虚拟机组网

11.06：目前的主要工作是完成静态ip配置，连接各虚拟机网络；学会使用scp工具，如何利用这个工具相互之间传输数据，包括ssh的配置；
下一步：配置linux的python相关环境；跑通脚本；

静态ip配置

系统：ubuntu 24.04
一些工具的说明：
net-tools工具，是Linux操作系统中用于网络配置和故障排除的命令行工具集。它提供了一系列实用程序，允许用户查看和修改网络接口、路由表、网络连接状态等。利用net-tools工具修改ip时修改的是当前运行时的网络配置，而不是任何配置文件。这意味着这些修改是临时性的，在系统重启后会丢失。目前linux更加主流的网络工具集是iproute2，一般系统自带。

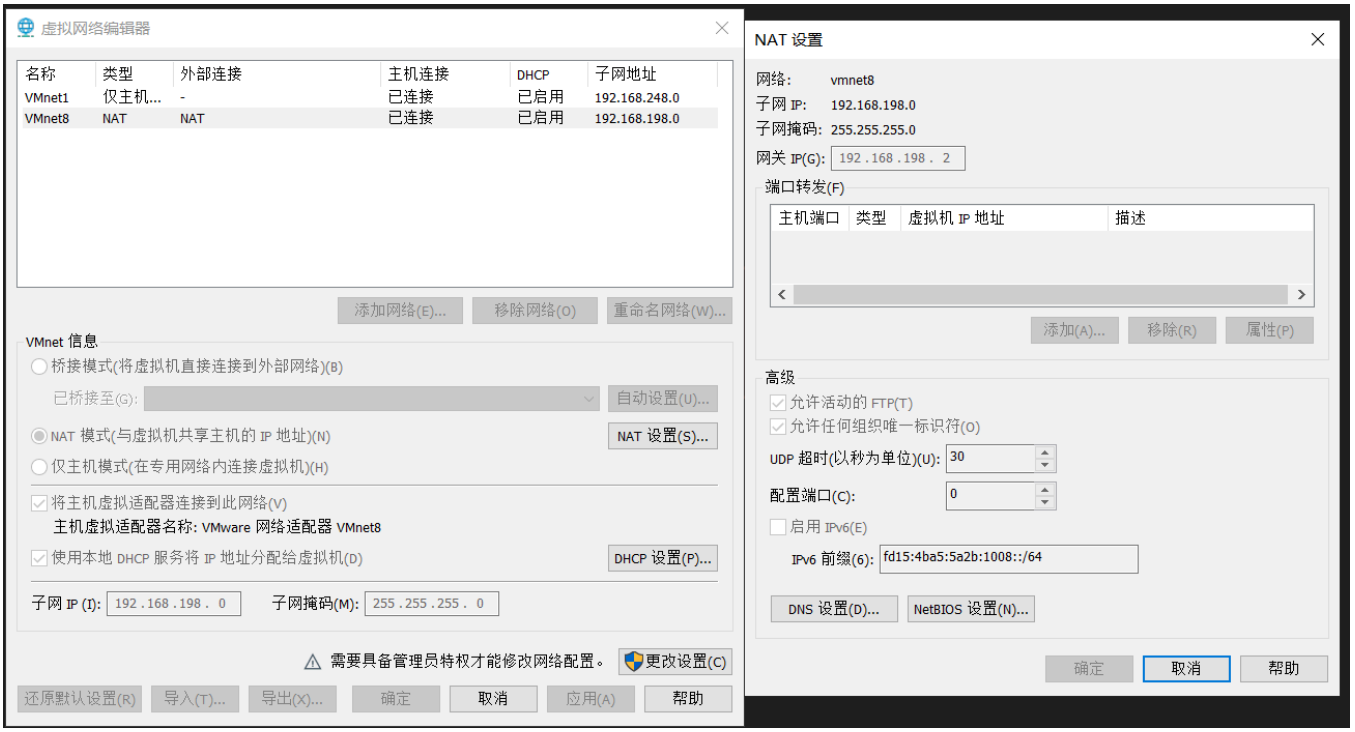
network-manager：linux中用于管理网络配置的工具；是一个动态网络配置守护进程，主要是维护网络连接；能够自动化管理网络连接，具有GUI界面；通常通过 GUI 或命令行工具（nmcli）进行交互式操作。是大多数 Linux 桌面发行版的默认网络管理工具。

netplan：linux中用于管理网络配置的工具；Netplan 是 Ubuntu 引入的一种新的网络配置抽象工具。它使用 YAML 文件来声明式地定义网络接口的配置。Netplan 本身并不直接管理网络，而是将 YAML 配置转换为底层网络管理工具（如 NetworkManager 或 systemd-networkd）能够理解的配置。声明式配置，有一套语法，能够进行自动化脚本化

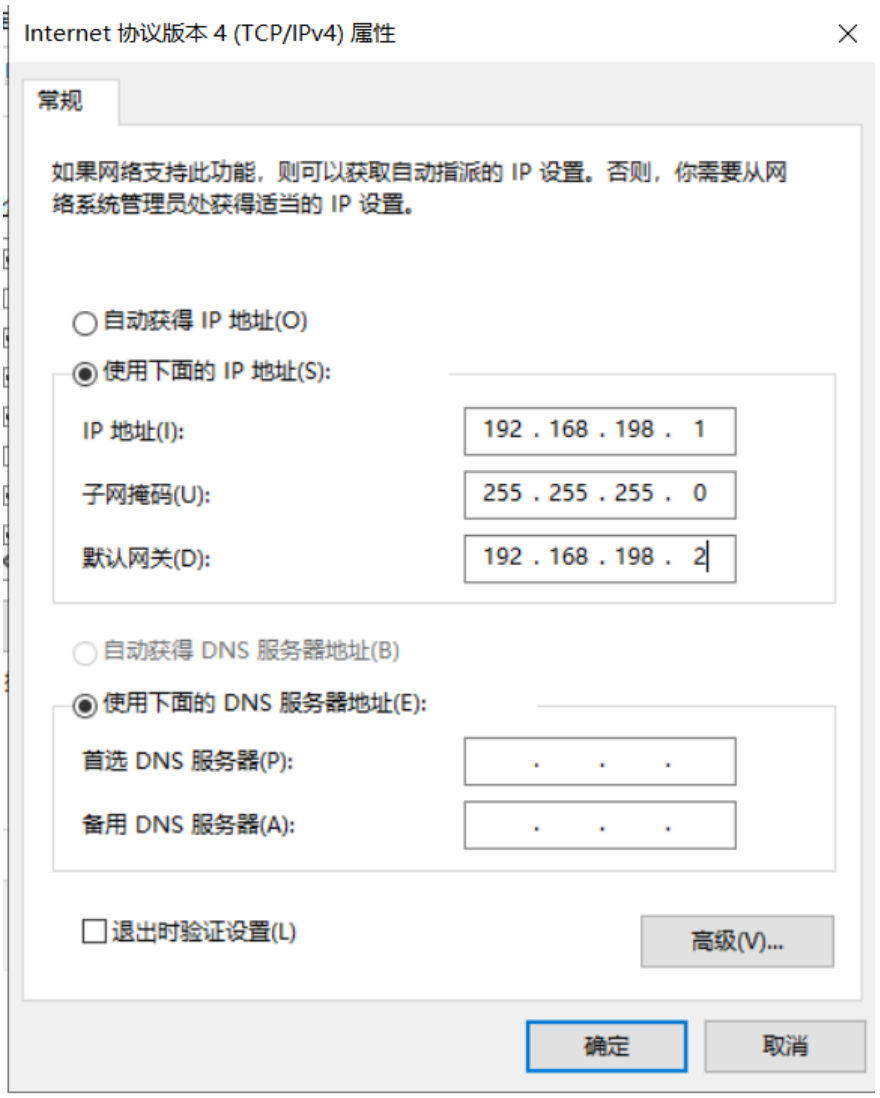
一般情况下，在桌面版 Ubuntu 中，通常 Netplan 会配置为使用NetworkManager作为其渲染器。这意味着你通过Netplan的YAML文件定义的网络设置，Netplan会将其转换为NetworkManager的配置文件，然后由NetworkManager来实际管理和应用这些网络连接。

宿主机需要进行的设置：

虚拟网关构建的虚拟网络vmnet8的一些设置如下：



虚拟机通过nat利用宿主机接到公网，注意对宿主机的网络vmnet8的ipv4进行配置，填写子网ip，子网掩码，网关如下：



此处我们直接使用的是nat模式来连接虚拟机，整个网络中没有对路由器的强调，各虚拟机都通过宿主机连接到公网
实际电台而言，设备相互之间应该是等价的，怎么去同时兼具路由主机角色可能是一个需要考虑的问题，这个可能要拿到电台才知道；实际电台可能要用到桥接的模式

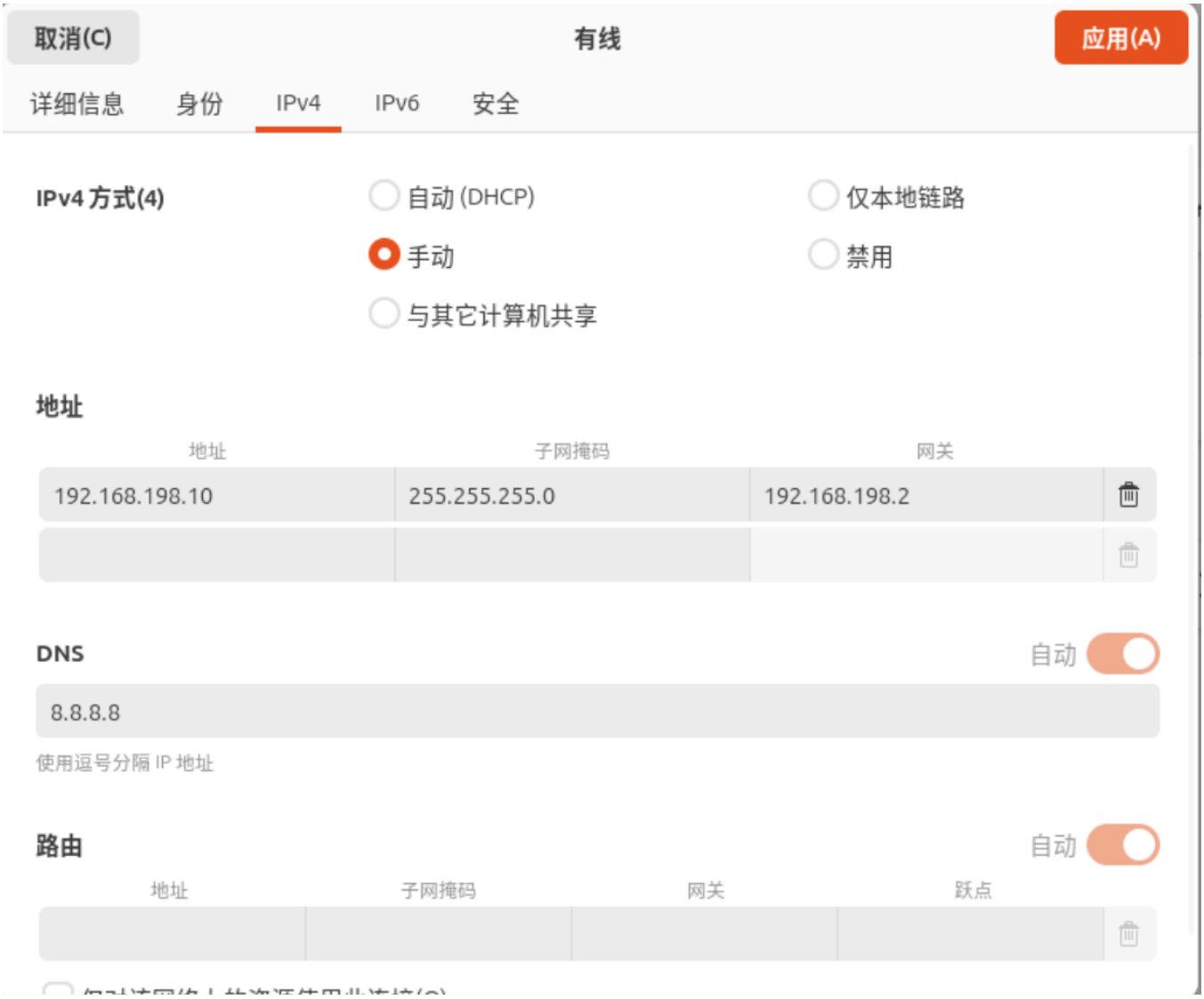
设置静态IP的方式：①终端修改网络配置文件；②直接在gui的设置界面修改；
对于第一种方式，ubuntu系统配置文件的地址为：/etc/netplan
在这里可以新建一个yaml的配置文件进行ip配置，大致格式如下：

```yaml
network:
  version: 2
  renderer: networkd # 或者 systemd-networkd (ubuntu server) , 也可以是NetworkManager (ubuntu desktop)
  ethernets:
    enp0s3: # 网卡名称，例如 eth0, enp0s3, eno1 等。请使用 `ip a` 命令查看实际网卡名称。
      dhcp4: no # 禁用 IPv4 DHCP
      addresses:
        - 192.168.1.100/24 # 你的静态 IP 地址和子网掩码
      routes:
        - to: default
          via: 192.168.1.1 # 你的网关地址
      nameservers:
        addresses:
          - 8.8.8.8 # 主 DNS 服务器
          - 8.8.4.4 # 备用 DNS 服务器
```

需要注意配置文件的名称前方的序号，代表了执行的优先级，数字越大，优先级越高

第二种在gui上进行修改则是本质上直接修改的networkmanager，会自动生成一个配置文件，对于Ubuntu desktop可以用这种方式，更加简单方便；如果是没有gui的server则是只能修改配置文件；

对于第二种方式，可能会出现修改ip以后无法访问公网的问题，一般是因为自动的DNS不行，需要改为手动的，如8.8.8.8等



这样修改的配置文件存放地址为：/etc/netplan

关于配置文件作以下说明：
这个地址里面会有三个配置文件
第一个是01-network-manager-all.yaml

```yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
```

这设置了Netplan的全局渲染器（renderer）为networkmanager；

第二个是50-cloud-init.yaml

```yaml
network:
  version: 2
  ethernets:
    ens33:
      dhcp4: true
```

配置了虚拟机的初始化网络设置，也就是让虚拟机通过接口ens33连接到了宿主机网络；
第三个文件为14f59568-5076-387a-aef6-10adfcca2e26.yaml
这一串名称的代表NetworkManager连接配置文件的唯一标识符

```yaml
network:
  version: 2
  ethernets:
    ens33:
      renderer: NetworkManager
      match: {}
      addresses:
        - "192.168.198.10/24"
      nameservers:
        addresses:
          - 8.8.8.8
      networkmanager:
        uuid: "14f59568-5076-387a-aef6-10adffcca2e26"
        name: "netplan-ens33"
        passthrough:
          connection.timestamp: "1762270035"
          ipv4.address1: "192.168.198.10/24,192.168.198.2"
          ipv4.method: "manual"
          ipv6.method: "disabled"
          ipv6.ip6-privacy: "-1"
          proxy._: ""
```

文件名以90-开头，具有更高的优先级，会覆盖之前的网络默认配置；这个文件配置了静态ip和dns等设置，是实际上起作用的配置文件；
另外，设置了静态的ipv4就会默认禁用hdcp，使用静态ip

这样就为每个vm设置好了静态ip，可以相互ping通过；

然后就是考虑相互之间怎么转发
linux中的各设备之间数据传输工具主要有scp、sftp、rsync等
前两者都是基于ssh协议，但是传输过程中不能重连，必须从头传到尾传完整个文件；rsync则是更像一个同步工具，更擅长传输大文件或者大量文件，只传输差异部分，支持断点传输，本质上传输的是修改了的部分；

主机之间的文件转发

然后就是考虑相互之间怎么转发
linux中的各设备之间数据传输工具主要有scp、sftp、rsync等
前两者都是基于ssh协议，但是传输过程中不能重连，必须从头传到尾传完整个文件；rsync则是更像一个同步工具，更擅长传输大文件或者大量文件，只传输差异部分，支持断点传输，本质上传输的是修改了的部分；
目前先用简单的scp试一下

scp工具隶属于openssh软件包，安装指令如下：

```shell
sudo apt update
sudo apt install openssh-client
```

文件传输命令行为：

```shell
scp [选项] [源文件/目录] [目标文件/目录]
```

具体可以示例为：

```shell
scp /path/to/local/file.txt username@remote_host:/path/to/remote/directory/
```

目前的传输为：

```shell
scp /home/adhoc1/image.jpg adhoc1@192.168.198.11:/home/adhoc1/
```

目前尝试一下一张图片，失败，需要配置一下ssh服务
安装ssh服务：

```shell
sudo apt update
sudo apt install openssh-server
```

启动ssh服务：

```shell
sudo systemctl start ssh
```

或者：

```shell
sudo service ssh start
```

设置ssh服务开机自启动：

```shell
sudo systemctl enable ssh
```

检查ssh状态：

```shell
sudo service ssh status
```

检查防火墙设置，允许ssh服务的22端口：

```shell
sudo ufw allow ssh
sudo ufw enable # 如果防火墙未启用
```

这样再进行传输

```shell
scp /home/adhoc1/image.jpg adhoc1@192.168.198.11:/home/adhoc1/
```

第一次连接需要将传输目标写入know_hosts名单，yes即可，然后还要输入用户密码（这个可以后续优化为密钥认证）

```
root@adhoc1-VMware-Virtual-Platform:~# scp /home/adhoc1/image.jpg adhoc1@192.168
.198.11:/home/adhoc1/
adhoc1@192.168.198.11's password:
image.jpg                                     100%  135KB  35.3MB/s   00:00
root@adhoc1-VMware-Virtual-Platform:~#
```

另外，这里最好使用非root用户，root用户一般是不允许使用密码进行，登录root需要使用密钥认证，密钥认证的步骤写在后面的一个模块。

ssh服务的密钥认证流程

安装ssh服务：

```shell
sudo apt update
sudo apt install openssh-server
```

启动ssh服务：

```shell
sudo systemctl start ssh
```

或者：

```shell
sudo service ssh start
```

设置ssh服务开机自启动：

```shell
sudo systemctl enable ssh
```

检查ssh状态：

```shell
sudo service ssh status
```

检查防火墙设置，允许ssh服务的22端口：

```shell
sudo ufw allow ssh
sudo ufw enable # 如果防火墙未启用
```

这样这台desktop/server便可以被ssh连接了
如果是要ssh密钥登录 root/其他 用户，需要进行密钥认证
首先，客户端生成密钥对

```shell
ssh-keygen -t rsa -b 4096
```

其中加密方式为rsa。也可以使用其他的
然后系统会提示保存密钥文件的位置，通常默认为 ~/.ssh/id_rsa（私钥）和 ~/.ssh/id_rsa.pub（公钥）
然=后会提示使用一个密码进行密钥连接，直接回车，不设置密码；

将生成的公钥复制到允许登录客户名单内：路径如下
对于普通用户 username：/home/username/.ssh/authorized_keys
对于 root 用户：/root/.ssh/authorized_keys
也可以使用命令行直接复制如下

```shell
ssh-copy-id user@remote_host
```

然后输入密码就添加成功

涉及到root用户的话然后可能还需要更改服务器端的ssh服务配置文件中的一些配置（具体哪些后续再写），让密钥连接root也是被允许的
这个配置文件的路径一般为：/etc/ssh/sshd_config.d

配置linux系统的python环境

详细配置一下linux中的python环境，记录一下配置过程
安装vscode：可以直接官网安装或者命令行安装

tips

DHCP服务：会为网络中的每个设备分配一个ip，随着租凭期的到期，会适当改变ip（本质就是一个动态分配）

计算机网络的回环接口：本地进程之间的相互通信；ens33一般是虚拟网络的接口名称

linux系统中互ping会一直发送icmp请求，终端会一直反馈实际的延迟为多少；Windows则是一般4次请求

olsrd：olsrd（the olsr daemon）是OLSR协议（Optimized Link State Routing Protocol，RFC 3626）的一个开源、稳定且功能丰富的实现。它是一个在后台运行的守护进程（daemon），专门用于在移动自组织网络（MANET）中自动建立和维护路由表，使得网络中的任何节点都能与其他节点通信。后面可能会用到这个工具。