

# Porter节点部署

## 一、依赖环境

- 支持ubuntu、mac OS、 windows操作系统
- Go 1.12.2 及以上版本
- 均衡性+高带宽型服务器（100G SSD + 8G RAM + 100M带宽）

## 二、安装

从代码库Release处可以下载最新的Porter

## 三、使用源代码进行编译

porter节点可使用Makefile进行编译。执行如下命令

```
$ make
```

编译完成之后会在当前目录生成 `porter` 可执行程序

## 四、配置文件

配置文件名字目前为 `config.json`，具体含义如下

```
{
  Configuration: {
    "InterfaceName": 网卡别名, 通过ifconfig查看
    "InnerIP": 在云服务器上, 会有内网IP和外网IP之分, InnerIP表示内网;
    "PortTimeout": 7200, 当没有数据交互时, porter为某个代理机器保留端口的最大时长, 单位是秒;
    "PublicIP": "40.73.103.72", 在云服务器上, 会有内网IP和外网IP之分, PublicIP表示外网;
    "RandomPortBegin": 30000, 随机端口的其实端口值;
    "RandomPortRange": 10000, 随机端口的范围宽度; 例如当前的值为10000, 则随机端口的范围是[30000, 30000+10000];
    "TPort": 6007, TCP协议监听端口;
    "UPort": 6008, UDP协议监听端口;
    "KPort": 6009, KCP 协议监听端口;
    "QPort": 6010, QUIC协议监听端口;
    "LogDir": "./Log/", Porter log日志文件目录;
```

```
"NetworkID":1565267317, 网络ID;
"LogLevel": 1, 日志等级;
"PorterDBPath": "./PorterDB" Porter持久化目录(leveldb);
}
}
```

## 五、启动Porter

1. 使用TCP协议的Porter启动命令为:

```
$ nohup ./porter -protocol=tcp >/dev/null 2>nohup.log &
```

2. 使用UDP协议的Porter启动命令为:

```
$ nohup ./porter -protocol=udp >/dev/null 2>nohup.log &
```

3. 使用QUIC协议的Porter启动命令为:

```
$ nohup ./porter -protocol=quic >/dev/null 2>nohup.log &
```

4. 使用KCP协议的Porter启动命令为:

```
$ nohup ./porter -protocol=kcp >/dev/null 2>nohup.log &
```

命令解释:

--protocol: 启动的网络代理协议