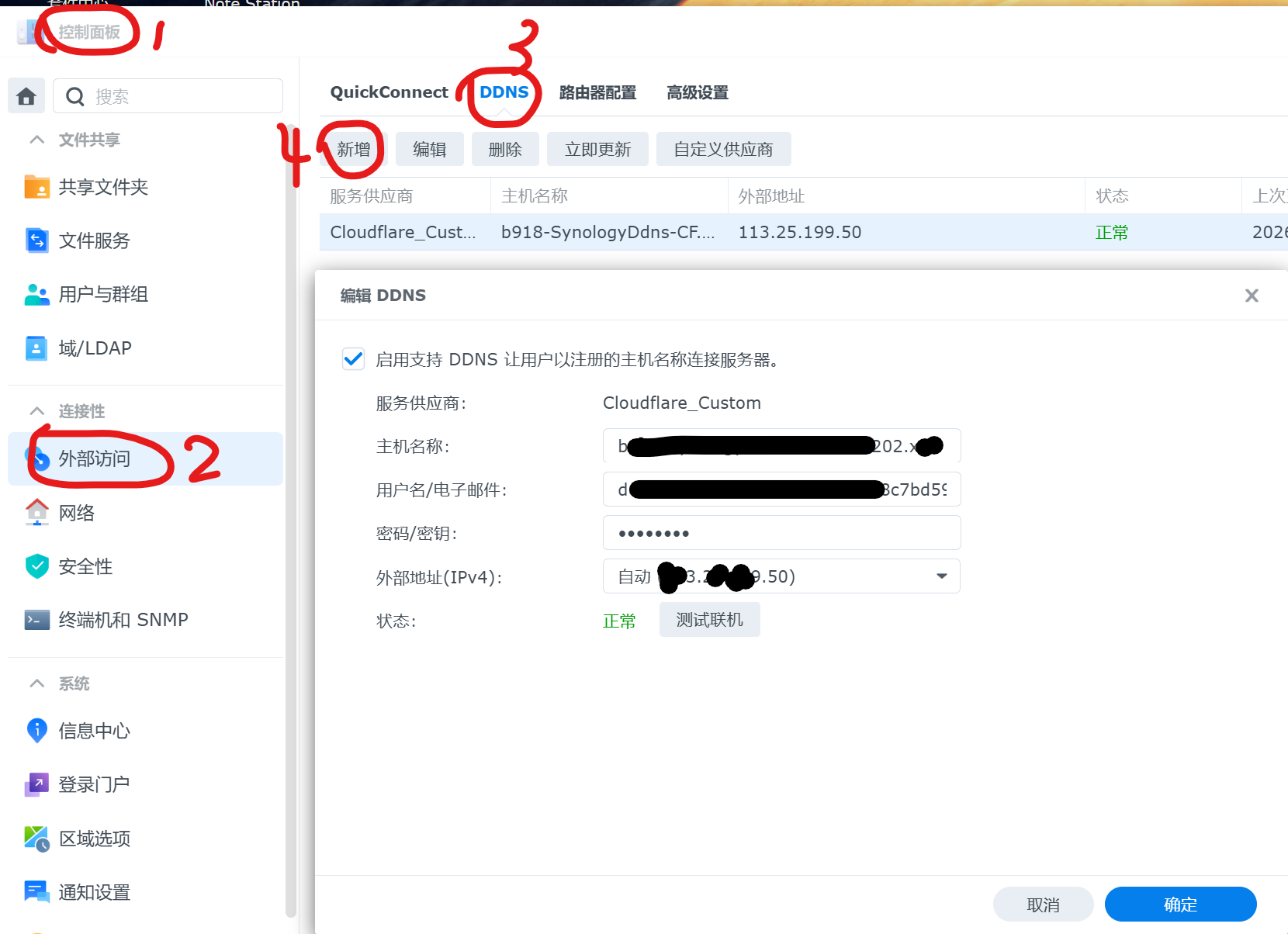
### 4. 验证与 Web 端设置

完成上述步骤后，进入群晖 DSM 界面：

1. **控制面板** > **外部访问** > **DDNS** > **新增**。
2. **服务提供商**：现在应该能看到 Cloudflare\_Custom 了，选中它。
3. **主机名称**：填写你的完整域名（如 nas.abc.com）。
4. **用户名/Email**：随意填写（脚本不验证此项，但 DSM 要求必填）。
5. **密码/Key**：填写 **Cloudflare API Token**。
   * *注意：不是 Global Key，是你在 CF 后台创建的只有 DNS 编辑权限的 Token。*
6. 点击 **测试联机**。

如果不成功（显示 badauth），请检查 Token 权限或 jq 是否安装成功。

—----------------------------------------------------------------



主机名称：你的域名（[nas.zhangsanfeng.com](http://nas.zhangsanfeng.com)）

用户名/电子邮件：你的域名- 区域 ID

密码/密钥：API令牌-DNS，api-tokens

外部地址ipv4：虽然你看的到ipv4，

但是在CF-你的域名-**dns记录**中会出现A和AAAA，**双解析**。