集群规划

kubeadm安装集群, 实际考试环境也是用的这种方式。

三个节点: 最低2C2G不然无法安装。配置尽量给高点。

主机名	ip地址	硬件配置	安装组件
ek8s-master01	192.168.56.111	2C,4G, 10G硬盘	api-server, etcd, kubelet, kubectl, kubeadm
ek8s-node01	192.168.56.121	2C,4G, 10G硬盘	kubelet, kubeadm, containerd
ek8s-node02	192.168.56.122	2C,4G, 10G硬盘	kubelet, kubadm, containerd

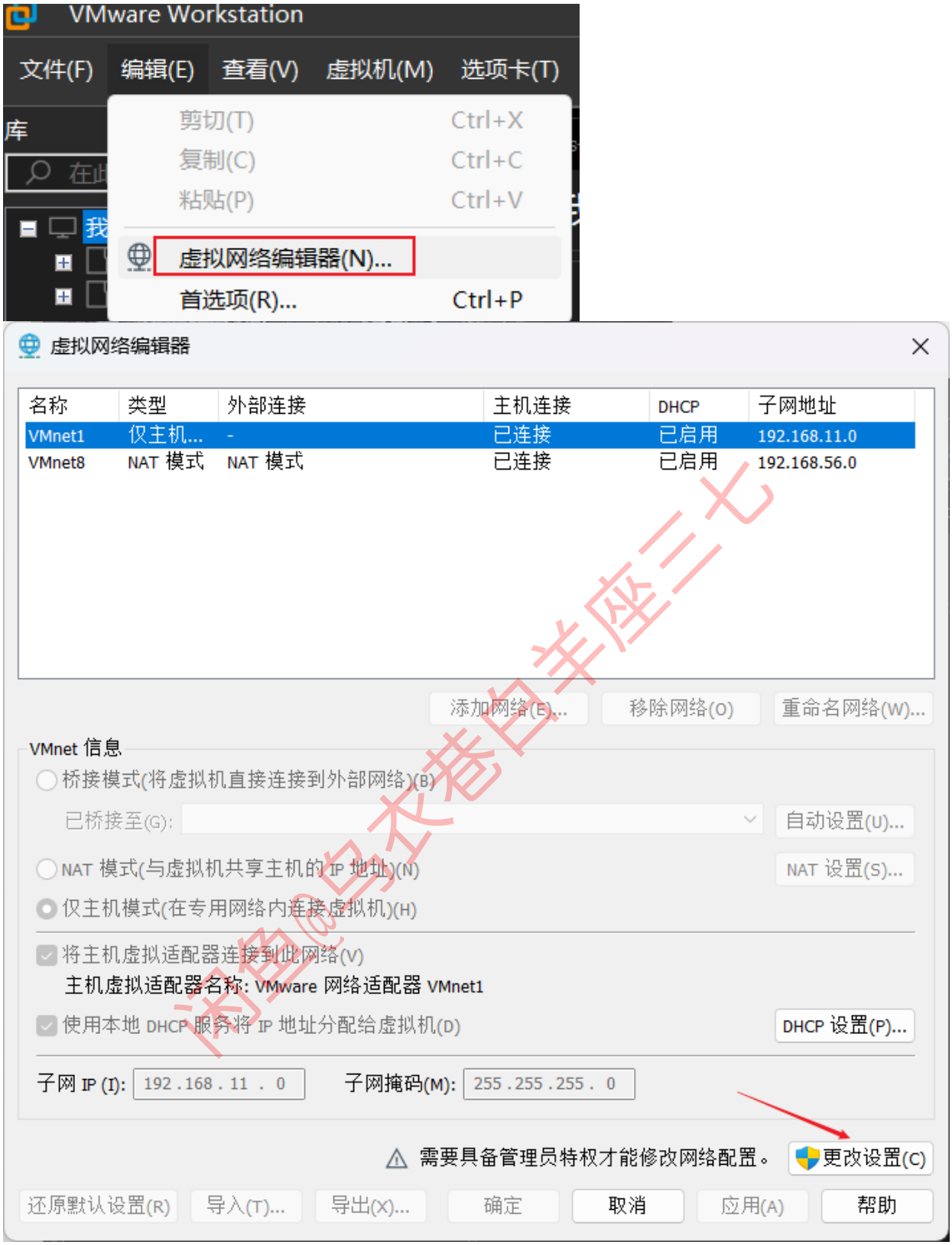
一、准备VMware workstation

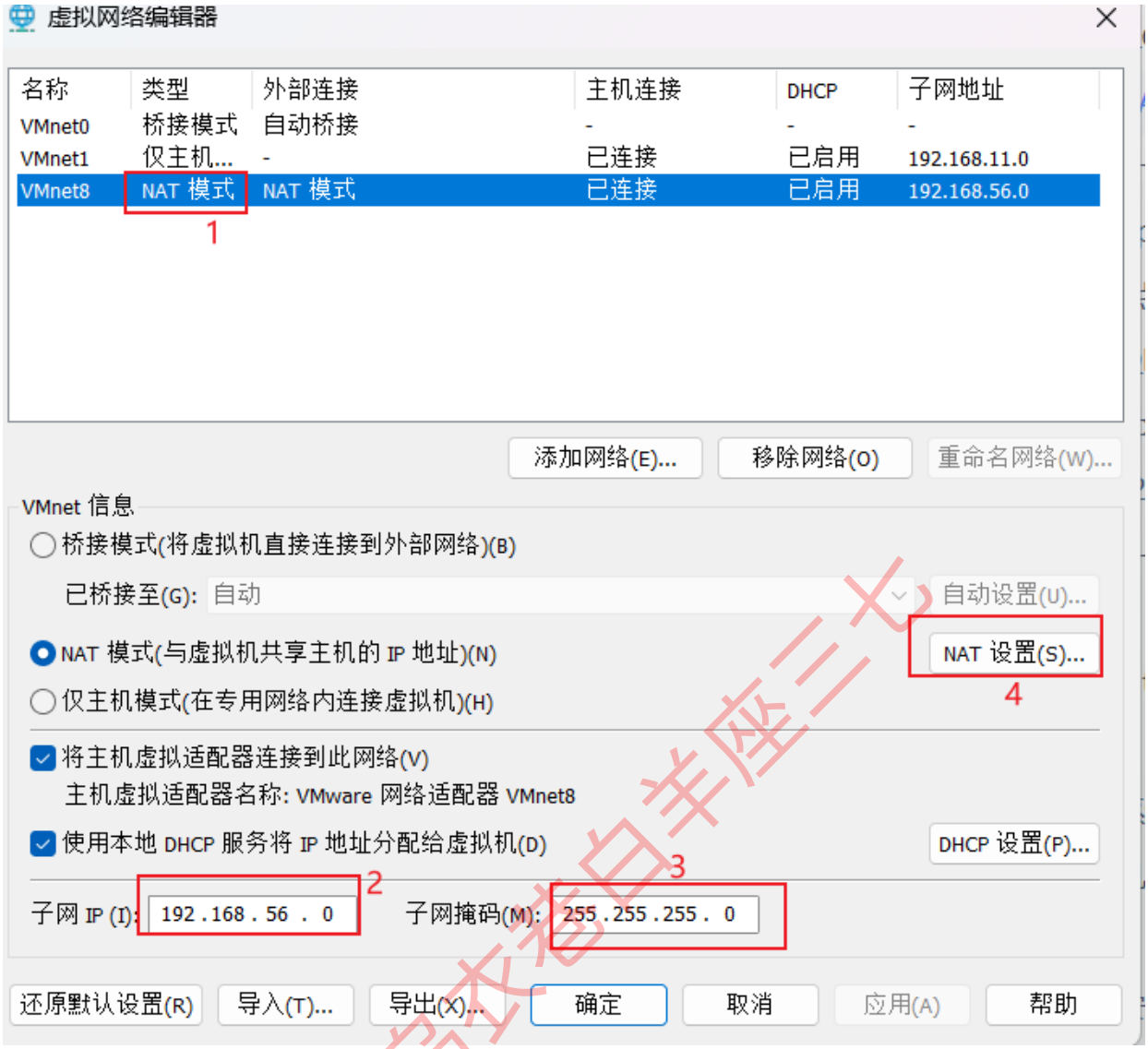
1.1 安装

推荐安装16版本，网盘中有安装包和序列号，点击下一步安装即可，这里不在赘述。

1.2 配置vmware workstation网络

配置网络, 需要与集群规划 部分保持一致。


VMware Workstation
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T)
库
剪切(T) Ctrl+X
复制(C) Ctrl+C
粘贴(P) Ctrl+V
虚拟网络编辑器(N)...
首选项(R)... Ctrl+P
虚拟网络编辑器
名称 类型 外部连接 主机连接 DHCP 子网地址
VMnet1 仅主机... - 已连接 已启用 192.168.11.0
VMnet8 NAT 模式 NAT 模式 已连接 已启用 192.168.56.0
添加网络(E)... 移除网络(O) 重命名网络(W)...
VMnet 信息
桥接模式(将虚拟机直接连接到外部网络)(B)
已桥接至(G): 自动设置(U)...
NAT 模式(与虚拟机共享主机的 IP 地址)(N) NAT 设置(S)...
仅主机模式(在专用网络内连接虚拟机)(H)
将主机虚拟适配器连接到此网络(V)
主机虚拟适配器名称: VMware 网络适配器 VMnet1
使用本地 DHCP 服务将 IP 地址分配给虚拟机(D) DHCP 设置(P)...
子网 IP (I): 192 . 168 . 11 . 0 子网掩码(M): 255 . 255 . 255 . 0
需要具备管理员特权才能修改网络配置。 更改设置(C)
还原默认设置(R) 导入(T)... 导出(X)... 确定 取消 应用(A) 帮助



然后点击确定。

以管理员打开cmd：

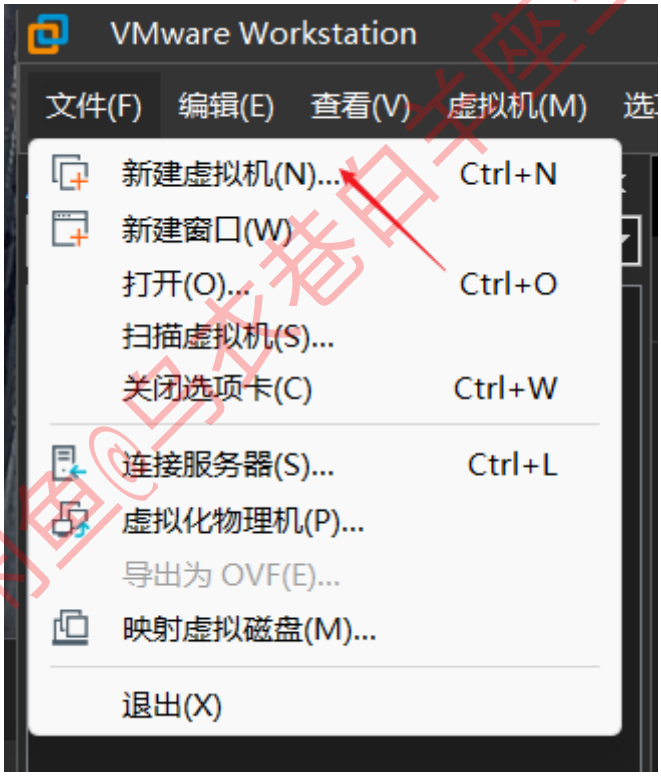
bcdedit /set hypervisorlaunchtype off

二、安装ubuntu20.0

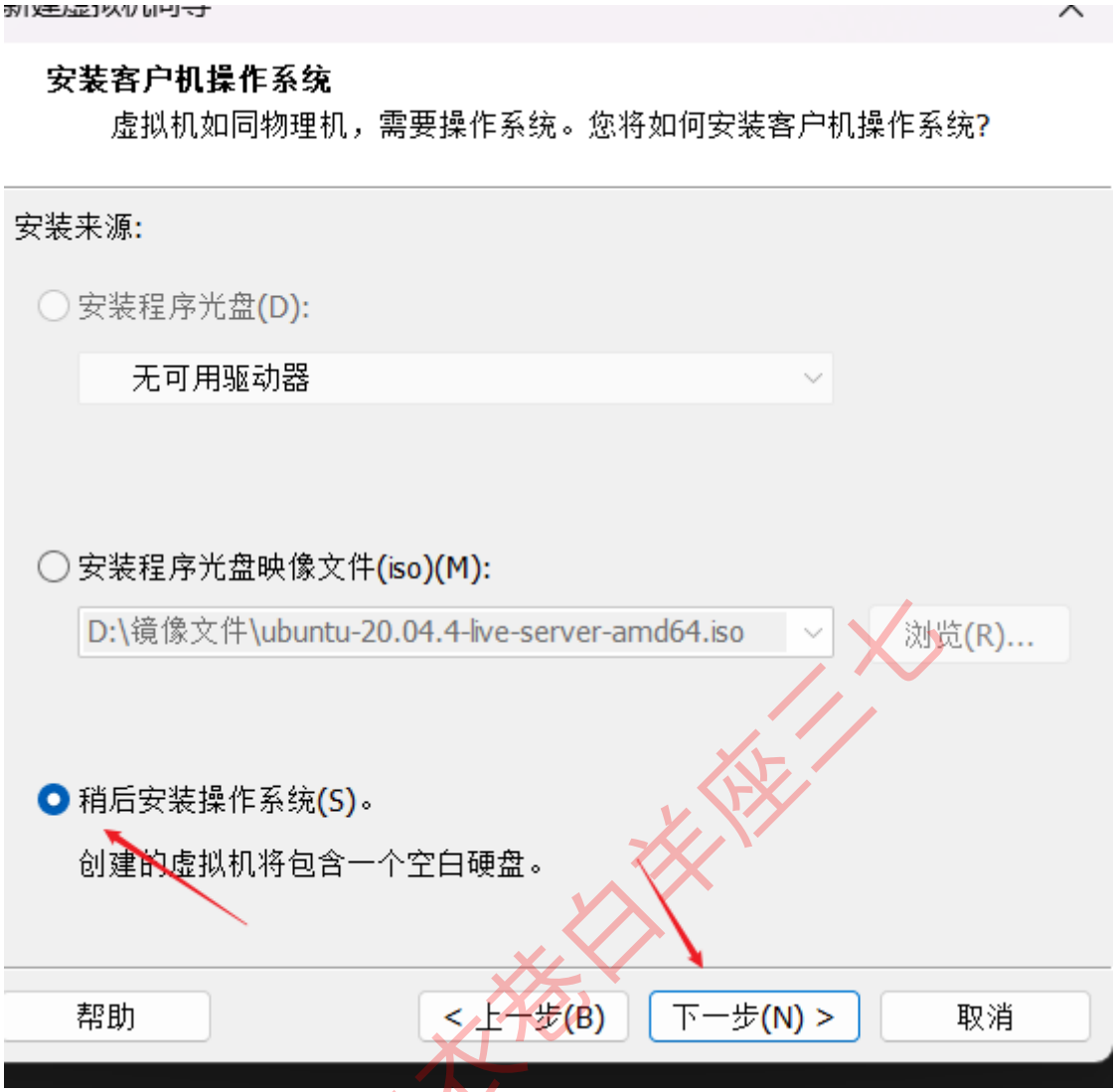
2.1 系统安装

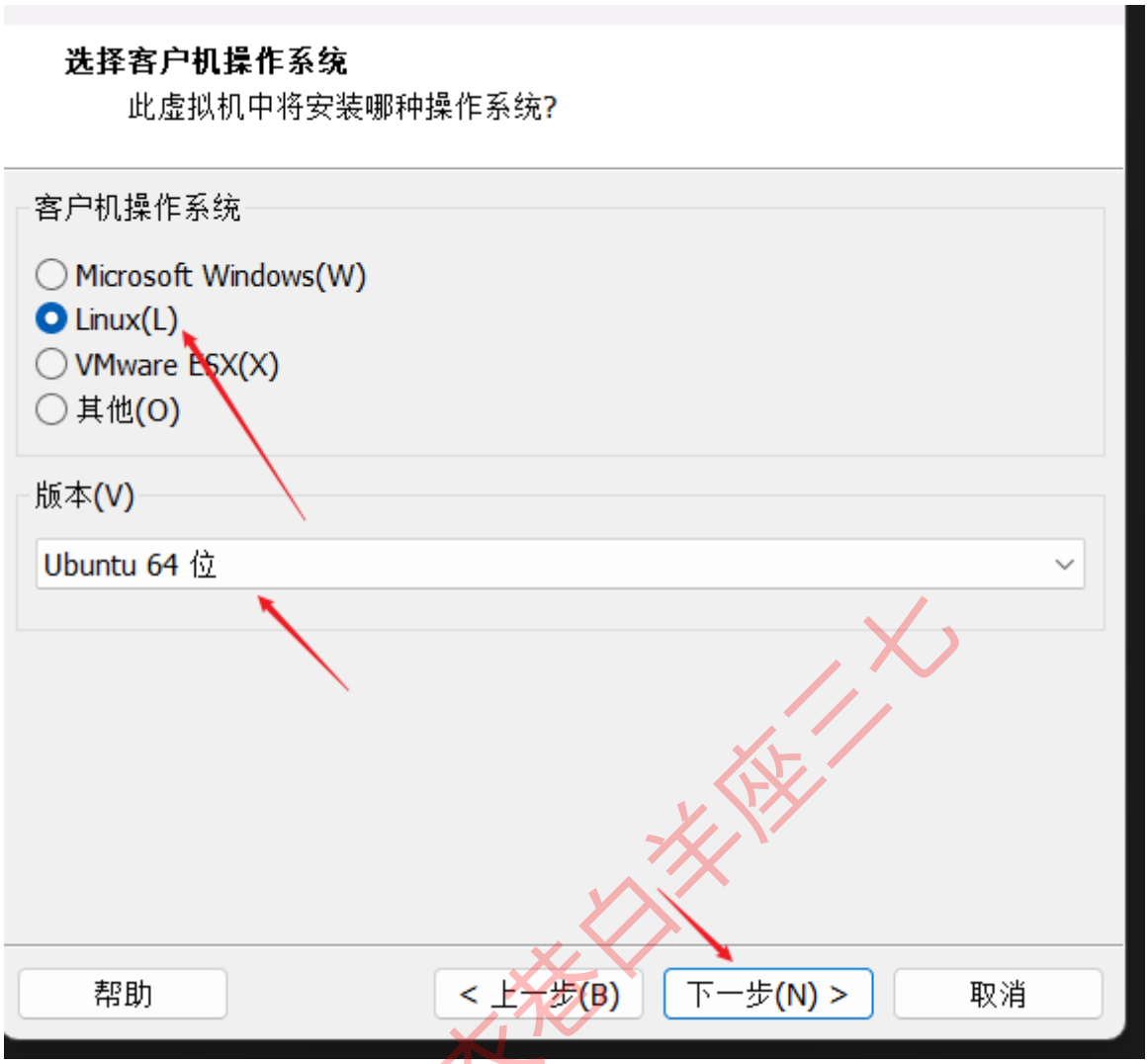
下载网盘中的安装镜像。

开始安装:

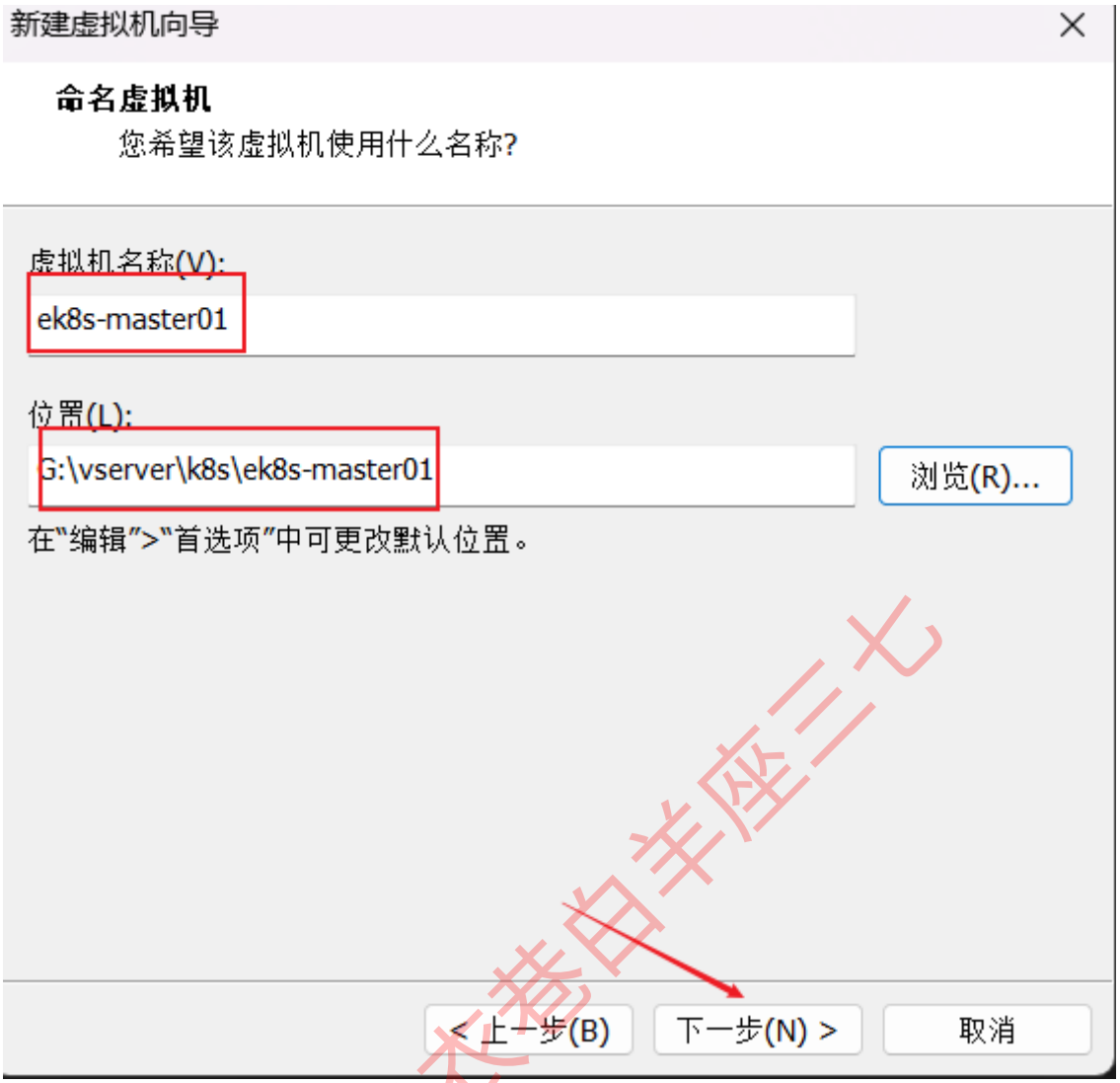



新建虚拟机向导
VMWARE
WORKSTATION
PRO™
16
欢迎使用新建虚拟机向导
您希望使用什么类型的配置?
典型(推荐)(T)
通过几个简单的步骤创建 Workstation 16.2.x 虚拟机。
自定义(高级)(C)
创建带有 SCSI 控制器类型、虚拟磁盘类型以及与旧版 VMware 产品兼容性等高级选项的虚拟机。
帮助
< 上一步(B)
下一步(N) >
取消

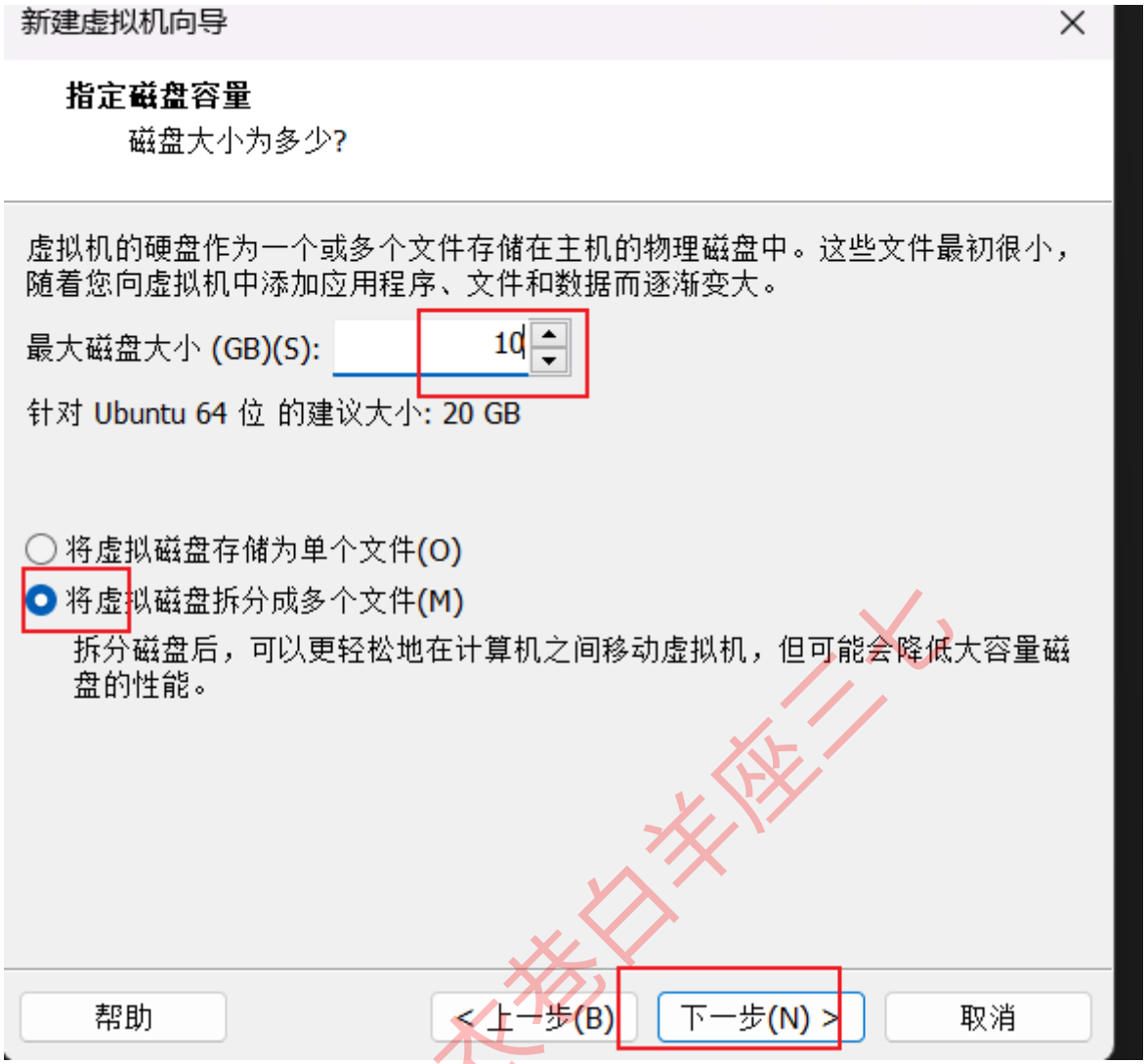

安装客户机操作系统
虚拟机如同物理机，需要操作系统。您将如何安装客户机操作系统?
安装来源:
安装程序光盘(D):
无可用驱动器
安装程序光盘映像文件(iso)(M):
D:\镜像文件\ubuntu-20.04.4-live-server-amd64.iso
浏览(R)...
稍后安装操作系统(S)。
创建的虚拟机将包含一个空白硬盘。
帮助
< 上一步(B)
下一步(N) >
取消

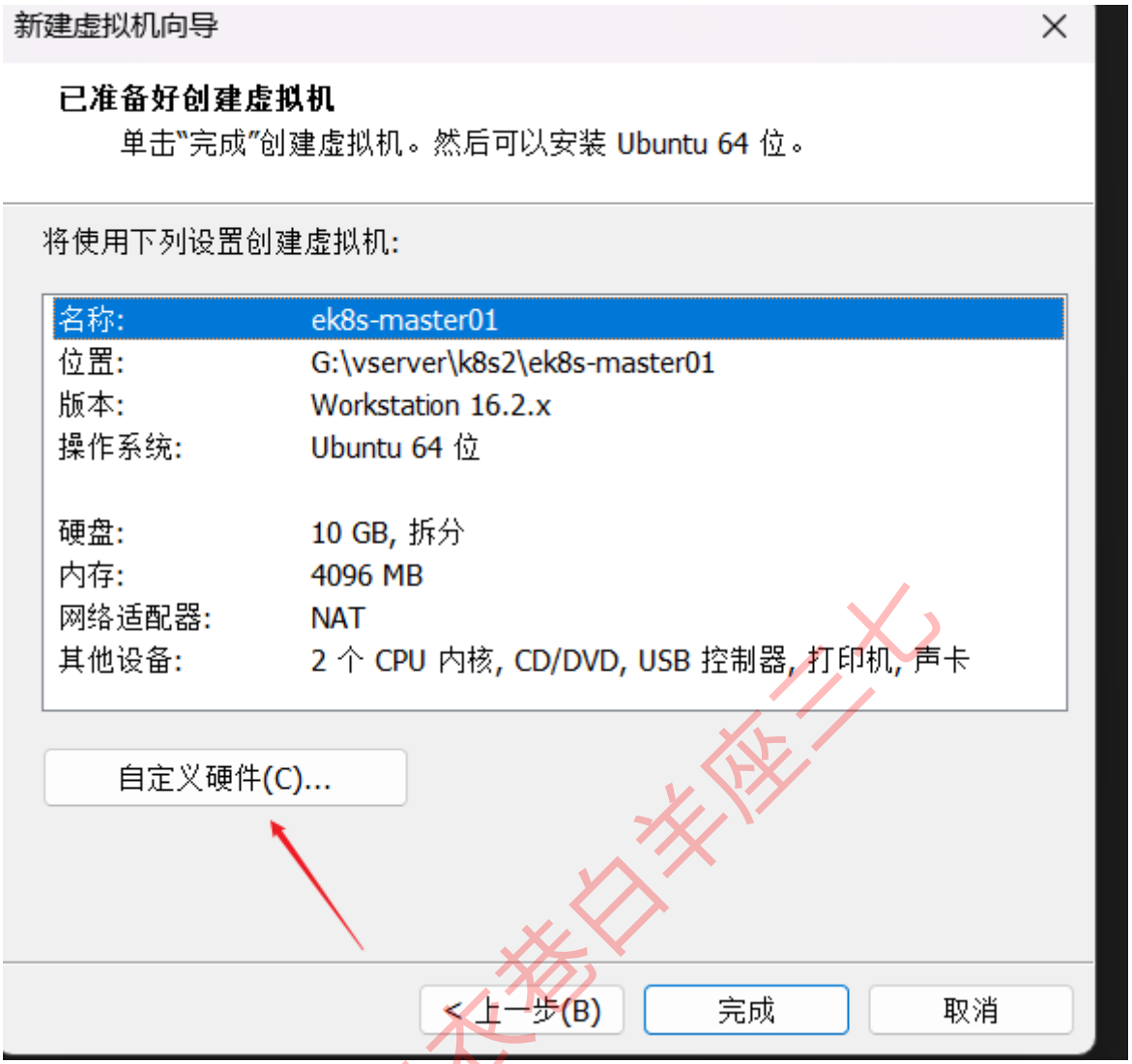


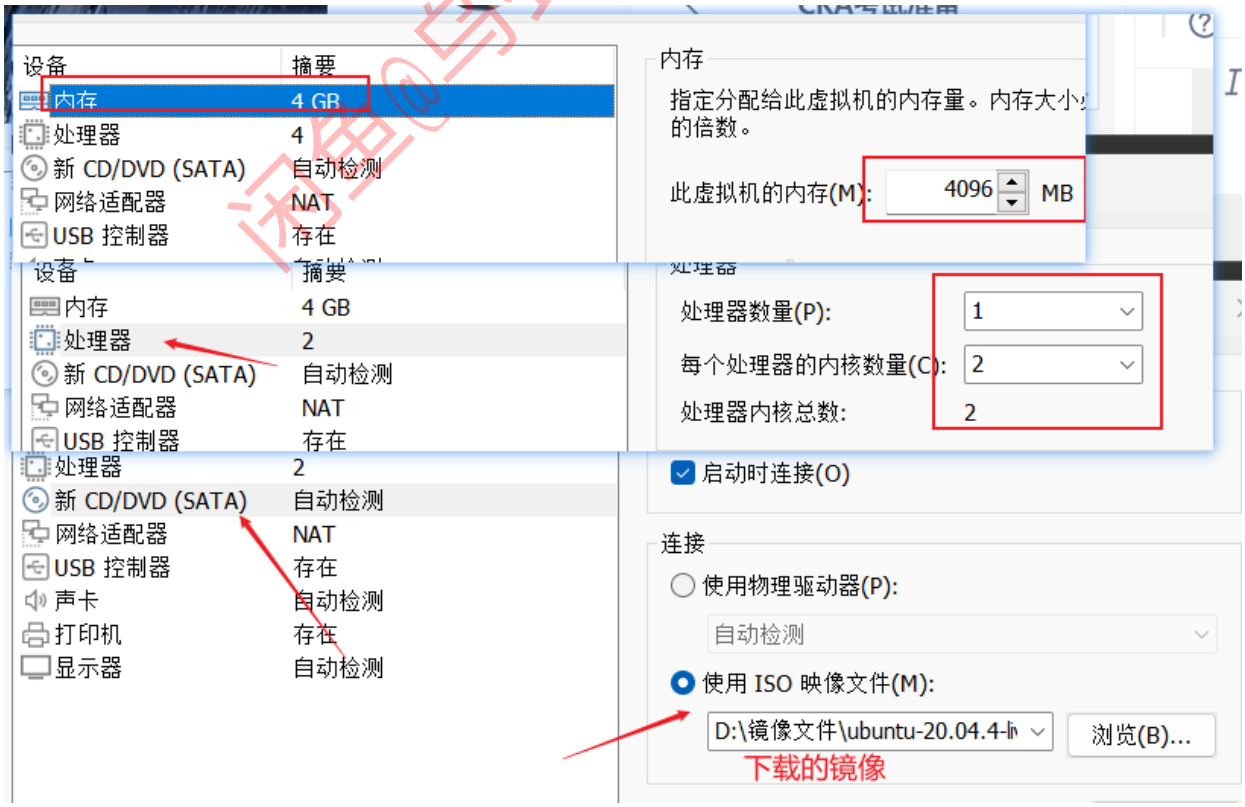
剩下两台替换下主机名和路径即可。

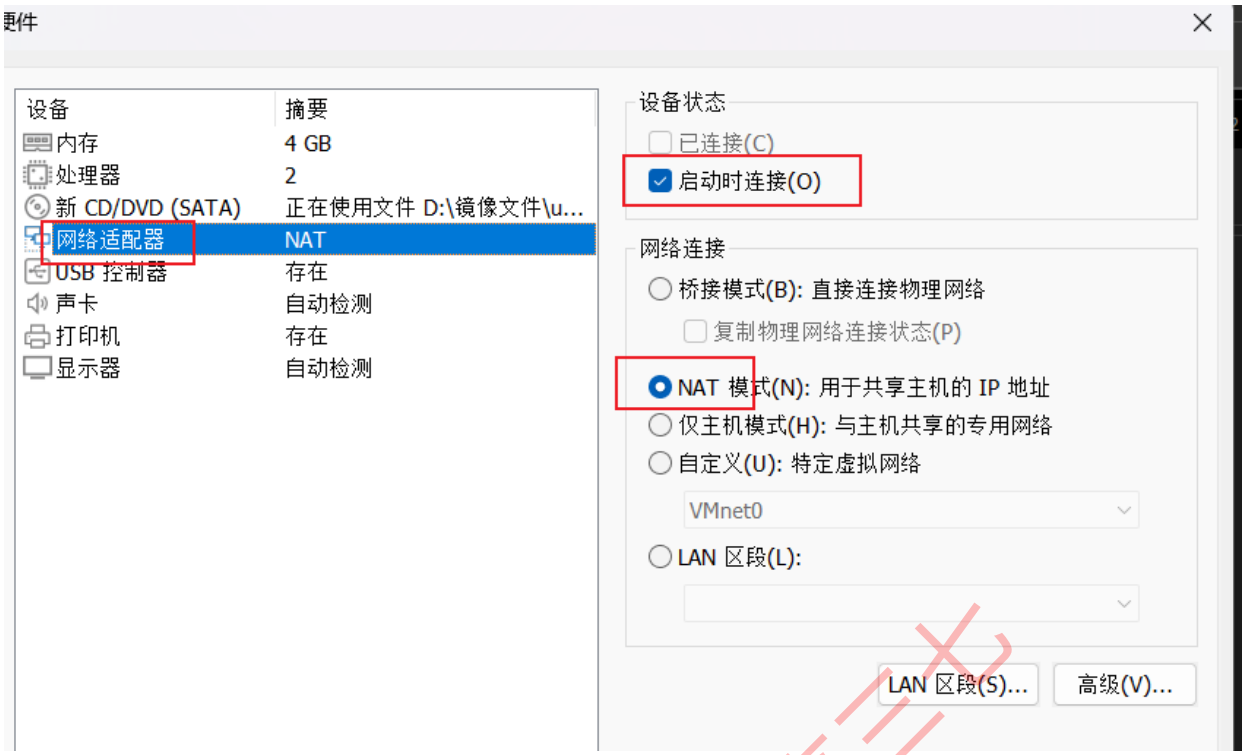


容量10G就够了

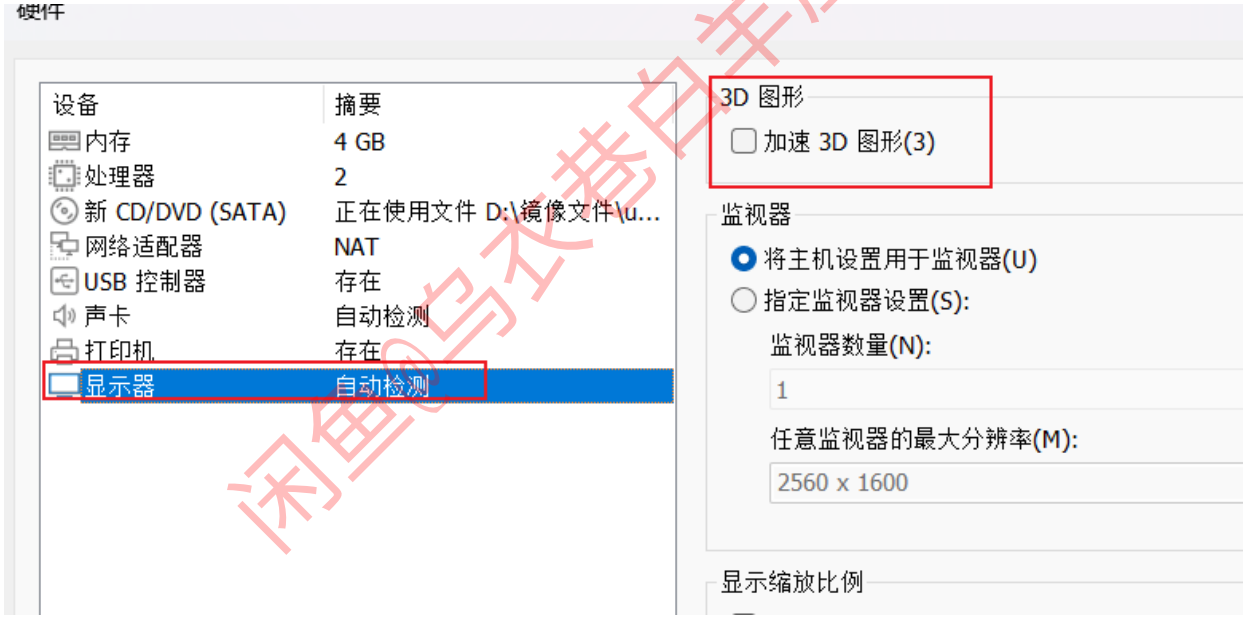

新建虚拟机向导
指定磁盘容量
磁盘大小为多少?
虚拟机的硬盘作为一个或多个文件存储在主机的物理磁盘中。这些文件最初很小，随着您向虚拟机中添加应用程序、文件和数据而逐渐变大。
最大磁盘大小 (GB)(S):
10
针对 Ubuntu 64 位 的建议大小: 20 GB
将虚拟磁盘存储为单个文件(O)
将虚拟磁盘拆分成多个文件(M)
拆分磁盘后，可以更轻松地在计算机之间移动虚拟机，但可能会降低大容量磁盘的性能。
帮助
< 上一步(B)
下一步(N) >
取消

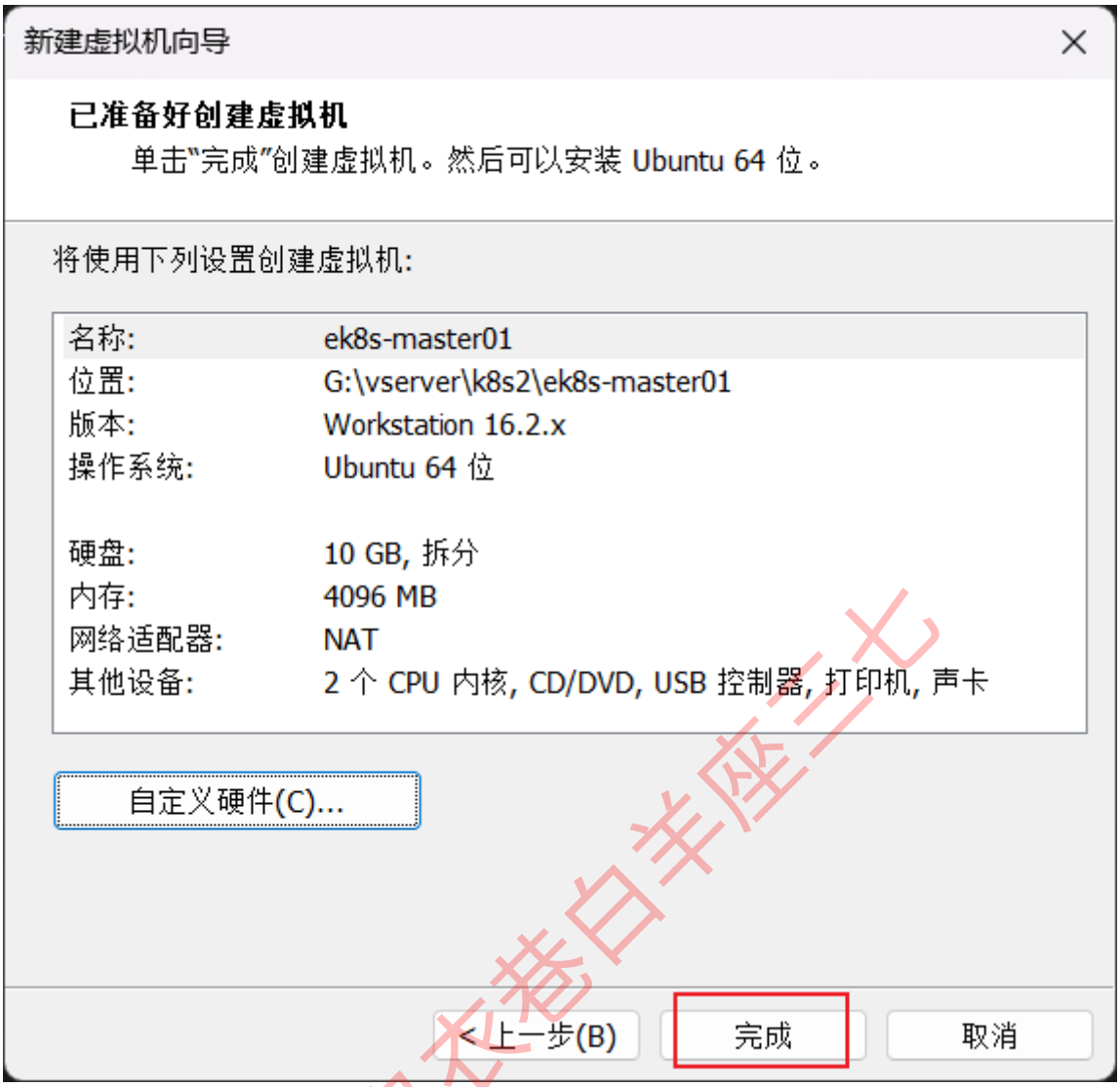

新建虚拟机向导
已准备好创建虚拟机
单击"完成"创建虚拟机。然后可以安装 Ubuntu 64 位。
将使用下列设置创建虚拟机:
名称: ek8s-master01
位置: G:\vserver\k8s2\ek8s-master01
版本: Workstation 16.2.x
操作系统: Ubuntu 64 位
硬盘: 10 GB, 拆分
内存: 4096 MB
网络适配器: NAT
其他设备: 2 个 CPU 内核, CD/DVD, USB 控制器, 打印机, 声卡
自定义硬件(C)...
< 上一步(B)
完成
取消

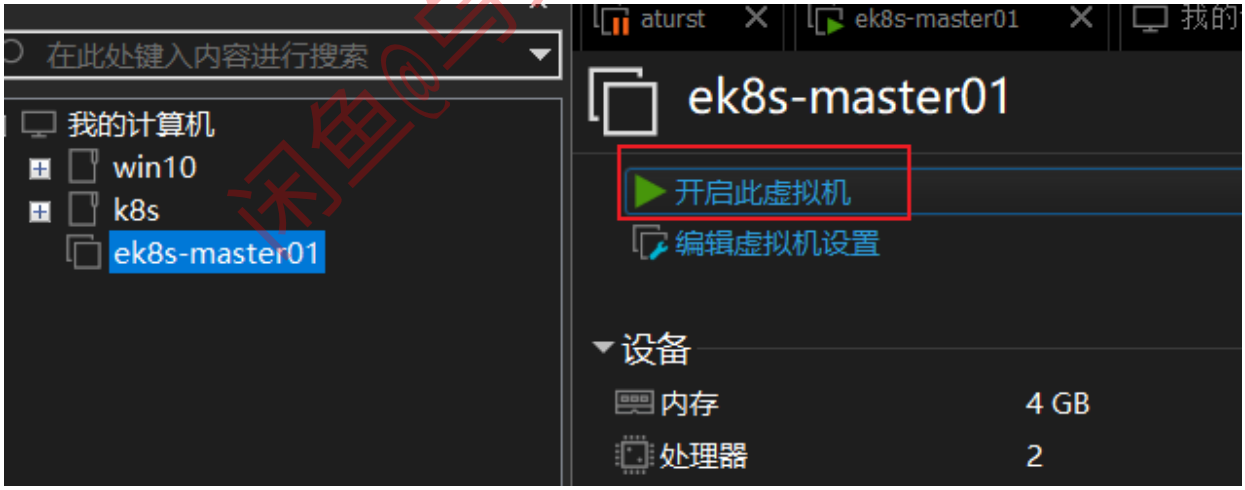

设备 摘要
内存 4 GB
处理器 4
新 CD/DVD (SATA) 自动检测
网络适配器 NAT
USB 控制器 存在
内存
指定分配给此虚拟机的内存量。内存大小
的倍数。
此虚拟机的内存(M): 4096 MB
设备 摘要
内存 4 GB
处理器 2
新 CD/DVD (SATA) 自动检测
网络适配器 NAT
USB 控制器 存在
处理器数量(P): 1
每个处理器的内核数量(C): 2
处理器内核总数: 2
处理器 2
新 CD/DVD (SATA) 自动检测
网络适配器 NAT
USB 控制器 存在
声卡 自动检测
打印机 存在
显示器 自动检测
启动时连接(O)
连接
使用物理驱动器(P):
自动检测
使用 ISO 映像文件(M):
D:\镜像文件\ubuntu-20.04.4-
浏览(B)...
下载的镜像



最重要的一点， 取消勾选， 不然虚拟机会一直崩溃。







进入安装界面,

Willkommen! Bienvenue! Welcome! Добро пожаловать! Welkom! [Help]

Use UP, DOWN and ENTER keys to select your language.

[Asturianu ▸]
[Bahasa Indonesia ▸]
[Català ▸]
[Deutsch ▸]
[English ▸]
[English (UK) ▸]
[Español ▸]
[Français ▸]
[Galego ▸]
[Hrvatski ▸]
[Latviski ▸]
[Lietuviškai ▸]
[Magyar ▸]
[Nederlands ▸]
[Norsk bokmål ▸]
[Polski ▸]
[Português ▸]
[Suomi ▸]
[Svenska ▸]
[Čeština ▸]
[Ελληνικά ▸]
[Беларуская ▸]
[Русский ▸]
[Српски ▸]
[Українська ▸]

Installer update available [Help]

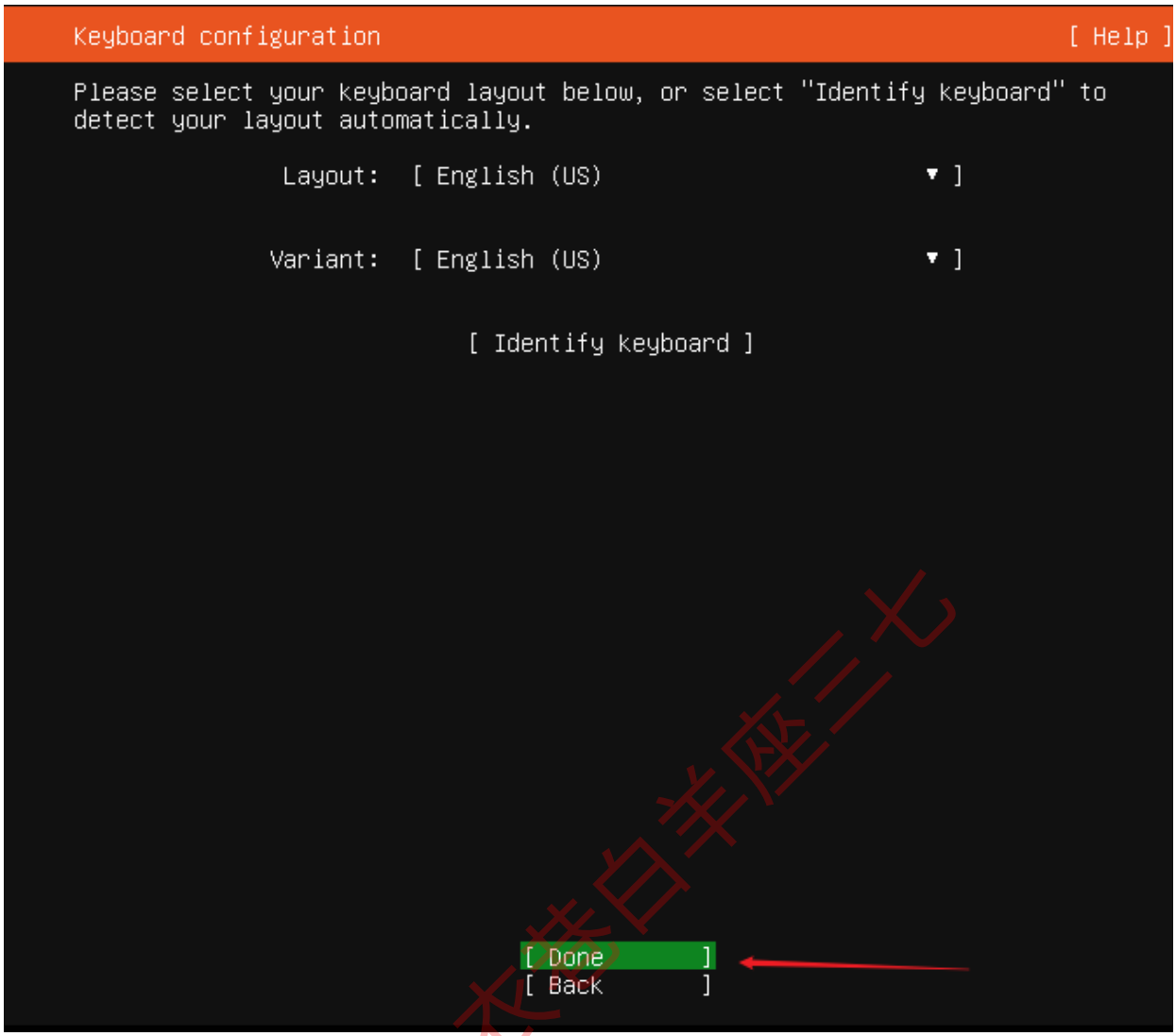
Version 23.02.1 of the installer is now available (22.02.2 is currently running).

You can read the release notes for each version at:

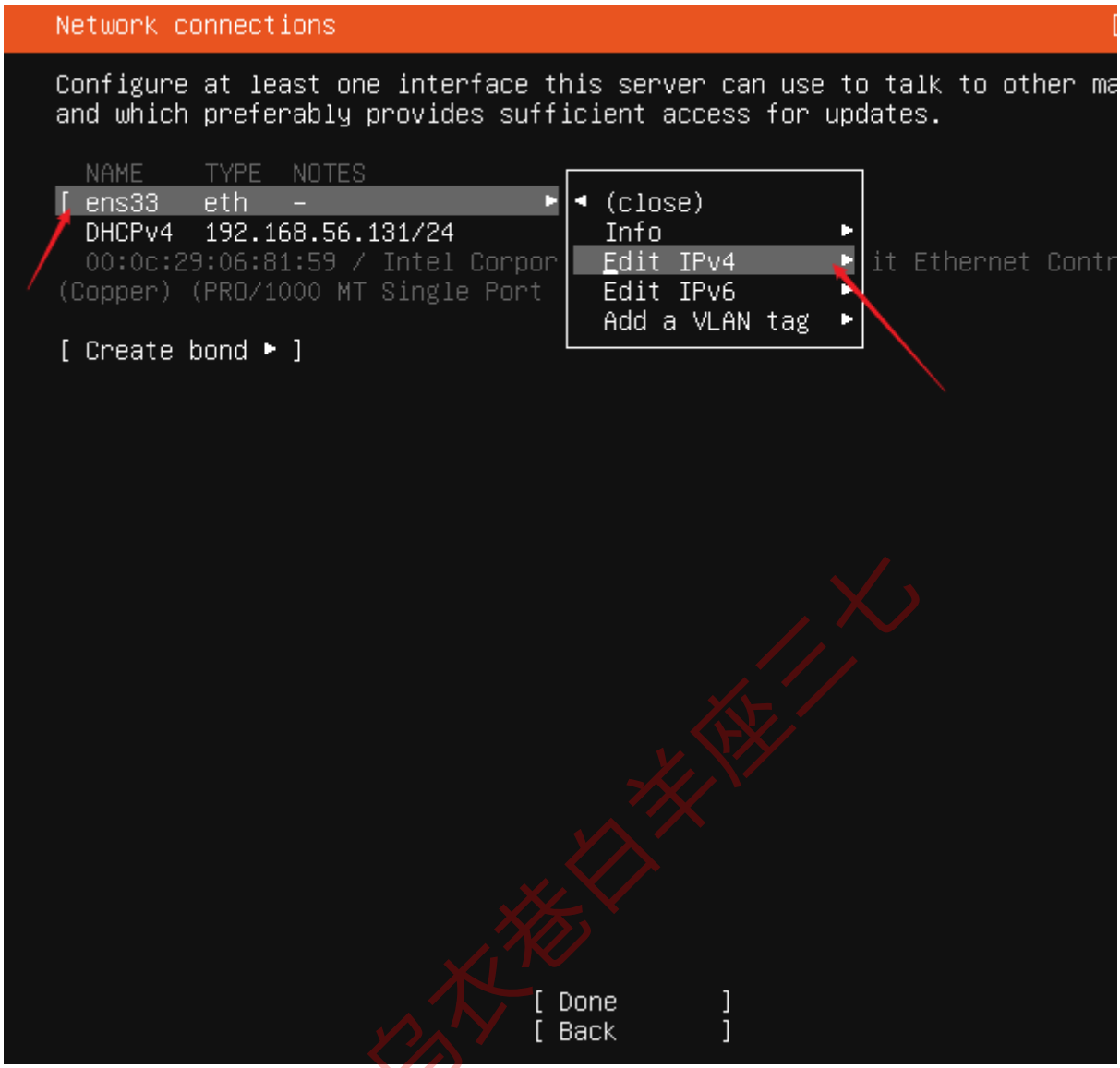
https://github.com/canonical/subiquity/releases

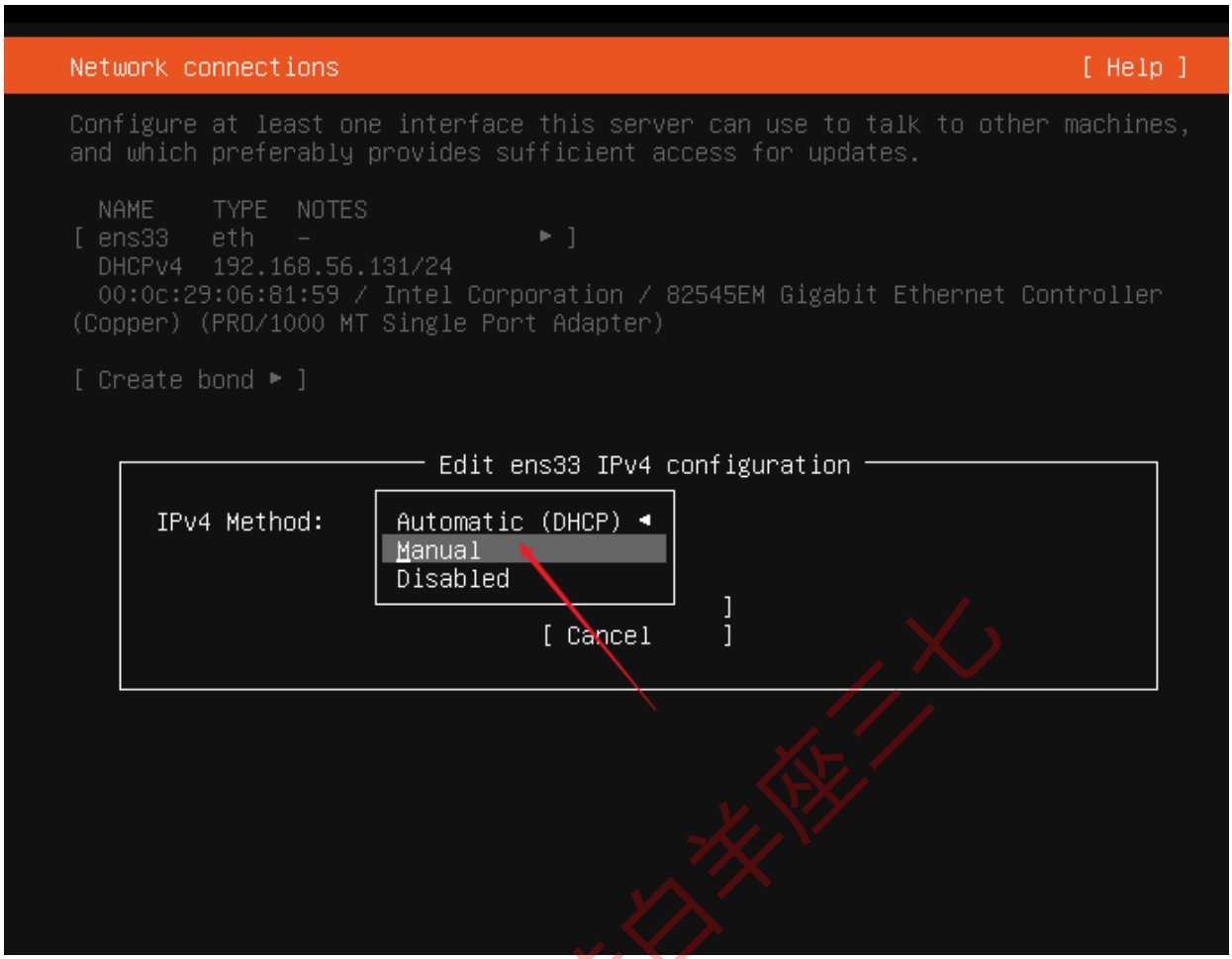
If you choose to update, the update will be downloaded and the installation will continue from here.

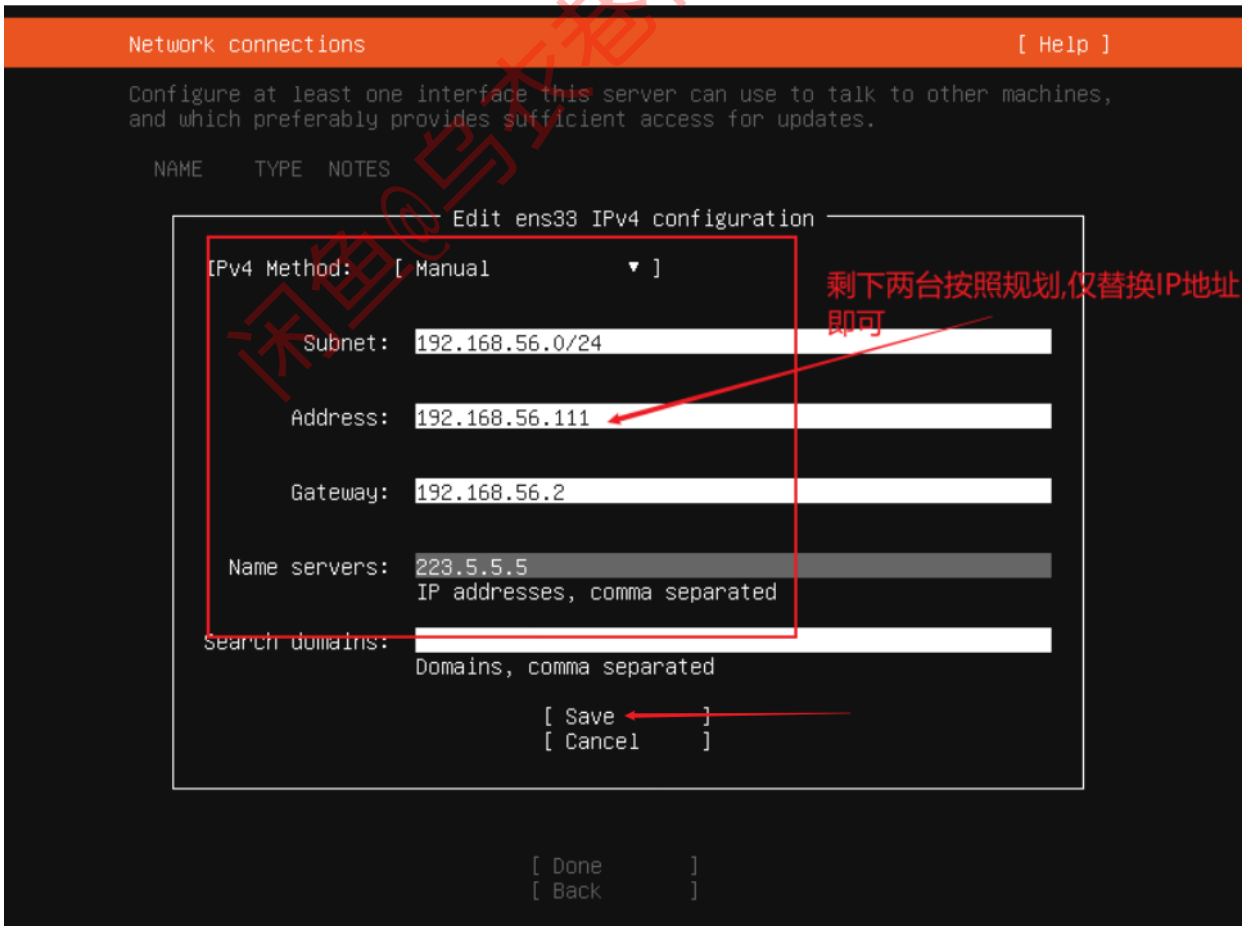
[Update to the new installer]
[Continue without updating]
[Back]



按方向键选中网卡, 然后按回车确认


Network connections
Configure at least one interface this server can use to talk to other ma
and which preferably provides sufficient access for updates.
NAME TYPE NOTES
[ens33 eth -
DHCPv4 192.168.56.131/24
00:0c:29:06:81:59 / Intel Corpor
(Copper) (PRO/1000 MT Single Port
[Create bond ▸]
(close)
Info
Edit IPv4
Edit IPv6
Add a VLAN tag
it Ethernet Contr
[Done]
[Back]


Network connections
[Help]
Configure at least one interface this server can use to talk to other machines, and which preferably provides sufficient access for updates.
NAME TYPE NOTES
[ens33 eth - ▸]
DHCPv4 192.168.56.131/24
00:0c:29:06:81:59 / Intel Corporation / 82545EM Gigabit Ethernet Controller (Copper) (PRO/1000 MT Single Port Adapter)
[Create bond ▸]
Edit ens33 IPv4 configuration
IPv4 Method:
Automatic (DHCP) ◂
Manual
Disabled
[Cancel]


Network connections
[Help]
Configure at least one interface this server can use to talk to other machines, and which preferably provides sufficient access for updates.
NAME TYPE NOTES
Edit ens33 IPv4 configuration
IPv4 Method: [Manual ▾]
Subnet: 192.168.56.0/24
Address: 192.168.56.111
Gateway: 192.168.56.2
Name servers: 223.5.5.5
IP addresses, comma separated
Search domains:
Domains, comma separated
[Save]
[Cancel]
剩下两台按照规划,仅替换IP地址即可
[Done]
[Back]

```
Network connections                                                [ Help ]

Configure at least one interface this server can use to talk to other machines,
and which preferably provides sufficient access for updates.

  NAME    TYPE  NOTES
[ ens33   eth   -                  ► ]
  static  192.168.56.111/24
  00:0c:29:06:81:59 / Intel Corporation / 82545EM Gigabit Ethernet Controller
(Copper) (PRO/1000 MT Single Port Adapter)

[ Create bond ► ]

                                  [ Done       ]
                                  [ Back       ]
```

```
Configure proxy                                                    [ Help ]

If this system requires a proxy to connect to the internet, enter its details
here.

Proxy address:
                If you need to use a HTTP proxy to access the outside world,
                enter the proxy information here. Otherwise, leave this blank.

                The proxy information should be given in the standard form of
                "http://[[user][:pass]@]host[:port]/".

                                  [ Done       ]
                                  [ Back       ]
```

```
Configure Ubuntu archive mirror                                  [ Help ]

If you use an alternative mirror for Ubuntu, enter its details here.

Mirror address:  http://cn.archive.ubuntu.com/ubuntu
                 You may provide an archive mirror that will be used instead of
                 the default.

                                  [ Done       ]
                                  [ Back       ]
```

```
Guided storage configuration                                     [ Help ]

Configure a guided storage layout, or create a custom one:

(X)  Use an entire disk

     [ /dev/sda local disk 10.000G ▾ ]

     [X]  Set up this disk as an LVM group

          [ ]  Encrypt the LVM group with LUKS

                    Passphrase:

                 Confirm passphrase:

( )  Custom storage layout

                                  [ Done       ]
                                  [ Back       ]
```

```
Storage configuration                                                    [ Help ]

FILE SYSTEM SUMMARY

  MOUNT POINT     SIZE     TYPE      DEVICE TYPE
[ /               9.109G   new ext4  new LVM logical volume      ▸ ]
[ /boot         907.000M   new ext4  new partition of local disk ▸ ]


AVAILABLE DEVICES

  No available devices

[ Create software RAID (md) ▸ ]
[ Create volume group (LVM) ▸ ]


USED DEVICES

  DEVICE                                      TYPE                 SIZE
[ ubuntu-vg (new)                             LVM volume group     9.109G  ▸ ]
  ubuntu-lv    new, to be formatted as ext4, mounted at /          9.109G  ▸

[ /dev/sda                                    local disk          10.000G  ▸ ]
  partition 1  new, BIOS grub spacer                                1.000M  ▸
  partition 2  new, to be formatted as ext4, mounted at /boot     907.000M  ▸
  partition 3  new, PV of LVM volume group ubuntu-vg                9.111G  ▸


                                  [ Done       ]
                                  [ Reset      ]
                                  [ Back       ]
```

```
Storage configuration                                                    [ Help ]

FILE SYSTEM SUMMARY

  MOUNT POINT     SIZE     TYPE      DEVICE TYPE
[ /               9.109G   new ext4  new LVM logical volume      ▸ ]
[ /boot         907.000M   new ext4  new partition of local disk ▸ ]


AVAILABLE DEVICES

                    Confirm destructive action

      Selecting Continue below will begin the installation process and
      result in the loss of data on the disks selected to be formatted.

      You will not be able to return to this or a previous screen once the
      installation has started.

      Are you sure you want to continue?

                                  [ No         ]
                                  [ Continue   ]


                                  [ Done       ]
                                  [ Reset      ]
                                  [ Back       ]
```

Profile setup [Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name: cc CC

Your server's name: ek8s-master01 剩下两头,在这里替换下主机名
The name it uses when it talks to other computers.

Pick a username: cc CC

Choose a password: ****** 123456

Confirm your password: ****** 123456

[Done]

Enable Ubuntu Advantage [Help]

Enter your Ubuntu Advantage token if you want to enroll this system.

Ubuntu Advantage token:

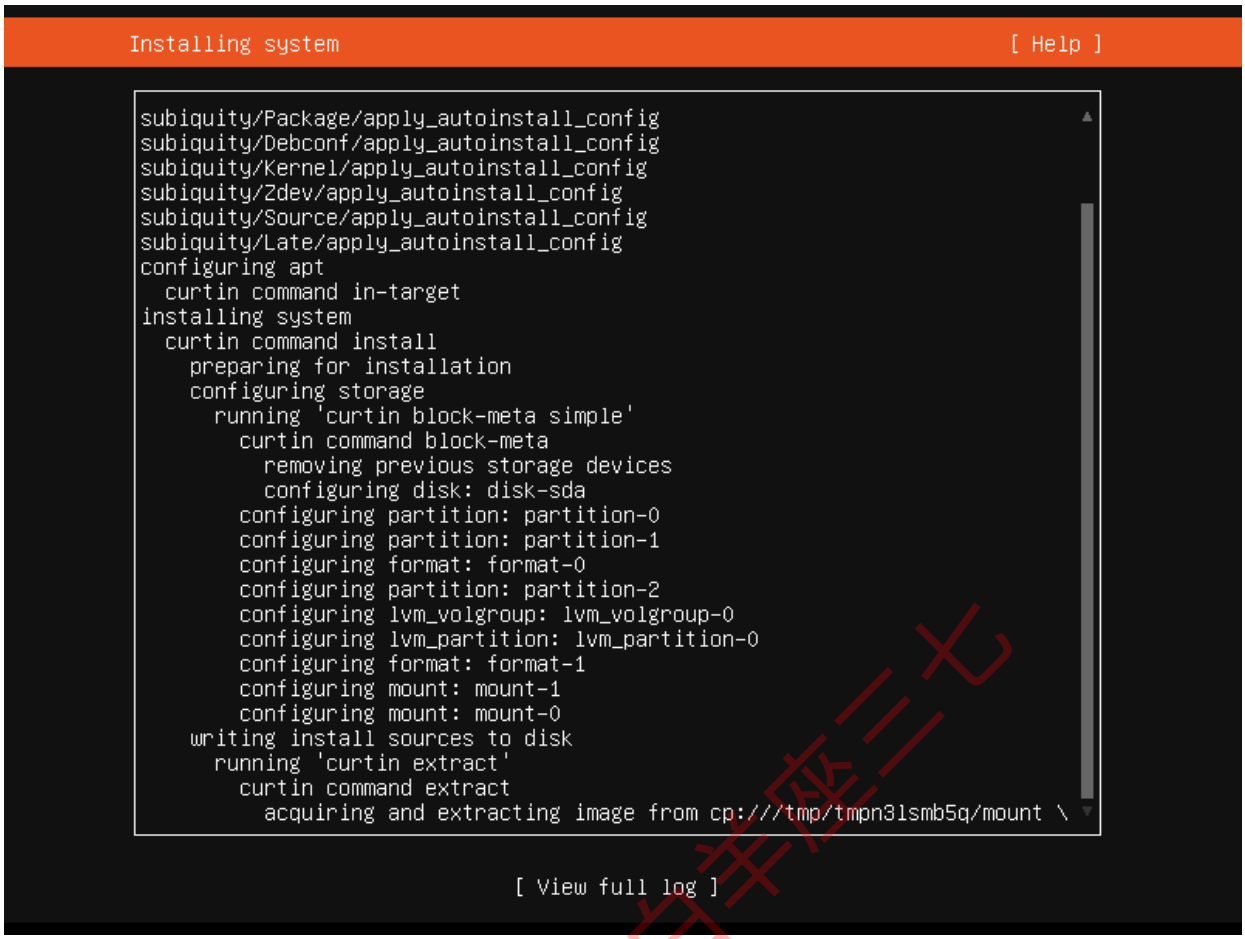
If you want to enroll this system using your Ubuntu Advantage subscription, enter your Ubuntu Advantage token here. Otherwise, leave this blank.

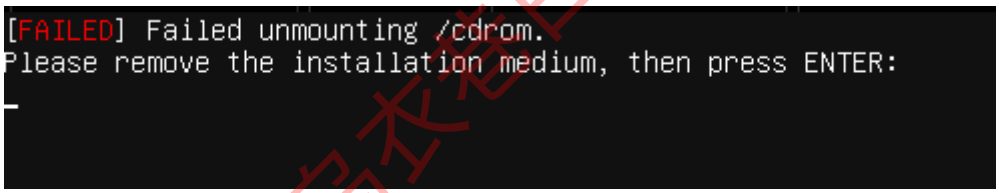
[Done]
[Back]

```
SSH Setup                                                    [ Help ]

You can choose to install the OpenSSH server package to enable secure remote
access to your server.

                [X]  Install OpenSSH server   按空格键勾选

Import SSH identity:  [ No              ▼ ]
                      You can import your SSH keys from GitHub or Launchpad.

    Import Username:

                  [X]  Allow password authentication over SSH

                                  [ Done       ]
                                  [ Back       ]
```

```
Featured Server Snaps                                        [ Help ]

These are popular snaps in server environments. Select or deselect with SPACE,
press ENTER to see more details of the package, publisher and versions
available.

[ ] microk8s            Kubernetes for workstations and appliances      ►
[ ] nextcloud           Nextcloud Server - A safe home for all your data ►
[ ] wekan               The open-source kanban                          ►
[ ] kata-containers     Build lightweight VMs that seamlessly plug into the c ►
[ ] docker              Docker container runtime                        ►
[ ] canonical-livepatch Canonical Livepatch Client                      ►
[ ] rocketchat-server   Rocket.Chat server                              ►
[ ] mosquitto           Eclipse Mosquitto MQTT broker                   ►
[ ] etcd                Resilient key-value store by CoreOS             ►
[ ] powershell          PowerShell for every system!                    ►
[ ] stress-ng           tool to load and stress a computer              ►
[ ] sabnzbd             SABnzbd                                         ►
[ ] wormhole            get things from one computer to another, safely ►
[ ] aws-cli             Universal Command Line Interface for Amazon Web Servi ►
[ ] google-cloud-sdk    Google Cloud SDK                                ►
[ ] slcli               Python based SoftLayer API Tool.                ►
[ ] doctl               The official DigitalOcean command line interface ►
[ ] conjure-up          Package runtime for conjure-up spells           ►
[ ] postgresql10        PostgreSQL is a powerful, open source object-relation ►
[ ] heroku              CLI client for Heroku                           ►
[ ] keepalived          High availability VRRP/BFD and load-balancing for Lin ►
[ ] prometheus          The Prometheus monitoring system and time series data ►
[ ] juju                Juju - a model-driven operator lifecycle manager for ►

                                  [ Done       ]
                                  [ Back       ]
```


至此等待安装完成即可，点击reboot接可。





重启时可能会有这个提示, 将鼠标点击去, 按下回车即可。

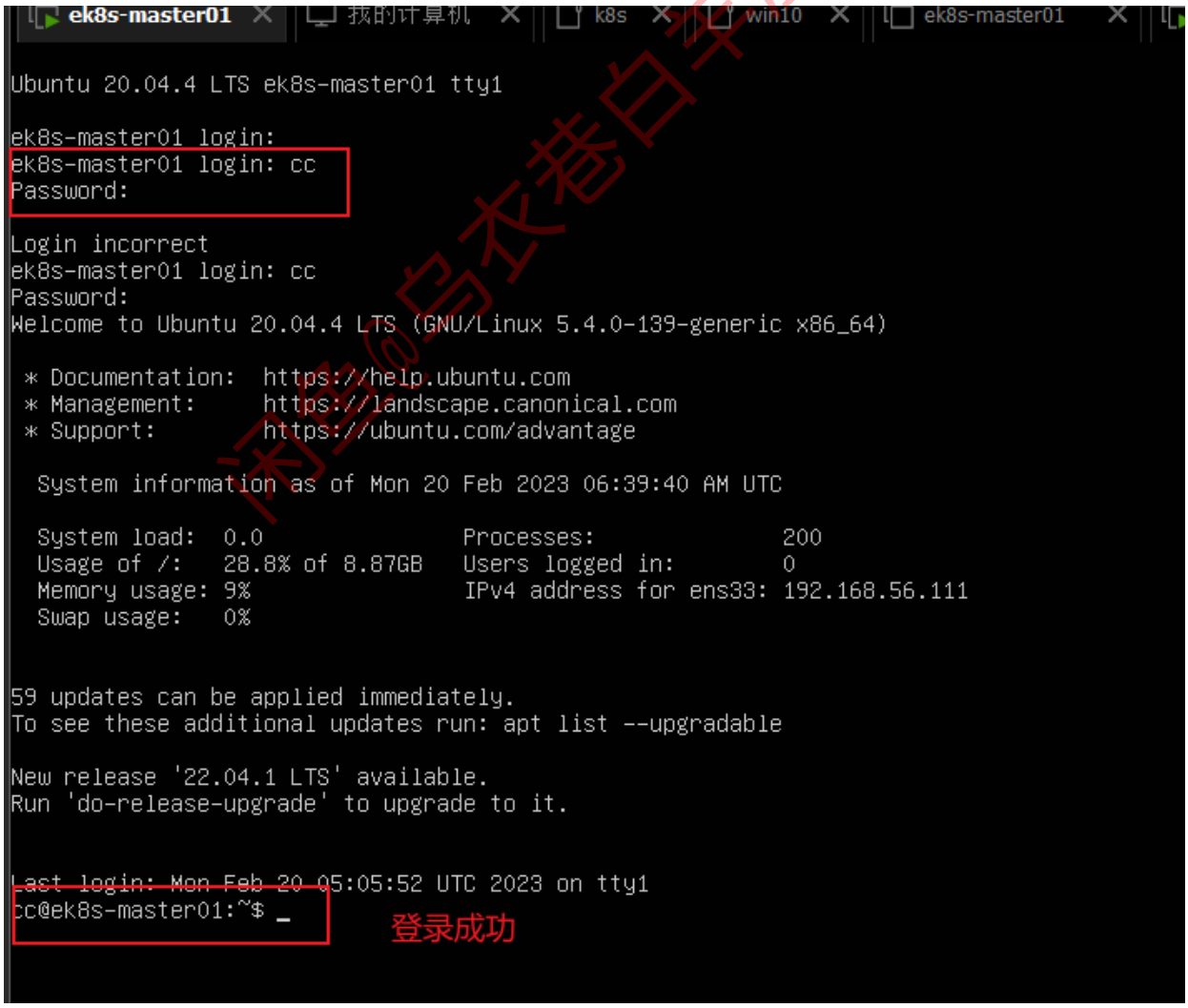
可以先关机创建虚拟机快照: 系统安装完成

2.2系统初始化

> 《2.2 系统初始化》这一章节三个节点都做一遍, 方法一样

点击开机

开机后使用账户cc密码123456登录



2.2.1开启root远程登录

执行sudo su - 输入123456切换到root

```
cc@ek8s-master01:~$ sudo su -
[sudo] password for cc:
root@ek8s-master01:~#
```

设置root密码123456

```
root@ek8s-master01:~# passwd
New password:
Retype new password:
passwd: password updated successfully
root@ek8s-master01:~#
```

#修改sshd配置文件

```
root@ek8s-master01:~# vi /etc/ssh/sshd_config
```

#第34行修改为这样,然后保存退出

```
32
33 #LoginGraceTime 2m
34 PermitRootLogin yes
35 #StrictModes yes
36 #MaxAuthTries 6
37 #MaxSessions 10
38
39 #PubkeyAuthentication yes
-- INSERT --
```

#重启网络服务

```
root@ek8s-master01:~#
root@ek8s-master01:~# systemctl restart sshd
```


2.2.2 配置从节点ssh免密登录

```
#主节点执行

#生成ssh密钥
ssh-keygen -t rsa   # 一直回车

#添加hosts解析
cat >>/etc/hosts<<EOF
192.168.56.111  ek8s-master01 em01
192.168.56.121  ek8s-node01 en01
192.168.56.122  ek8s-node02 en02
EOF

ssh-copy-id ek8s-node01  #输入yes 然后是123456
ssh-copy-id ek8s-node02  #输入yes 然后是123456
```

```

#验证从节点免密登录
root@ek8s-master01:~# ssh ek8s-node01
root@ek8s-master01:~# ssh ek8s-node02
```

2.2.3 配置时间同步服务

> 可以使用xshell工具的“发生键输入到所有会话”功能同时配置三个节点。

root远程登录开启后, 就可以使用ssh工具远程连接服务器了。

主机名	ip地址
ek8s-master01	192.168.56.111
ek8s-node01	192.168.56.121
ek8s-node02	192.168.56.122

账户root, 密码123456

```
apt install -y chrony
sed -i '/^pool/d' /etc/chrony/chrony.conf
echo 'pool ntp.aliyun.com iburst' >>/etc/chrony/chrony.conf
systemctl enable chronyd --now
timedatectl set-timezone Asia/Shanghai
chronyc sources
```

```
root@ek8s-master01:~# chronyc sources
210 Number of sources = 1
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* 203.107.6.88                  2   6    37    19    -37us[ -648us] +/-   16ms
root@ek8s-master01:~#
```

```
shutdown -h now
#将所有节点关机，建立快照“系统初始化完成”
```

三、kubeadm安装k8s

3.1安装containerd(三个节点全部执行)

```
#配置yum源
apt update
apt install -y ca-certificates curl gnupg lsb-release
mkdir -p /etc/apt/keyrings

rm -f /etc/apt/keyrings/docker.gpg
curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://mirrors.aliyun.com/docker-ce/linux/ubuntu \
$(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list

apt update
```

K8s 已经默认不支持Docker了(需要安装额外插件)，这里runc选择Containerd

```
#安装Containerd
apt install containerd.io -y

#配置Containerd的内核
cat <<EOF | tee /etc/modules-load.d/containerd.conf
overlay
br_netfilter
EOF

modprobe overlay
modprobe br_netfilter

cat <<EOF | tee /etc/sysctl.d/99-kubernetes-cri.conf
net.bridge.bridge-nf-call-iptables  = 1
net.ipv4.ip_forward                 = 1
net.bridge.bridge-nf-call-ip6tables = 1
EOF

sysctl --system
```

```
#创建Containerd的配置文件
mkdir /etc/containerd
containerd config default | tee /etc/containerd/config.toml

sed -i 's#SystemdCgroup = false#SystemdCgroup = true#g' /etc/containerd/config.toml
sed -i 's#k8s.gcr.io/pause#registry.cn-hangzhou.aliyuncs.com/google_containers/pause#g'  /etc/containerd/config.toml
sed -i 's#registry.gcr.io/pause#registry.cn-hangzhou.aliyuncs.com/google_containers/pause#g'  /etc/containerd/config.toml
sed -i 's#registry.k8s.io/pause#registry.cn-hangzhou.aliyuncs.com/google_containers/pause#g'  /etc/containerd/config.toml
```

```
# 启动Containerd
systemctl daemon-reload
systemctl restart containerd
systemctl enable containerd
ctr plugin ls
```

```
root@ek8s-master01:~# ctr plugin ls
TYPE                                    ID                    PLATFORMS      STATUS
io.containerd.content.v1                content                -              ok
io.containerd.snapshotter.v1            aufs                   linux/amd64    ok
io.containerd.snapshotter.v1            btrfs                  linux/amd64    skip
io.containerd.snapshotter.v1            devmapper              linux/amd64    error
io.containerd.snapshotter.v1            native                 linux/amd64    ok
io.containerd.snapshotter.v1            overlayfs              linux/amd64    ok
io.containerd.snapshotter.v1            zfs                    linux/amd64    skip
io.containerd.metadata.v1               bolt                   -              ok
io.containerd.differ.v1                 walking                linux/amd64    ok
io.containerd.event.v1                  exchange               -              ok
io.containerd.gc.v1                     scheduler              -              ok
io.containerd.service.v1                introspection-service  -              ok
io.containerd.service.v1                containers-service     -              ok
```


3.2安装kubeadm, kubelet, kubectl

```
#三个节点全部执行
# 添加apt-key
curl -s https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg | apt-key add -
# 添加源
echo "deb https://mirrors.aliyun.com/kubernetes/apt/ kubernetes-xenial main" | tee /etc/apt/sources.list.d/kubernetes.list
```

```
#主节点执行 ek8s-master01
apt update
apt install -y kubelet=1.25.0-00 kubeadm=1.25.0-00 kubectl=1.25.0-00
apt-mark hold kubelet kubeadm kubectl
```

```
#从节点执行 ek8s-node01, ek8s-node02
apt update
apt install -y kubelet=1.25.0-00 kubeadm=1.25.0-00
apt-mark hold kubelet kubeadm
```

3.3 关闭swap

#所有节点执行

```
swapoff -a
#注释自动挂载swap
vim /etc/fstab
```

```
root@ek8s-node01:~# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/ubuntu-vg/ubuntu-lv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-dDcN846GASaUKPtg1WwiudSgwU2HDWOXwHCsL9ZeokyDDufAFHpNP1fgxeybyvWm / ex
1
# /boot was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/5b8a9943-b809-465c-a257-a536a46dbe6c /boot ext4 defaults 0 1
#/swap.img      none    swap    sw      0       0
root@ek8s-node01:~#
```

> 如果没有swap可以忽略

```
root@ek8s-node01:~# free -h
              total        used        free      shared  buff/cache   available
Mem:          1.9Gi       300Mi       520Mi       1.0Mi       1.1Gi       1.4Gi
Swap:            0B          0B          0B
root@ek8s-node01:~#
```

使用free命令检查有无swap

3.4 集群初始化

```
#主节点执行
#拉取镜像
kubeadm config images pull \
--image-repository registry.cn-hangzhou.aliyuncs.com/google_containers --
kubernetes-version 1.25.0
```

```
root@ek8s-master01:~# kubeadm config images pull \
> --image-repository registry.cn-hangzhou.aliyuncs.com/google_containers --kubernetes-version 1.25.0
[config/images] Pulled registry.cn-hangzhou.aliyuncs.com/google_containers/kube-apiserver:v1.25.0
[config/images] Pulled registry.cn-hangzhou.aliyuncs.com/google_containers/kube-controller-manager:v1.25.0
[config/images] Pulled registry.cn-hangzhou.aliyuncs.com/google_containers/kube-scheduler:v1.25.0
[config/images] Pulled registry.cn-hangzhou.aliyuncs.com/google_containers/kube-proxy:v1.25.0
[config/images] Pulled registry.cn-hangzhou.aliyuncs.com/google_containers/pause:3.8
[config/images] Pulled registry.cn-hangzhou.aliyuncs.com/google_containers/etcd:3.5.4-0
[config/images] Pulled registry.cn-hangzhou.aliyuncs.com/google_containers/coredns:v1.9.3
root@ek8s-master01:~#
```

```
#初始化主节点，只在主节点上执行
kubeadm init --apiserver-advertise-address 192.168.56.111 --image-repository registry.cn-hangzhou.aliyuncs.com/google_containers --cri-socket "unix:///var/run/containerd/containerd.sock" --kubernetes-version 1.25.0
```

如果前面的操作没有错误，到这一步结果如下:

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 192.168.56.111:6443 --token 4t5k1v.rugnajpzbruen1u1 \
        --discovery-token-ca-cert-hash sha256:bfd677ea524282aa52de6749493d93cb97dfacb75338757cf251cf26e061979f
root@ek8s-master01:~#
```

保存好上面的 `kubeadm join` 命令

```
#从节点加入集群；两个从节点分别执行；
#这个命令每个人的都不一样，复制自己的命令(就是上面那一行)
kubeadm join 192.168.56.111:6443 --token 4t5k1v.rugnajpzbruen1u1 \
    --discovery-token-ca-cert-hash sha256:bfd677ea524282aa52de6749493d93cb97dfacb75338757cf251cf26e061979f

#后续如果要新增加从节点，从节点上做完前面的初始化后执行这条命令就可以了。
```

从节点已加入集群:

```
This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

root@ek8s-node01:~#
```

```
#主节点执行
#主节点配置kubeconfig(从节点按需配置，不懂就不用管)
mkdir -p $HOME/.kube
cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
chown $(id -u):$(id -g) $HOME/.kube/config
```

这时候验证下，如果能列出三个节点，那么到此你做的是对的。

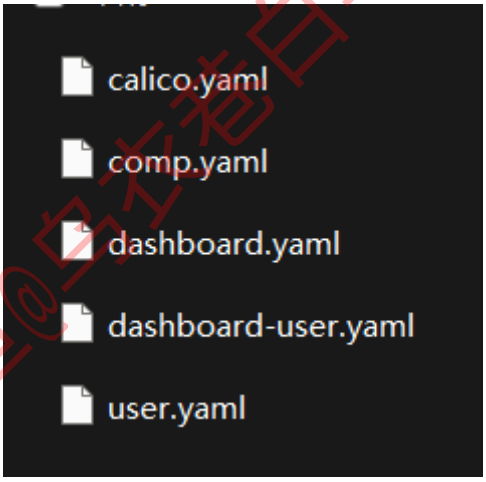
```
root@ek8s-master01:~# kubectl get node
NAME            STATUS     ROLES           AGE     VERSION
ek8s-master01   NotReady   control-plane   4m46s   v1.25.0
ek8s-node01     NotReady   <none>          2m40s   v1.25.0
ek8s-node02     NotReady   <none>          2m40s   v1.25.0
root@ek8s-master01:~#
```

可以看到有三个节点，但是它们的状态都是NotReady，这个可以暂时忽略，后面安装完网络插件后即可恢复。

> 后面题目中的**kubectl**命令都是在主节点执行。

3.5 Addons安装

将云盘中doc下的五个文件上传到ek8s-master01




```
#主节点执行上传
apt install lrzsz -y
mkdir /home/tools
cd /home/tools
rz
ll
```

```
root@ek8s-master01:/home/tools# ll
total 272
drwxr-xr-x 2 root root   4096 Feb 20 16:09 ./
drwxr-xr-x 4 root root   4096 Feb 20 16:08 ../
-rw-r--r-- 1 root root 244494 Feb 16 18:52 calico.yaml
-rw-r--r-- 1 root root   4872 Feb 16 18:52 comp.yaml
-rw-r--r-- 1 root root    449 Feb 16 18:52 dashboard-user.yaml
-rw-r--r-- 1 root root   8001 Feb 16 18:52 dashboard.yaml
-rw-r--r-- 1 root root    429 Jan 31 20:32 user.yaml
```

```
#主节点执行
#创建calico.yaml(网络插件使用calico)
kubectl create -f calico.yaml

#创建metrics-server
#先将/etc/kubernetes/pki/front-proxy-ca.crt复制到所有的node节点
scp /etc/kubernetes/pki/front-proxy-ca.crt ek8s-node01:/etc/kubernetes/pki/
scp /etc/kubernetes/pki/front-proxy-ca.crt ek8s-node02:/etc/kubernetes/pki/
kubectl  create -f comp.yaml

#建dashboard(可选安装,后面用不到)
kubectl create -f dashboard.yaml -f user.yaml
kubectl create token admin-user -n kube-system
```

这时候在检查集群状态, 可以看到三个节点都已经ready了。

```
root@ek8s-master01:/home/tools# kubectl get node
NAME            STATUS   ROLES           AGE   VERSION
ek8s-master01   Ready    control-plane   12m   v1.25.0
ek8s-node01     Ready    <none>          10m   v1.25.0
ek8s-node02     Ready    <none>          10m   v1.25.0
root@ek8s-master01:/home/tools#
```

三个node的metrics数据也都采集到

```
root@ek8s-master01:/home/tools# kubectl top node
NAME            CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
ek8s-master01   198m         9%     1360Mi          35%
ek8s-node01     115m         5%     776Mi           41%
ek8s-node02     102m         5%     967Mi           25%
```

3.6 安装kubelet自动补全

```
#主节点执行
apt-get install bash-completion
source <(kubectl completion bash)
echo "source <(kubectl completion bash)" >> ~/.bashrc
```

集群安装完成, 然后对三个节点做一下快照。

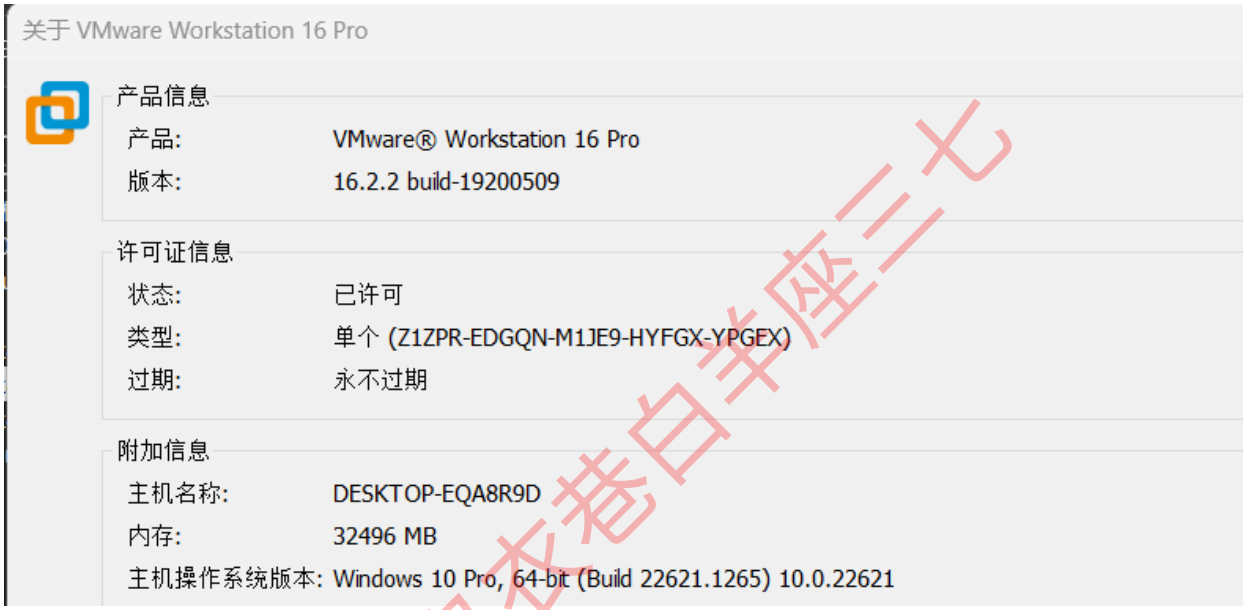
3.7 配置vim yaml缩进

```
echo 'autocmd FileType yaml setlocal ai nu ru ts=2 sw=2 et' >~/.vimrc
```

四、究极安装方法

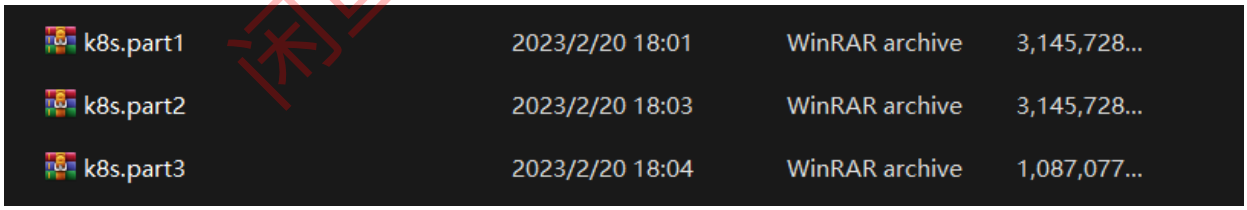
使用此方法前需要先完成《一、准备VMware workstati》章节, 安装好VMware workstation并配置 nat网络。

并且VMware workstation版本必须 ≥ 16.2.2, 如果低于16.2.2会有虚拟机兼容性问题。



4.1 下载虚拟机压缩包

云盘中找到这个四个压缩包, 下载到固态硬盘。

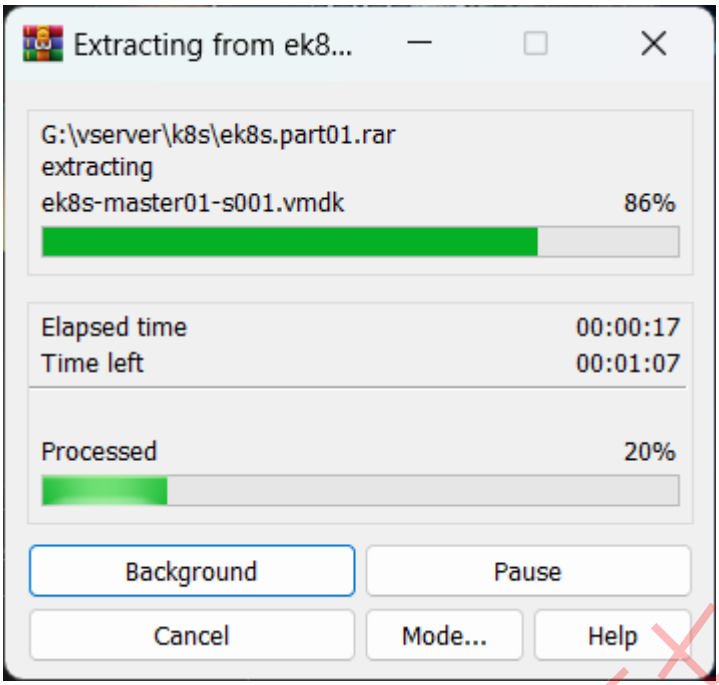


k8s.part1	2023/2/20 18:01	WinRAR archive	3,145,728...
k8s.part2	2023/2/20 18:03	WinRAR archive	3,145,728...
k8s.part3	2023/2/20 18:04	WinRAR archive	1,087,077...

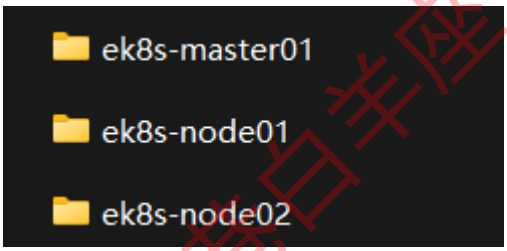
右键全部选中解压

解压密码

等待解压完成。



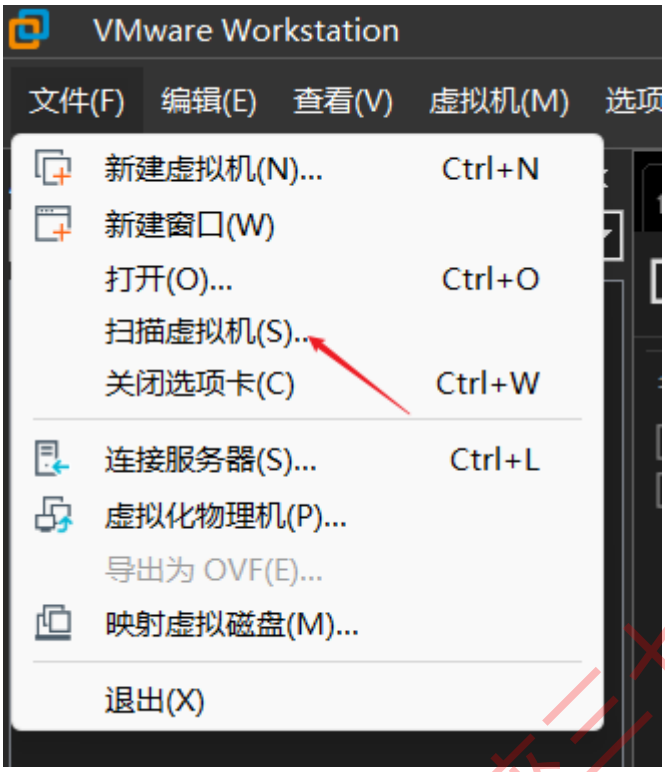
解压完成后是这三个虚拟机目录



总大小在30G左右。

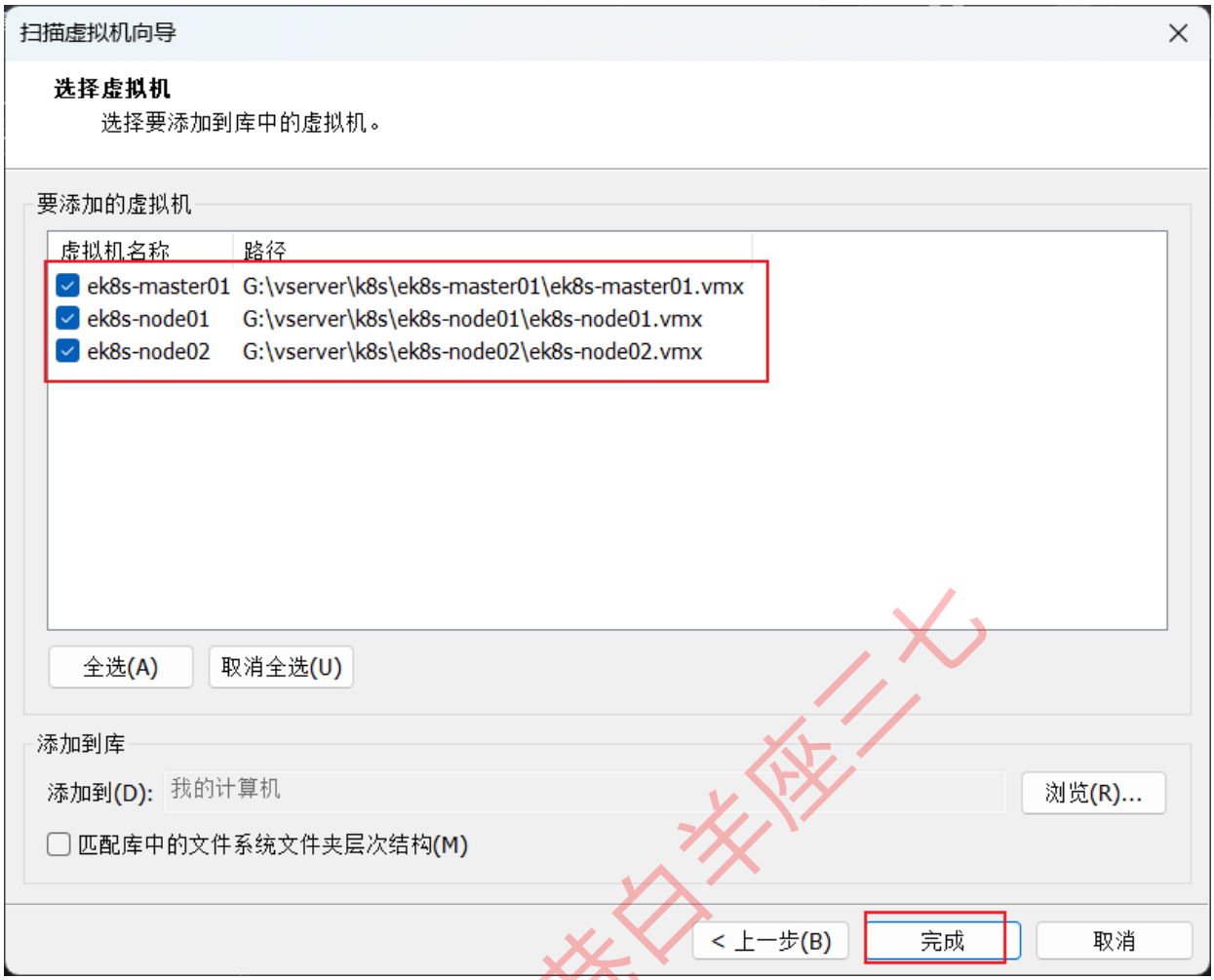
4.2 导入虚拟机

打开vmware workstation

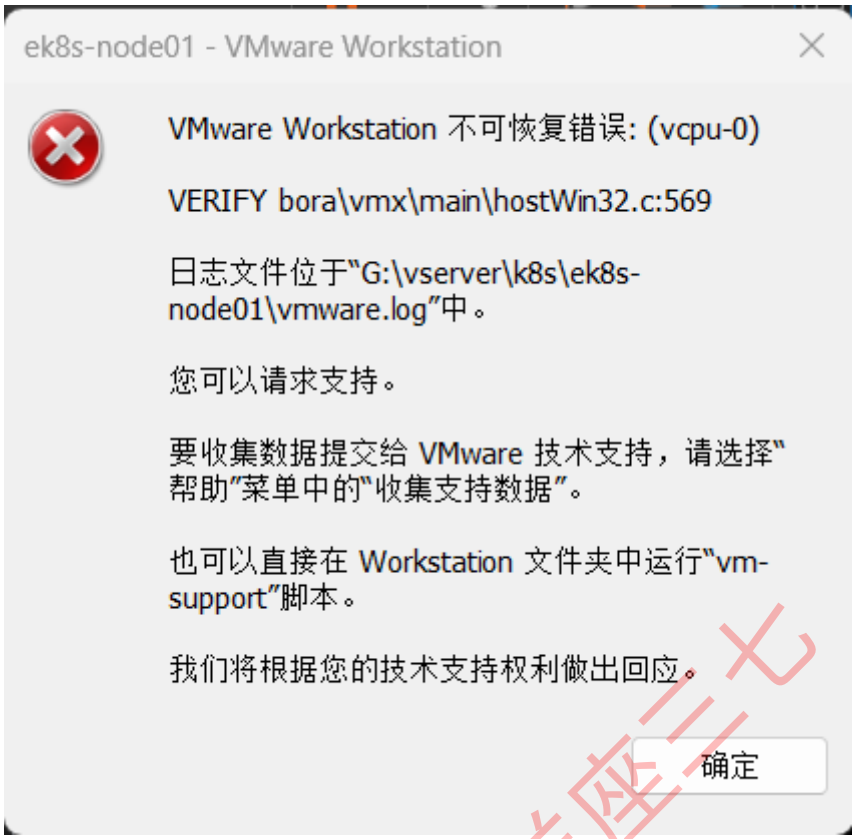


设置为解压目录

扫描到三台虚拟机, 点击完成。



如果打开虚拟机时出现这个报错, 打开虚拟机设置---关闭加速3D图形(取消勾选)



然后以管理员打开cmd：

bcdedit /set hypervisorlaunchtype off

重启笔记本。

4.4 登录集群

开机后ssh登录三个节点

主机名	ip地址
ek8s-master01	192.168.56.111
ek8s-node01	192.168.56.121
ek8s-node02	192.168.56.122

```
#检查集群状态
kubectl get node
kubectl top node
```