

一分钟实现内网穿透 (ngrok 服务器搭建)

2017 年 09 月 05 日 10:47:45 张果 阅读数 307628 标签: ubuntu 内网穿透 ngrok 更多

个人分类: 服务器

版权声明: 欢迎转载, 请声明作者信息与原文链接

<https://blog.csdn.net/zhangguo5/article/details/77848658>

简单来说内网穿透的目的是: 让外网能访问你本地的应用, 例如在外网打开你本地 <http://127.0.0.1> 指向的 Web 站点。

最近公司的花生壳到期了, 要续费, 发现价格一直在涨, 都是 5 年以上的老用户, 旗舰版都没有实现内网完全穿透, 打算自己动手替换这个服务, 中间走了不少的弯路, 这里记录一些文字为大家提供参考。

随着开发与运行移动互联网的应用越来越多对打通内外网的需要也更加迫切, 如微信开发、IOS 与 Android 开发等。

虽然租用 VPS、ECS 等服务器可以解决很多问题但高性能的外网服务器价格非常贵还有数据安全, 我选择的是公网服务器仅做代理与轻量应用, 复杂的应用部署到内网服务器再穿透访问。

一、内网穿透概要

为了理解内网穿透我们先来了解几个概念:

1.1、IP 地址

网络中唯一定位一台设备的逻辑地址, 类似我们的电话号码

在互联网中我们访问一个网站或使用一个网络服务最终都需要通过 IP 定位到每一台主机, 如访问 baidu 网站:

```
C:\> 管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping www.baidu.com

正在 Ping www.a.shifen.com [119.75.213.61] 具有 32 字节的数据:
来自 119.75.213.61 的回复: 字节=32 时间=44ms TTL=52
来自 119.75.213.61 的回复: 字节=32 时间=43ms TTL=52
来自 119.75.213.61 的回复: 字节=32 时间=45ms TTL=52
来自 119.75.213.61 的回复: 字节=32 时间=43ms TTL=52
```

其中 119.75.213.61 就是一个公网的 IP 地址，他最终指向了一台服务器。

IP 地址是 IP 协议提供的一种统一的地址格式，它为互联网上的每一个网络和每一台主机分配一个逻辑地址，以此来屏蔽物理地址的差异。

内网 IP 可以同时出现在多个不同的局域网络中，如 A 公司的 U1 用户获得了 192.168.0.5，B 公司的 U3 用户也可以获得 192.168.0.5；但公网 IP 是唯一的，因为我们只有一个 Internet。

//局域网可使用的网段（私网地址段）有三大段：10.0.0.0~10.255.255.255（A 类）
172.16.0.0~172.31.255.255（B 类）192.168.0.0~192.168.255.255（C 类）

1.2、域名

域名是 IP 的别名，便于记忆，域名最终通过 DNS 解析成 IP 地址。

```
管理员: C:\Windows\system32\cmd.exe

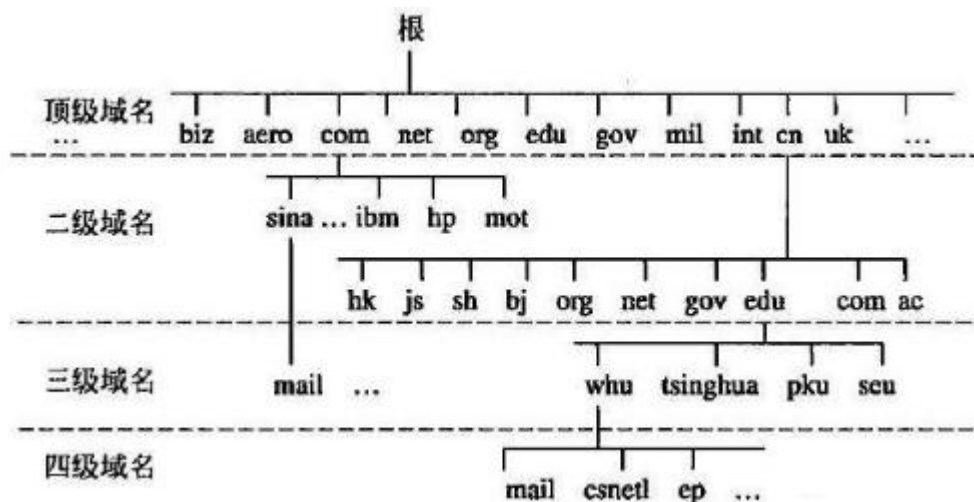
C:\Users\Administrator>nslookup www.baidu.com
服务器:  public1.114dns.com
Address:  114.114.114.114

非权威应答:
名称:      www.a.shifen.com
Addresses:  119.75.216.20
            119.75.213.61
Aliases:    www.baidu.com

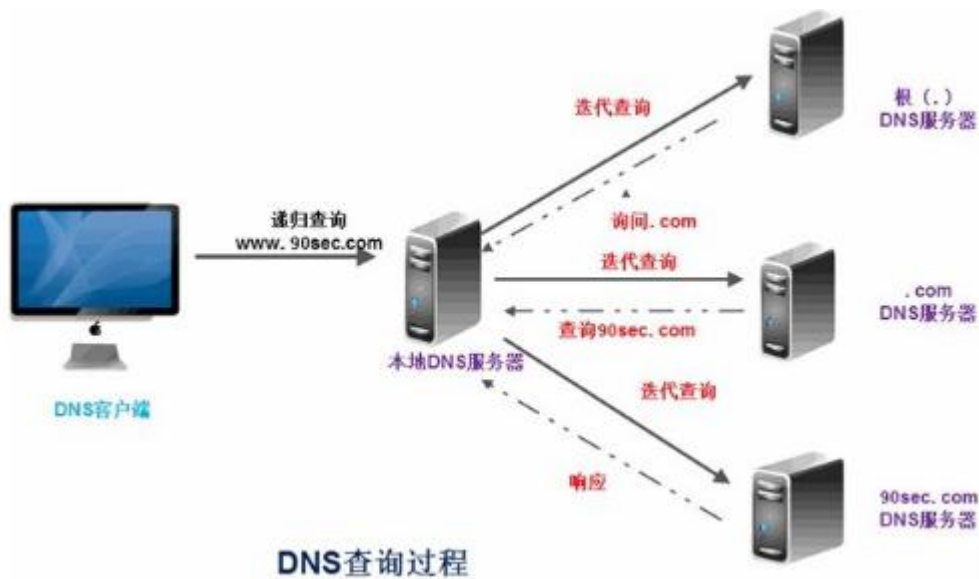
C:\Users\Administrator>
```

IP V4 是一个 32 位的数字，IP V6 有 128 位，要记住一串毫无意义的数字非常困难，域名解决了这个问题。

如 www.zhangguo.com.cn 就是一个域名，cn 表示地区，com 表示商业机构，zhangguo 是公司名称，www 是主机名



DNS 查询过程如下，最终将域名变成 IP 地址



1.3、NAT

NAT (Network Address Translation) 即网络地址转换，NAT 能将其本地地址转换成全球 IP 地址。

内网的一些主机本来已经分配到了本地 IP 地址（如局域网 DHCP 分配的 IP），但现在又想和因特网上的主机通信（并不需要加密）时，可使用 NAT 方法。

通过使用少量的公有 IP 地址代表较多的私有 IP 地址的方式，将有助于减缓可用的 IP 地址空间的枯竭。

NAT 不仅能解决了 IP 地址不足与共享上网的问题，而且还能够有效地避免来自网络外部的攻击，隐藏并保护网络内部的计算机。

多路由器可完成 NAT 功能。

NAT 的实现方式：

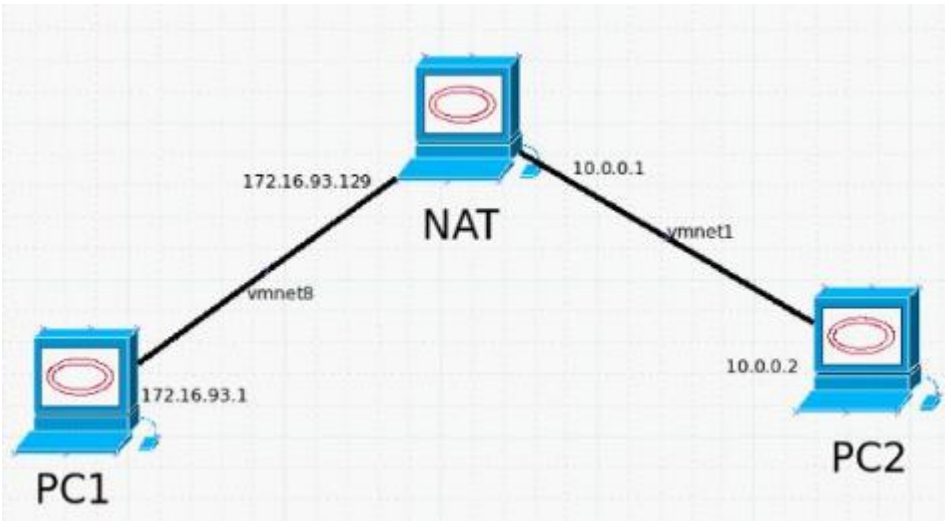
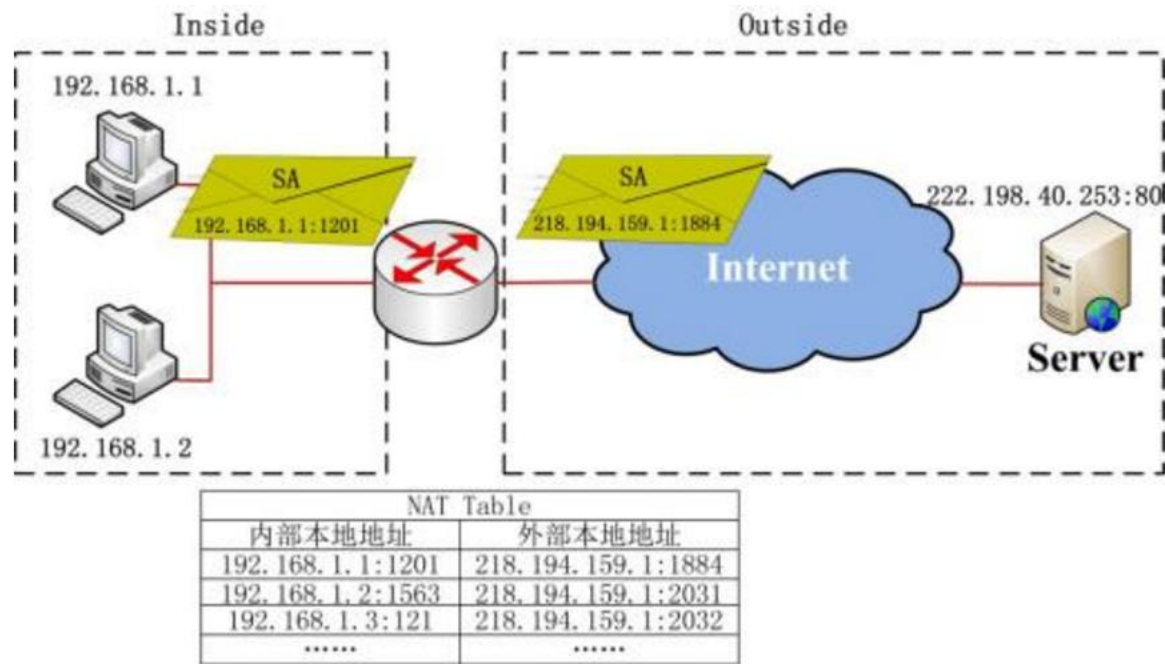
静态转换是指将内部网络的私有 IP 地址转换为公有 IP 地址，IP 地址对是一对一。

动态转换是指将内部网络的私有 IP 地址转换为公用 IP 地址时，IP 地址是不确定的，是随机的。

端口多路复用（Port address Translation,PAT），内部网络的所有主机均可共享一个合法外部 IP 地址实现对 Internet 的访问，从而可以最大限度地节约 IP 地址资源。同时又可隐藏

网络内部的所有主机，有效避免来自 internet 的攻击。因此，目前网络中应用最多的就是端口多路复用方式。

应用程序级网关技术（Application Level Gateway）ALG：传统的 NAT 技术只对 IP 层和传输层头部进行转换处理，ALG 它能为这些应用程序在通信时所包含的地址信息也进行相应的 NAT 转换。



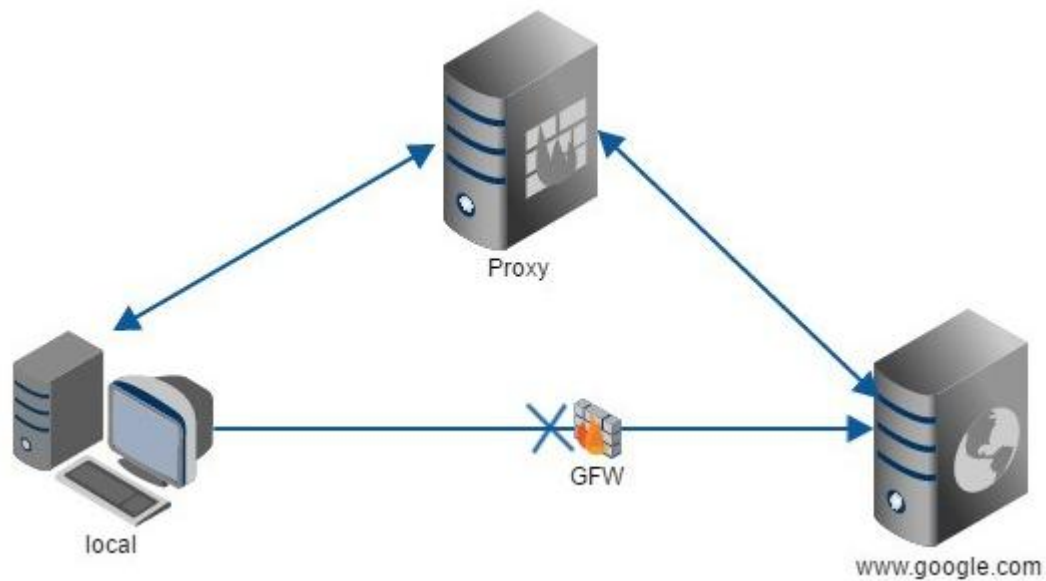
1.4、Proxy

Proxy 即代理，被广泛应用于计算机领域，主要分为正向代理与反向代理：

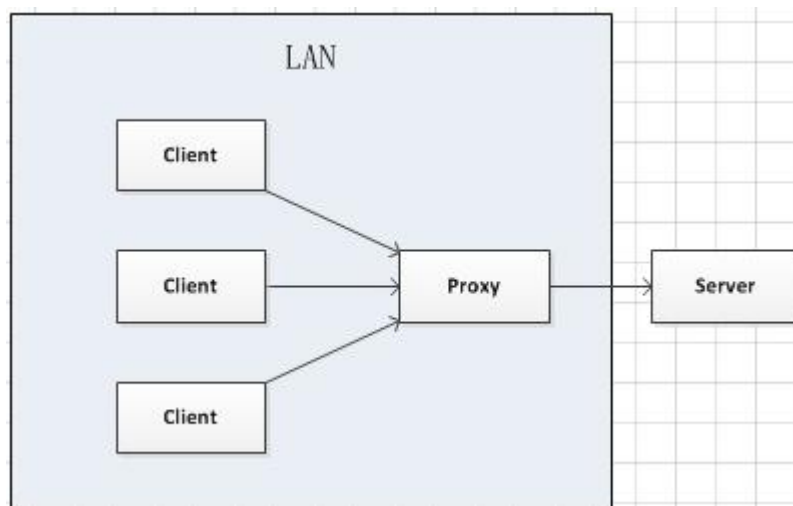
1.4.1、正向代理

比如 X 花店代 A,B,C,D,E 五位男生向 Candy 女生送匿名的生日鲜花，这里的 X 花店就是 5 位顾客的代理，花店代理的是客户，隐藏的是客户。这就是我们常说的代理。

正向代理隐藏了真实的请求客户端。服务端不知道真实的客户端是谁，客户端请求的服务都被代理服务器代替来请求，某些科学上网工具扮演的就是典型的正向代理角色。用浏览器访问 <http://www.google.com> 时被墙了，于是你可以在国外搭建一台代理服务器，让代理帮我去请求 [google.com](http://www.google.com)，代理把请求返回的相应结构再返回给我。



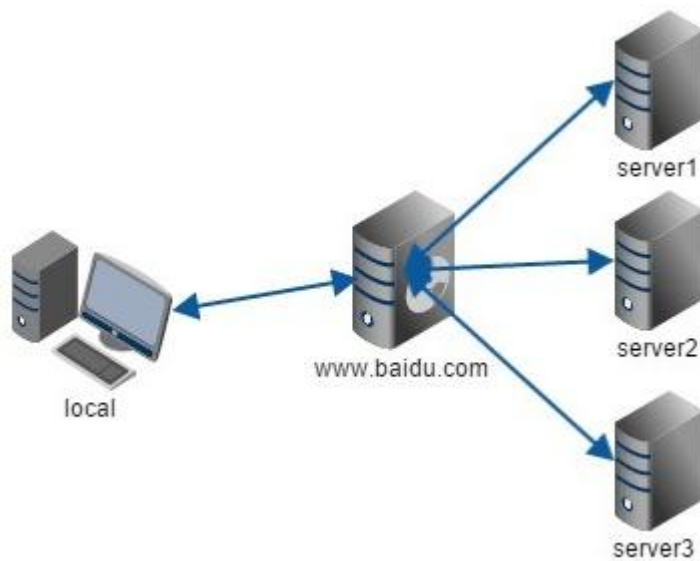
当多个客户端访问服务器时服务器不知道真正访问自己的客户端是那一台。正向代理中,proxy 和 client 同属一个 LAN,对 server 透明;



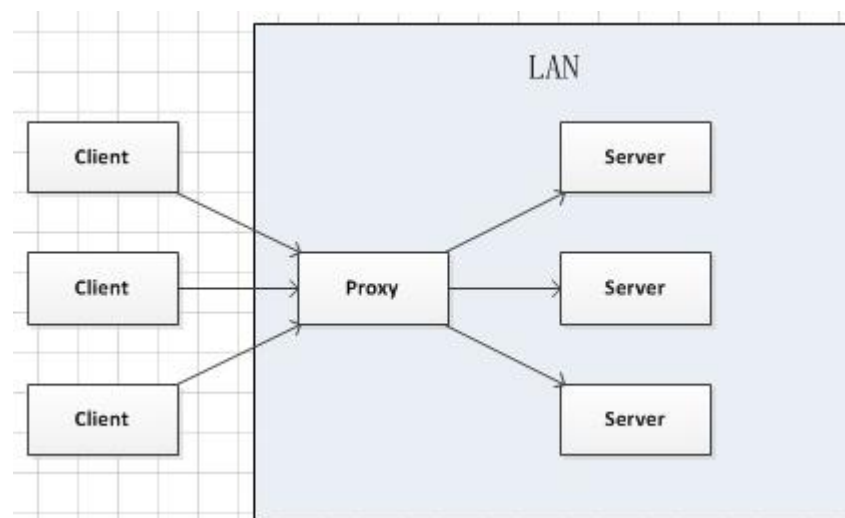
1.4.2、反向代理

拨打 10086 客服电话，接线员可能有很多个，调度器会智能的分配一个接线员与你通话。这里的调度器就是一个代理，只不过他代理的是接线员，客户端不能确定真正与自己通话的人，隐藏与保护的是目标对象。

反向代理隐藏了真实的服务端，当我们请求 `www.baidu.com` 的时候，就像拨打 10086 一样，背后可能有成千上万台服务器为我们服务，但具体是哪一台，你不知道，也不需要知道，你只需要知道反向代理服务器是谁就好了，`www.baidu.com` 就是我们的反向代理服务器，反向代理服务器会帮我们吧请求转发到真实的服务器那里去。Nginx 就是性能非常好的反向代理服务器，用来做负载均衡。



反向代理中,proxy 和 server 同属一个 LAN,对 client 透明。



[了解更多关于代理内容请点击这里。](#)

1.5、DDNS

DDNS 即动态域名解析，是**将用户的动态 IP 地址映射到一个固定的域名解析服务上**，用户每次连接网络的时候，客户端程序就会通过信息传递把该主机的动态 IP 地址传送给位于服务商主机上的服务器程序，服务程序负责提供 DNS 服务并实现动态域名解析。就是说 DDNS 捕获用户每次变化的 IP 地址，然后将其与域名相对应，这样域名就可以始终解析到非固定 IP 的服务器上，互联网用户通过本地的域名服务器获得网站域名的 IP 地址，从而可以访问网站的服务。

1.6、为什么需要内网穿透

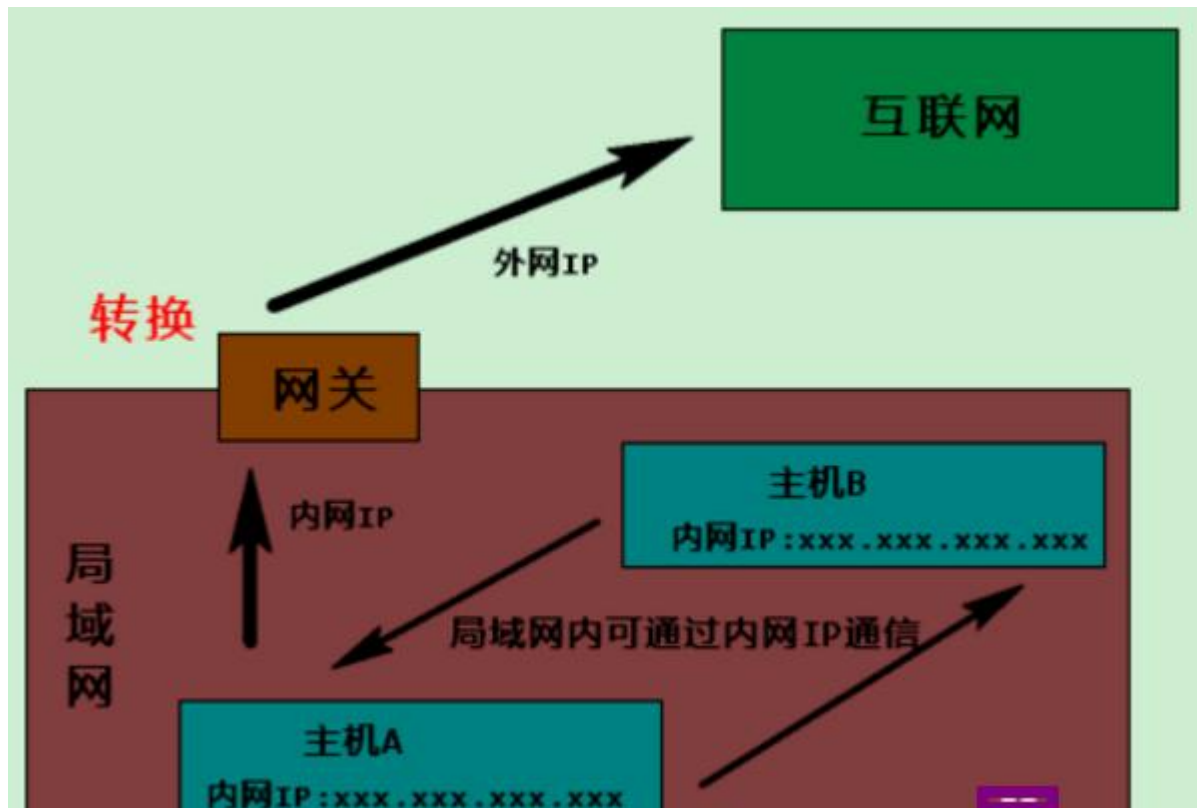
当内网中的主机没有静态 IP 地址要被外网稳定访问时可以使用内网穿透

在互联网中唯一定位一台主机的方法是通过公网的 IP 地址，但固定 IP 是一种非常稀缺的资源，不可能给每个公司都分配一个，且许多中小公司不愿意为高昂的费用买单，多数公司直接或间接的拨号上网，电信部门会给接入网络的用户分配 IP 地址，以前上网用户少的时候基本分配的都是临时的静态 IP 地址，租约过了之后可能会更换成另一个 IP 地址，这样外网访问就不稳定，因为内网的静态 IP 地址一直变化，为了解决这个问题可以使用动态域名解析的办法变换域名指向的静态 IP 地址。但是现在越来越多的上网用户使得临时分配的静态 IP 地址也不够用了，电信部门开始分配一些虚拟的静态 IP 地址，这些 IP 是公网不能直接访问的，如以 125 开头的一些 IP 地址，以前单纯的动态域名解析就不好用了。

1.7、内网穿透的定义与障碍

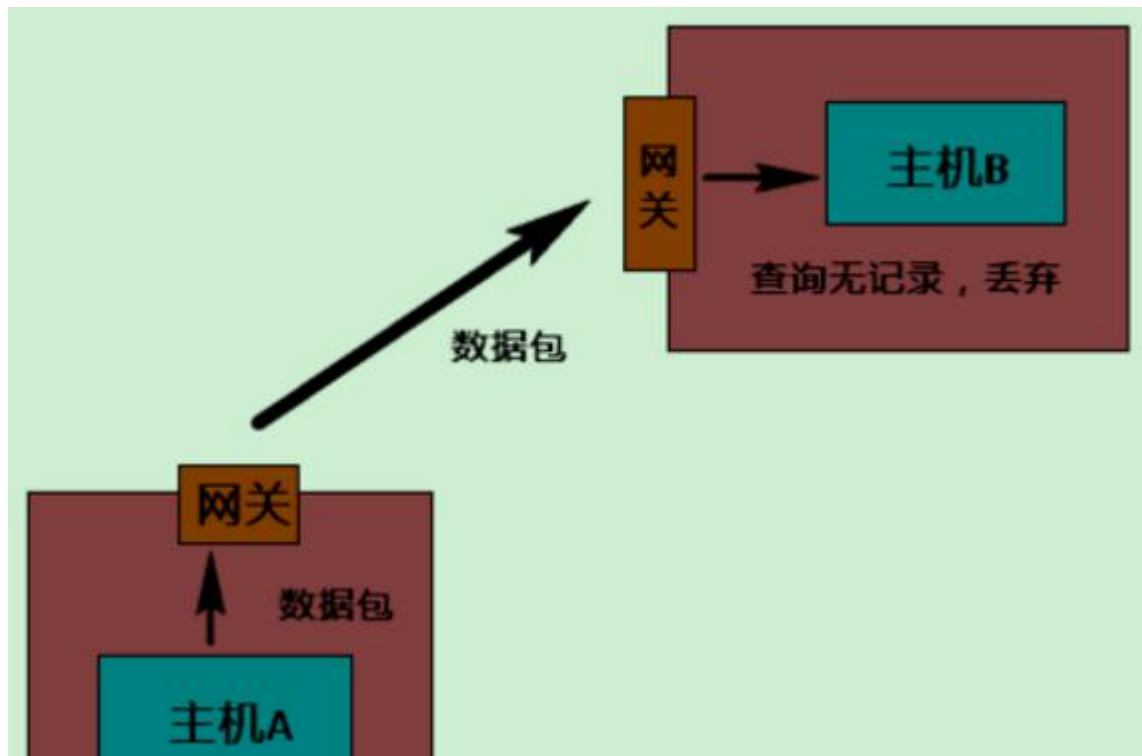
简单来说实现不同局域网内的主机之间通过互联网进行通信的技术叫内网穿透。

障碍一：位于局域网内的主机有两套 IP 地址，一套是局域网内的 IP 地址，通常是动态分配的，仅供局域网内的主机间通信使用；一套是经过网关转换后的外网 IP 地址，用于与外网程序进行通信。



障碍二：位于不同局域网内的两台主机，即使是知道了对方的 IP 地址和端口号，“一厢情愿”地将数据包发送过去，对方也是接收不到的。

因为出于安全起见，除非是主机主动向对方发出了连接请求（这时会在该主机的数据结构中留下一条记录），否则，当主机接收到数据包时，如果在其数据结构中查询不到对应的记录，那些不请自来的数据包将会被丢弃。



解决办法：要想解决以上两大障碍，我们需要借助一台具有公网 IP 的服务器进行桥接。

二、常见的内网穿透产品

2.1、花生壳

花生壳既是内网穿透软件、内网映射软件,也是端口映射软件。规模最大，较正规，完善。

收费高，使用简单

官网：<http://www.oray.com/>

尊享7.5折



铂金版

适用于高可用性的大型企业应用

享顶级BGP线路、价值980元VIP服务

购买年限满1年，送免费注册域名授权并协助备案

稳定性

解析速度

心跳包

服务器数量

按需选购

皇冠

🔍

5秒/次

专属服务器群

内网穿透（铂金版）
支持120个并发量
含4条映射（可升级）
6Mbps/映射（带宽）
不限流量

¥1260 4680元/年（无内网穿透）
¥1485 4980元/年（含内网穿透）

立即购买

了解无忧版 ^

新购满2年，送普通端口号（限9月）



旗舰版

适用于中、小型企业

享受顶级BGP线路

稳定性

解析速度

心跳包

服务器数量

按需选购

💎

🔍

15秒/次

70台

内网穿透（旗舰版）
支持60个并发量
含4条映射（可升级）
4Mbps/映射（带宽）
不限流量

668元/年（无内网穿透）
868元/年（含内网穿透）

立即购买

了解无忧版 ^

新购

2.2、Nat123

nat123 是内网端口映射与动态域名解析软件，在内网启动映射后，可在外网访问连接内网网站等应用。整个网站我都没有找到客服电话，网友发了一些反面的评价

收费，使用简单

官网：<http://www.nat123.com>



优V服务
助你 无忧

VIP

首页

端口映射

域名解析

免费域名

远程控制

套餐服务

VIP帮助

教程

免费注册

如何用动态域名解析

如何用端口映射

如何发布网站

如何用自主域名

如何找回用户名或密码

用户中心

▶ 端口映射列表

▶ 端口映射添加

▶ DNS解析列表

▶ DNS解析添加

▶ 远程开机列表

▶ 用户资料设置

▶ 用户参数设置

▶ 用户认证使用

▶ 用户优惠卡券

▶ 域名认证管理

▶ 域名解封申请

▶ 最近操作历史

▶ 发票索取流程

映射服务交易中心

阿里云 奥运会全球指定云服务商

云数据库 MySQL 最高32核128G，自建免

您当前的位置：首页 > 个人中心 > T币充值

当前T币： 0

充值平台： 支付宝  支持普通储蓄卡、信用卡、支付宝余额、支付宝余额

充值T币：

充值金额： 30.00元 快捷选择： ☒ 30元 ☐ 50元 ☐ 100元 ☐ 200元

优惠活动： 即时到账送等额N币

限时欢惠： 单次充值满3000T币送4%T币 单次充值满6000T币送8%T币

使用协议： ☒ 我已阅读并同意《T币使用协议》《退款提现协议》

确认充值

返回

2.3、NATAPP

NATAPP 基于 ngrok 的国内内网穿透服务，免费版会强制更换域名，临时用一下可以

收费，使用简单

官网：<https://natapp.cn/>

免费隧道(注册用户)	VIP_1型隧道	VIP_2型隧道
<h2>完全免费</h2> <p>提供http,https,tcp全隧道穿透 随机域名/TCP端口 不定时强制更换域名/端口 自定义本地端口</p> <p>立即注册</p>	<h2>¥5 元/月</h2> <p>提供http,https,tcp全隧道穿透 自定义子域名/TCP端口 服务期间永久固定 自定义本地端口 绑定自主域名 HTTPS泛域名及本地SSL证书</p> <p>立即购买</p>	<h2>¥10 元/月</h2> <p>100M带宽 流量计费</p> <p>提供http,https,tcp全隧道穿透 自定义子域名/TCP端口 服务期间永久固定 自定义本地端口 绑定自主域名 HTTPS泛域名及本地SSL证书</p> <p>立即购买</p>

2.4、frp 与其它

frp 是一个高性能的反向代理应用，可以帮助您轻松地进行内网穿透，对外网提供服务，支持 tcp, http, https 等协议类型，并且 web 服务支持根据域名进行路由转发。

开源免费

使用相对复杂，需要代理服务器支持

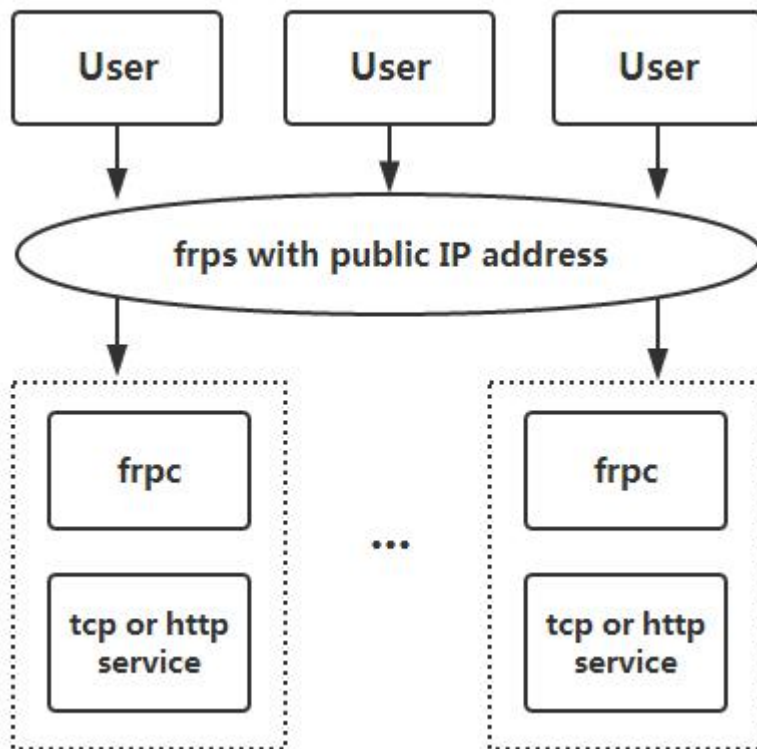
官网：<https://github.com/fatedier/frp>

文档：[查看帮助文档](#)，[简书示例](#)

利用处于内网或防火墙后的机器，对外网环境提供 http 或 https 服务。

对于 http, https 服务支持基于域名的虚拟主机，支持自定义域名绑定，使多个域名可以共用一个 80 端口。

利用处于内网或防火墙后的机器，对外网环境提供 tcp 和 udp 服务，例如在家里通过 ssh 访问处于公司内网环境内的主机。



因为 frp 仍然处于前期开发阶段，未经充分测试与验证，不推荐用于生产环境，所以我选择了 ngrok，资料比较多。

还有如圣剑内网通、ngrok（开源免费）、[更多办法](#)

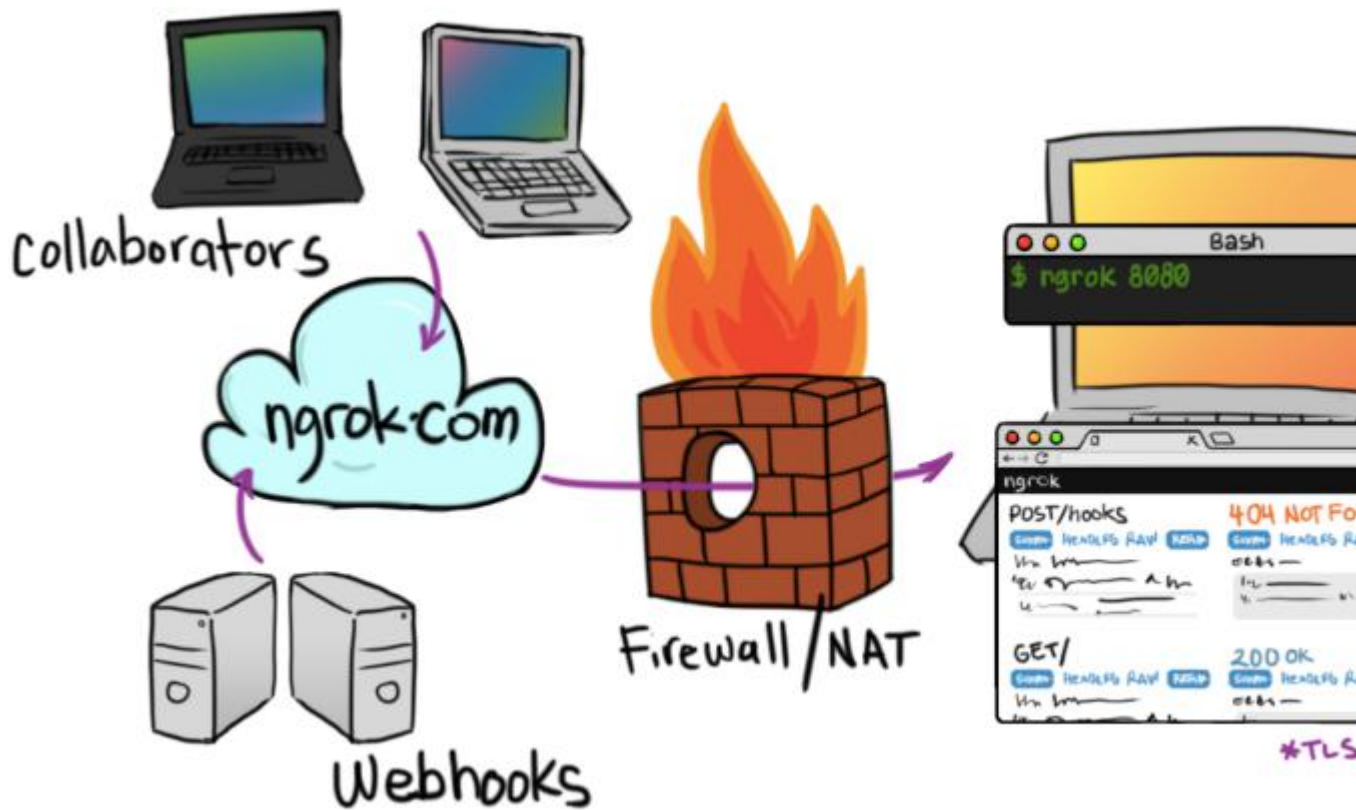
三、ngrok

ngrok 是一个反向代理，通过在公共的端点和本地运行的 Web 服务器之间建立一个安全的通道。ngrok 可捕获和分析所有通道上的流量，便于后期分析与响应。

开源免费

官网：<https://ngrok.com/>

源码：<https://github.com/inconshreveable/ngrok>



ngrok1.x 开源，ngrok2.x 不开源

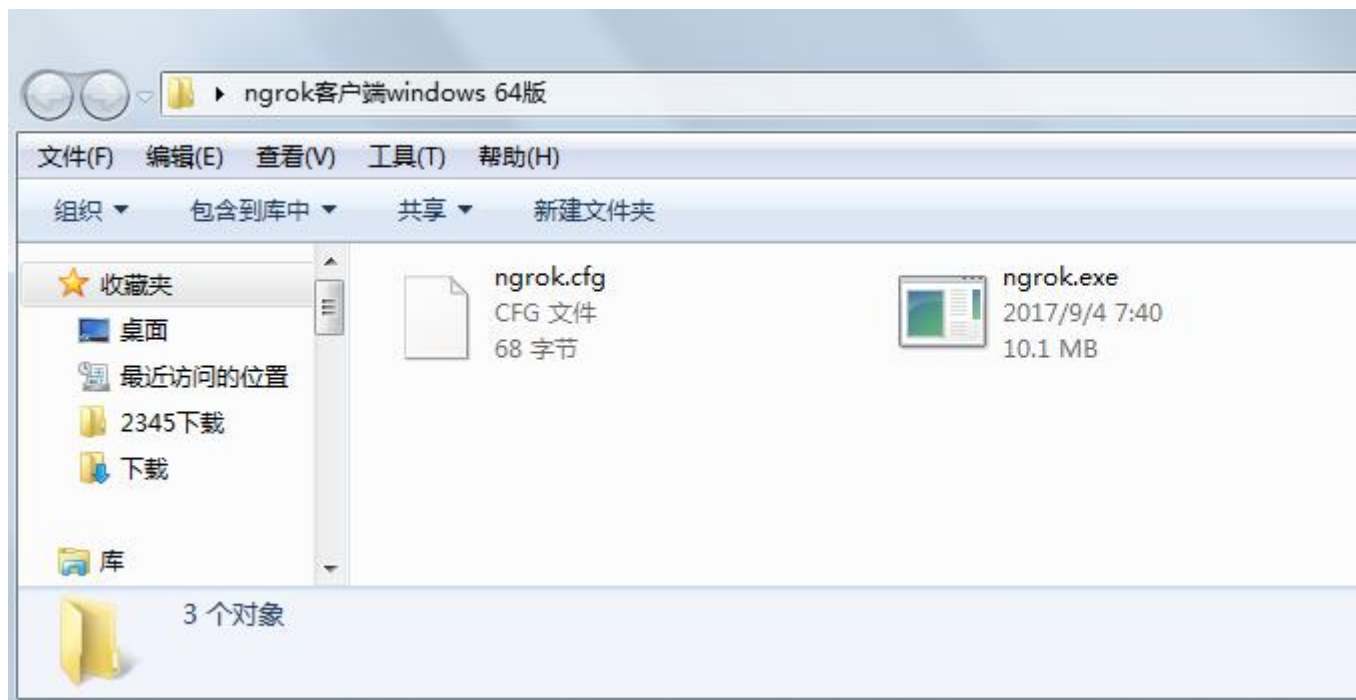
ngrok 使用 go 语言开发，源代码分为客户端与服务器端。

国内免费服务器：<http://ngrok.ciqiuwl.cn/>，更多免费服务器请大家挖掘，资源共享，我随时更新：)

如果有服务器，仅客户端的使用是不复杂的，以上面的免费服务器为示例完成内网穿透

现在假定我的本地已成功部署了一个网站，访问地址为 127.0.0.1，想内网穿透后被公网上的用户访问，一般步骤如下：

步骤 1、下载 windows 版本的客户端，解压。一般在为你提供代理服务器的网站上找你要下载的客户端：



步骤 2、在命令（cmd）行下进入到 ngrok 客户端目录下

```
管理员: C:\Windows\system32\cmd.exe

C:\>cd C:\Users\Administrator\Desktop\ngrok客户端windows 64版

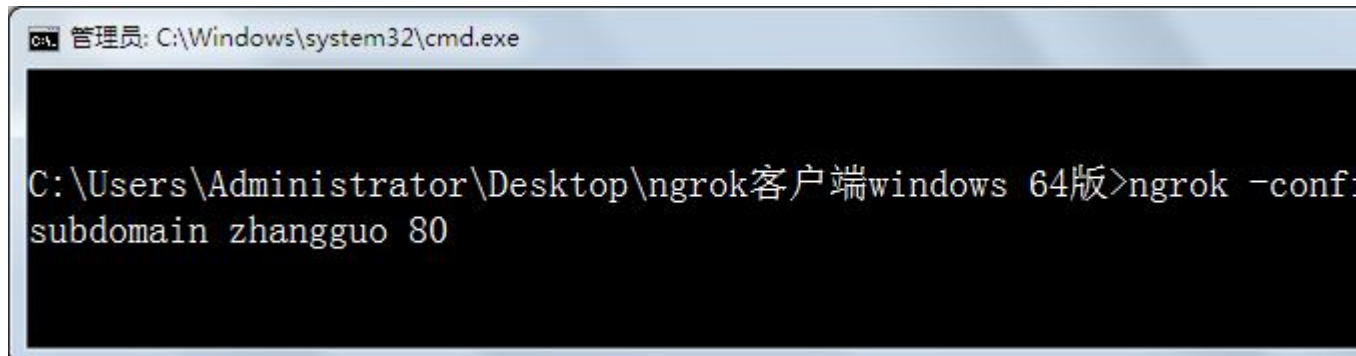
C:\Users\Administrator\Desktop\ngrok客户端windows 64版>dir
驱动器 C 中的卷没有标签。
卷的序列号是 8CC1-8C9C

C:\Users\Administrator\Desktop\ngrok客户端windows 64版 的目录

2017/09/04  07:40    <DIR>          .
2017/09/04  07:40    <DIR>          ..
2017/05/10  14:10                68 ngrok.cfg
2017/05/10  14:06          10,665,984 ngrok.exe
2017/05/10  14:53                984 start.bat
               3 个文件      10,667,036 字节
               2 个目录   3,796,570,112 可用字节

C:\Users\Administrator\Desktop\ngrok客户端windows 64版>
```

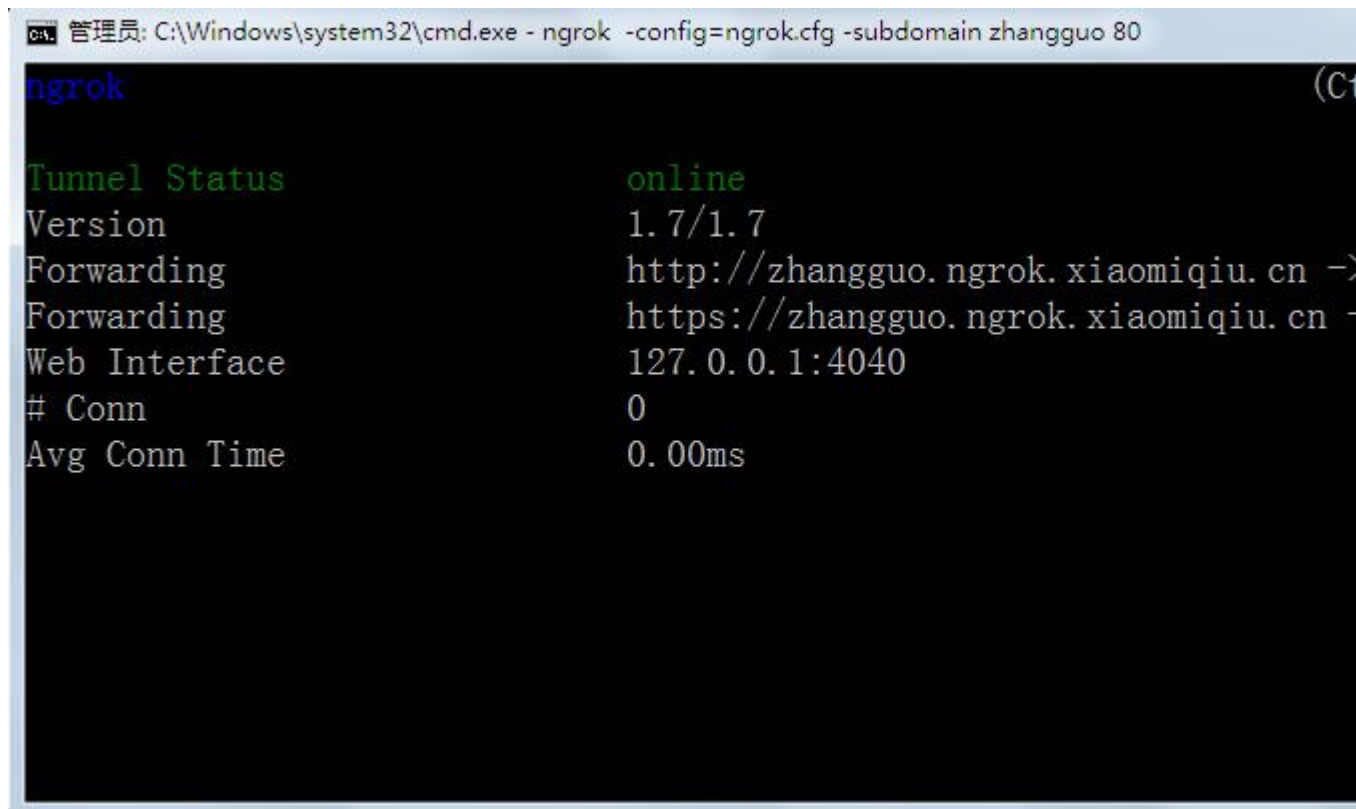
步骤 3、执行 `ngrok -config=ngrok.cfg -subdomain xxx 80` //(xxx 是你自定义的域名前缀)，建议批处理



```
管理员: C:\Windows\system32\cmd.exe

C:\Users\Administrator\Desktop\ngrok客户端windows 64版>ngrok -config=ngrok.cfg -subdomain zhangguo 80
```

如果连接成功，会提示如下信息：



```
管理员: C:\Windows\system32\cmd.exe - ngrok -config=ngrok.cfg -subdomain zhangguo 80

ngrok

Tunnel Status      online
Version            1.7/1.7
Forwarding          http://zhangguo.ngrok.xiaomiqu.cn ->
Forwarding          https://zhangguo.ngrok.xiaomiqu.cn ->
Web Interface       127.0.0.1:4040
# Conn              0
Avg Conn Time       0.00ms
```

这一步如果你认为太麻烦，可以直接运行目录下的 `start.bat` 批处理文件就不用进 DOS 环境了。运行 `start.bat` 直接跳过 2，3 步

步骤 4、如果开启成功 你就可以使用 `xxx.ngrok.xiaomiqu.cn` 来访问你本机的 `127.0.0.1:80` 的服务了，当然你必须确定的是你本机的 **Web** 是可以正常访问的。



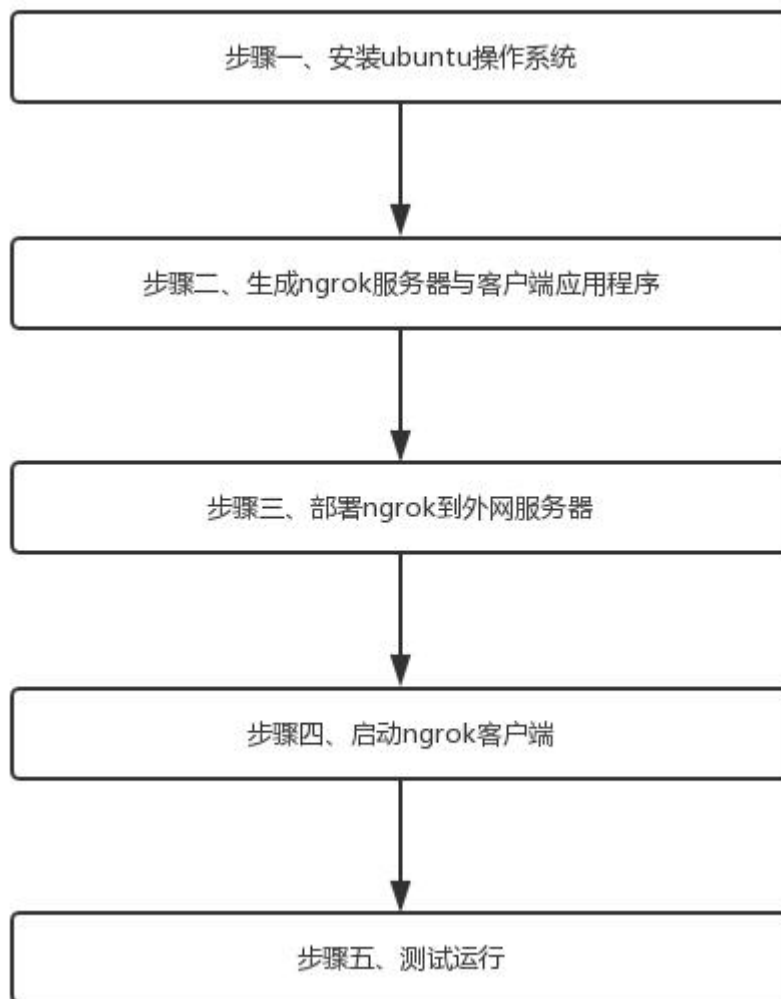
注意：

如果你自己有顶级域名，想通过自己的域名来访问本机的项目，那么先将自己的顶级域名解析到 120.25.161.137(域名需要已备案哦，80 端口必须备案)，然后执行 `ngrok -config=ngrok.cfg -hostname xxx.xxx.xxx 80` //(xxx.xxx.xxx 是你自定义的顶级域名)

四、ubuntu 下生成 ngrok 服务器主程序

4.1、步骤与先决条件

如果你只是临时穿透或调试用，到第三步基本就可以了，但如果想作为稳定的商业服务，用别人的服务器还是受制于人，这里我们准备搭建自己的 ngrok 服务器。大致的步骤如下：



ngrok 服务器可以是多种平台，如 windows、Linux（CentOS、Debian、Ubuntu 等）、Mac OS 等。

编译源代码生成应用强烈建议大家使用 linux 环境，windows 肯定可以成功，但非常麻烦，我在 windows 操作系统上兜了一个大圈圈。

先决条件：

- a)、您有一台公网上的服务器，如阿里云的 ECS
- b)、您有一个域名，最好 ICP 备案成功，不然 80 端口没有办法使用，不过像微信开发是不使用 80 端口的，可以用 nginx 代理转换。

4.2、安装 ubuntu 操作系统

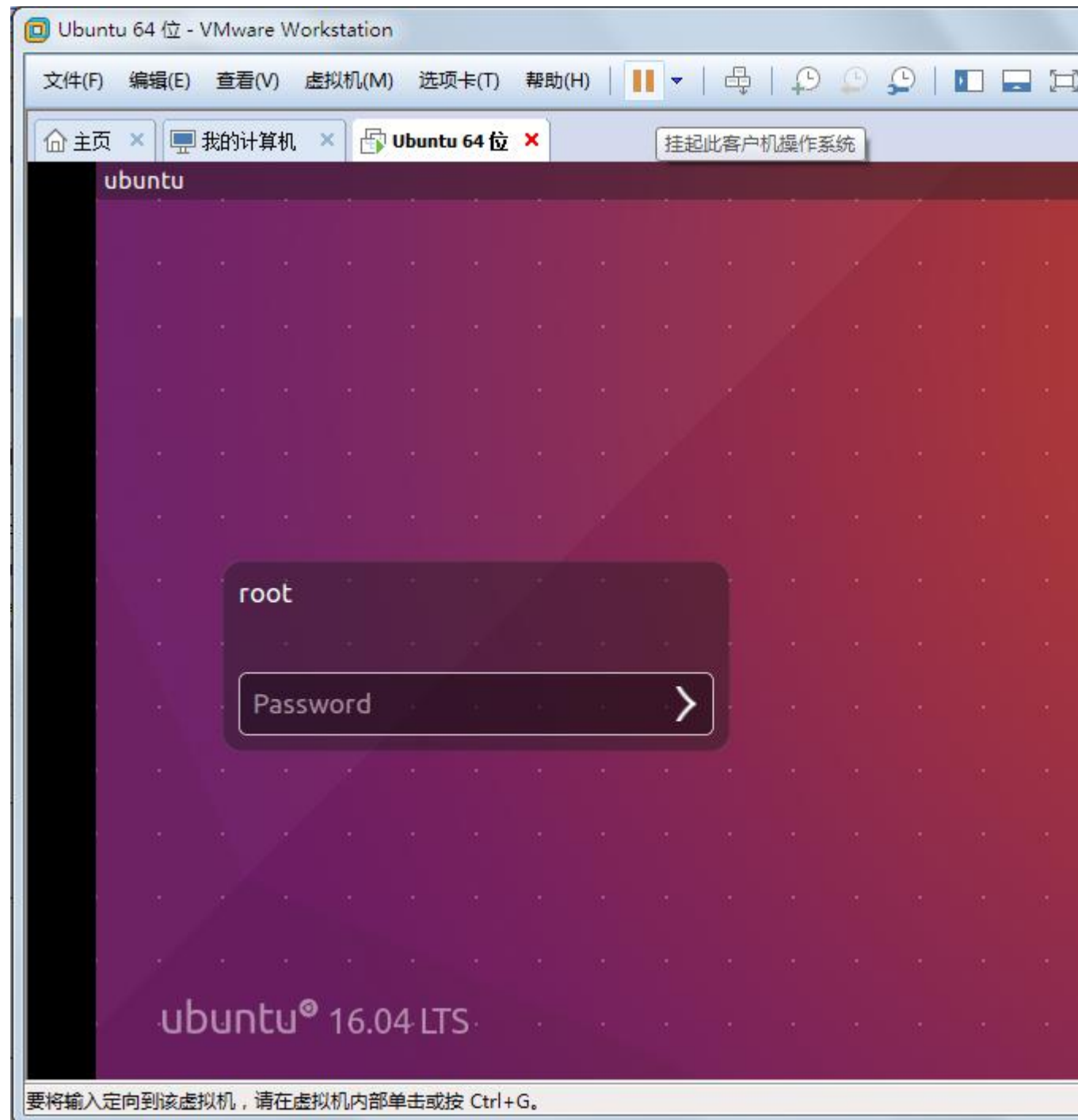
在 linux 环境下编译 ngrok 的源代码比 windows 下方便很多，这里我们选择使用 ubuntu，获得 ubuntu 的方法有如下几种：

1)、全新安装 ubuntu 系统

2)、申请 VPS 服务器，阿里云、腾讯云、华为云、百度云、新浪云等，仅编译一下这种方法不错

3)、在虚拟机中安装 ubuntu 系统

综合考虑我选择了在虚拟机中安装 ubuntu 操作系统



4.2.1、安装 VMware 虚拟机

VMware Workstation 是一款功能强大的虚拟机软件，在不影响本机操作系统的情况下，用户可以在虚拟机中同时运行不同版本的操作系统，用于开发、测试以及部署工作。

VMware Workstation 12 pro 下载：[VMware-workstation-full-12.1.0-3272444.exe](#)

序列号：5A02H-AU243-TZJ49-GTC7K-3C61N（商业应用请购买正式版权，这里仅为学习使用）

1）、双击 VMware Workstation 12 安装文件，或者右键管理员身份打开，提示是否允许更改，点击是；

2）、打开 VMware 安装向导，点击下一步；

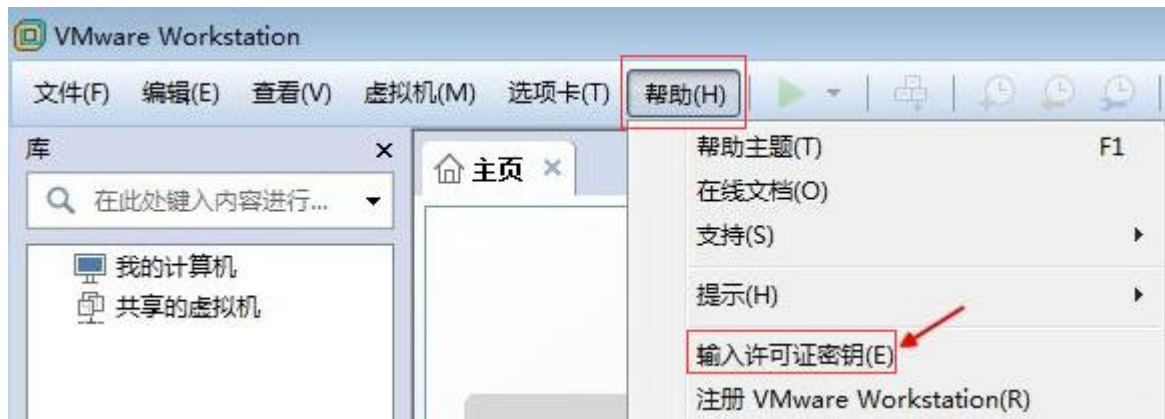


3）、VMware Workstation 12 激活步骤：

方法一、首次开启直接输入上文密钥，即可激活；

方法二、首次开启选择试用，进入试用后按一下步骤激活：

a、打开虚拟机主界面，点击“帮助”—“输入许可证密钥”；



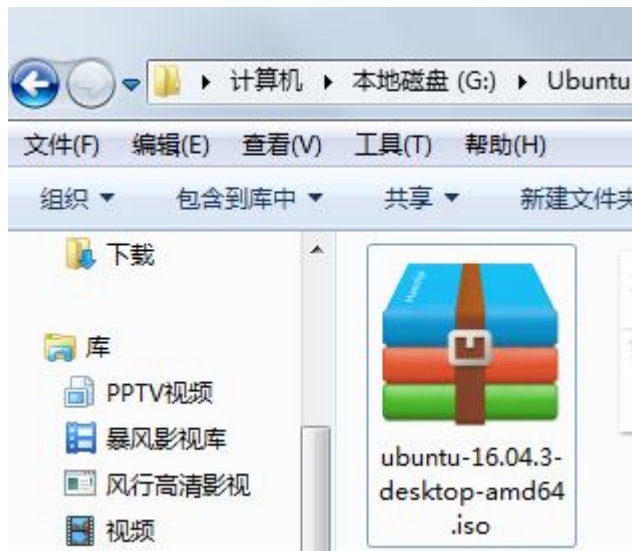
b、在密钥输入框输入永久许可证密钥 5A02H-AU243-TZJ49-GTC7K-3C61N，确定；
[更多](#)

4.2.2、安装 ubuntu 到虚拟机

1)、下载 ubuntu 操作系统镜像

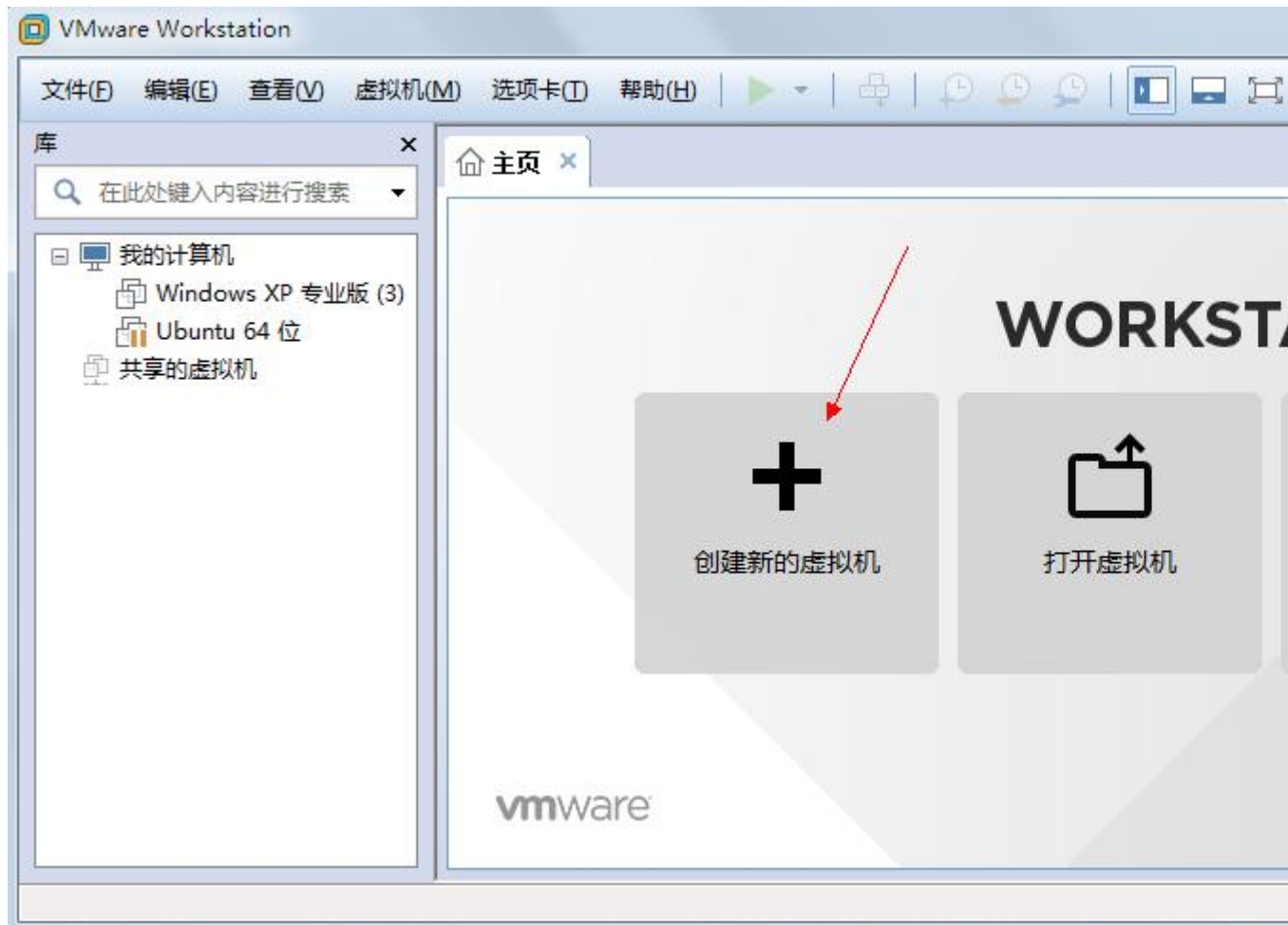
下载地址：<https://www.ubuntu.com/download/desktop>

这里我下载的是 ubuntu-16.04.3-desktop-amd64.iso

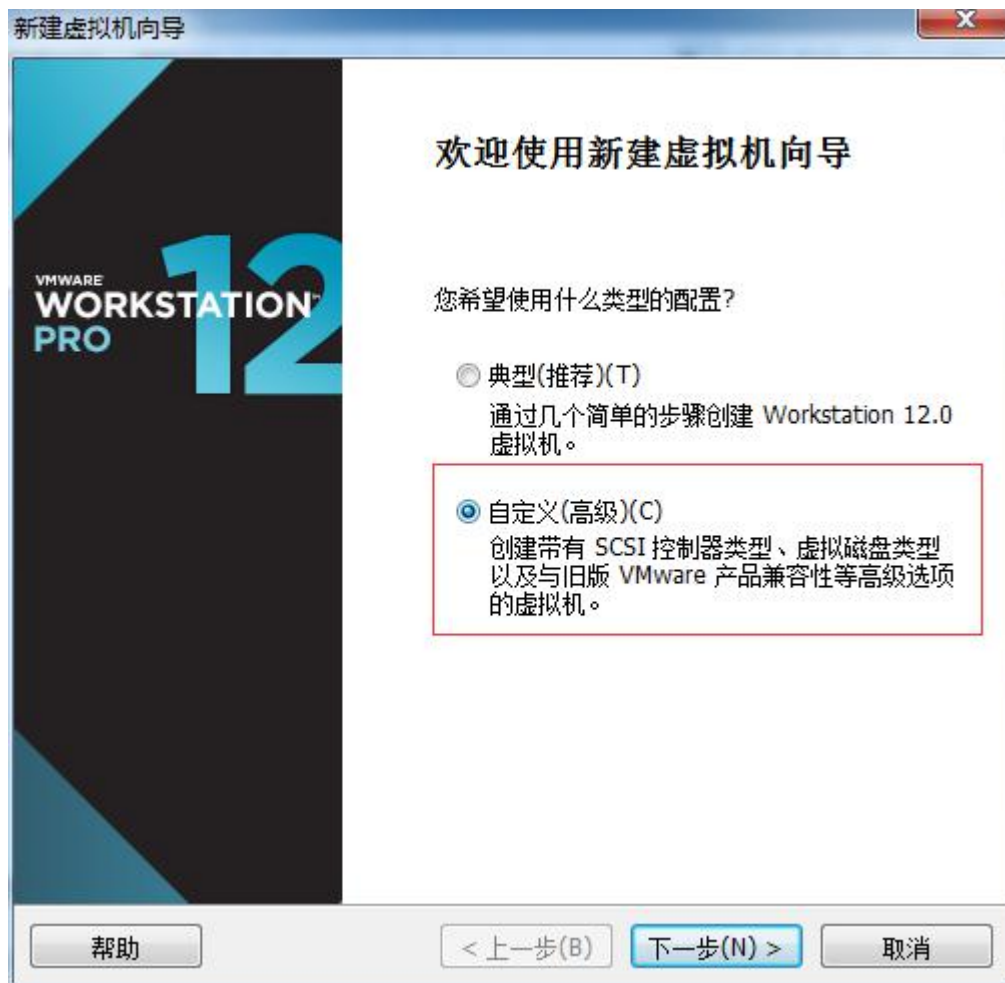


2)、在 VMware 中安装 ubuntu

打开 VMware 点击“创建新的虚拟机”



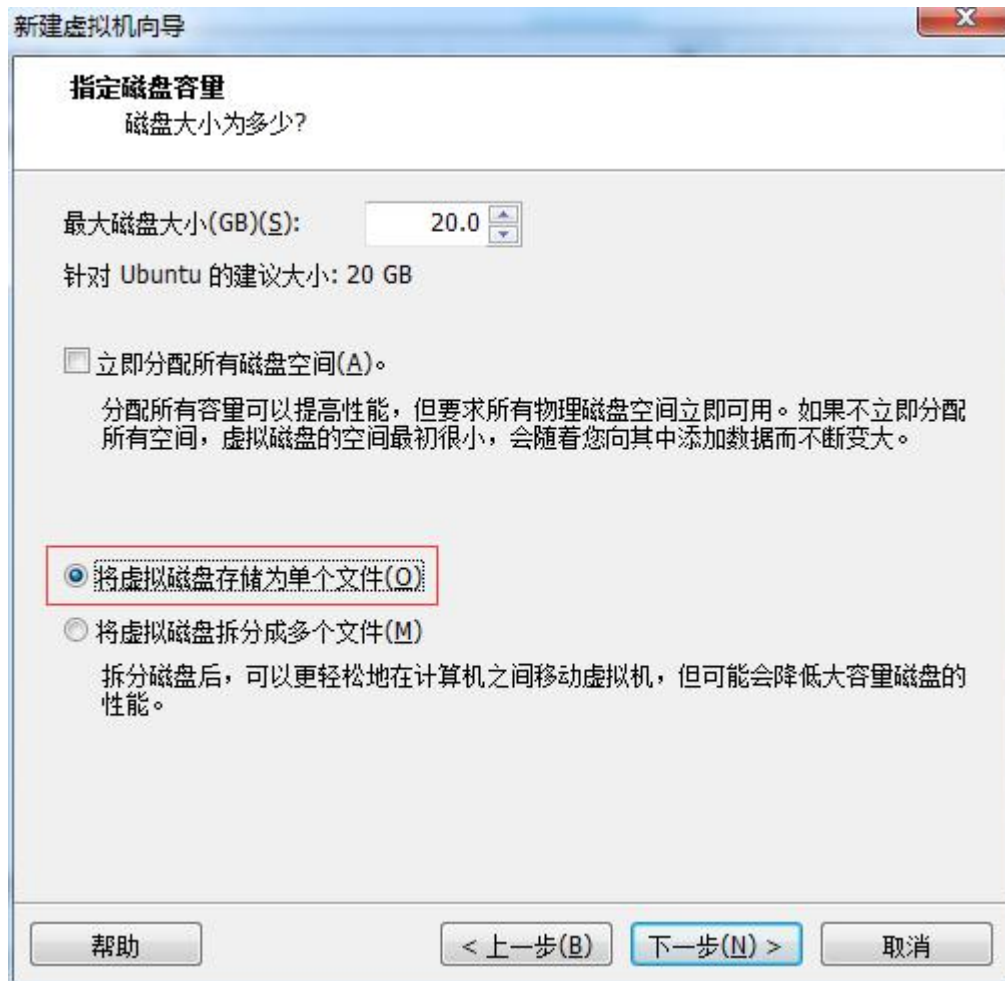
向导选择自定义



然后下一步再下一步，直到这里，稍后再安装系统



后面设置处理器和内存的，电脑配置好的可以试试，否则采用默认的，博主这里是采用默认的，然后下一，直到这里，选择将虚拟机存储为单个磁盘：

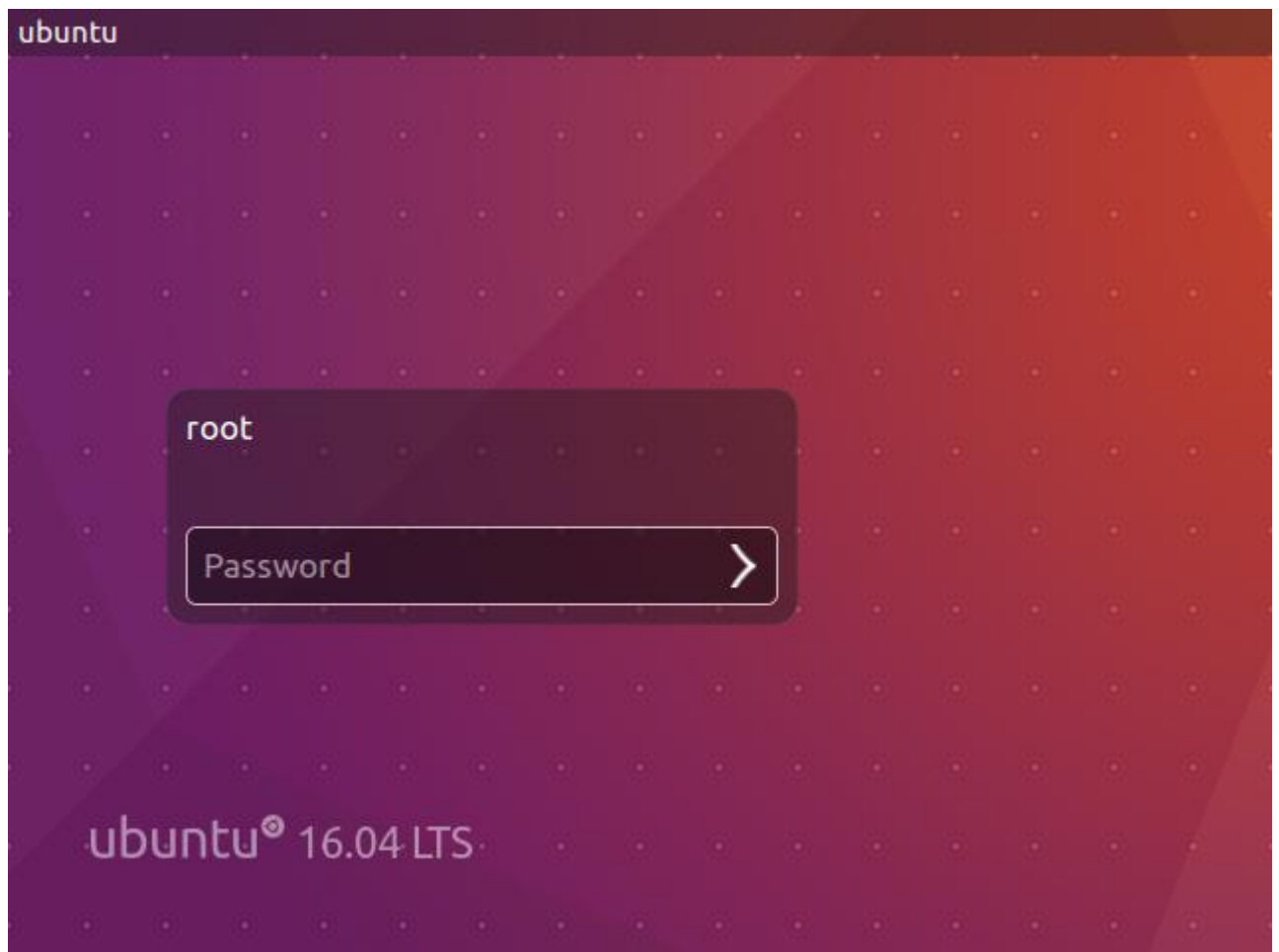


个人建议至少 20G 硬盘空间，内存建议给 1.5G，当然也要看电脑本身的配置，1G 的内存跑起来比较卡。

其它的步骤比较简单，更多细节可以参考这里，[《VMware Ubuntu 安装详细过程》](#)。

4.2.3、配置 ubuntu 系统

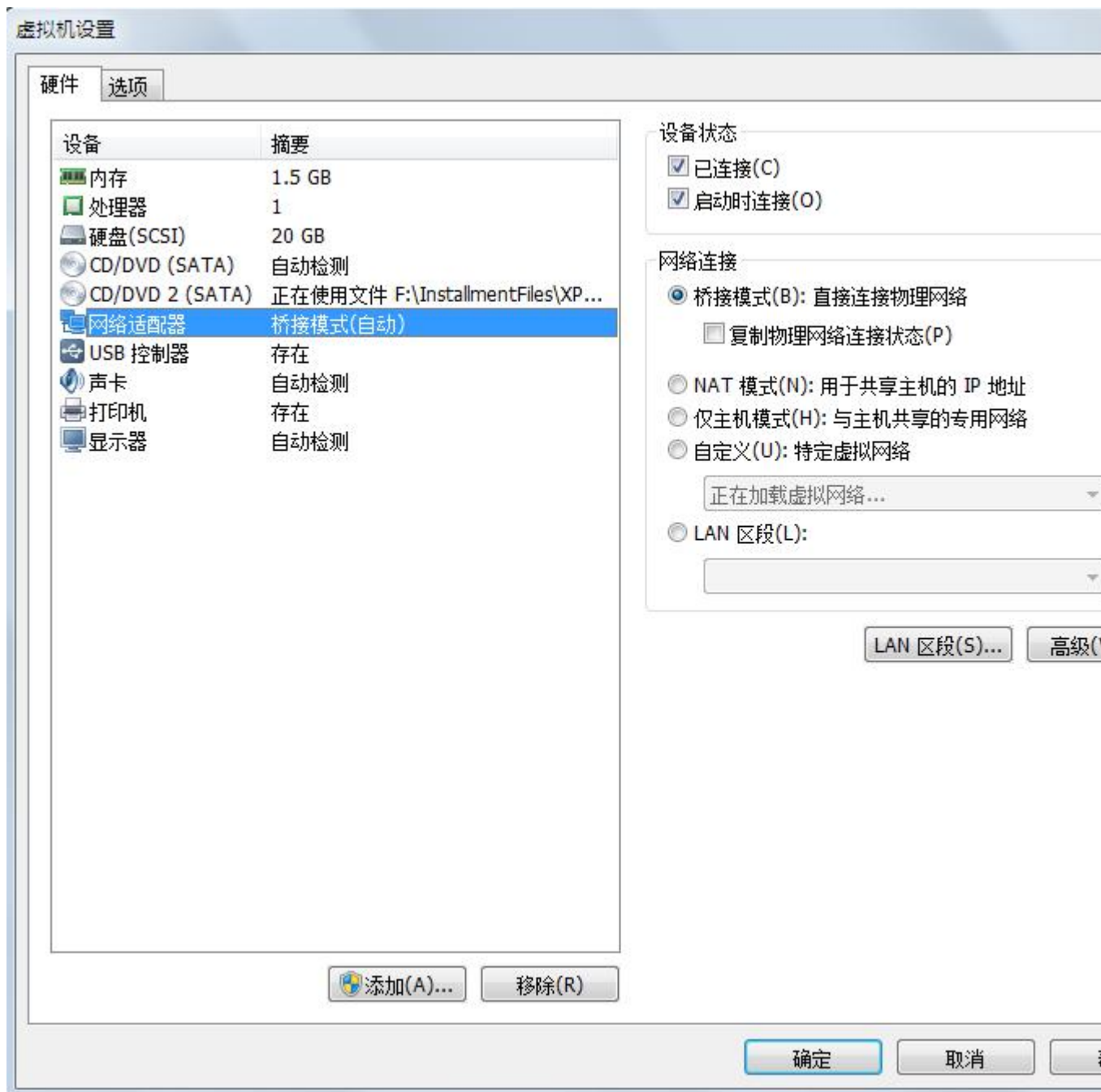
当 ubuntu 系统安装成功后，在虚拟机中可以启动 ubuntu 系统，启动后的系统如下：



ubuntu 系统的使用还是有许多内容的，这里需要设置的内容如下：

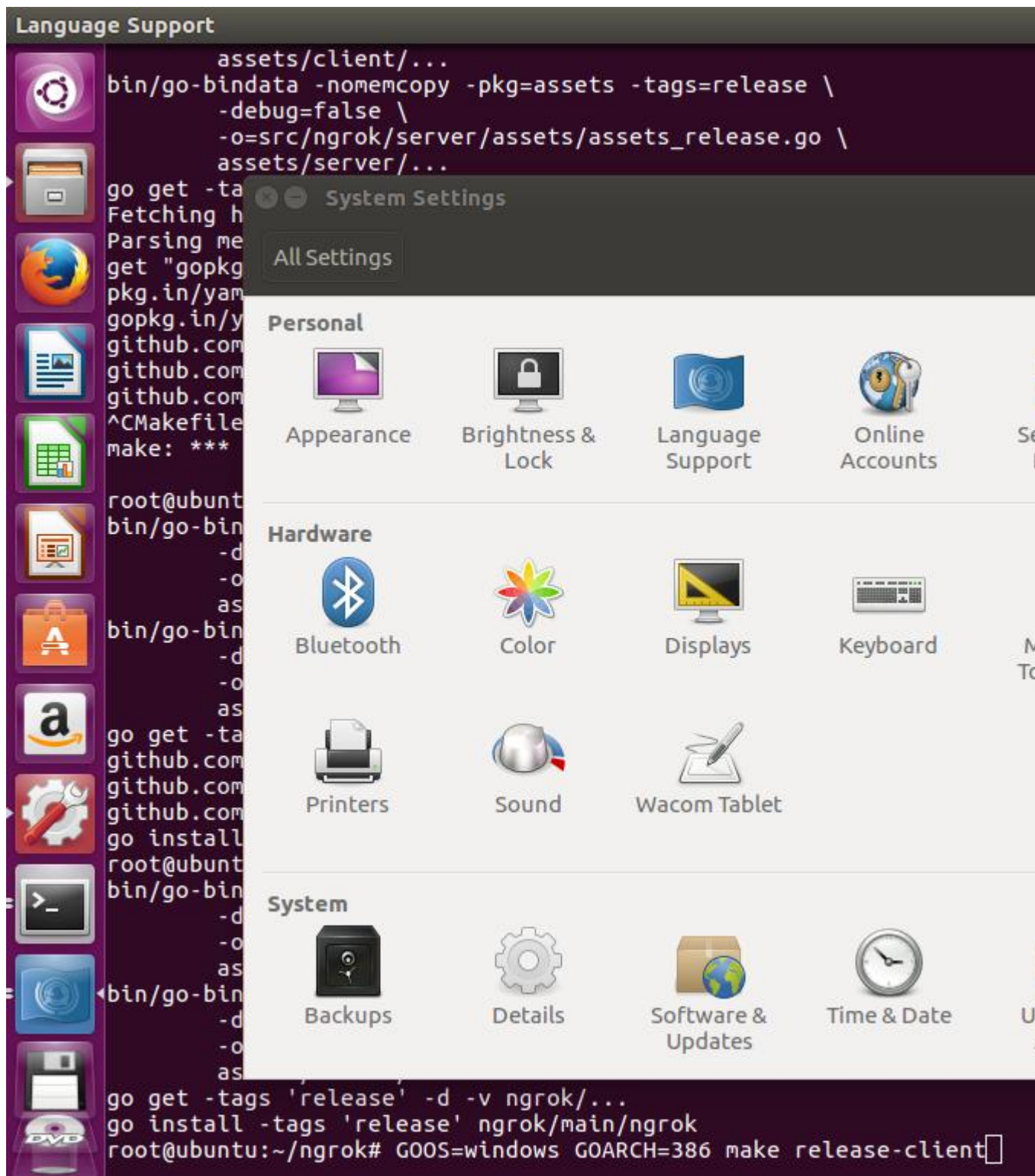
a)、设置上网

就是在 ubuntu 中可以访问外网，可以使用多种形式



b)、设置语言

可以选择使用中文版的 ubuntu 语言环境



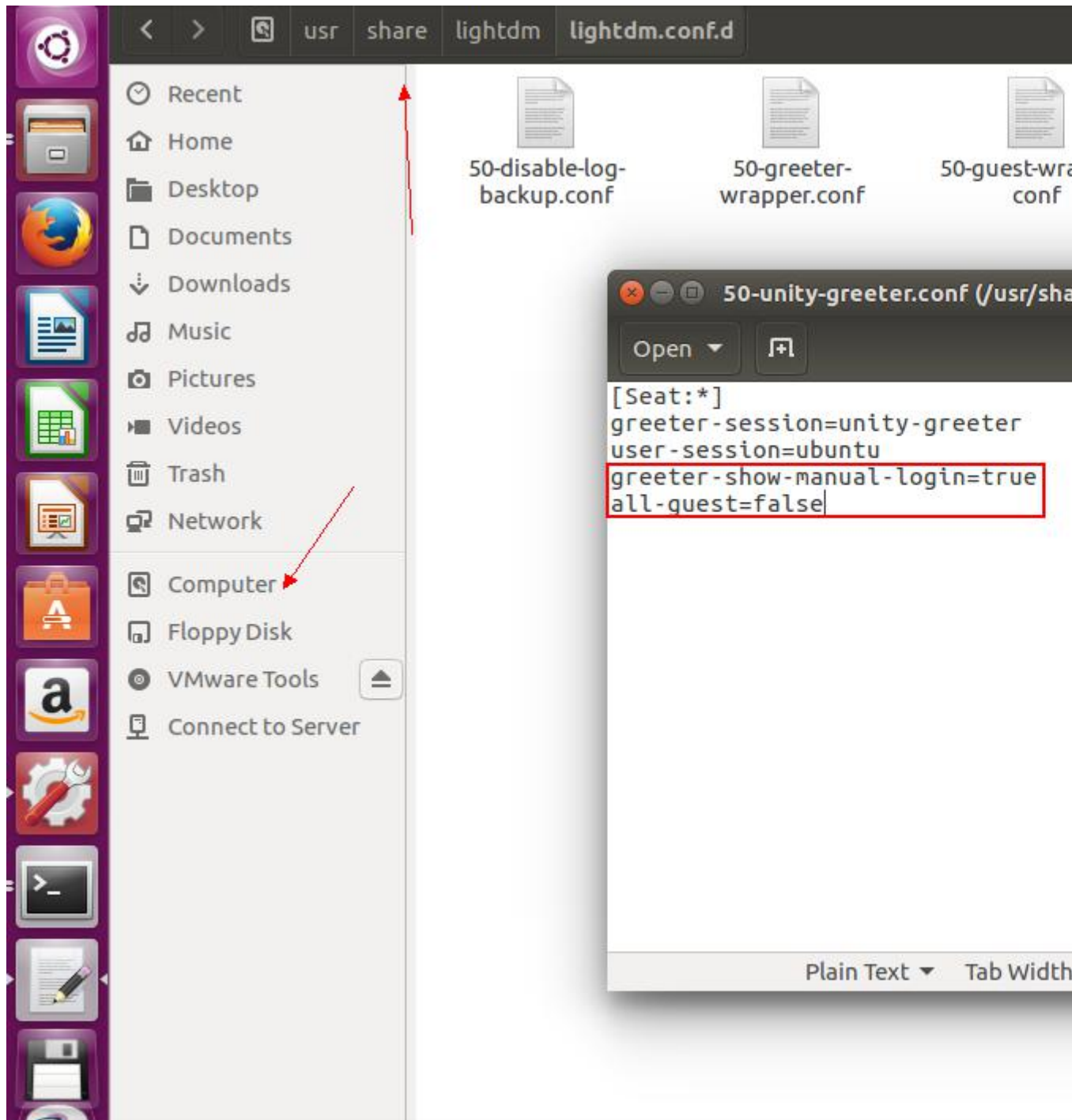
c)、设置屏幕分辨率

如果不设置默认屏幕比较小



d)、设置以 root 超级管理员的身份登录

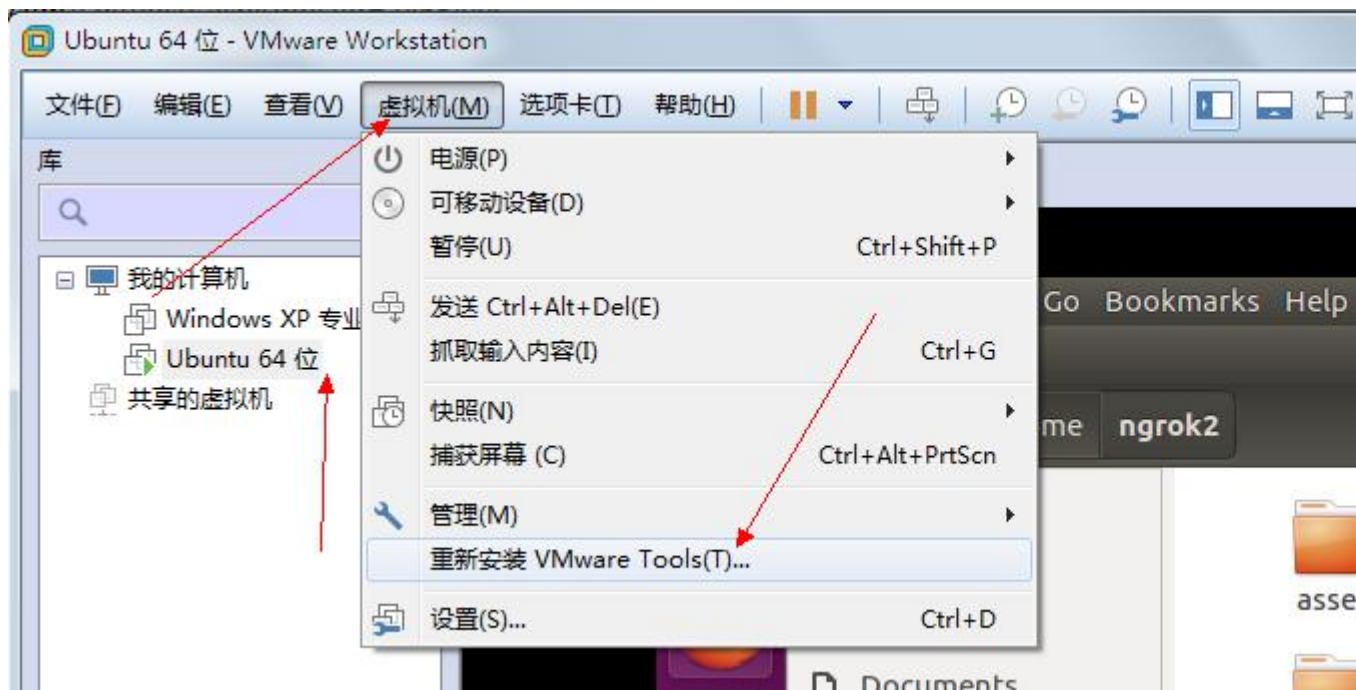
许多操作要求管理身份



e)、安装 VMware Tools 工具

只有在 VMware 虚拟机中安装好了 VMware Tools，才能实现主机与虚拟机之间的文件共享，同时可支持自由拖拽的功能，鼠标也可在虚拟机与主机之间自由移动（不用再按 `ctrl+alt`），且虚拟机屏幕也可实现全屏化。

VMware Tools 是 VMware 虚拟机中自带的一种增强工具，相当于 VirtualBox 中的增强功能（Sun VirtualBox Guest Additions），是 VMware 提供的增强虚拟显卡和硬盘性能、以及同步虚拟机与主机时钟的驱动程序。



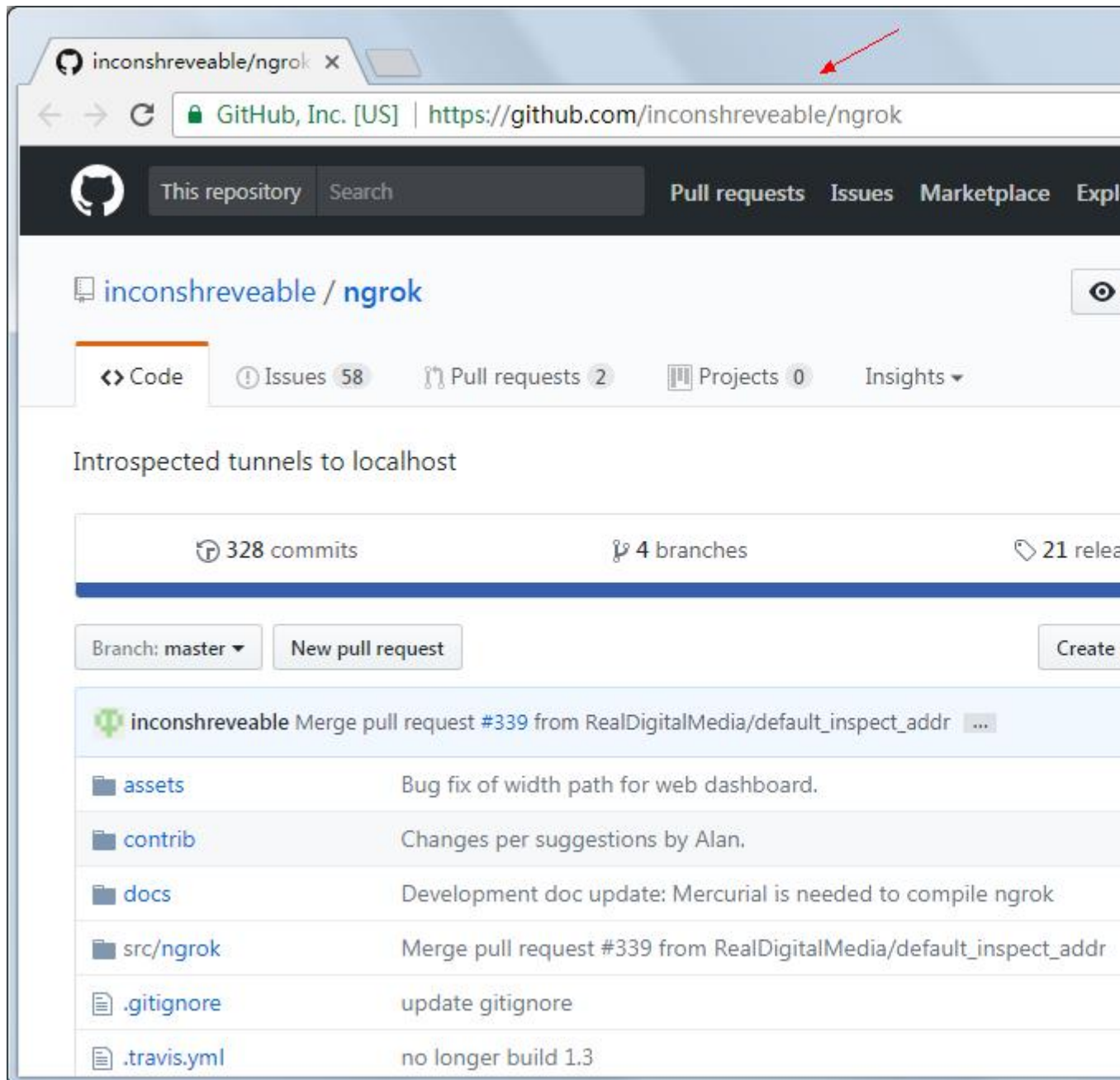
注意如果这里是灰色的需要您将 linux.iso 镜像加载到虚拟光驱中，一般在 VM 的安装目录下有，如果没有您需要自行下载。

说明：ubuntu 的使用不是本文的重点，相关操作请大家自行查找。

4.3、生成 ngrok 服务器与客户端应用程序

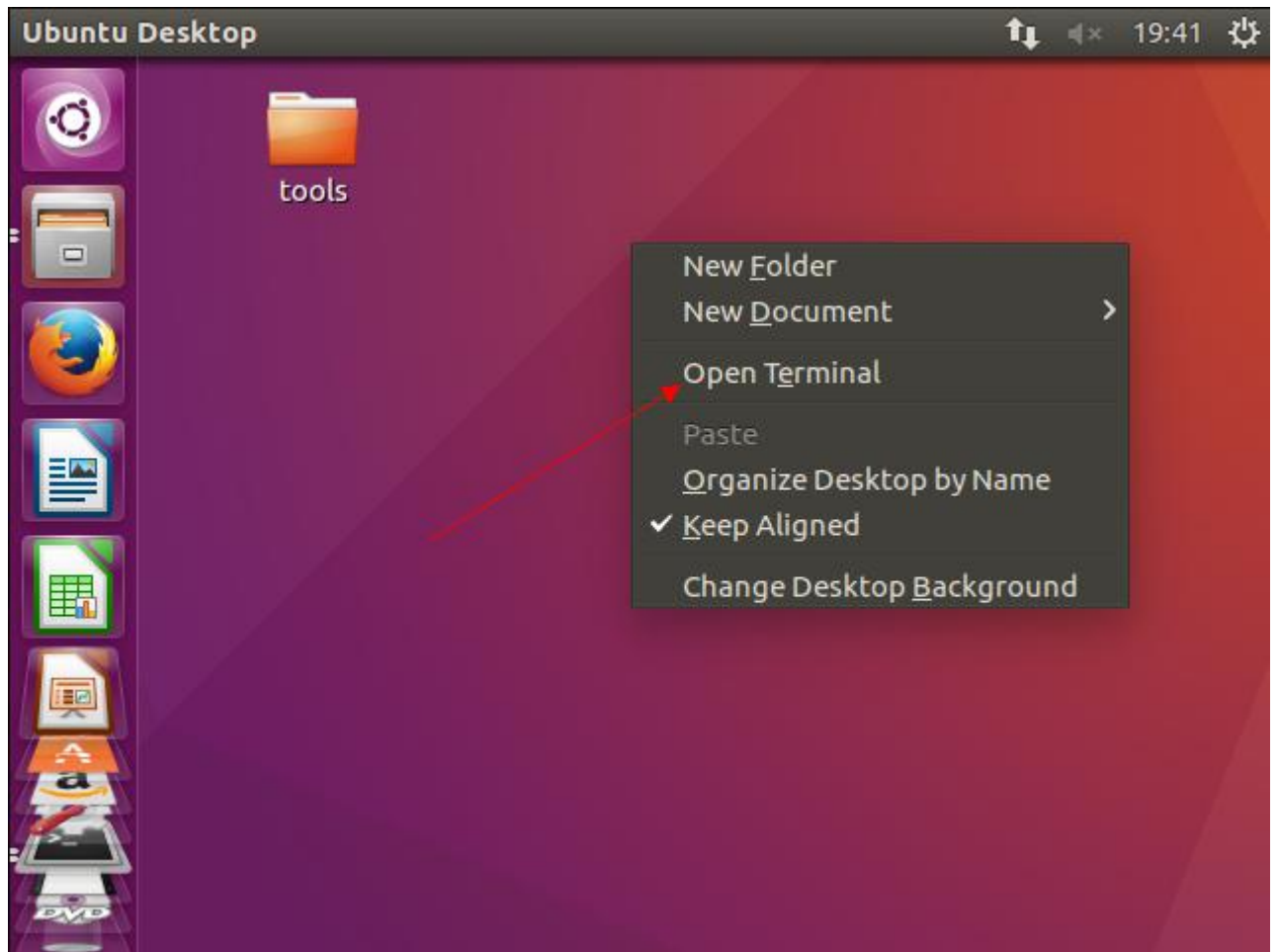
4.3.1. 导出源代码

ngrok 的源代码托管在 github 上，可以先在 ubuntu 下安装 git 再将 ngrok 的源代码克隆到本地。



其实也可以直接下载到本地后解压，这里使用命令行完成。

启动 ubuntu，开打命令行（终端），如下所示：



以 root 身份执行如下命令：

```
mkdir ngrok #创建名称为 ngrok 的目录
```

```
apt-get update #更新包管理器
```

```
apt-get install git #安装 git
```

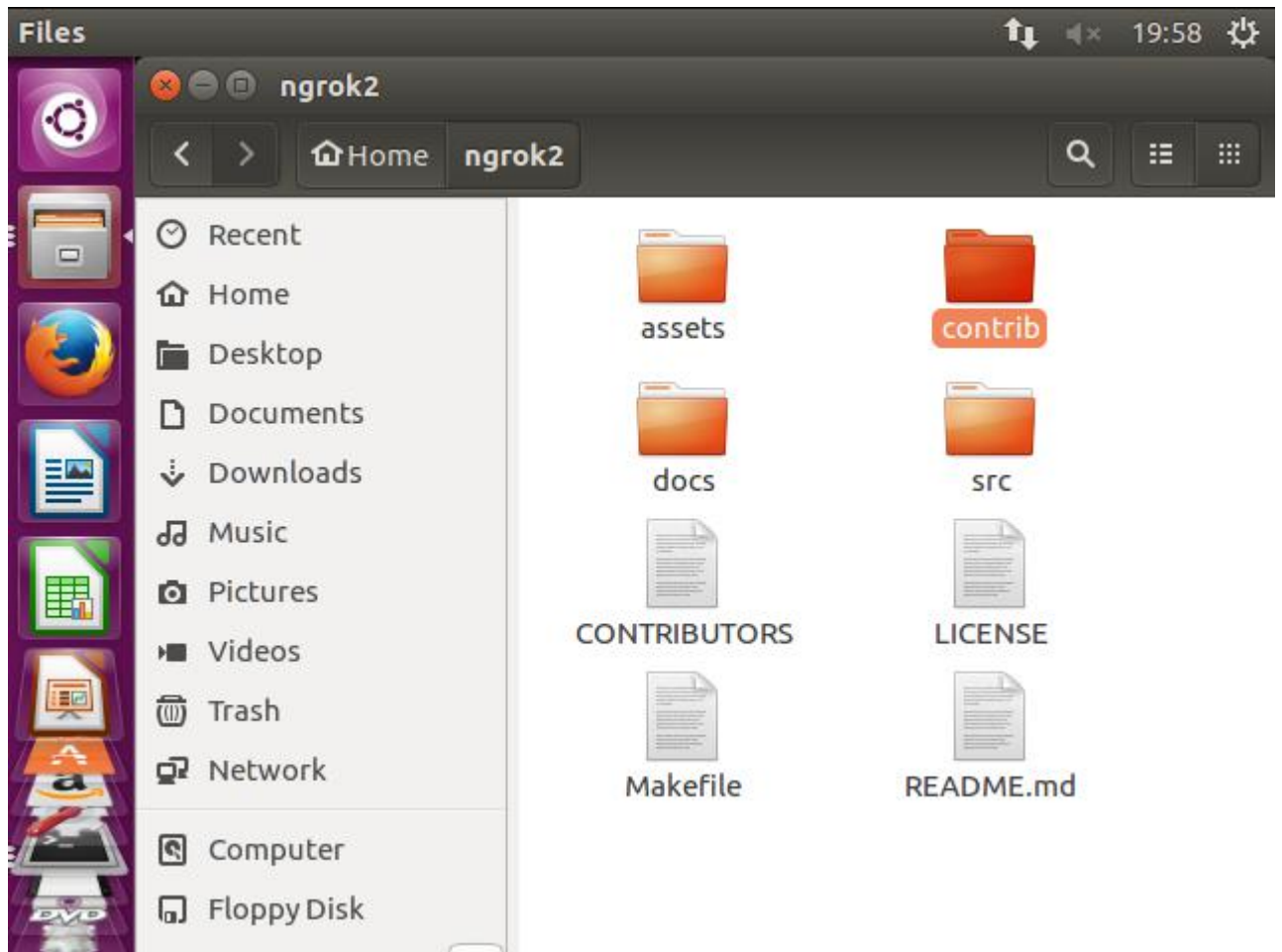
```
git clone https://github.com/inconshreveable/ngrok.git ngrok2 #将  
ngrok 源代码克隆回本地
```

成功执行后如下所示：


```
root@ubuntu: ~
root@ubuntu:~# mkdir ngrok2 #create dir
root@ubuntu:~# apt-get update #更新包管理器
Hit:1 http://mirrors.aliyun.com/ubuntu xenial InRelease
Hit:2 http://mirrors.aliyun.com/ubuntu xenial-updates InRelease
Hit:3 http://mirrors.aliyun.com/ubuntu xenial-backports InRelease
Hit:4 http://mirrors.aliyun.com/ubuntu xenial-security InRelease
Reading package lists... Done
root@ubuntu:~# clear

root@ubuntu:~# mkdir ngrok2 #创建名称为ngrok2的目录
root@ubuntu:~# apt-get update #更新包管理器
Hit:1 http://mirrors.aliyun.com/ubuntu xenial InRelease
Hit:2 http://mirrors.aliyun.com/ubuntu xenial-updates InRelease
Hit:3 http://mirrors.aliyun.com/ubuntu xenial-backports InRelease
Hit:4 http://mirrors.aliyun.com/ubuntu xenial-security InRelease
Reading package lists... Done
root@ubuntu:~# apt-get install git #安装git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.7.4-0ubuntu1.2).
0 upgraded, 0 newly installed, 0 to remove and 38 not upgraded.
root@ubuntu:~# git clone https://github.com/inconshreveable/ngrok.git
Cloning into 'ngrok2'...
remote: Counting objects: 2732, done.
Receiving objects: 16% (438/2732), 84.01 KiB | 123.00 KiB/s Receiving objects: 17% (465/2732), 168.01 KiB | 142.00 KiB/s Receiving objects: 18% (492/2732), 204.01 KiB | 109.00 KiB/s Receiving objects: 19% (523/2732), 204.01 KiB | 109.00 KiB/s Receiving objects: 21% (574/2732), 204.01 KiB | 109.00 KiB/s Receiving objects: 23% (629/2732), 204.01 KiB | 109.00 KiB/s Receiving objects: 26% (711/2732), 292.01 KiB | 109.00 KiB/s Receiving objects: 28% (765/2732), 292.01 KiB | 115.00 KiB/s Receiving objects: 30% (820/2732), 292.01 KiB | 115.00 KiB/s Receiving objects: 33% (902/2732), 292.01 KiB | 115.00 KiB/s Receiving objects: 34% (929/2732), 372.01 KiB | 110.00 KiB/s Receiving objects: 36% (984/2732), 372.01 KiB | 110.00 KiB/s Receiving objects: 39% (1066/2732), 372.01 KiB | 110.00 KiB/s Receiving objects: 41% (1121/2732), 416.01 KiB | 107.00 KiB/s Receiving objects: 43% (1175/2732), 416.01 KiB | 107.00 KiB/s Receiving objects: 45% (1230/2732), 416.01 KiB | 107.00 KiB/s Receiving objects: 47% (1285/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 49% (1339/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 51% (1394/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 53% (1449/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 55% (1503/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 57% (1558/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 59% (1612/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 61% (1667/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 63% (1721/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 65% (1776/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 67% (1830/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 69% (1885/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 71% (1939/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 73% (1994/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 75% (2048/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 77% (2103/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 79% (2157/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 81% (2212/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 83% (2266/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 85% (2321/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 87% (2375/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 89% (2430/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 91% (2484/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 93% (2539/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 95% (2593/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 97% (2648/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 99% (2702/2732), 436.01 KiB | 97.00 KiB/s Receiving objects: 100% (2732/2732), 436.01 KiB | 97.00 KiB/s
```

导出成功后的源代码:



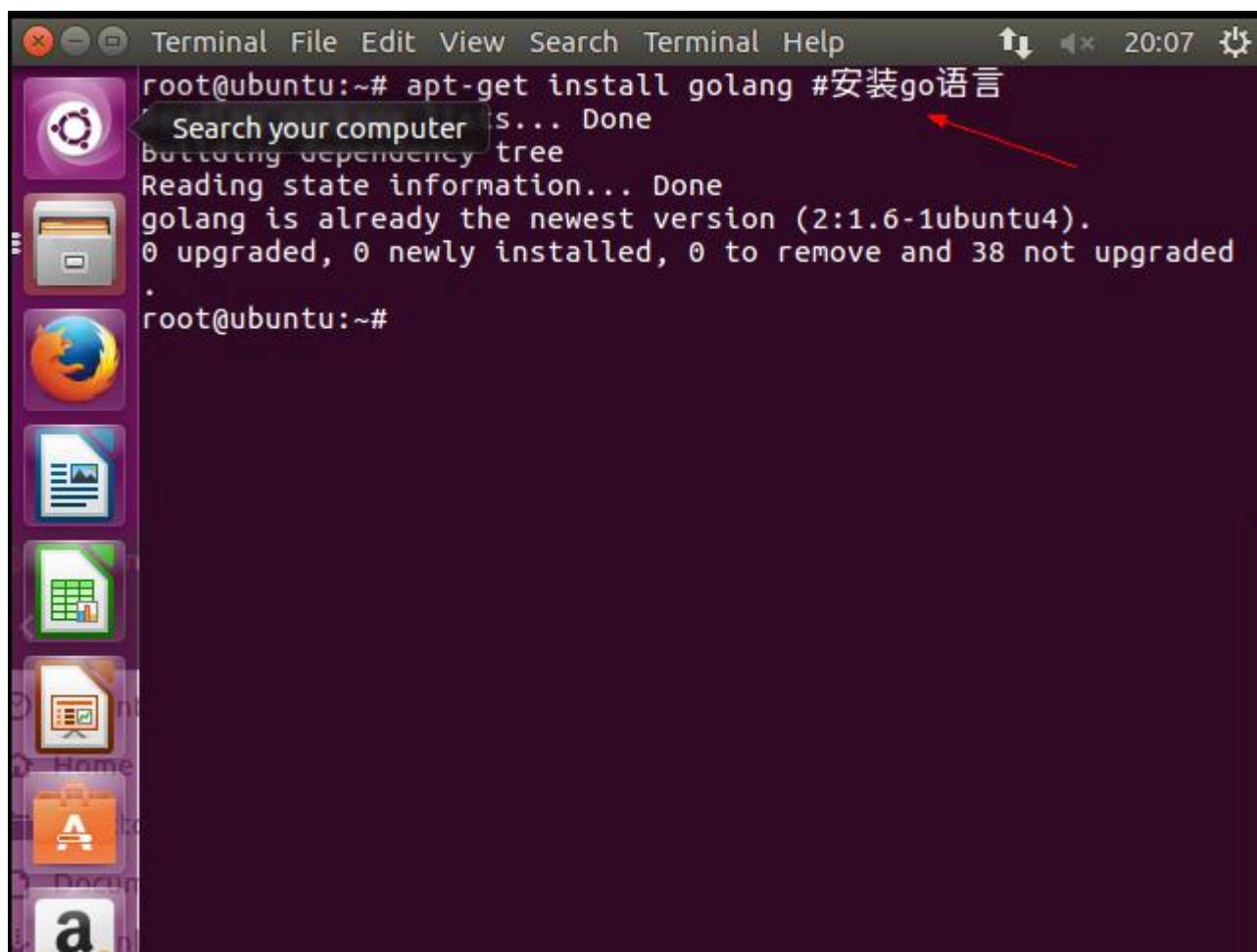
PS. 直接在服务器上下载的话实在太慢，可以先在本地下载好，然后用 `ftp` 放到服务器上去直接用，如果安装了 `VMware tools` 直接拖进去就可以了。

4.3.2. 安装 Go 语言开发环境

直接在命令模式下执行如下指令：

```
apt-get install golang #安装 go 语言
```

执行结果如下：



```
Terminal File Edit View Search Terminal Help
root@ubuntu:~# apt-get install golang #安装go语言
Building dependency tree
Reading state information... Done
golang is already the newest version (2:1.6-1ubuntu4).
0 upgraded, 0 newly installed, 0 to remove and 38 not upgraded
root@ubuntu:~#
```

4.3.3. 更改 ngrok 域名

在自己的域名管理中添加解析 A 记录，如下所示：

[添加解析](#) [批量导入解析](#) [导出解析记录](#) [新手引导设置](#)

• 建议您在电脑上修改公共DNS，让解析设置实时生效。 [下载DNS修改工具](#) [什么是公共DNS，如何修改？](#)

<input type="checkbox"/> 记录类型 ▲	主机记录 ▲	解析线路(运营商) ▲	记录值
<input type="checkbox"/> A	*.ngrok	默认	
<input type="checkbox"/> A	ngrok	默认	0.24. / 8.14
<input type="checkbox"/> A	www	默认	

将*.ngrok 与 ngrok 都指向您的主机 IP。

默认的域名是 ngrok 自己的，要替换成您自己的域名

```
export GOPATH=/usr/local/ngrok/ #设置环境变量，Go 语言的安装位置
```

```
export NGROK_DOMAIN="ngrok.yourdomain.com" #设置环境变量，ngrok 域名
```

PS. ngrok 名称可以任意，推荐名称为 ngrok 或者 tunnel

4.3.4. 为域名生成证书

```
openssl genrsa -out rootCA.key 2048
```

```
openssl req -x509 -new -nodes -key rootCA.key -subj  
"/CN=$NGROK_DOMAIN" -days 5000 -out rootCA.pem
```

```
openssl genrsa -out server.key 2048
```

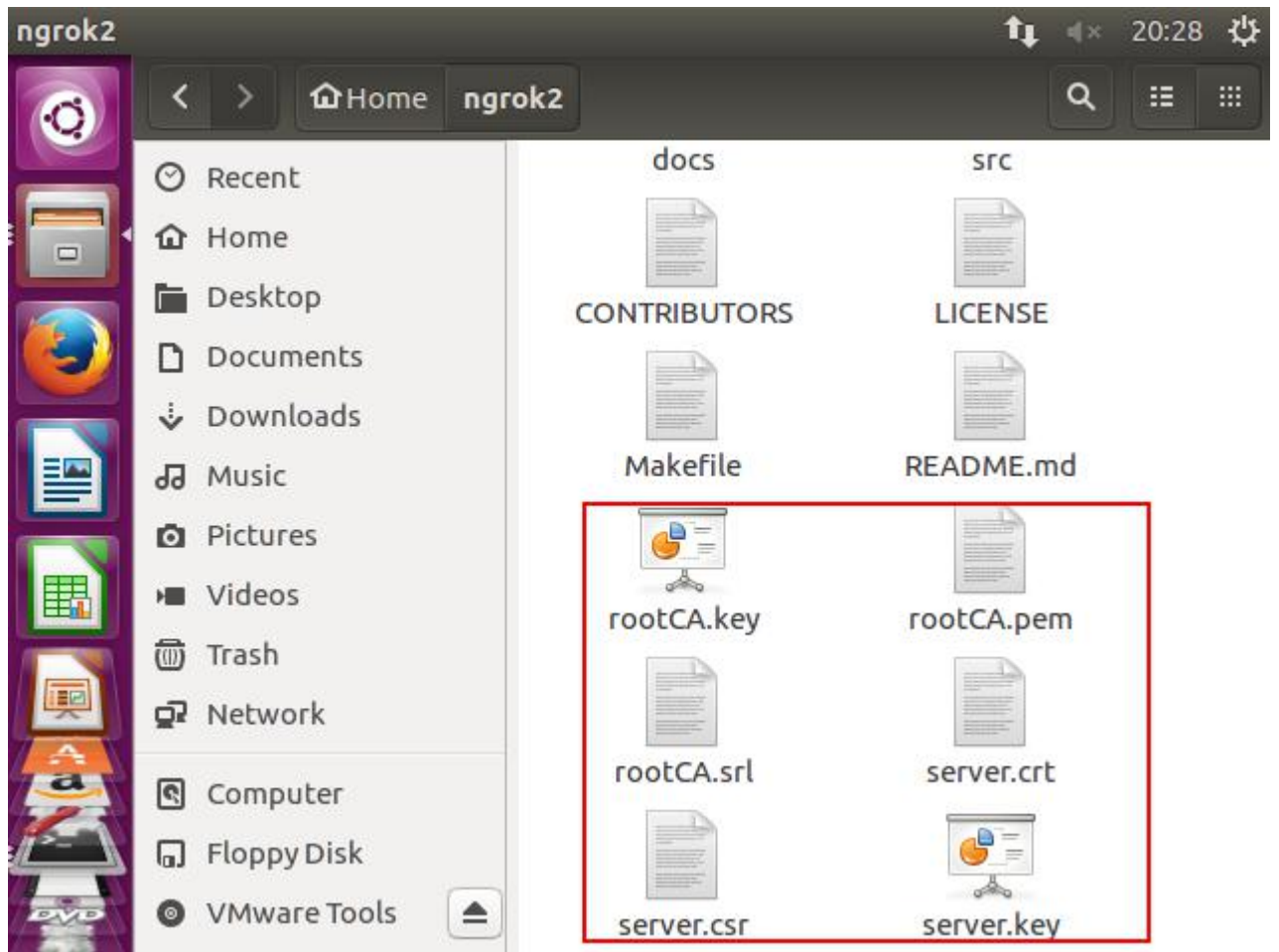
```
openssl req -new -key server.key -subj "/CN=$NGROK_DOMAIN" -out  
server.csr
```

```
openssl x509 -req -in server.csr -CA rootCA.pem -CAkey rootCA.key -  
CAcreateserial -out server.crt -days 5000
```

生成后的结果如下：


```
root@ubuntu: ~/ngrok2
server.csr: No such file or directory
root@ubuntu:~/ngrok2# export GOPATH=/usr/local/ngrok/ #设置环境变量, Go语言的安装位置
root@ubuntu:~/ngrok2# export NGROK_DOMAIN="ngrok.yourdomain.com" #设置环境变量, ngrok域名
Files
root@ubuntu:~/ngrok2# openssl genrsa -out rootCA.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
root@ubuntu:~/ngrok2# openssl req -x509 -new -nodes -key rootCA.key -subj "/CN=$NGROK_DOMAIN" -days 5000 -out rootCA.pem
root@ubuntu:~/ngrok2# openssl genrsa -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
root@ubuntu:~/ngrok2# openssl req -new -key server.key -subj "/CN=$NGROK_DOMAIN" -out server.csr
root@ubuntu:~/ngrok2# openssl x509 -req -in server.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out server.crt -days 5000
Signature ok
subject=/CN=ngrok.yourdomain.com
Getting CA Private Key
root@ubuntu:~/ngrok2#
```

证书如下:



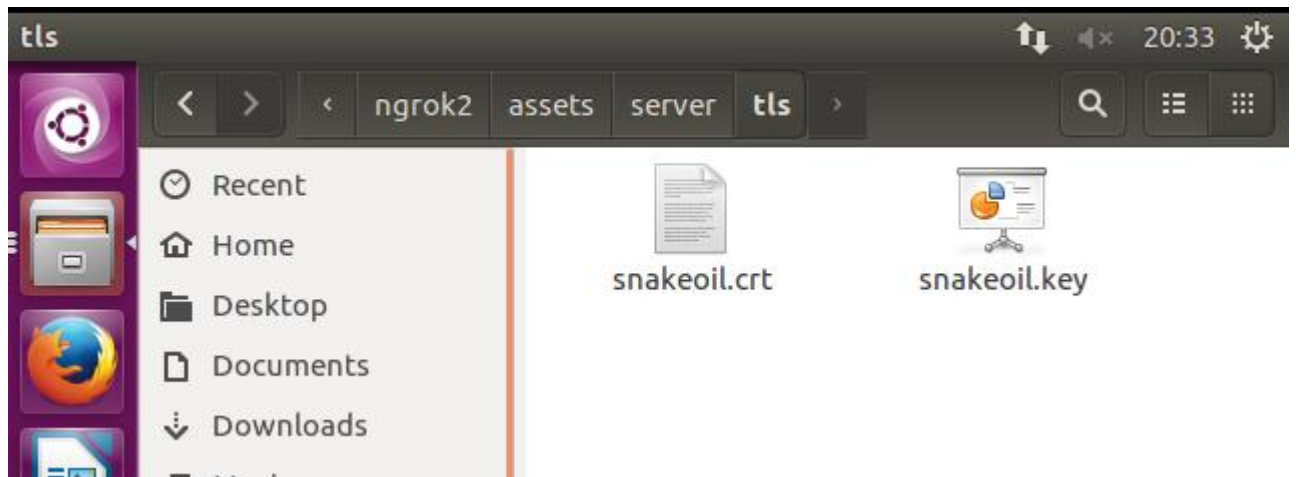
4.3.5. 拷贝证书到指定位置

```
cp rootCA.pem assets/client/tls/ngrokroot.crt #复制 rootCA.pem 到  
assets/client/tls/并更名为 ngrokroot.crt
```

```
cp server.crt assets/server/tls/snakeoil.crt #复制 server.crt 到  
assets/server/tls/并更名为 snakeoil.crt
```

```
cp server.key assets/server/tls/snakeoil.key #复制 server.key 到  
assets/server/tls/并更名为 snakeoil.key
```

运行结果：



4.3.6. 编译

由于 go 语言的特性，在编译时直接生成机器码，所以在运行过程中并不需要 go 的环境（非托管应用）。在 ngrok 目录下，运行一下命令分别生成对应的客户端与服务端。

#win 服务端

```
GOOS=windows GOARCH=386 make release-server #win 客户端
```

```
GOOS=windows GOARCH=386 make release-client #linux 服务端
```

```
GOOS=linux GOARCH=386 make release-server #linux 客户端
```

```
GOOS=linux GOARCH=386 make release-client
```

生成完成后，在工作目录的 bin 文件夹下，产生对应的文件。以编译 windows 平台为例，会产生“ngrok.exe”与“ngrokd.exe”这两个文件，前者客户端，后者需要运行在公网服务器上。

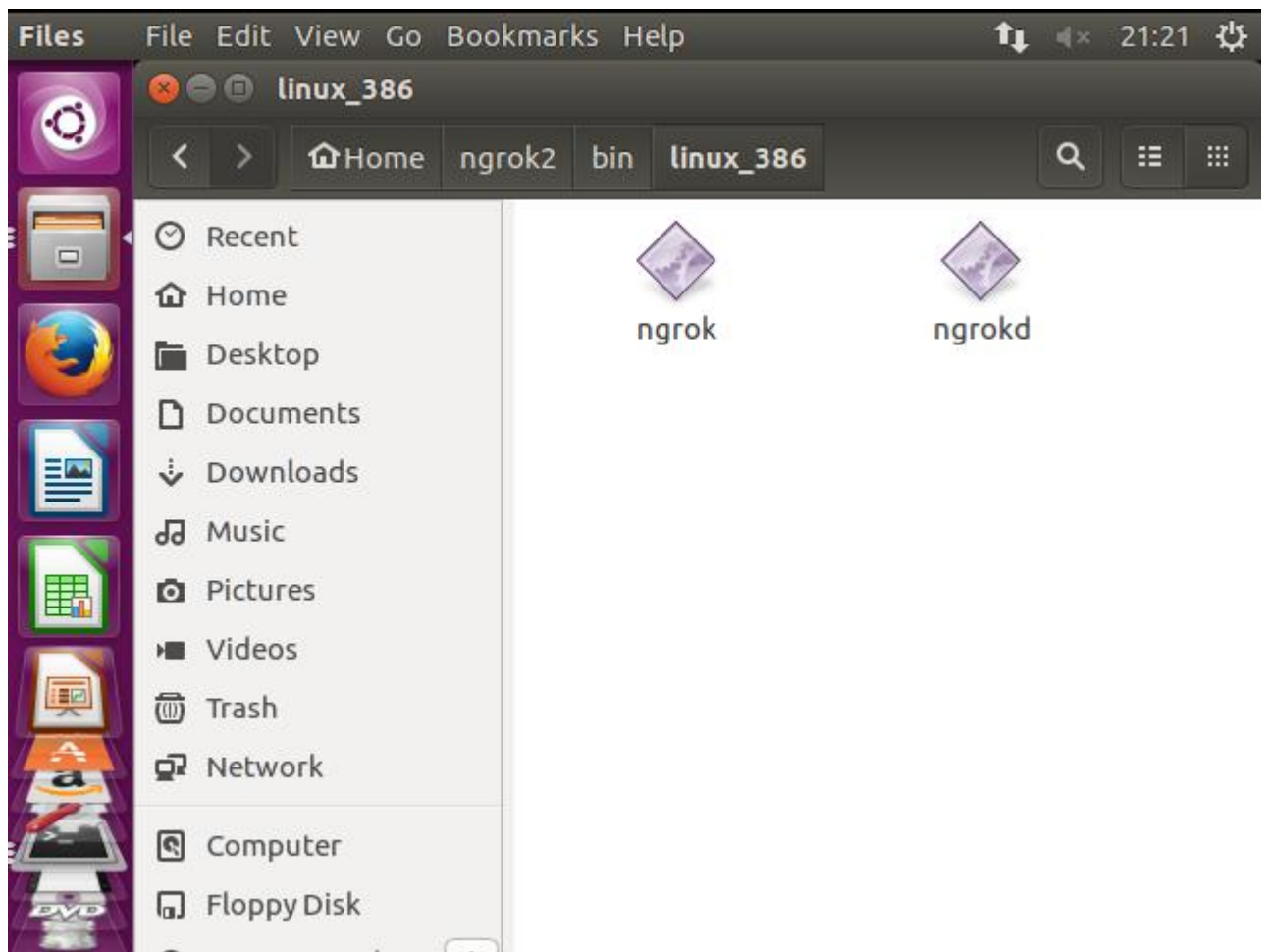
因为项目中引用了一些外部资源，生成会耗费一些时间，对网络也有一定的要求，太慢会中短，命令执行如下：


```

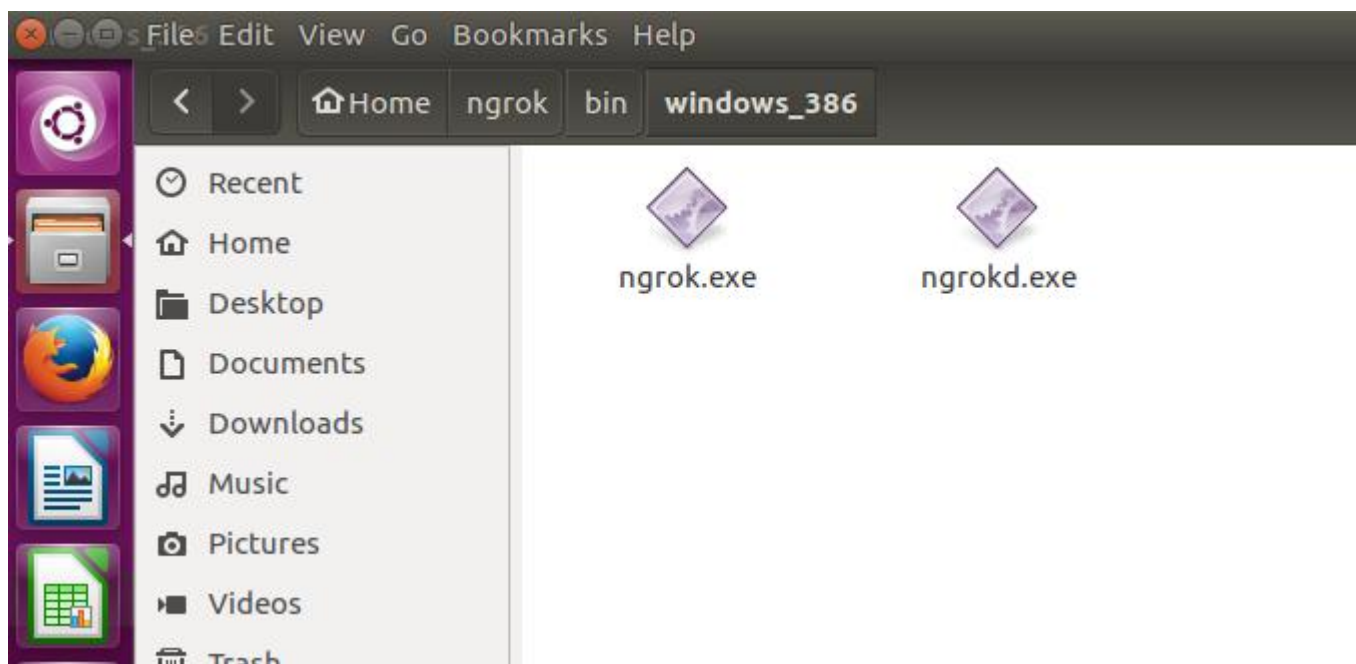
root@ubuntu: ~/ngrok2
assets/server/...
go get -tags 'release' -d -v ngrok/...
Fetching https://gopkg.in/yaml.v1?go-get=1
Parsing meta tags from https://gopkg.in/yaml.v1?go-get=1 (status code 200)
get "gopkg.in/yaml.v1": found meta tag main.metaImport{Prefix:"gopkg.in/yam
Root:"https://gopkg.in/yaml.v1"} at https://gopkg.in/yaml.v1?go-get=1
gopkg.in/yaml.v1 (download)
go install -tags 'release' ngrok/main/ngrok
root@ubuntu:~/ngrok2# #linux服务端
root@ubuntu:~/ngrok2# GOOS=linux GOARCH=386 make release-server
bin/go-bindata -nomemcopy -pkg=assets -tags=release \
    -debug=false \
    -o=src/ngrok/client/assets/assets_release.go \
    assets/client/...
bin/go-bindata -nomemcopy -pkg=assets -tags=release \
    -debug=false \
    -o=src/ngrok/server/assets/assets_release.go \
    assets/server/...
go get -tags 'release' -d -v ngrok/...
go install -tags 'release' ngrok/main/ngrokd
root@ubuntu:~/ngrok2# #linux客户端
root@ubuntu:~/ngrok2# GOOS=linux GOARCH=386 make release-client
bin/go-bindata -nomemcopy -pkg=assets -tags=release \
    -debug=false \
    -o=src/ngrok/client/assets/assets_release.go \
    assets/client/...
bin/go-bindata -nomemcopy -pkg=assets -tags=release \
    -debug=false \
    -o=src/ngrok/server/assets/assets_release.go \
    assets/server/...
go get -tags 'release' -d -v ngrok/...
go install -tags 'release' ngrok/main/ngrok
root@ubuntu:~/ngrok2#

```

生成结果:

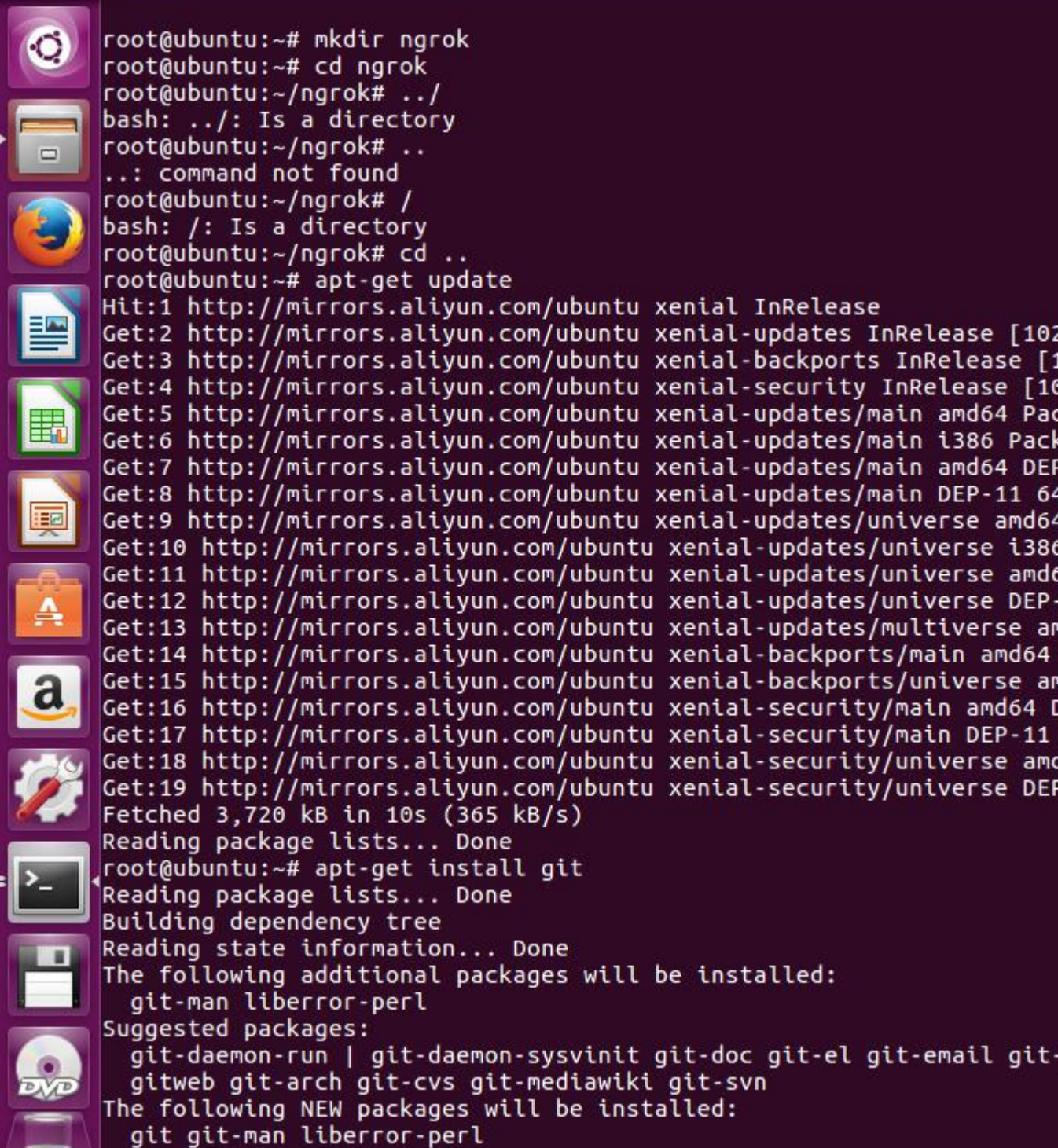


这里我还生成了两个运行在 windows 服务器与客户端的应用：




ngrok.exe 是客户端，ngrokd.exe 是服务端，下面是比较连续的操作结果。

root@ubuntu: ~



The image shows a terminal window on an Ubuntu desktop. On the left side of the terminal, there is a vertical column of desktop icons: a gear (system settings), a folder (home), a globe (internet), a document with a magnifying glass (search), a calendar, a presentation (slides), a shopping bag (software center), an Amazon logo, a wrench and gear (tools), a terminal icon (highlighted), a floppy disk, a DVD, and a laptop. The terminal text shows the following sequence of commands and outputs:

```
root@ubuntu:~# mkdir ngrok
root@ubuntu:~# cd ngrok
root@ubuntu:~/ngrok# ../
bash: ../: Is a directory
root@ubuntu:~/ngrok# ..
..: command not found
root@ubuntu:~/ngrok# /
bash: /: Is a directory
root@ubuntu:~/ngrok# cd ..
root@ubuntu:~# apt-get update
Hit:1 http://mirrors.aliyun.com/ubuntu xenial InRelease
Get:2 http://mirrors.aliyun.com/ubuntu xenial-updates InRelease [102
Get:3 http://mirrors.aliyun.com/ubuntu xenial-backports InRelease [1
Get:4 http://mirrors.aliyun.com/ubuntu xenial-security InRelease [16
Get:5 http://mirrors.aliyun.com/ubuntu xenial-updates/main amd64 Pac
Get:6 http://mirrors.aliyun.com/ubuntu xenial-updates/main i386 Pack
Get:7 http://mirrors.aliyun.com/ubuntu xenial-updates/main amd64 DEP
Get:8 http://mirrors.aliyun.com/ubuntu xenial-updates/main DEP-11 64
Get:9 http://mirrors.aliyun.com/ubuntu xenial-updates/universe amd64
Get:10 http://mirrors.aliyun.com/ubuntu xenial-updates/universe i386
Get:11 http://mirrors.aliyun.com/ubuntu xenial-updates/universe amd6
Get:12 http://mirrors.aliyun.com/ubuntu xenial-updates/universe DEP-
Get:13 http://mirrors.aliyun.com/ubuntu xenial-updates/multiverse an
Get:14 http://mirrors.aliyun.com/ubuntu xenial-backports/main amd64
Get:15 http://mirrors.aliyun.com/ubuntu xenial-backports/universe an
Get:16 http://mirrors.aliyun.com/ubuntu xenial-security/main amd64 D
Get:17 http://mirrors.aliyun.com/ubuntu xenial-security/main DEP-11
Get:18 http://mirrors.aliyun.com/ubuntu xenial-security/universe amc
Get:19 http://mirrors.aliyun.com/ubuntu xenial-security/universe DEP
Fetched 3,720 kB in 10s (365 kB/s)
Reading package lists... Done
root@ubuntu:~# apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-
  gitweb git-arch git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
```


The image shows a screenshot of an Ubuntu desktop environment. On the left side, there is a vertical dock containing several application icons: Dash, Home Folder, Firefox, LibreOffice Writer, LibreOffice Impress, Amazon, System Settings, Terminal, Files, and a DVD icon. The main area of the screen is occupied by a terminal window titled 'Terminal'. The terminal has a menu bar with 'Terminal', 'File', 'Edit', 'View', 'Search', and 'Help'. The terminal output shows the process of installing Git and its dependencies using apt-get. It lists suggested packages, shows the disk space requirements, and displays the progress of downloading and unpacking the packages. Finally, it shows the execution of 'git clone' to clone a repository from GitHub.

```
root@ubuntu:~# apt-get install git
Fetched 3,720 kB in 10s (365 kB/s)
Reading package lists... Done
root@ubuntu:~# apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-
  gitweb git-arch git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 92 not upgraded.
Need to get 3,918 kB of archives.
After this operation, 25.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirrors.aliyun.com/ubuntu xenial/main amd64 liberror-pe
Get:2 http://mirrors.aliyun.com/ubuntu xenial-updates/main amd64 git
Get:3 http://mirrors.aliyun.com/ubuntu xenial-updates/main amd64 git
Fetched 3,918 kB in 38s (101 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 183752 files and directories currently install
Preparing to unpack .../liberror-perl_0.17-1.2_all.deb ...
Unpacking liberror-perl (0.17-1.2) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.7.4-0ubuntu1.2_all.deb ...
Unpacking git-man (1:2.7.4-0ubuntu1.2) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.7.4-0ubuntu1.2_amd64.deb ...
Unpacking git (1:2.7.4-0ubuntu1.2) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up liberror-perl (0.17-1.2) ...
Setting up git-man (1:2.7.4-0ubuntu1.2) ...
Setting up git (1:2.7.4-0ubuntu1.2) ...
root@ubuntu:~# git clone https://github.com/inconshreveable/ngrok.git
Cloning into 'ngrok'...
remote: Counting objects: 2732, done.
remote: Total 2732 (delta 0), reused 0 (delta 0), pack-reused 2732
Receiving objects: 100% (2732/2732), 665.40 KiB | 53.00 KiB/s, done.
Resolving deltas: 100% (1298/1298), done.
Checking connectivity... done.
root@ubuntu:~#
```

五、部署服务器端主程序

5.1、部署到 Windows Server 服务器

将生成的 ngrokd.exe 文件复制到 windows 服务器中，当然如果要部署到 linux 中也是没有问题的。

这里我将 ngrokd.exe 放在 c:\grokeServer 目录下：



为了方便，我编写了一个批处理文件：ngrokserver2.bat

```
ngrokd.exe -tlsKey="snakeoil.key" -tlsCrt="snakeoil.crt" -domain="ngrok.  
你的域名.com" -httpAddr=":801" -httpsAddr=":802"
```

点击批处理运行结果如下：

```
C:\Windows\system32\cmd.exe

C:\grokeServer>ngrok.exe -tlsKey="snakeoil.key" -tlsCrt="snakeoil.crt" -domain="ngrok.
.com" -httpAddr=":801" -httpsAddr=":802"
[12:37:40 CST 2017/09/04] [INFO] <ngrok/log.(*PrefixLogger).Info:83> [registry]
[tun] No affinity cache specified
[12:37:40 CST 2017/09/04] [INFO] <ngrok/log.Info:112> Listening for public http
connections on [::]:801
[12:37:40 CST 2017/09/04] [INFO] <ngrok/log.Info:112> Listening for public https
connections on [::]:802
[12:37:40 CST 2017/09/04] [INFO] <ngrok/log.Info:112> Listening for control and
proxy connections on [::]:4443
[12:37:40 CST 2017/09/04] [INFO] <ngrok/log.(*PrefixLogger).Info:83> [metrics] R
eporting every 30 seconds
```

绑定的域名换成自己的域名，http 使用 801 端口，https 使用 802 端口，供客户端连接的管道端口设置为 4443 端口，必须前面的域名相同。

为了安全许多服务器会将端口屏蔽，我使用的是 ECS 服务器，默认 801，802 都是关闭的，需要手动开启，在阿里云的后台添加开放的端口就可了：

管理控制台

产品与服务

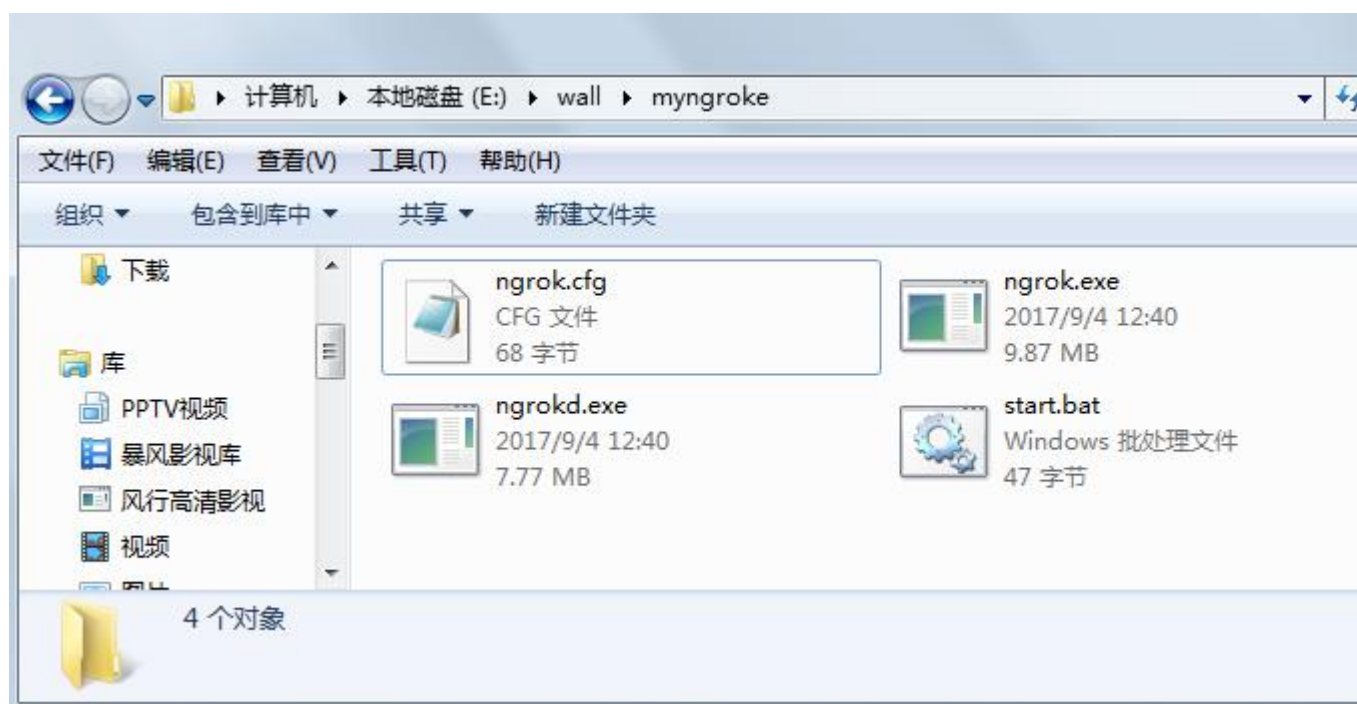
5.2、一键部署 ngrok 服务器（CentOS、Debian、Ubuntu）

如果编译生成 ngrok 的源代码生成应用太麻烦，你可以选择网友写的工具，支持一键部署到安装平台：CentOS、Debian、Ubuntu。

<https://github.com/clangcn/ngrok-one-key-install>

六、部署 ngrok 客户端

这里的客户端就是您的 web 应用程序所运行的主机，将 ubuntu 生成的 ngrok.exe 客户端应用复制到您的系统中：



添加配置文件 ngrok.cfg:

```
server_addr: "ngrok.你的域名.com:4443"
```

```
trust_host_root_certs: false
```

添加批处理 start.bat，如果只运行一次直接在命令行下输入命令也是一样的效果，内容如下：

```
ngrok.exe -subdomain kyt -config=ngrok.cfg 8987
```

其中 8987 为端口号，运行成功的结果如下所示：

```
C:\Windows\system32\cmd.exe
ngrok

Tunnel Status      online
Version           1.7/1.7
Forwarding        https://kyt.██████████.com:801 ->
Forwarding        http://kyt.██████████.com:801 ->
Web Interface     127.0.0.1:4040
# Conn            22
Avg Conn Time     92027.40ms

HTTP Requests
-----
GET /favicon.ico           200 OK
GET /Images/loginbg.jpg   200 OK
GET /js/jquery/jquery-1.12.3. 200 OK
GET /js/jscookie/js.cookie.js 200 OK
GET /Images/loadingLogin.gif 200 OK
GET /Images/dept_bg.jpg    200 OK
GET /Images/password_bg.jpg 200 OK
GET /Images/user_bg.jpg    200 OK
GET /Images/login_rptx.jpg 200 OK
```

看到这个界面时说明已成功了。

七、启动客户端并测试

打开浏览器，输入您映射后的域名就可以穿透内网访问您的 web 服务器了。



八、总结

一开始选择错了平台，在 windows 花了不少时间，在 ubuntu 下顺利完成。

无论是客户端还是服务器端最好都做成服务，更方便与稳定。

由于服务器上同时运行着 IIS，故服务端 Ngrok 启动时无法使用 80 端口，所以在上面，我使用了 801 作为 Ngrok 服务器的 http 端口，使用 IIS 的代理功能可以解决这个问题，[点击这里](#)。当然也可以使用 nginx 将 80 转换成其它端口。

许多内容都参考了网友的文章。

如果服务器搭建好了，只运行客户端穿透内网一分钟够了：）。

欢迎您提供更加好的解决方案，欢迎您提供更多的免费代理服务器，我随时更新，谢谢！

作者：张果

原文：<http://www.cnblogs.com/best/p/7465444.html>