Ubuntu Server 18.04 LTS 使用Shadowsocks-ShadowsocksR访问互联网

mystery0.vip/2018/08/23/Ubuntu Server 18.04 LTS 使用Shadowsocks-ShadowsocksR访问互联网

说在前面

由于一时兴起，搞了一台服务器主机，然后想试试在这台服务器主机上编译一下 LineageOS 15.1 看看效果。所以在弄好其他的东西之后开始拉源码，首先是编译工具的安装，这个基本上没问题，但是在拉源码的时候出了问题，要么是速度只有 20k ，要么就是直接出问题，清华源实在是……emmmmmmmm……
在经过两天的尝试之后，算是摸出了一个办法可以使得拉源码的速度达到带宽的满速（我的带宽是 20m ），故此记录。

配置列表

1. 服务器主机系统：**Ubuntu 18.04.1 LTS x86_64**
2. 一个SS/SSR节点（如果没有的可以通过我的推广链接购买一个）
   对于翻墙服务的可靠性不做保证
   对于翻墙服务的可靠性不做保证！
   对于翻墙服务的可靠性不做保证！
   这个机场是在TG群里随手收藏的一个，今天是我第一次购买，也是想试一下是不是因为我的自建节点有问题所以导致速度慢。

开始

安装Shadowsocks/ShadowsocksR

根据你的节点是 SS 还是 SSR 选择安装

Shadowsocks安装

由于是 server ，所以在这里我只介绍命令行的安装。

shadowsocks 运行的时候通过 `python` ，所以需要先安装 `python`

```
1 sudo apt-get install
  python
```

接着安装 python 的包管理器 pip

```
1 sudo apt-get install
  python-pip
```

之后我们通过 pip 直接安装 shadowsocks

```
sudo pip install
shadowsocks
```

配置Shadowsocks

在任意地方新建一个配置文件`shadowsocks.json`（在这里我放在`/etc/shadowsocks/shadowsocks.json`中），然后配置相关参数

```
{
  "server": "{your-
server}",
  "server_port": 12345,
  "local_port": 1080,
  "password": "{your-
password}",
  "timeout": 600,
  "method": "aes-256-cfb"
}
```

注意替换配置中的相关节点信息
注意替换配置中的相关节点信息
注意替换配置中的相关节点信息

启动shadowsocks服务

```
sudo sslocal -c /etc/shadowsocks/shadowsocks.json -d
start
```

至此， Shadowsocks 的安装和配置完成

ShadowsocksR安装

在此我们不讨论SSR和SS的爱恨情仇，你想用哪个就用哪个，喜欢哪个就用哪个
在客户端安装 SSR 按理说也很简单，但是我当初找了很久才找到一个脚本，其他的基本上都是服务端的安装。

安装git

这个 SSR 脚本会使用git自动将SSR下载到本地，所以在这里我们先安装git

```
sudo apt-get
install git
```

安装ShadowsocksR

或者 SSR 脚本

```
wget
https://github.com/the0demiurge/CharlesScripts/raw/master/charles/bin/ssr
```

这个脚本算是写的比较完善了，里面封装了 SSR 的安装、配置、启动、关闭等功能。
为了方便操作，我们将脚本放进 `/usr/local/bin` 中

```
sudo mv ssr /usr/local/bin
sudo chmod 766
/usr/local/bin/ssr
```

接下来我们通过脚本安装SSR

```
ssr
instal
l
```

配置ShadowsocksR

第一次安装完成之后回自动打开配置文件，如下所示：

```
{
    "server": "{your-server}",
    "server_ipv6": "::",
    "server_port": 12345,
    "local_address": "127.0.0.1",
    "local_port": 1080,

    "password": "{your-password}",
    "method": "aes-256-cfb",
    "protocol": "origin",
    "protocol_param": "",
    "obfs": "plain",
    "obfs_param": "",
    "speed_limit_per_con": 0,
    "speed_limit_per_user": 0,

    "additional_ports" : {}, // only works under multi-user mode
    "additional_ports_only" : false, // only works under multi-user
mode
    "timeout": 120,
    "udp_timeout": 60,
    "dns_ipv6": false,
    "connect_verbose_info": 0,
    "redirect": "",
    "fast_open": false
}
```

注意替换配置中的相关节点信息
注意替换配置中的相关节点信息
注意替换配置中的相关节点信息

配置好之后会自动启动SSR，如果输出的日志没有提示错误的话，那么 ShadowsocksR 的配置就完成了。

将SOCKS代理转换为HTTP代理

我尝试过 `privoxy` 和 `polipo` ，最终发现 `privoxy` 好一些。

安装配置privoxy

如果你想使用 `polipo` ，请跳过本节。

```
1 sudo apt-get install privoxy #安装
2 privoxy
  sudo vim /etc/privoxy/config #配置
  privoxy
```

通过 `/` 搜索 `listen-address 192.168.0.1:8118` ，然后将前面的 `#` 去掉，然后修改为 `listen-address 127.0.0.1:8118` （端口号可以自行更改，后面同理）；同理，取消 `listen-address [::1]:8118` 前面的注释，不用修改。
搜索 `forward-socks5t` ，然后取消注释（没找到可以新建一行），然后将内容修改为如下内容：

```
1 forward-socks5t   /
  127.0.0.1:1080 .
```

注意最后的那个点是必须写的。
注意最后的那个点是必须写的。
注意最后的那个点是必须写的。

重启privoxy

```
1 sudo service privoxy
  start
```

安装配置polipo

如果你已经安装了 `privoxy` 可以跳过这一步。

```
1 sudo apt-get install
2 polipo
  sudo vim
  /etc/polipo/config
```

配置文件修改如下：

```
logSyslog = true
logFile =
/var/log/polipo/polipo.log

proxyAddress = "0.0.0.0"

socksParentProxy =
"127.0.0.1:1080"
socksProxyType = socks5

chunkHighMark = 50331648
objectHighMark = 16384

serverMaxSlots = 64
serverSlots = 16
serverSlots1 = 32
```

由于对 `polipo` 不是很了解，所以具体是什么意思我也不清楚。

重启polipo

```
sudo /etc/init.d/polipo
restart
```

使用 `privoxy` 默认的HTTP代理端口是 `8118` ，使用 `polipo` 默认的HTTP代理端口是 `8123`
使用 `privoxy` 默认的HTTP代理端口是 `8118` ，使用 `polipo` 默认的HTTP代理端口是 `8123`
使用 `privoxy` 默认的HTTP代理端口是 `8118` ，使用 `polipo` 默认的HTTP代理端口是 `8123`

剩下的就看你想怎么使用了，现在我的HTTP代理的端口是 `8118` ，如果我想让 apt 走代理，那么执行下面的操作配置环境变量

```
# 设置http 和 https 全局代理
export
http_proxy='http://localhost:8118'
export
https_proxy='http://localhost:8118'
```

取消代理：

```
unset
http_proxy
unset
https_proxy
```

测试代理是否可用：

```
curl
www.google.com.hk
```

如果有输出证明代理成功。

如果是像我一样拉LineageOS的源码编译，那么就给 `git` 配置代理，不用 `HTTP` 代理，直接使用 `SOCKS` 代理：

```
1 git config --global http.proxy
2 'socks5://127.0.0.1:1080'
  git config --global https.proxy
  'socks5://127.0.0.1:1080'
```

取消代理：

```
1 git config --global --unset
2 http.proxy
  git config --global --unset
  https.proxy
```

至此，本文完~