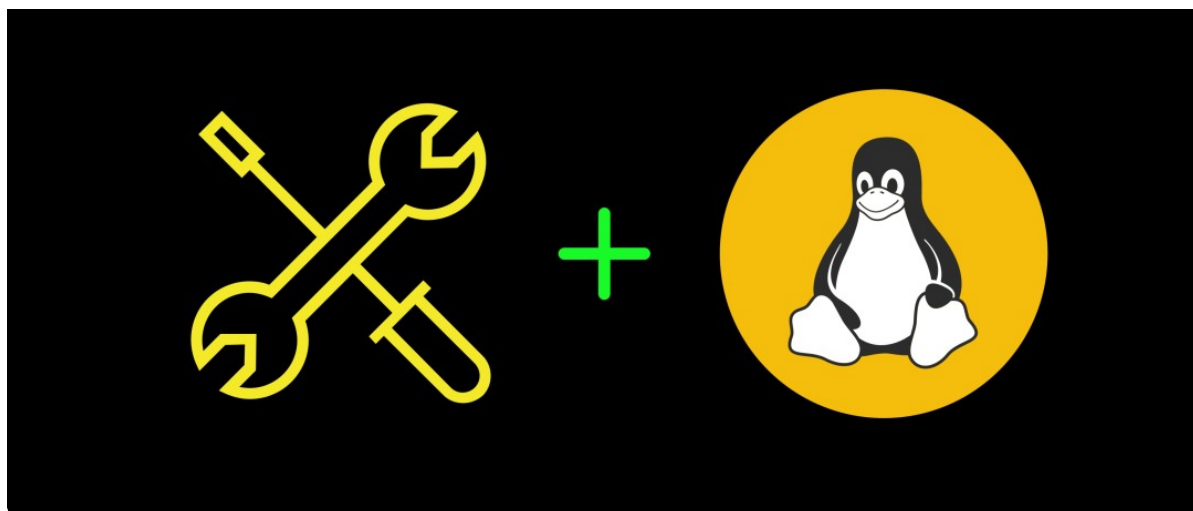


# 人手一套Linux环境之：Windows版本教程

原创 hansonwong99 CodeSheep 5月3日



## 缘由

---

之前写的一篇 《[Linux环境搭建保姆级操作](#)》 中是基于macOS宿主机系统搭建的，评论区有很多小伙伴要Windows版本的操作教程。

所以这不就趁着五一假期，从箱底掏出了那台破Windows电脑操作了一遍嘛！



其实Windows版的操作总体也大差不差，只是部分操作界面有点小变化。

还是那句话，既然学编程，提前备好Linux系统环境非常重要，建议人手一套，这样方便后续 **学Linux、用Linux、Linux环境编程、应用和项目部署、工具实验**等一系列学习和实践

## 软件版本

---

- 物理宿主机系统： Windows 10 专业版
- 虚拟机软件： VMware Workstation 10.0.1 版本
- CentOS 操作系统 ISO 镜像： CentOS 7.4 64位
- SSH终端软件： SecureCRT
- SFTP文件传输工具： WinSCP

## 1、创建新的虚拟机



## 2、选择虚拟机硬件兼容性



默认即可

**3、加载Linux系统ISO镜像**



**4、虚拟机命名并存储**



**5、自定义虚拟机配置**

处理器按需配置：



内存按需配置：

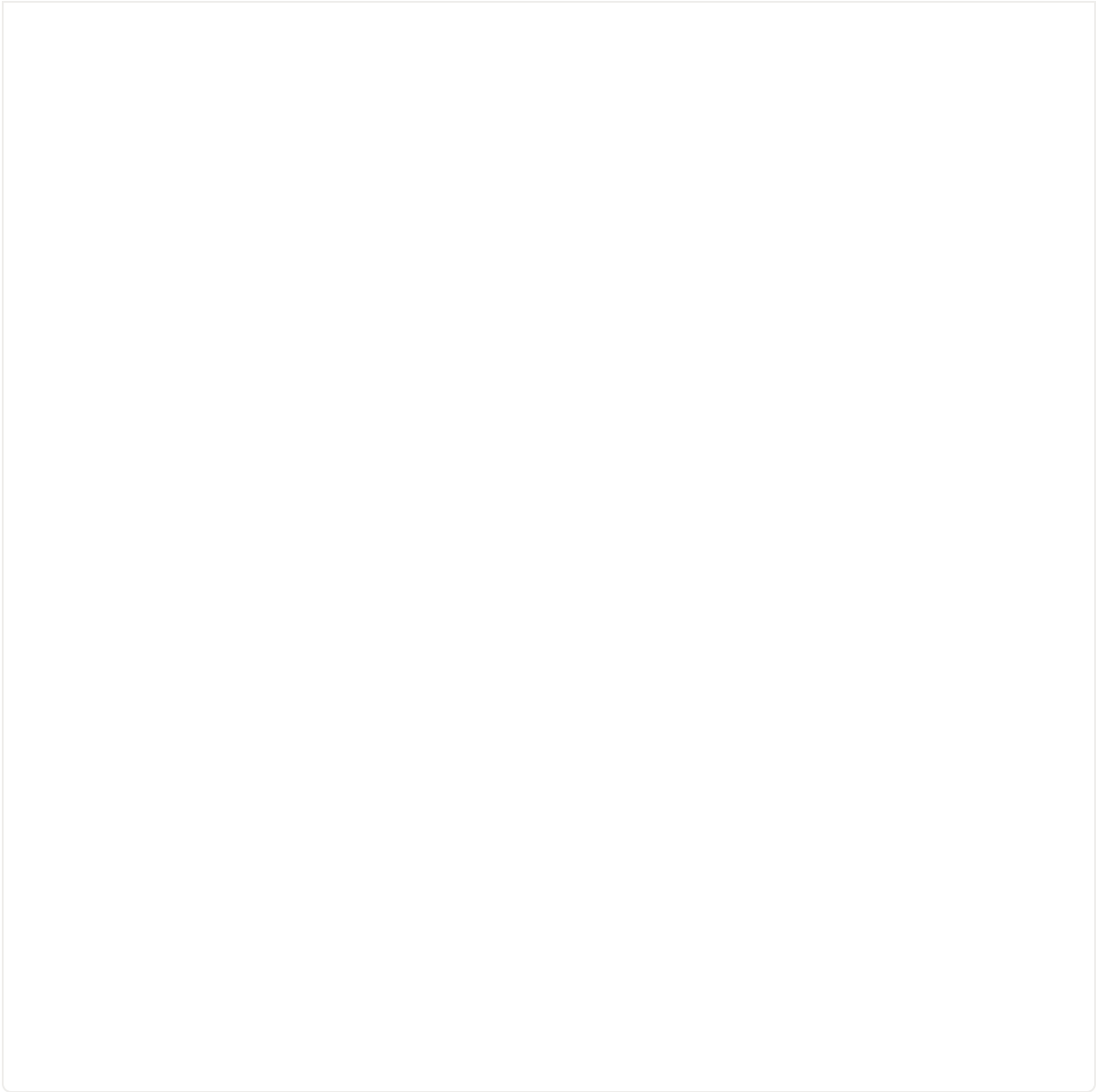
网络部分选择「桥接模式」即可：

I/O控制器选择默认即可：

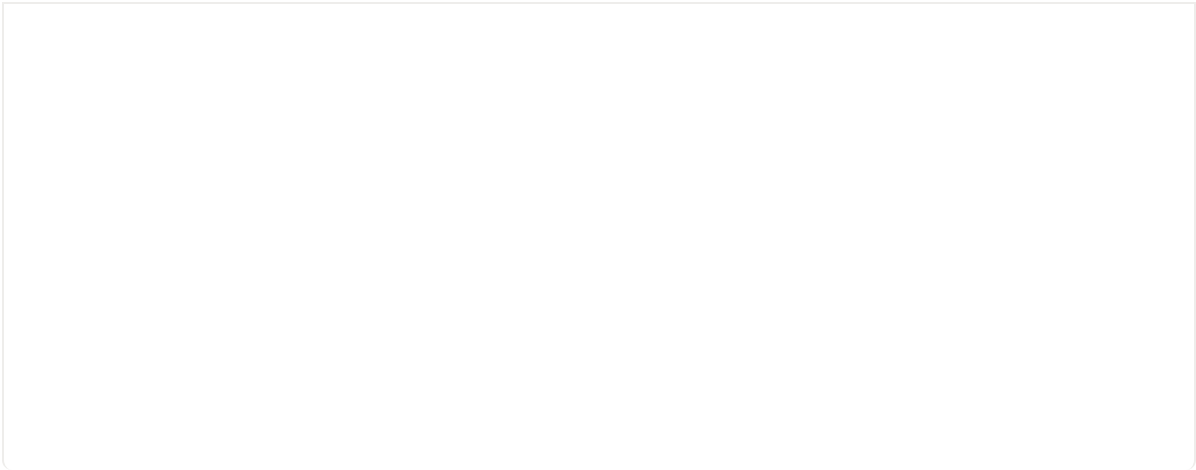


硬盘类型选择默认即可：

硬盘容量按需分配：

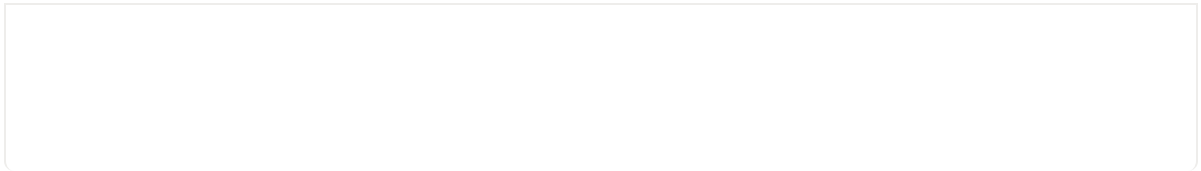


**6、安装设置，开启安装**





**7、进入系统安装界面**



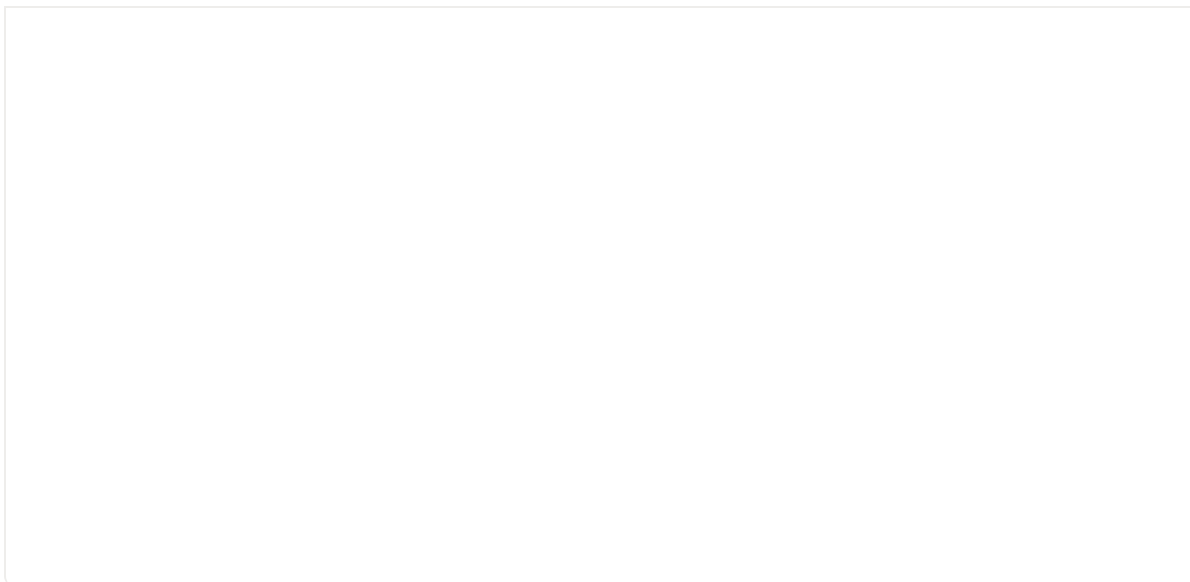


---

## 8、选择安装语言



## 9、选择预安装的软件

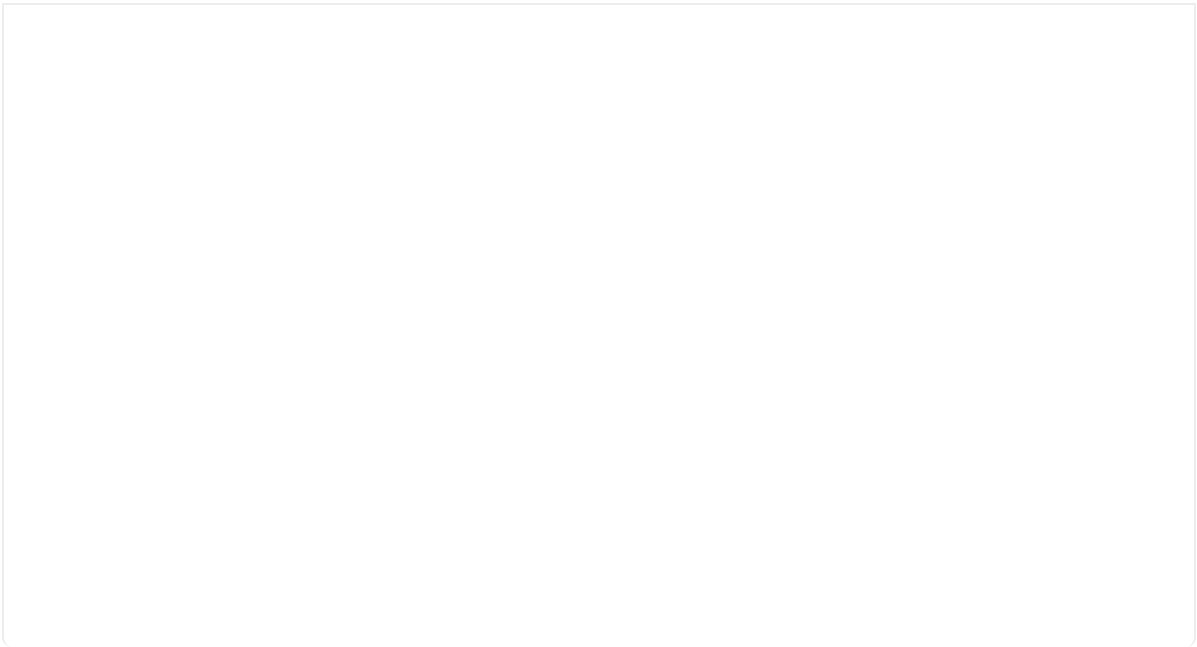


10、配置分区



没有特别需求可以选择自动分区，大家如果有需要可以自定义分区。

**11、进入正式安装过程**



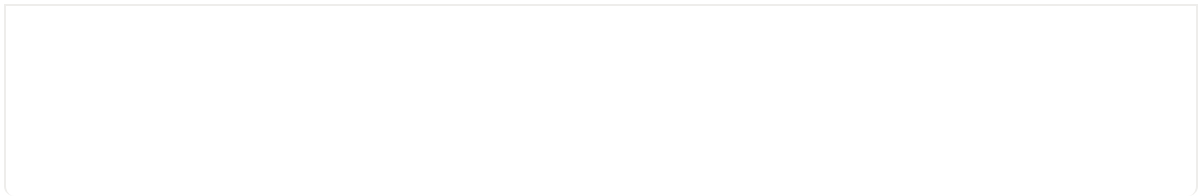


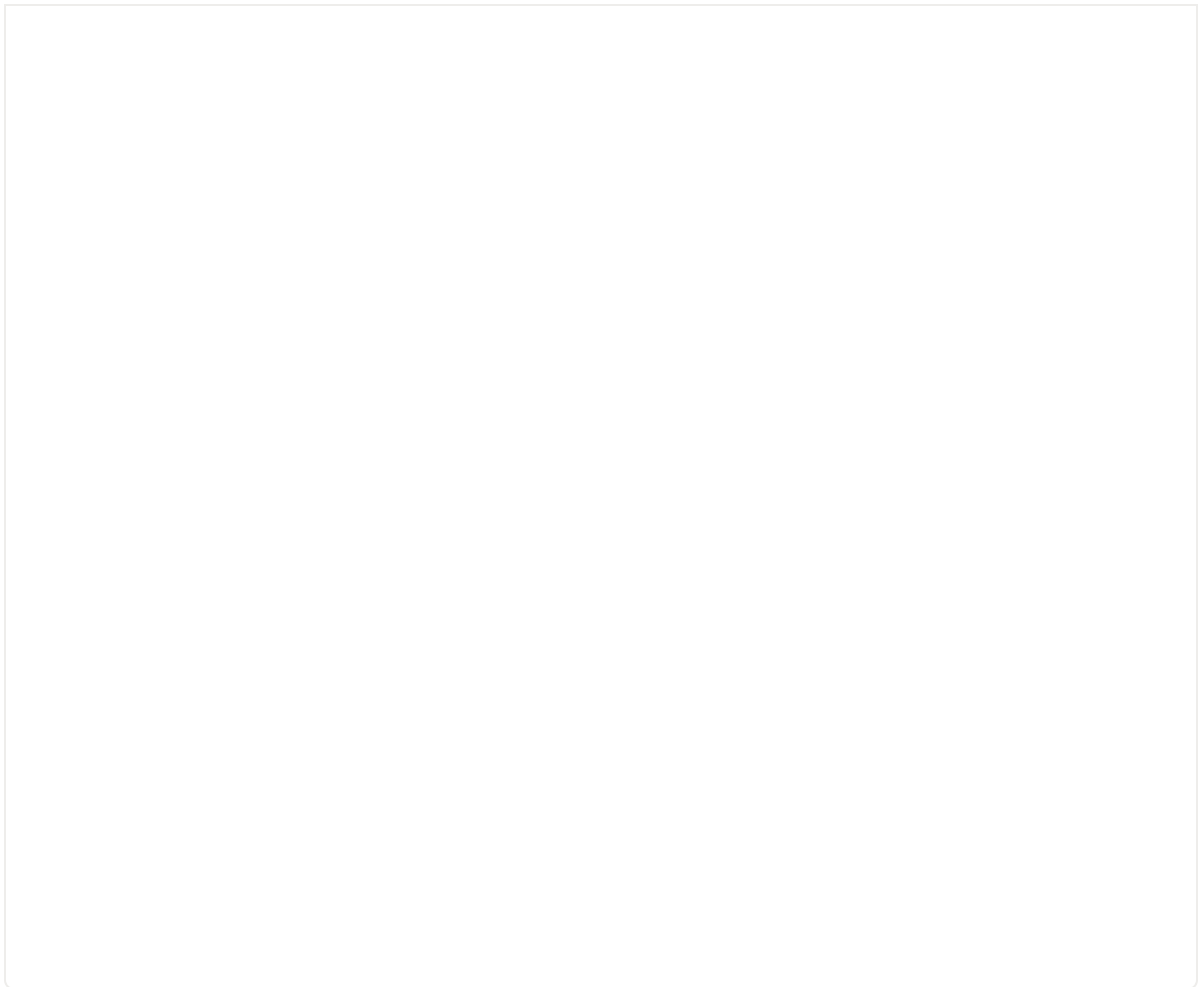


**12、安装完成并重启**



**13、进入新系统**





系统是装好了，但还有几个问题

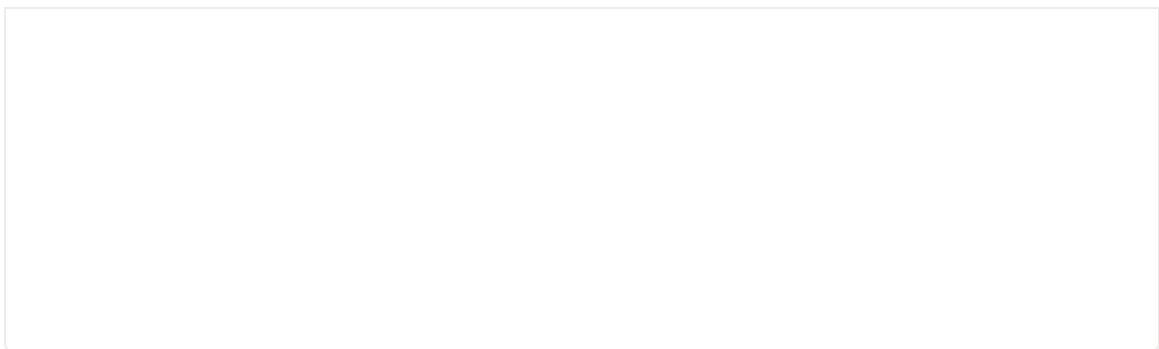
---

**问题一：** 虚拟机内Linux系统与外网无法连通



**问题二：** 虚拟机内Linux系统与外部宿主机无法连通

比如我这里的物理宿主机的IP地址为： 192.168.31.156



**问题三：** 虚拟机内Linux系统节点与节点之间无法连通（如果装了多个Linux节点的话）

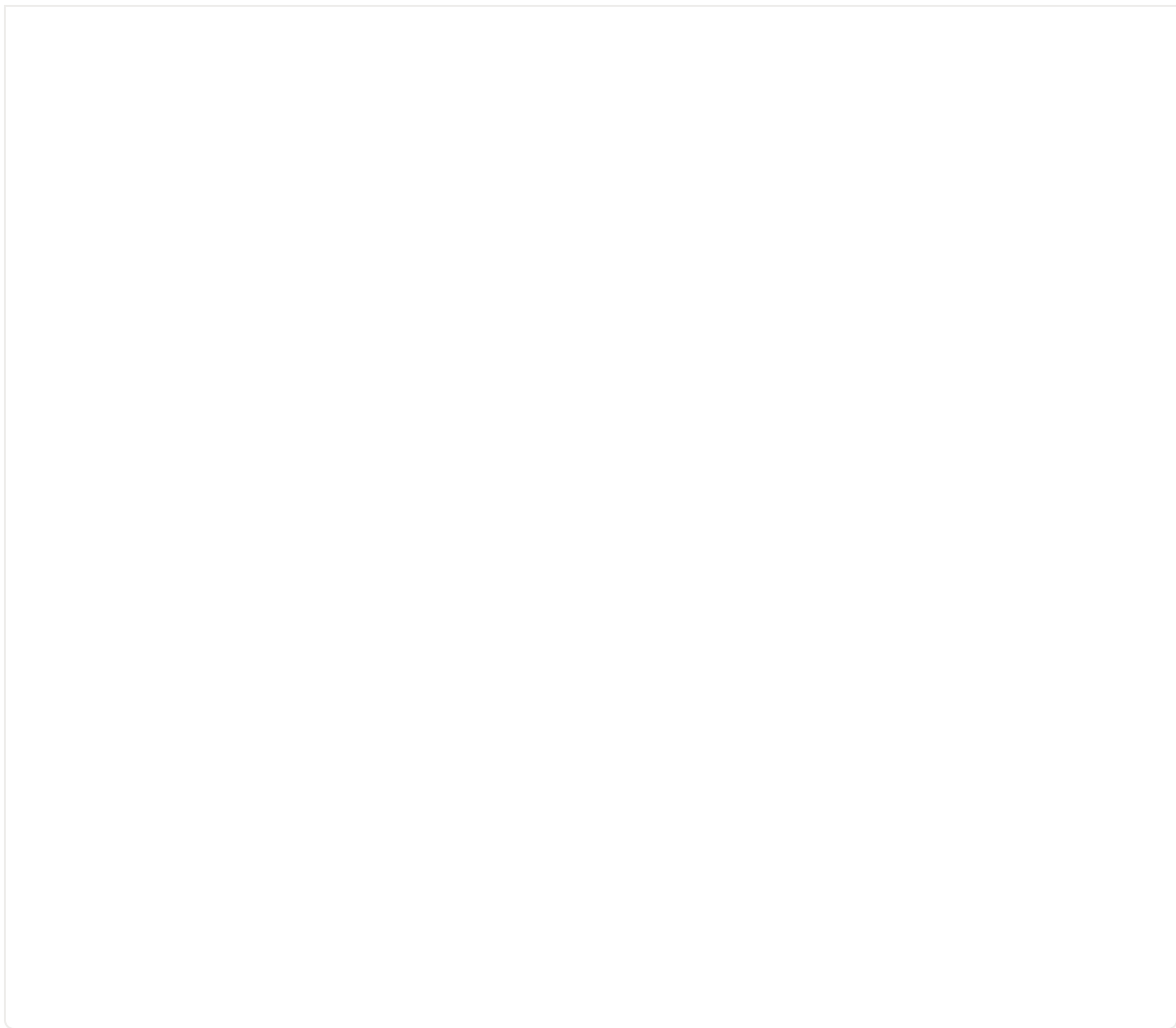
## 网络配置（极其重要！）

---

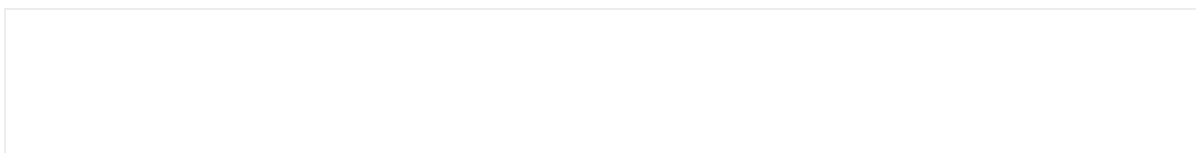
### 1、首先尝试查看虚拟机系统的IP地址

使用命令 `ifconfig` 进行查看。我们会发现装好的系统并没有为它设置IP地址。

### 2、设置虚拟机与物理宿主机的网络连接



选择**桥接模式**，并选择桥接到物理宿主机的上网网卡即可：



小伙伴们可以按实际情况进行选择。

### 3、为虚拟机配置固定静态IP

首先使用 `dhclient` 工具为本机分配一个网络内可用的IP地址：

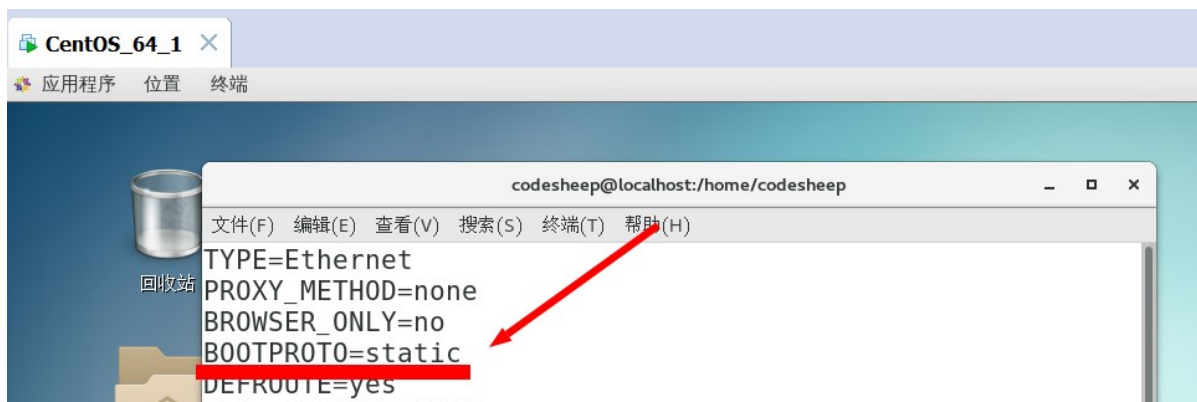
接下来编辑虚拟机系统网卡配置，将上面分配所得的IP地址配置进去：

使用命令编辑：`vim /etc/sysconfig/network-scripts/ifcfg-ens33`

修改配置如下：

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=824ec4bd-a9ae-4410-8346-17ce7f3dd111
DEVICE=ens33
ONBOOT=yes
IPADDR=192.168.31.21
NETMASK=255.255.255.0
GATEWAY=192.168.31.1
DNS1=119.29.29.29
```

尤其注意下图红色标记部分的配置：





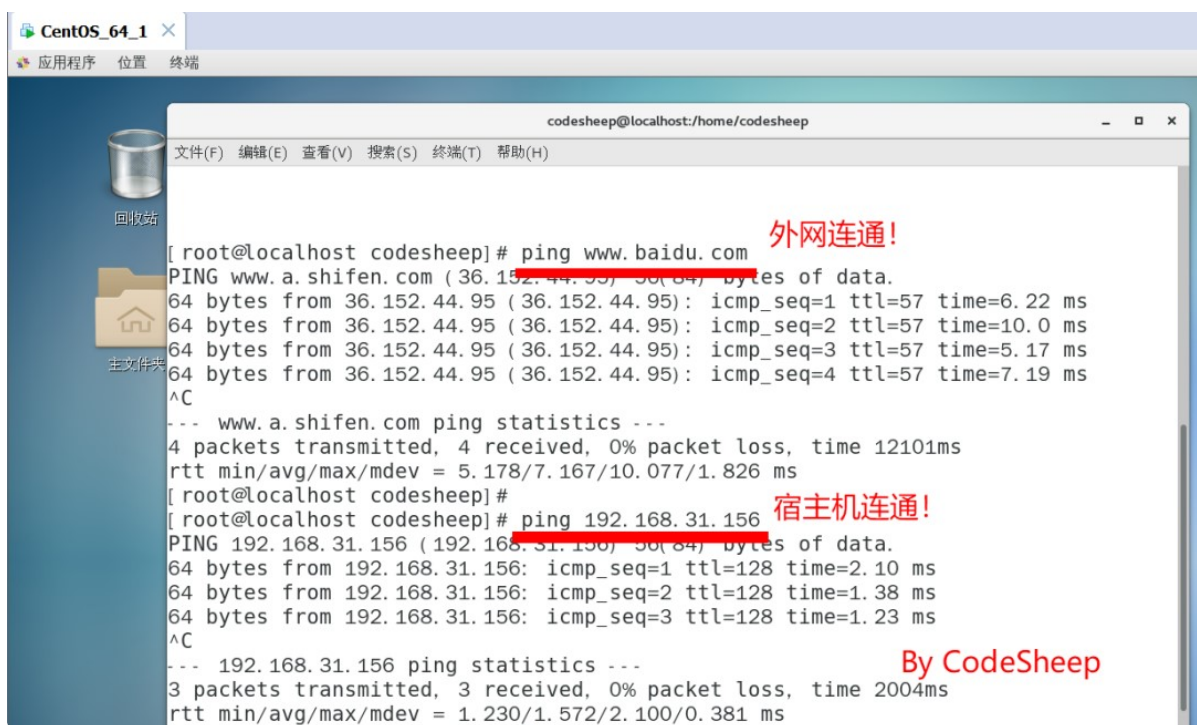
编辑完成，重启网络设置即可

```
systemctl restart network.service
```

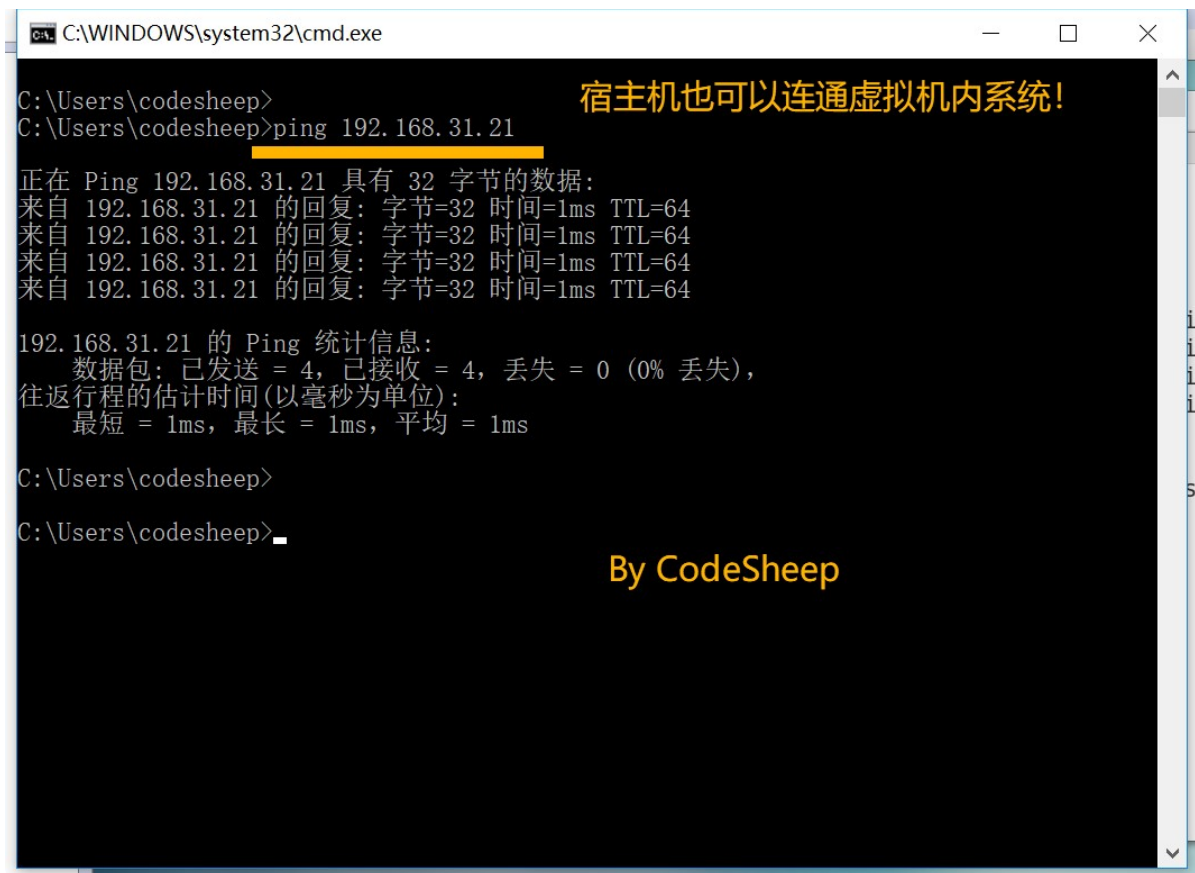
## 检查安装配置结果

### 1、检验虚拟机系统网络和外界的连通性

包括检查和外网的连通、和物理宿主机的连通、以及和兄弟节点（前提是你安装了多个虚拟机系统节点的话）之间的连接



## 2、反向检查物理主机和虚拟机系统网络的连接性



A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The prompt shows the user "C:\Users\codesheep" running the command "ping 192.168.31.21". The output shows four successful replies with 32 bytes of data, 1ms time, and TTL=64. The statistics show 4 packets sent, 4 received, 0% loss, and 1ms round-trip time. A yellow highlight is under the command. A yellow text overlay on the right says "宿主机也可以连通虚拟机内系统!". At the bottom right, it says "By CodeSheep".

```
C:\Users\codesheep>ping 192.168.31.21

正在 Ping 192.168.31.21 具有 32 字节的数据:
来自 192.168.31.21 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.31.21 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.31.21 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.31.21 的回复: 字节=32 时间=1ms TTL=64

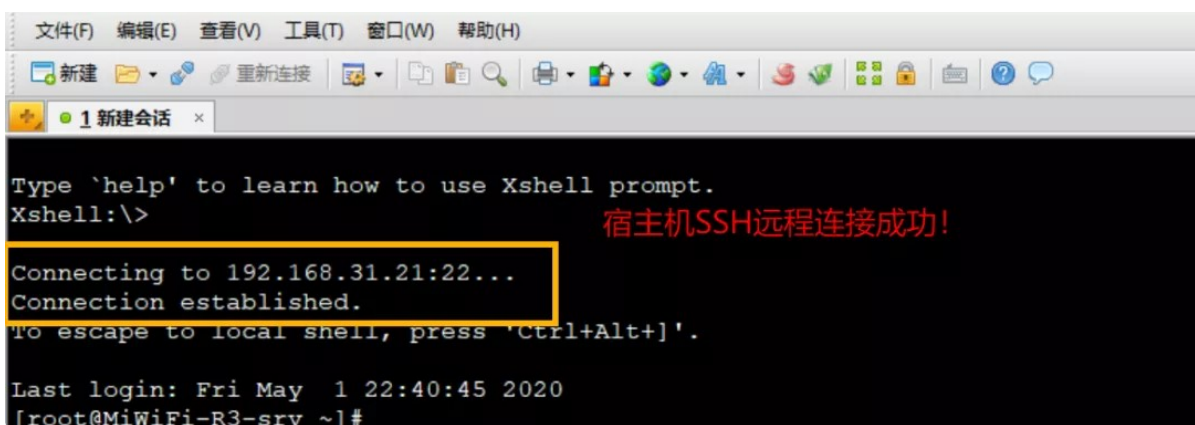
192.168.31.21 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 1ms, 最长 = 1ms, 平均 = 1ms

C:\Users\codesheep>
C:\Users\codesheep>
```

至此，大功告成！

## SSH远程连接

在宿主机通过SSH终端连接虚拟机内Linux系统，成功！



A screenshot of an Xshell terminal window. The title bar shows "文件(F) 编辑(E) 查看(V) 工具(T) 窗口(W) 帮助(H)". The menu bar has "新建", "重新连接", and "1 新建会话". The terminal shows the Xshell prompt, a help message, and the command "ssh root@192.168.31.21". The output shows "Connecting to 192.168.31.21:22...", "Connection established.", and "To escape to local shell, press 'Ctrl+Alt+J'.". The last login is "Fri May 1 22:40:45 2020". The prompt is "[root@MiWiFi-R3-srv ~]#". A yellow highlight is under the connection message. A red text overlay on the right says "宿主机SSH远程连接成功!".

```
Type 'help' to learn how to use Xshell prompt.
Xshell:\>
Connecting to 192.168.31.21:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

Last login: Fri May 1 22:40:45 2020
[root@MiWiFi-R3-srv ~]#
```



```

[root@MiWiFi-R3-srv ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.31.21 netmask 255.255.255.0 broadcast 192.168.31.255
    inet6 fe80::afb9:480f:1a57:b42e prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ab:35:74 txqueuelen 1000 (Ethernet)
    RX packets 5986 bytes 368157 (359.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 255 bytes 28400 (27.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

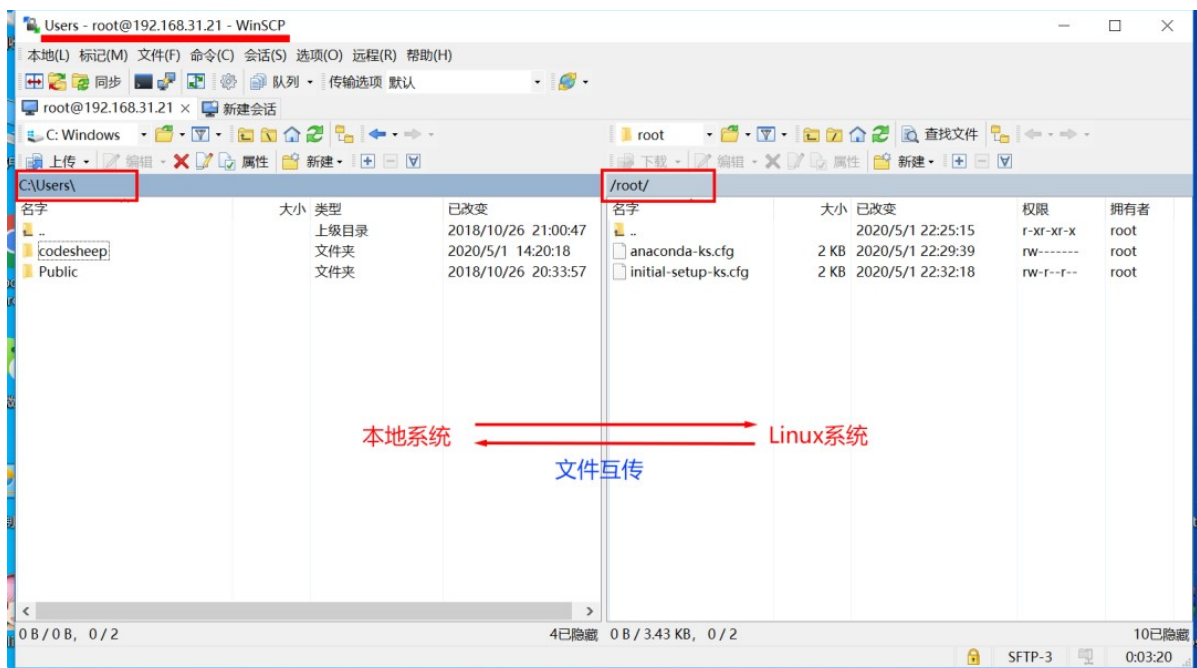
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1 (Local Loopback)
    RX packets 416 bytes 36032 (35.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 416 bytes 36032 (35.1 KiB)

```

By CodeSheep

## SFTP文件传输

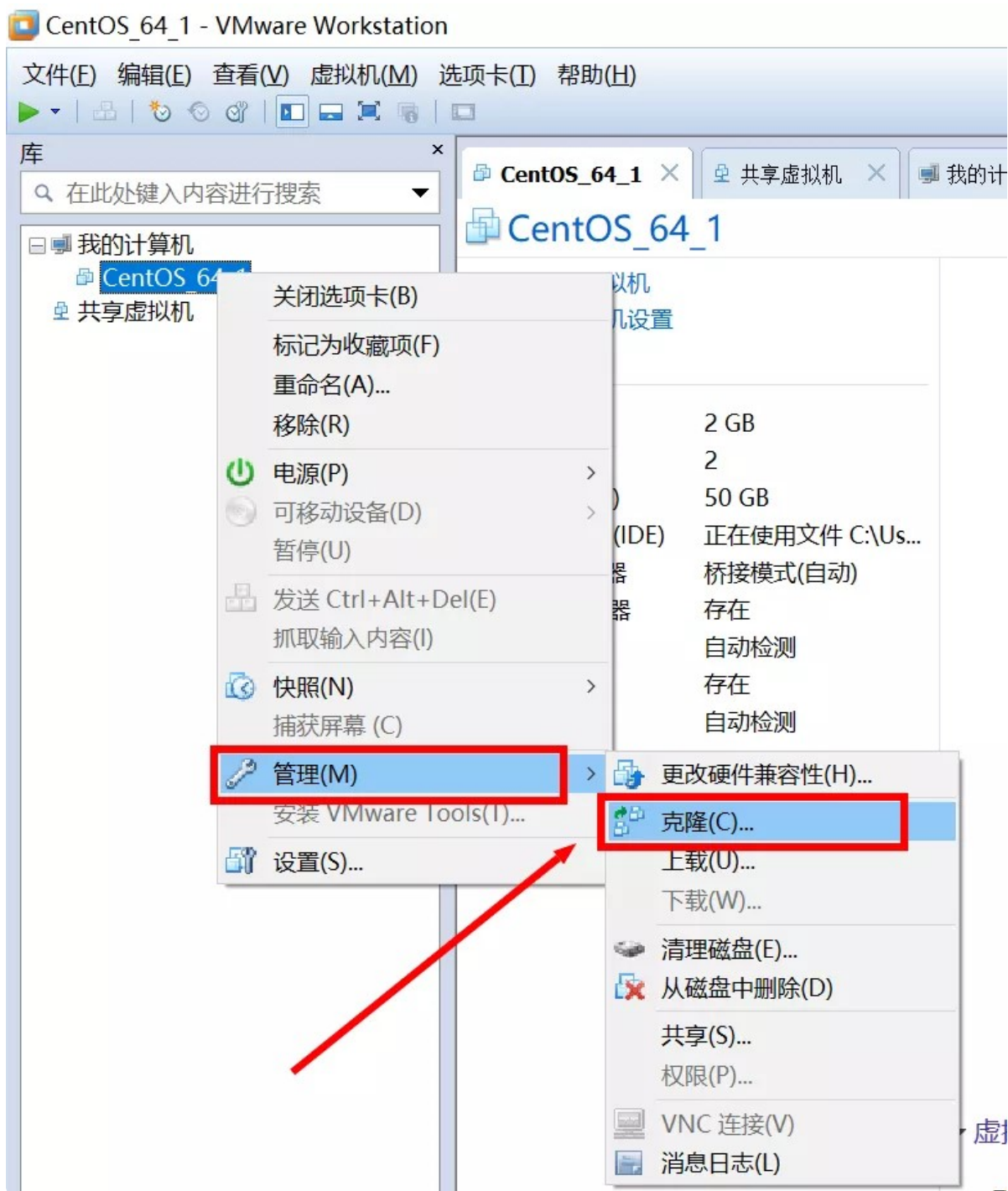
在宿主机通过SFTP工具即可访问虚拟机内Linux节点，从而可以实现本地和服务器的文件互传



这样一来，一个可用的Linux节点就打造完成了！

## 再多弄几个节点

可以完全重复以上步骤再打造出多个Linux节点，当然**更简单的方式**则是直接通过上面已经装好了的虚拟机节点**直接克隆**，来快速生成其他节点。



克隆完成之后，重新配置一下新节点的网络即可。

## 后记

---

好啦，现在**多节点的Linux环境**终于搭建完成了，后续不管是 **学Linux、用Linux**，还是**Linux环境编程、应用和项目部署、工具实验**，都有可以动手实践的地方了。

每天进步一点点，Peace！

2020.05.02 下午

五一假期快乐！

❤ 给个[在看]，是对程序羊最大的支持