

申请免费的 Amazon 云服务器

【操作步骤】

1. 访问亚马逊云服务，注册账号
2. 申请 ec2 云服务器
3. 使用 puttygen 生成访问密钥
4. 使用 putty 访问云服务器
5. 云服务器防火墙设置

【参考】

- 亚马逊云服务网站

<https://aws.amazon.com/>

- puttygen 下载地址

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

- SUSE 下 nano 的安装

suse 的很多操作命令与其它 linux 有差别（大家可以选熟悉的 ubuntu 系统，它的操作基本与树莓派一致）

很多同学问 SUSE 下 nano 的安装，参考：<https://software.opensuse.org/download.html?project=editors&package=nano>

命令：

```
sudo zypper addrepo https://download.opensuse.org/repositories/editors/SLE_15/editors.repo
```

```
sudo zypper refresh
```

```
sudo zypper install nano
```

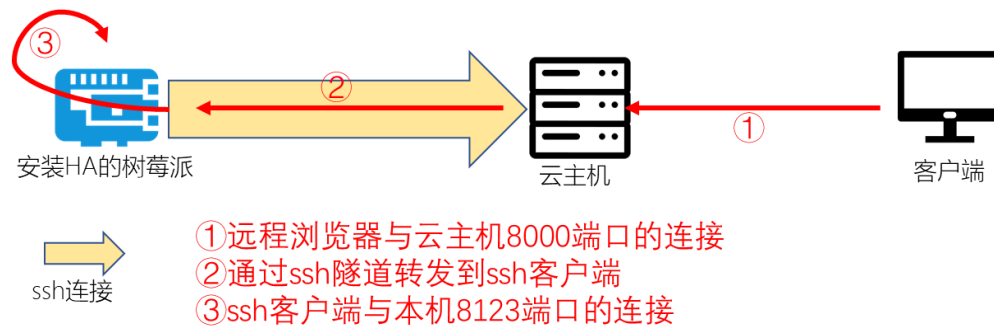
ssh 隧道构建

【操作步骤】

1. 在树莓派上使用 ssh 登录云主机
2. 构建 ssh 隧道
3. 安装与使用 autossh
4. 启动时自动执行 autossh 命令

【参考】

- 连接示意图



- 自启动/etc/rc.local 添加内容

```
sudo -u pi /usr/bin/autossh -i "/home/pi/etc/amazon_xinjiapo.pem" -R 0.0.0.0:8000:127.0.0.1:8123 ec2-user@ec2-54-251-155-96.ap-southeast-1.compute.amazonaws.com -N -f
```

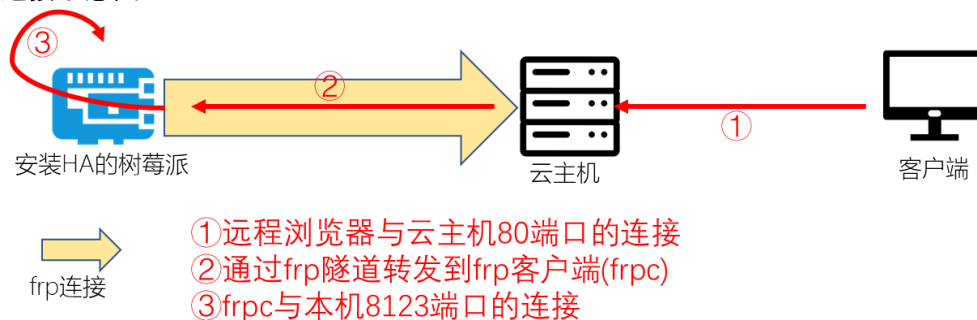
frp 隧道构建

【操作步骤】

1. 打开防火墙规则
2. 在云主机上下载、配置、运行 frp 服务器端
3. 在树莓派上下载、配置、运行 frp 客户端
4. 将 frps 和 frpc 命令分别加入云主机与树莓派的启动执行中

【参考】

- 连接示意图



- frp 软件 github 地址

<https://github.com/fatedier/frp/>

- frp 软件下载地址

<https://github.com/fatedier/frp/releases>

- frp 服务器端配置

[common]

bind_port = 7000

token = a1234

- frp 客户端配置

[common]

server_addr = ec2-54-251-155-96.ap-southeast-1.compute.amazonaws.com

server_port = 7000

token = a1234

[HA]

type = tcp

local_ip = 127.0.0.1

local_port = 8123

remote_port = 80

- frp 客户端自启动添加内容

注：前半部分在等待网络启动后才运行对应命令，其中网关地址需要修改为你实际环境中的地址

```
(
until ping -nq -c3 192.168.31.1; do
# Waiting for network
sleep 5
done
/home/pi/frp_0.21.0_linux_arm/frpc -c /home/pi/frp_0.21.0_linux_arm/frpc.ini
)&
```

为 HA 配上域名与数字证书

【操作步骤】

1. 申请一个免费的 duckdns.org 子域名
2. 下载 certbot-auto
3. 申请数字证书
4. 在 homeassistant 中配置 https 访问
5. 设置 frp 隧道中 443 远程端口映射
6. 更新数字证书

【参考】

- duckdns 官网
<https://www.duckdns.org/>
- Let's Encrypt 官网
<https://letsencrypt.org>
- certbot 官网
<https://certbot.eff.org/>
- certbot-auto 下载地址
<https://dl.eff.org/certbot-auto> 或 <https://github.com/certbot/certbot/raw/master/certbot-auto>
- 申请证书命令

```
sudo certbot-auto certonly --standalone --preferred-challenges http-01 --http-01-port 8123 --email your@email.address -d examplehome.duckdns.org
```
- 增加证书文件访问权限命令

```
sudo chmod 755 /etc/letsencrypt/live  
sudo chmod 755 /etc/letsencrypt/archive  
sudo chmod +r /etc/letsencrypt/archive -R
```
- HomeAssistant 中证书的配置样例
http:

```
ssl_certificate: /etc/letsencrypt/live/examplehome.duckdns.org/fullchain.pem  
ssl_key: /etc/letsencrypt/live/examplehome.duckdns.org/privkey.pem  
base_url: examplehome.duckdns.org
```
- 更新证书命令

```
sudo certbot-auto renew --standalone --preferred-challenges http-01 --http-01-port 8123  
sudo certbot-auto renew --standalone --preferred-challenges tls-sni-01 --tls-sni-01-port 8123 (已过期)
```

注:

以上命令执行前需要先停止 hass 以释放 8123 端口。

其中的 http-01 用于远程 80 端口映射到本地的 8123 端口的情况, tls-sni-01 用于远程的 443 端口映射到本地 8123 端口的情况 (tls-sni-01 更新方式已经不再被支持)。你也可以使用非 8123 端口, 这样就不用事先停止 home assistant 了, 但需要额外构建通讯隧道。

命令中可以加入 --pre-hook 和 --post-hook 参数用于指定更新前自动执行的停止 hass 和更新后启动 hass 的命令, 如-

```
-pre-hook "sudo systemctl stop home-assistant@pi"
```

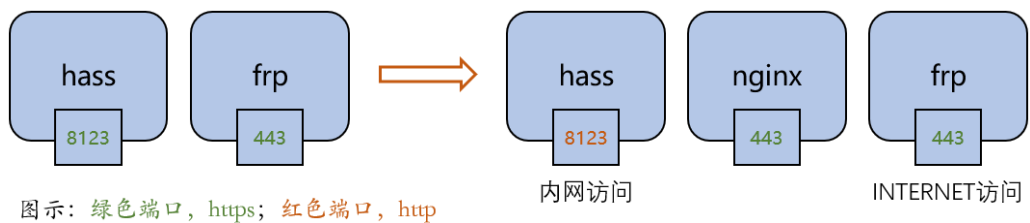
nginx 代理

【操作步骤】

1. 问题与解决方案（为什么要用 nginx 代理）
2. 安装 nginx
3. 修改 nginx 的配置
4. 修改 HA 配置
5. 修改 frp 配置
6. 演示

【参考】

- 示意图



- nginx 官网
<http://nginx.org/>

- 安装 nginx
`sudo apt-get install nginx`

- nginx 配置内容

```
server {  
    listen 443;  
    server_name zjs.duckdns.org;  
  
    ssl on;  
    ssl_certificate /etc/letsencrypt/live/zjs.duckdns.org/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/zjs.duckdns.org/privkey.pem;  
    ssl_prefer_server_ciphers on;  
  
    location / {  
        proxy_pass http://127.0.0.1:8123;  
        proxy_set_header Host $host;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection "upgrade";  
    }  
}
```

- 修改 nginx 配置命令

```
#编辑配置文件 ha_ssl  
sudo vi /etc/nginx/sites-available/ha_ssl  
#在 sites-enabled 目录下建立配置文件链接  
sudo ln -sf /etc/nginx/sites-available/ha_ssl /etc/nginx/sites-enabled/default  
#重新加载 nginx 配置  
sudo nginx -s reload
```