

Hyper-V 给虚拟机换磁盘-centos

怎么开头呢，咱也不知道，先上图吧：



某实体服务器使用了 Hyper-V 作为基础虚拟化设施，其上运行若干虚拟机，其中有台虚拟机运行的是 Centos7 的系统，主要运行 web 服务器，刚开始给它分配了 100GB 的磁盘，后来在实际的使用过程中发现这台虚拟机用不了那么多的存储空间，所以想给它降一下配置，只分配 10GB 的磁盘。因为 Hyper-V 本身只能给虚拟机的虚拟磁盘扩容，不支持缩小，所以没别的办法，只能换一个虚拟磁盘，把原来的位于 100GB 磁盘上的 centos7 系统整体迁移到新的 10GB 的磁盘上。（这和给实体机的 centos7 系统换磁盘是差不多的步骤）

操作的主要流程如下：

- 0.先进入目标虚拟机（centos7_web_serv1），查看运行的服务及某些关键配置，再关机
- 1.给目标虚拟机（centos7_web_serv1）添加一块虚拟磁盘，容量为 10GB
- 2.给目标虚拟机添加一个虚拟光驱，使用 centos7 的镜像 iso 文件
- 3.设置目标虚拟机的启动顺序，让 CD/DVD 优先启动，并进入 centos7 光盘的救援系统
- 4.在 centos7 光盘的救援系统里，给新添加的 10GB 磁盘进行分区操作（旧磁盘有几个分区，这个新磁盘也创建对应数量及类型的分区）并格式化至对应的文件系统
- 5.在 centos7 光盘的救援系统里，把旧的 100G 磁盘里的各分区里的文件分别复制到新磁盘里的对应分区里（要保留原文件的各属性及权限）
- 6.修改新磁盘里的 2 个文件（grub2 的配置文件和 centos7 的分区挂载文件），主要是修改里面目标分区的 uuid 或 lvm 分区的卷名，因为磁盘变了，其分区的 uuid 也变了
- 7.为了系统能正常启动，要创建及复制几个文件，主要是受 selinux 和 hyper-v 的预启动程序（固件）的影响
- 8.退出 centos7 光盘的救援系统，然后在 hyper-v 里删除旧的 100G 磁盘（这个其实只是卸载磁盘，并不是删除真实的虚拟磁盘文件），还要把 centos7 的虚拟光驱也卸载了
- 9.修改目标虚拟机的启动顺序，从新的磁盘启动，如无意外，可正常进入系统，如有意外，请见文档末尾的常见问题

**本文档发布时为 pdf 文档，请下载文档至本地查看，点击书签进入相应章节

**请不要在真实的生产环境中直接做测试，先在自己本地的 hyper-v 里做一下实验

0.进入目标虚拟机



在 hyper-v 管理器界面，双击目标虚拟机，进入它的 console 界面



输入用户名和密码，进入 centos7 系统，查看磁盘挂载情况及分区结构以及各分区目前使用的存储大小，方便规划新磁盘的大小和分区

```
[root@localhost ~]#  
[root@localhost ~]# df -Th  
Filesystem                Type      Size  Used Avail Use% Mounted on  
/dev/mapper/centos-root    xfs       90G   979M   89G   2% /  
devtmpfs                  devtmpfs  903M    0   903M   0% /dev  
tmpfs                     tmpfs     914M    0   914M   0% /dev/shm  
tmpfs                     tmpfs     914M  8.6M   906M   1% /run  
tmpfs                     tmpfs     914M    0   914M   0% /sys/fs/cgroup  
/dev/sda2                  xfs       494M  127M   368M  26% /boot  
/dev/sda1                  vfat      200M   9.8M  191M   5% /boot/efi  
/dev/mapper/centos-home    xfs       8.0G    33M   8.0G   1% /home  
tmpfs                     tmpfs     183M    0   183M   0% /run/user/0  
[root@localhost ~]#  
[root@localhost ~]# lsblk  
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
sda           8:0    0 100G  0 disk  
├─sda1        8:1    0  200M  0 part /boot/efi  
├─sda2        8:2    0  500M  0 part /boot  
└─sda3        8:3    0   90G  0 part  
   └─centos-root 253:0    0   90G  0 lvm /  
      └─centos-home 253:1    0    8G  0 lvm /home  
sdb           8:16    0   10G  0 disk  
sr0          11:0    1  906M  0 rom  
[root@localhost ~]#  
[root@localhost ~]# _
```

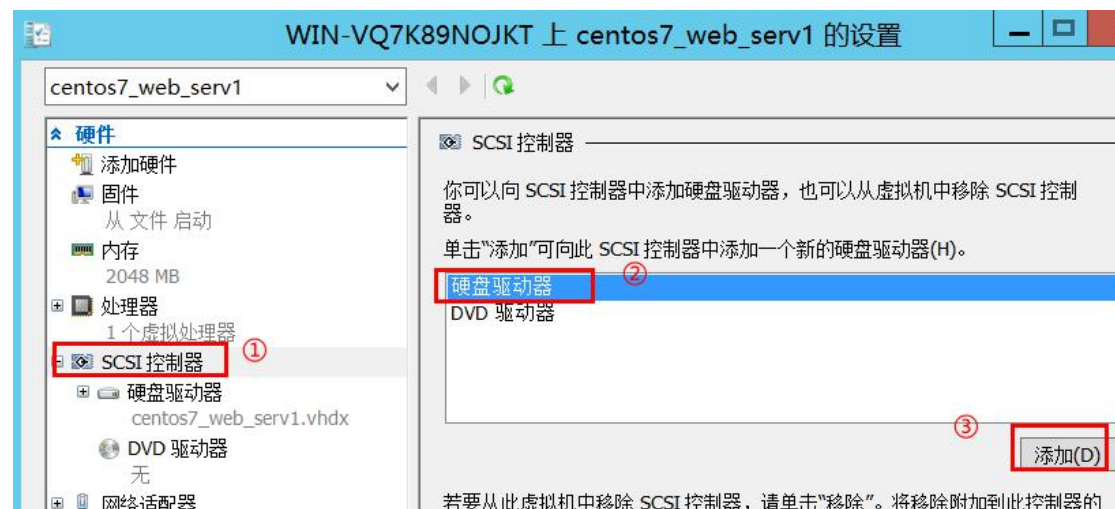
上图可见，原磁盘有 3 个分区，sda1 和 sda2 是标准分区，sda3 是 Lvm 分区，创建了卷组 centos 并在此卷组里划分 2 个逻辑卷 root 和 home，分别挂载到/根目录和/home 目录
sda1 挂载到/boot/efi 目标，sda2 挂载到/boot 目录

本例中是这种情况，大家的实际情况可能不一样，记住这些分区的相关信息就行了，比如分区大小，挂载的目录，目前实际使用的存储大小；还要知道它运行的服务都有哪些，再关机

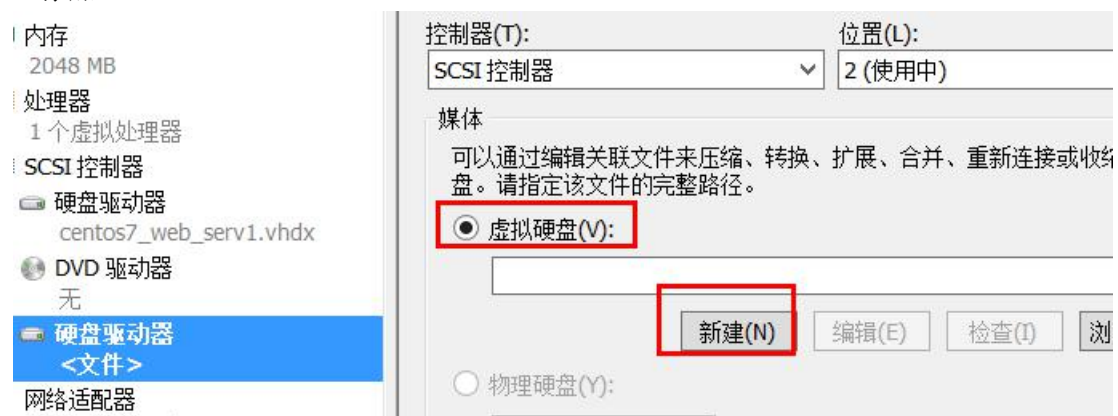
1. 添加新的虚拟磁盘



如上图，在 Hyper-V 管理器主界面，选中目标虚拟机，点击右边的“设置”



在虚拟机的设置界面里，先点击左边的“SCSI 控制器”，再点击右边的“硬盘驱动器”，点击“添加”



然后扩展出一个虚拟硬盘，点击“新建”



选择磁盘类型

开始之前

选择磁盘类型

指定名称和位置

配置磁盘

摘要

你要创建哪种类型的虚拟硬盘？

☐ 固定大小(X)

这种类型的磁盘可提供更好的性能，建议用于运行具有高级别磁盘器。虚拟硬盘文件在最初创建时使用虚拟硬盘的大小，并且当删除更改。

☒ 动态扩展(D)

这种类型的磁盘可更好地利用物理存储空间，建议用于运行不含有序的服务器。虚拟硬盘文件在最初创建时很小，但随着向其中添加

☐ 差异(I)


这种类型的磁盘与要保持原样的另一个磁盘之间存在父子关系。你情况下对数据或操作系统进行更改，以便可以轻松还原更改。所有盘相同的虚拟硬盘格式(VHD 或 VHDX)。

< 上一步(P)

下一步(N) >

完

在“选择磁盘类型”里，一般就用动态扩展的磁盘，方便以后的扩容，下一步



指定名称和位置

开始之前

选择磁盘类型

指定名称和位置

配置磁盘

摘要

指定虚拟硬盘文件的名称和位置。

名称(M)

位置(L):

< 上一步(P)

下一步(N) >

指定名称最好是容易分辨的，比如以目标虚拟名为前缀，再加个 new 之类的，“位置”表示此虚拟磁盘存储在实体机里的位置，建议和原来的旧磁盘位于同一目录，下一步



配置磁盘

开始之前
选择磁盘类型
指定名称和位置
配置磁盘
摘要

你可以创建一个空白虚拟硬盘，也可以复制现有物理磁盘的内容

☒ 新建空白虚拟硬盘(B)
大小(S): GB (最大值: 64 TB)

☐ 复制指定物理磁盘的内容(C):

物理硬盘	大小
\\.\PHYSICALDRIVE0	39 GB
\\.\PHYSICALDRIVE1	119 GB

☐ 复制指定的虚拟硬盘的内容(V)
路径(A):

< 上一步(P) **下一步(N) >**

新的磁盘大小设置为 10GB，这个要根据实际情况来，本例中的情况是目标虚拟机的系统一共只用了 2GB 左右的存储空间，且估计以后也不会用到太多，所以选择了 10GB 的容量



正在完成新建虚拟硬盘向导

开始之前
选择磁盘类型
指定名称和位置
配置磁盘
摘要

你已经成功完成了新建虚拟硬盘向导。即将创建下列虚拟硬盘。

描述:

格式: VHDX
类型: 动态扩展
名称: centos7_web_serv1_newDisk.vhdx
位置: D:\hyper_v\vm_Disk
大小: 10 GB

若要创建虚拟硬盘并关闭向导，请单击“完成”。

< 上一步(P) 下一步(N) > **完成(F)**

然后点击“完成”



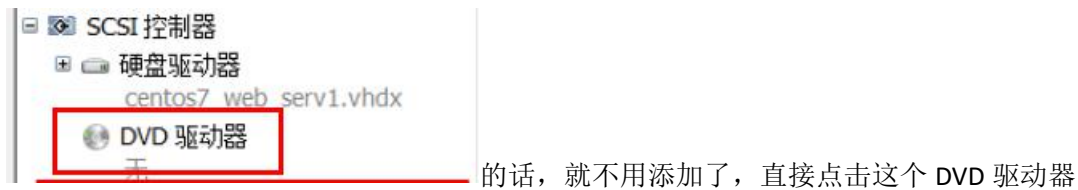
然后发现左侧多了刚刚添加的磁盘，点击右下角的“应用”

2. 添加虚拟光驱，加载 centos7 的 iso 文件

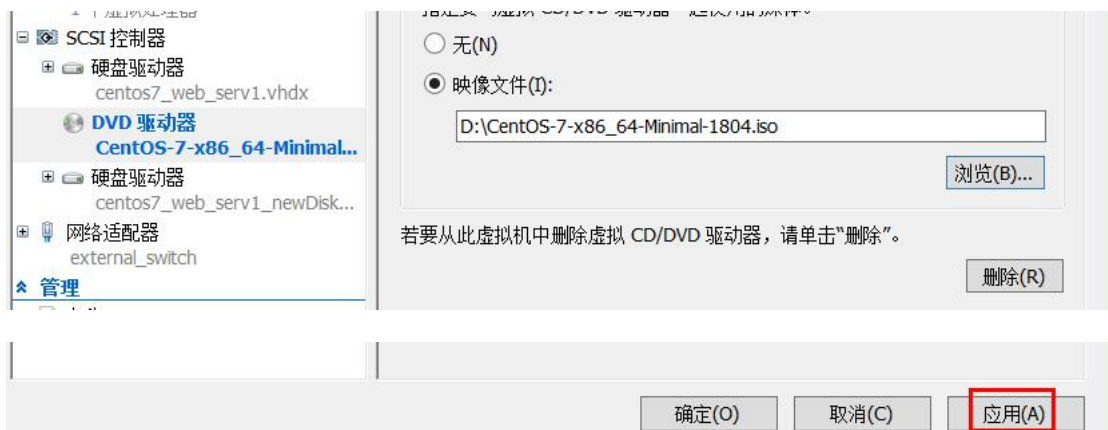
还是在目标虚拟机的设置界面里，



点击左侧的“SCSI 控制器”，右边选择 DVD 驱动器，点击“添加”（如果原本有 dvd 驱动器

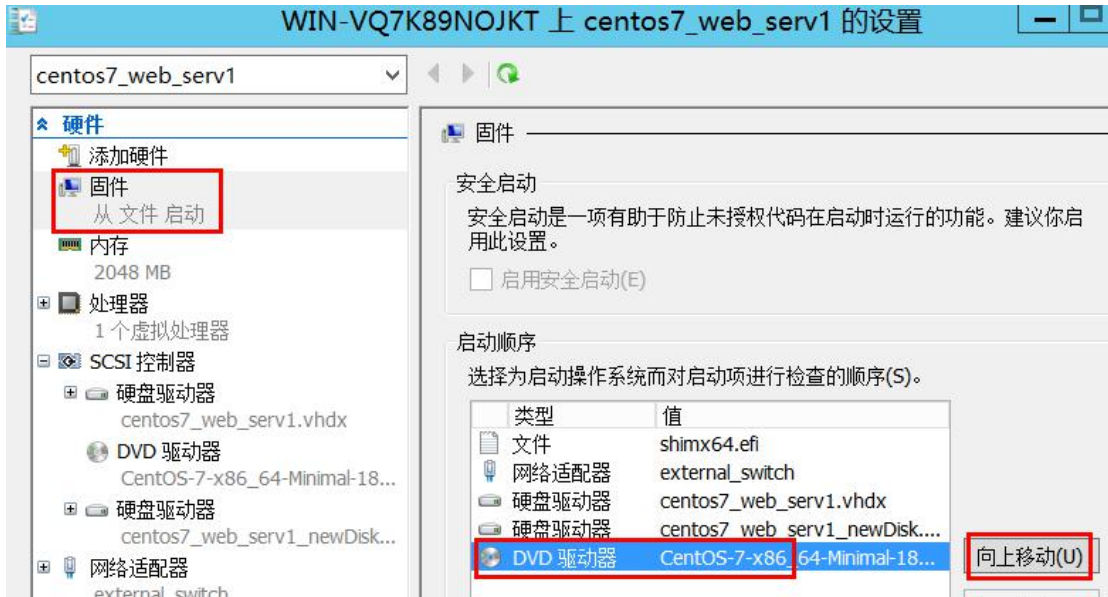


选择使用媒体为“映像文件”，点击“浏览”进入文件夹，选择 centos7 的安装镜像 iso 文件

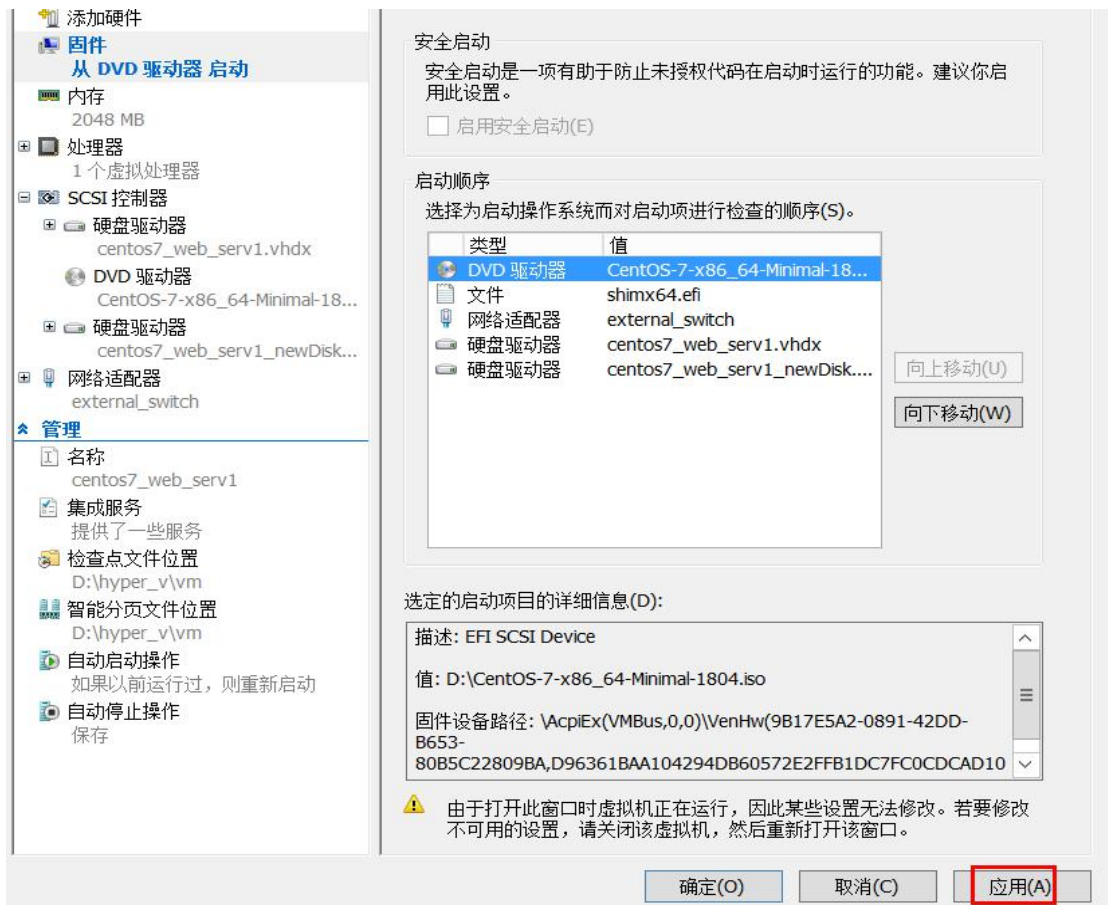


然后点击右下角的“应用”

3.设置目标虚拟机的启动顺序



在目标虚拟机的设置界面，点击左侧的“固件”，再点击右边的“DVD 驱动器 centos7xxx”的虚拟硬件，点击“向上移动”，移动至最顶上



再点击右下角的“应用”，最后“确定”退出配置界面

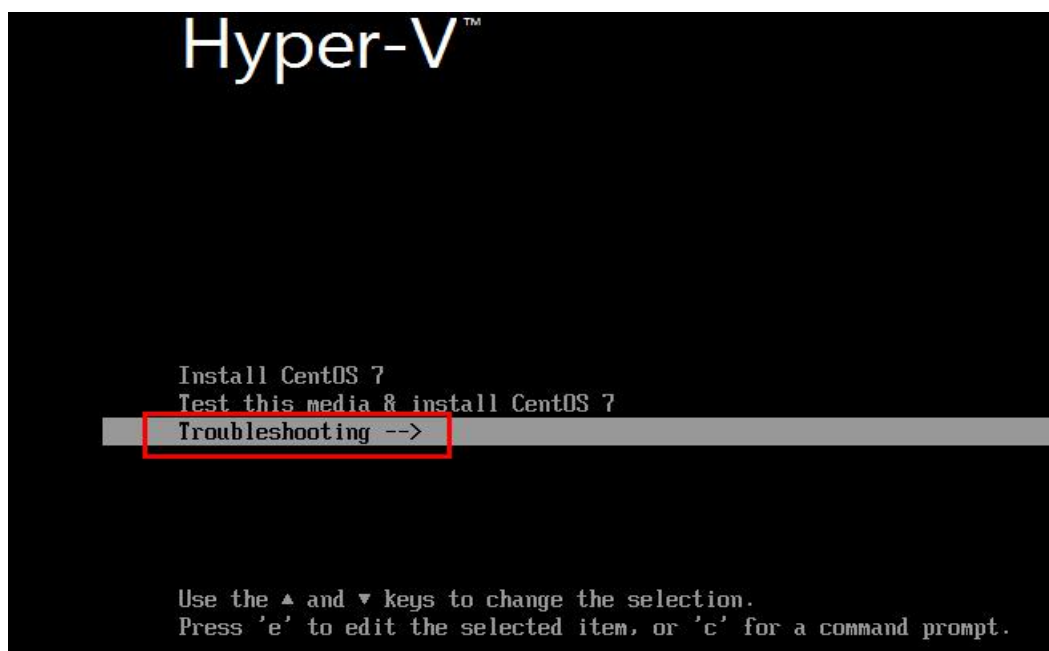
4.进入 centos7 光盘的救援系统



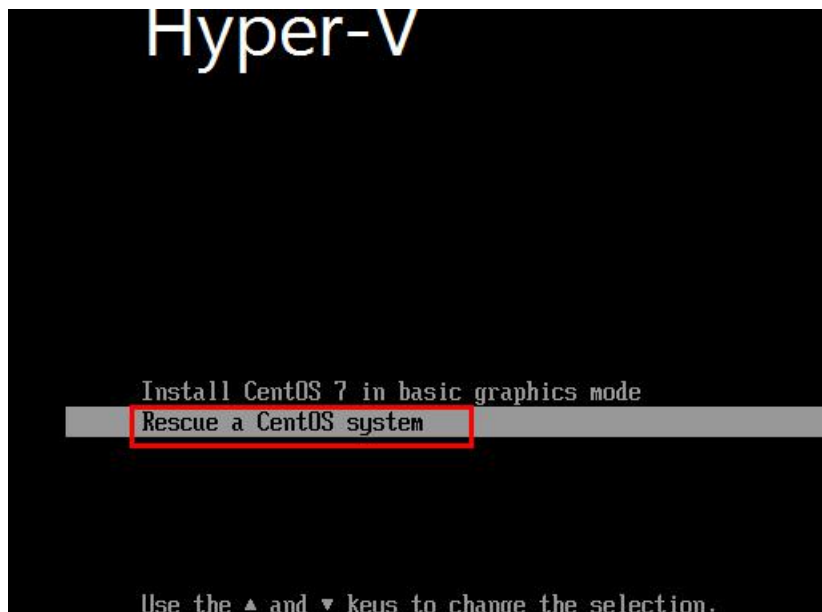
在 hyper-v 管理器界面，双击目标虚拟机，进入它的 console 界面



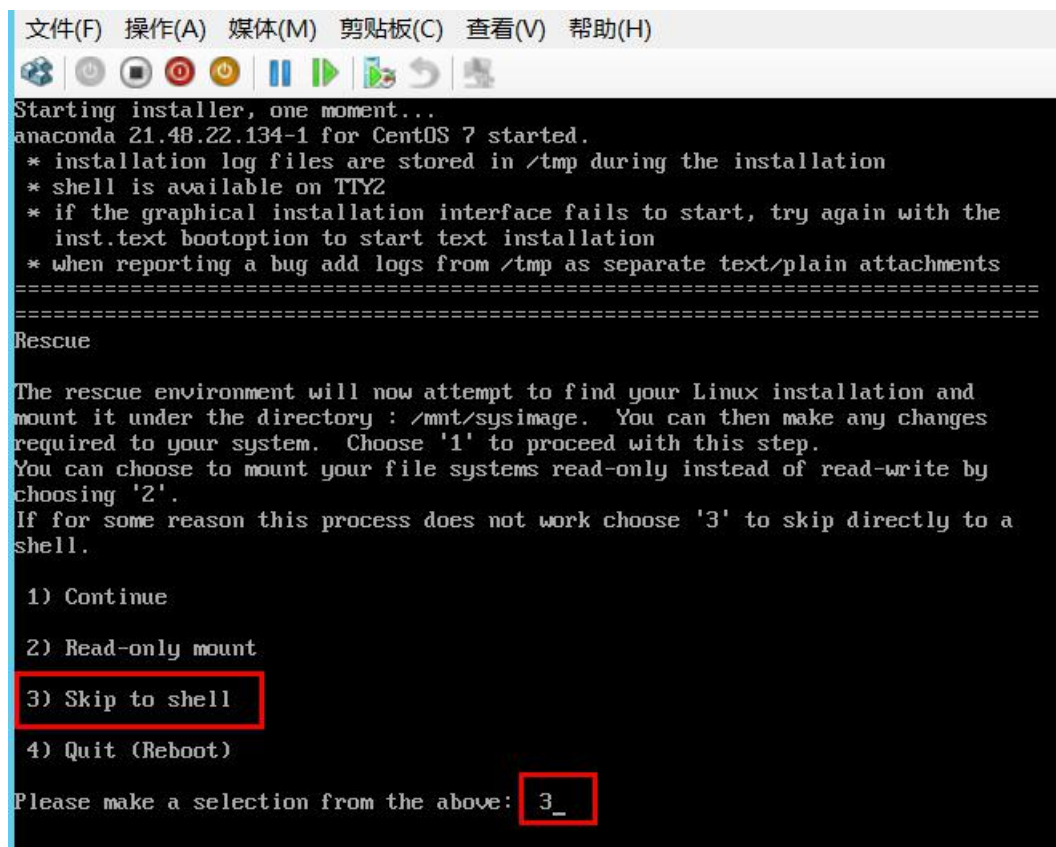
先启动此虚拟机，可以点击“操作”→“启动”，也可直接点击工具栏左边的那个绿色开关



然后就进入了虚拟机的启动界面，上图可见已成功从 centos7 的安装光盘启动，选择 “TroubleShooting”，回车



再选中“Rescue a CentOS system”，回车进入救援模式



如上图，我们输入 3，回车，直接进入 shell 命令行

```

3) Skip to shell
4) Quit (Reboot)

Please make a selection from the above: 3
When finished, please exit from the shell and your system will reboot.
sh-4.2#
sh-4.2# fdisk -l | more

```

然后输入命令:

`fdisk -l | more` //查看磁盘,之所以要用 `|more` 是因为输出的信息可能比较多,导致前面的被覆盖了,而我们要看的目标磁盘一般就在前面 2 个

```

Disk /dev/sda: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk label type: gpt
Disk identifier: 83B82DFE-EFD6-4143-B8AD-82AE7A9130C8

#           Start          End      Size  Type          Name
1           2048          411647    200M  EFI System    EFI System Partition
2           411648        1435647    500M  Microsoft basic
3          1435648        206964735    98G   Linux LVM

Disk /dev/sdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes

```

果然,输出的第一页就有目标磁盘,

第一个 `/dev/sda` 是原来的旧磁盘,大小 107.4GB,有三个分区,因为计算方式的差异所以显示的不是恰好 100GB。

第二个磁盘 `/dev/sdb` 大小 10.7GB,是我们新添加的磁盘

并不是说 原有的旧磁盘盘号就一定是 `sda`,也有可能是其他的,一定要根据实际情况记下新旧磁盘对应的盘号

4.2 给新磁盘创建分区

```

sh-4.2#
sh-4.2# gdisk /dev/sdb
[anaconda] 1:main* 2:shell 3:log

```

输入命令:

`gdisk /dev/sdb` //使用 `gdisk` 工具对新磁盘进行分区,一定要先确认新磁盘的盘号

```

Command (? for help): p
Disk /dev/sdb: 20971520 sectors, 10.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): AA0DF03A-A269-4F5B-B945-EE8E326CAD04
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 20971486
Partitions will be aligned on 2048-sector boundaries
Total free space is 20971453 sectors (10.0 GiB)

Number  Start (sector)    End (sector)  Size      Code  Name

```

在 `gdisk` 交互界面，输入字符 `p` 并回车，列出目标磁盘的所有分区，上图可见为空，说明这是一块新的磁盘，我们才放心继续操作，

```
Command (? for help): o
This option deletes all partitions and creates a new protective MBR.
Proceed? (Y/N) y
Command (? for help):
```

在 `gdisk` 交互界面，输入字母 `o` 并回车，将目标磁盘设置为使用 GPT 分区表，因为 Hyper-V 的第二代虚拟机使用的都是 UEFI 启动模式，所以要使用 GPT 类型的磁盘，输入 `y` 确认

```
Command (? for help): n
Partition number (1-128, default 1):
First sector (34-20971486, default = 2048) or {+}-size{KMGT}:
Last sector (2048-20971486, default = 20971486) or {+}-size{KMGT}: +200M
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300):
```

输入字符命令 `n` 创建一个新分区，使用默认分区编号 1，使用默认起始 LBA 地址 2048，Last Sector 那里，输入 `+200M` 表示该分区大小为 200M，回车

默认是 Linux filesystem 的分区类型，我们可继续输入字母 `l` 列出 GPT 分区类型对应的 code

```
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): l
0700 Microsoft basic data      0c01 Microsoft reserved      2700 Windows RE
4200 Windows LDM data          4201 Windows LDM metadata    7501 IBM GPFS
7f00 ChromeOS kernel          7f01 ChromeOS root           7f02 ChromeOS reserved
8200 Linux swap                8300 Linux filesystem        8301 Linux reserved
8e00 Linux LVM                 a500 FreeBSD disklabel      a501 FreeBSD boot
a502 FreeBSD swap              a503 FreeBSD UFS             a504 FreeBSD ZFS
a505 FreeBSD Vinum/RAID        a580 Midnight BSD data      a581 Midnight BSD boot
a582 Midnight BSD swap         a583 Midnight BSD UFS       a584 Midnight BSD ZFS
a585 Midnight BSD Vinum        a800 Apple UFS               a901 NetBSD swap
a902 NetBSD FFS                 a903 NetBSD LFS              a904 NetBSD concatenated
a905 NetBSD encrypted          a906 NetBSD RAID            ab00 Apple boot
af00 Apple HFS/HFS+            af01 Apple RAID              af02 Apple RAID offline
af03 Apple label               af04 AppleTV recovery        af05 Apple Core Storage
be00 Solaris boot              bf00 Solaris root            bf01 Solaris /usr & Mac Z
bf02 Solaris swap              bf03 Solaris backup          bf04 Solaris /var
bf05 Solaris /home             bf06 Solaris alternate se    bf07 Solaris Reserved 1
bf08 Solaris Reserved 2        bf09 Solaris Reserved 3      bf0a Solaris Reserved 4
bf0b Solaris Reserved 5        c001 HP-UX data              c002 HP-UX service
ed00 Sony system partition     ef00 EFI System              ef01 MBR partition scheme
ef02 BIOS boot partition       fb00 VMware UFS              fb01 VMware reserved
fc00 VMware kcore crash p      fd00 Linux RAID
```

```
Hex code or GUID (L to show codes, Enter = 8300): ef00
Changed type of partition to 'EFI System'
```

我们的新磁盘第一个分区也要和旧磁盘的第一个分区相对应，所以输入 `ef00` 表示使用 EFI System 类型

接着创第 2 个分区，操作如下：

```
Command (? for help): n
Partition number (2-128, default 2):
First sector (34-20971486, default = 411648) or {+}-size{KMGT}:
Last sector (411648-20971486, default = 20971486) or {+}-size{KMGT}: +500M
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): 0700
Changed type of partition to 'Microsoft basic data'
```

只是在分区类型这里 输入 `0700` 表示使用 Microsoft basic data 的分区类型，对，centos 的数

据分区的 **gpt** 分区类型也和 **Microsoft** 的 **windows** 一样，而不是 **Linux filesystem** 这个只是表示 **gpt** 分区的类型，而不是文件系统的类型，**gpt** 磁盘的分区类型和文件系统是弱相关的，和 **mbr** 磁盘的分区类型不太一样。

```
Command (? for help): n
Partition number (3-128, default 3):
First sector (34-20971486, default = 1435648) or {+-}size{KMGT}:
Last sector (1435648-20971486, default = 20971486) or {+-}size{KMGT}:
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): 8e00
Changed type of partition to 'Linux LVM'
```

创建完后，输入 p 列出分区：

```
sh-4.2# fdisk -l /dev/sdb
WARNING: fdisk GPT support is currently new, and therefore in an experimental phase.

Disk /dev/sdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk label type: gpt
Disk identifier: 5B985015-A12D-4A04-97DC-16A9EECD681C


#          Start          End      Size  Type              Name
1          2048          411647    200M  EFI System        EFI System
2          411648        1435647    500M  Microsoft basic   Microsoft basic data
3          1435648        20971486    9.3G  Linux LVM          Linux LVM
sh-4.2#
```


将 sdb1 分区格式化为 vfat 文件系统，centos7 的 EFI 分区使用 vfat 文件系统（和 fat16 一样）

`mkfs.vfat /dev/sdb1`

```
sh-4.2# mkfs.vfat /dev/sdb1
mkfs.fat 3.0.20 (12 Jun 2013)
sh-4.2#
```

再将 sdb2 格式化为 xfs 文件系统

`mkfs.xfs /dev/sdb2`

```
sh-4.2# mkfs.xfs /dev/sdb2
meta-data=/dev/sdb2            isize=512    agcount=4, agsize=32000 blks
      =                       sectsz=4096   attr=2, projid32bit=1
      =                       crc=1        finobt=0, sparse=0
data      =                       bsize=4096   blocks=128000, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0 ftype=1
log       =internal log        bsize=4096   blocks=1605, version=2
      =                       sectsz=4096   sunit=1 blks, lazy-count=1
realtime  =none                extsz=4096   blocks=0, rtextents=0
sh-4.2#
sh-4.2#
```

因为旧磁盘的 sda3 是用了 lvm，卷组名为 centos，有 2 个逻辑卷 root 和 home
所以新磁盘的 sdb3 也要用 lvm，卷组名在这里没法和原有的 centos 同名，所以用其他的名字，比如 centos2，逻辑卷名也用 root 和 home

```
sh-4.2# pvcreate /dev/sdb3
Physical volume "/dev/sdb3" successfully created.
sh-4.2# vgcreate centos2 /dev/sdb3
Volume group "centos2" successfully created
sh-4.2# lvcreate -L 7G -n root centos2
Logical volume "root" created.
sh-4.2#
sh-4.2# lvcreate -L 2G -n home centos2
Logical volume "home" created.
sh-4.2#
```

`pvcreate /dev/sdb3` //把 sdb3 分区做成 pv

`vgcreate centos2 /dev/sdb3` //创建名为 centos2 的卷组，并把 sdb3 加入卷组

`lvcreate -L 7G -n root centos2` //创建 root 逻辑卷，大小为 7G

`lvcreate -L 2G -n home centos2` //创建 home 逻辑卷，大小 2G，

要根据实际情况来分配，再创建文件系统

```
sh-4.2# mkfs.xfs /dev/centos2/root
meta-data=/dev/centos2/root    isize=512    agcount=4, agsize=458752 blks
      =                       sectsz=4096   attr=2, projid32bit=1
      =                       crc=1        finobt=0, sparse=0
data      =                       bsize=4096   blocks=1835000, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0 ftype=1
log       =internal log        bsize=4096   blocks=2560, version=2
      =                       sectsz=4096   sunit=1 blks, lazy-count=1
realtime  =none                extsz=4096   blocks=0, rtextents=0
sh-4.2#
```

`mkfs.xfs /dev/centos2/root`

```
sh-4.2# mkfs.xfs /dev/centos2/home
meta-data=/dev/centos2/home    isize=512    agcount=4, agsize=131072 blks
=                               sectsz=4096   attr=2, projid32bit=1
=                               crc=1         finobt=0, sparse=0
data      =                     bsize=4096   blocks=524288, imaxpct=25
=                               sunit=0         swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0 ftype=1
log       =internal log       bsize=4096   blocks=2560, version=2
=                               sectsz=4096   sunit=1 blks, lazy-count=1
realtime  =none                extsz=4096   blocks=0, rtextents=0
sh-4.2#
```

```
# mkfs.xfs /dev/centos2/home
```

给逻辑卷创建完文件系统，再挂载到某目录下，准备把旧磁盘上的文件移到新磁盘上
**不过，旧磁盘的 lvm 卷组在救援模式下没法直接使用，它还未激活卷组，要先激活旧磁盘的 centos 卷组才能使用

```
# vgchange -ay
```

5.复制原磁盘所有文件至新磁盘

先在 centos7 光盘的救援系统下创建挂载目录

```
sh-4.2# mkdir /mnt/sda1
sh-4.2# mkdir /mnt/sda2
sh-4.2# mkdir /mnt/sda_root
sh-4.2# mkdir /mnt/sda_home
sh-4.2#
sh-4.2# mkdir /mnt/sdb1
sh-4.2# mkdir /mnt/sdb2
sh-4.2# mkdir /mnt/sdb_root
sh-4.2# mkdir /mnt/sdb_home
sh-4.2#
sh-4.2#
```

再分别挂载各分区到相应目录下

```
sh-4.2# mount /dev/sda1 /mnt/sda1
sh-4.2# mount /dev/sda2 /mnt/sda2
sh-4.2# mount /dev/centos/root /mnt/sda_root
sh-4.2# mount /dev/centos/home /mnt/sda_home
sh-4.2#
sh-4.2# mount /dev/sdb1 /mnt/sdb1
sh-4.2# mount /dev/sdb2 /mnt/sdb2
sh-4.2# mount /dev/centos2/root /mnt/sdb_root
sh-4.2# mount /dev/centos2/home /mnt/sdb_home
sh-4.2#
```

然后查看一下新磁盘的 root 分区里是否有文件：

```
sh-4.2# ls /mnt/sdb_root
sh-4.2#
sh-4.2#
```

没有文件，确定这是新的磁盘，再复制旧磁盘里的文件到新磁盘里

`cp -r -p /mnt/旧磁盘分区挂载目录/* /mnt/新磁盘分区挂载目录`

```
sh-4.2# cp -r -p /mnt/sda1/* /mnt/sdb1
sh-4.2# cp -r -p /mnt/sda2/* /mnt/sdb2
sh-4.2# cp -r -p /mnt/sda_root/* /mnt/sdb_root
sh-4.2# cp -r -p /mnt/sda_home/* /mnt/sdb_home
```

如果旧磁盘的 root 分区和 home 分区文件较多的话，需要点时间去复制，请耐心等待

复制完后查看一下新磁盘的 root 分区里是否有/下的所有目录

```
sh-4.2# ls /mnt/sdb_root
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
sh-4.2#
```

有就 ok 了

6. 修改 2 个配置文件的磁盘分区 uuid

blkid //先列出磁盘及分区的 uuid

```
sh-4.2# blkid
/run/install/repo/LiveOS/squashfs.img: TYPE="squashfs"
/dev/sr0: UUID="2018-05-03-21-07-04-00" LABEL="CentOS 7 x86_64" TYPE="iso9660" PTTYPE="dos"
/dev/sdb1: SEC_TYPE="msdos" UUID="6899-F9B0" TYPE="vfat" PARTLABEL="EFI System" PARTUUID="65588a7e-9f66-4e82-be35-a36b98cdec12"
/dev/sdb2: UUID="a1a1e7ca-e7fc-4349-a070-6a59108fd347" TYPE="xfs" PARTLABEL="Microsoft basic data" PARTUUID="6113f045-74e9-42cd-7"
/dev/sdb3: UUID="gRaZku-DFqY-7tv7-WJxq-eYn3-K2i4-ZCjguZ" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="975285cd-9f8f-4135-"
/dev/sda1: SEC_TYPE="msdos" UUID="E280-64F5" TYPE="vfat" PARTLABEL="EFI System Partition" PARTUUID="2c00685e-69a7-4807-99f3-7e2"
/dev/sda2: UUID="3d7c4d34-b156-426c-b140-ccd7dd8f9246" TYPE="xfs" PARTUUID="7c727b15-e120-4401-9439-53661772662b"
/dev/sda3: UUID="TLFTpp-vFxx-Wjlc-RrCz-Kz10-10TE-UNcv3r" TYPE="LVM2_member" PARTLABEL="Linux LVM" PARTUUID="14129de0-4923-4931-a2af-edd74263d677"
/dev/loop0: TYPE="squashfs"
/dev/loop1: LABEL="Anaconda" UUID="d9beaa77-2f65-4264-8aa4-c4f0b300688d" TYPE="ext4"
/dev/loop2: TYPE="DM_snapshot_cow"
/dev/mapper/live-rw: LABEL="Anaconda" UUID="d9beaa77-2f65-4264-8aa4-c4f0b300688d" TYPE="ext4"
/dev/mapper/live-base: LABEL="Anaconda" UUID="d9beaa77-2f65-4264-8aa4-c4f0b300688d" TYPE="ext4"
/dev/zram0: UUID="4577a37c-445a-4e5e-81hf-9ca945ea070a" TYPE="swap"
/dev/mapper/centos-root: UUID="d50b6e81-efb9-4f38-9059-2ff7977cac59" TYPE="xfs"
/dev/mapper/centos-home: UUID="7b5b5a77-74c5-4f4d-86d0-7590d117d350" TYPE="xfs"
/dev/mapper/centos2-root: UUID="9a28792c-6f94-4b76-9ff0-a51cc47ab067" TYPE="xfs"
/dev/mapper/centos2-home: UUID="a3ac37c7-4f9e-45c3-866a-ed58eaf509f" TYPE="xfs"
sh-4.2#
```

对于旧磁盘分区的 uuid 可以只记前面几个字符，知道是它的 uuid 就行，对于新磁盘的 sdb1 和 sdb2 的分区 uuid 要记全。最好是截个图或用手机拍照，也可以写在纸上，要确保正确 uuid 是由 16 进制字符和横杠组成的：

遇到 l 则表示数字 1，不可能是字母 l 或 i

遇到 o 或 0 则表示数字 0，不可能是字母 o

字母 b 和数字 6 要写清楚，切不可搞混了

首先，修改 grub2 的配置文件，在新磁盘的 efi 分区上，即 sdb1 上的/EFI/centos/grub.cfg

```
sh-4.2# vi /mnt/sdb1/EFI/centos/grub.cfg
if [ x$feature_all_video_module = xy ]; then
  insmod all_video
else
  insmod efi_gop
  insmod efi_uga
  insmod ieee1275_fb
  menuentry
  anaconda 1:main* 2:shell 3:log 4:storage-log 5:prog
```

查找 menuentry 所在行，即启动菜单，默认是有 2 个启动菜单

```

set superusers= root
export superusers
password_pbkdf2 root ${GRUB2_PASSWORD}
fi
fi
### END /etc/grub.d/01_users ###

### BEGIN /etc/grub.d/10_linux ###
menuentry 'CentOS Linux (3.10.0-862.el7.x86_64) ? (Core)' --class centos --class gnu-linux --class gnu --class os --t
option 'gnulinux-3.10.0-862.el7.x86_64-advanced-d58b6e81-efb9-4f38-9059-2ff797cac59' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_gpt
    insmod xfs
    set root='hd0,gpt2'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt2 --hint-efi=hd0,gpt2 --hint-baremetal=ahci0,gpt
    else
        search --no-floppy --fs-uuid --set=root 3d7c4d34-b156-426c-b140-ccd7dd8f9246
    fi
    linuxefi /vmlinuz-3.10.0-862.el7.x86_64 root=/dev/mapper/centos-root ro crashkernel=auto rd.lvm.lv=centos/root
    initrdefi /initramfs-3.10.0-862.el7.x86_64.img
}

menuentry 'CentOS Linux (0-rescue-2f3422584f4346e68b55498c13ad1999) ? (Core)' --class centos --class gnu-linux --class gnu --class os --t
option 'gnulinux-0-rescue-2f3422584f4346e68b55498c13ad1999-advanced-d58b6e81-efb9-4f38-9059-2ff797cac59' {
    load_video
    insmod gzio
    insmod part_gpt
    insmod xfs
    set root='hd0,gpt2'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt2 --hint-efi=hd0,gpt2 --hint-baremetal=ahci0,gpt
    else
        search --no-floppy --fs-uuid --set=root 3d7c4d34-b156-426c-b140-ccd7dd8f9246
    fi
    linuxefi /vmlinuz-0-rescue-2f3422584f4346e68b55498c13ad1999 root=/dev/mapper/centos-root ro crashkernel=auto
    initrdefi /initramfs-0-rescue-2f3422584f4346e68b55498c13ad1999.img
}
### END /etc/grub.d/10_linux ###

```

一般进入正常系统的启动菜单是上面那个不带“rescue”字样的，先修改该项下的 search 后的目标分区 uuid，以及 linuxefi 后的 root=设备和 rd.lvm.lv=设备，改成新磁盘的
对比 uuid 可知这里 search 的是旧磁盘的 sda2 这个分区，所以把它改成 sdb2 的就行，下面的 linuxefi 加载内核时输入的参数 root=后是旧磁盘的 lvm 逻辑卷名，改为新磁盘的

```

fi
### END /etc/grub.d/01_users ###

### BEGIN /etc/grub.d/10_linux ###
menuentry 'CentOS Linux (3.10.0-862.el7.x86_64) ? (Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_o
option 'gnulinux-3.10.0-862.el7.x86_64-advanced-d58b6e81-efb9-4f38-9059-2ff797cac59' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_gpt
    insmod xfs
    set root='hd0,gpt2'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt2 --hint-efi=hd0,gpt2 --hint-baremetal=ahci0,gpt2 a1a1e7ca-e7fc-4349-a070-6a59108fd347
    else
        search --no-floppy --fs-uuid --set=root a1a1e7ca-e7fc-4349-a070-6a59108fd347
    fi
    linuxefi /vmlinuz-3.10.0-862.el7.x86_64 root=/dev/mapper/centos2-root ro crashkernel=auto rd.lvm.lv=centos2/root rhgb quiet LANG=en_US.L
    initrdefi /initramfs-3.10.0-862.el7.x86_64.img
}

menuentry 'CentOS Linux (0-rescue-2f3422584f4346e68b55498c13ad1999) ? (Core)' --class centos --class gnu-linux --class gnu --class os --unrestri
cted $menuentry_id_option 'gnulinux-0-rescue-2f3422584f4346e68b55498c13ad1999-advanced-d58b6e81-efb9-4f38-9059-2ff797cac59' {

```

其他的不变，先修改这一条启动菜单项吧，如果旧系统有多个启动项，要先确认之前是用的哪个启动项，若时间有限可先修改这条启动菜单，等能正常进入系统再去系统里改其他的启动菜单的 uuid
保存，退出 vi

再修改系统启动后的分区挂载配置文件，位于新磁盘的 root 根分区的 etc/fstab 文件

```
sh-4.2# vi /mnt/sdb_root/etc/fstab

# /etc/fstab
# Created by anaconda on Sat Oct 17 11:30:52 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos2-root / xfs defaults 0 0
UUID=a1a1e7ca-e7fc-4349-a070-6a59108fd347 /boot xfs defaults 0 0
UUID=6899-F9BA /boot/efi vfat umask=0077,shortname=winnt 0 0
/dev/mapper/centos2-home /home xfs defaults 0 0
```

/dev/mapper/centos2-root 挂载到/根目录

/dev/mapper/centos2-home 挂载到/home 目录

uuid 为 6899-F9BA 的是新磁盘的 sdb1 分区，efi 分区，挂载到/boot/efi

uuid 为 a1a1e7ca-xxxx 是新磁盘的 sdb2 分区，挂载到/boot 目录下

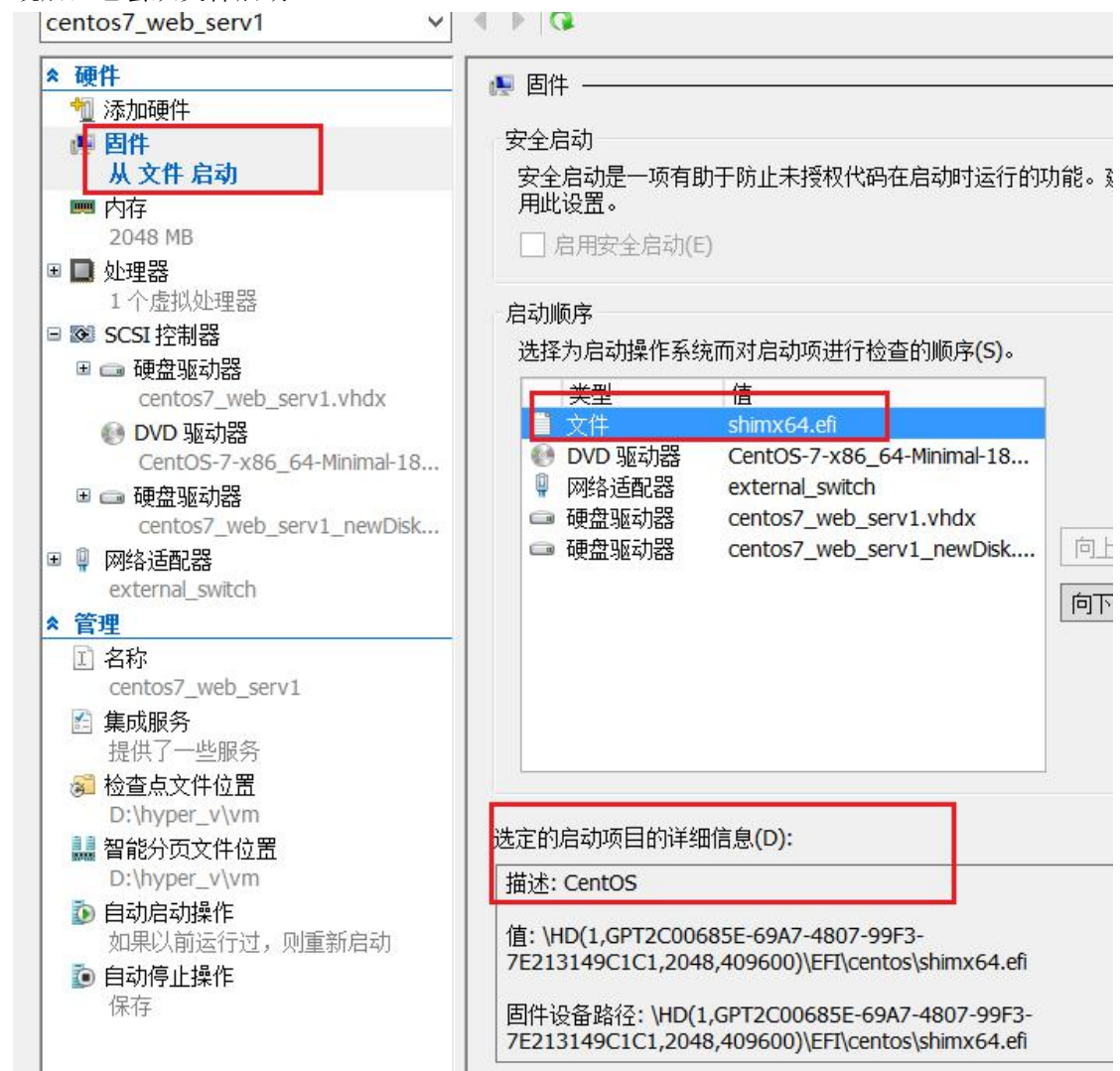
保存退出

7.创建及复制几个文件

因为原来的 centos 系统可能启用了 selinux 功能，但使用 cp 命令复制时，没能保住原文件的 selinux 标签，就算能有其他方法保留 selinux 标签，但我们刚刚也修改了 2 个配置文件，所以还是会改变原有文件的 selinux 标签，得想办法让 selinux 在系统启动时重新打标签，不然无法正常进入系统，会一直重复出现登录界面，操作方法为在新磁盘的 root 分区根目录下创建一个文件，名为 .autorelabel，前面有个点，表示隐藏文件。不需要写什么内容进去。有这个文件，selinux 就知道要干什么了。

```
sh-4.2# touch /mnt/sdb_root/.autorelabel
sh-4.2#
sh-4.2#
```

然后，因为我们目标系统是在 hyper-v 的虚拟机里，用的是第二代虚拟机，即使用 uefi 启动的，Hyper-V 的 uefi 固件比较智能（自作聪明），当我们最初创建虚拟机并安装好 centos7 系统后，它会从文件启动：



文件：shimx64.efi 或 shim.efi

从文件 shimx64.efi 启动意思是：固件自动去找目标磁盘（原来的那个 100G）的 efi 分区的/EFI/centos/shimx64.efi 文件，并从此文件启动，此 efi 文件再去引导其他文件进入最终系统

但是，我们现在换了新磁盘了，磁盘信息变了，而这个原来的从文件启动，还是找的旧磁盘，所以如果我们过一会儿卸载旧磁盘，它就找不到启动磁盘，无法进入新磁盘的系统。

所以，我们要在目标虚拟机的配置界面的“固件”里，选择从新磁盘启动，问题又来了，因为 hyper-V 太智能（自作聪明）了，它不按套路出牌，uefi 的规范是要求从启动磁盘上的 efi 分区的/EFI/BOOT/bootx64.efi 文件启动，而 hyper-v 虚拟机的配置文件里已经写了，这个目标虚拟之前是 centos7 系统，所以它又直接去找/EFI/BOOT/grubx64.efi 文件，而 centos7 系统的 efi 分区的 boot 目录下又没有这个文件，所以无法启动。

正确的做法是把新磁盘 efi 分区（本例中是 sdb1）下的/EFI/centos/grubx64.efi 复制 2 份到 /EFI/BOOT/目录下，一个命名为 grubx64.efi，另一个命名为 grub.efi

为什么要复制 2 份，因为目标虚拟机的原来的从文件启动那里，可能是从 shimx64.efi 启动，也可能是从 shim.efi 启动，所以不同的情况下，若从新磁盘启动，它就要从 grubx64.efi 或 grub.efi 这个文件启动。所以我们最好 2 个文件都有吧，也就是说：

新磁盘 efi 分区下的/EFI/BOOT/grubx64.efi 和 grub.efi 都是/EFI/centos/grubx64.efi 这个文件，只是名字不一样。

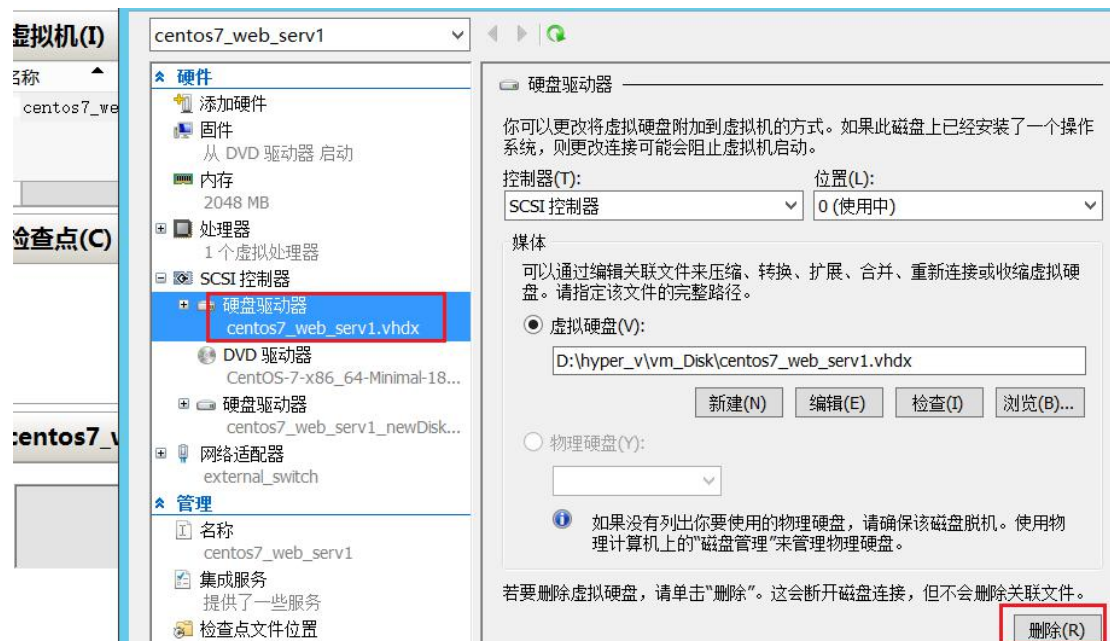
```
sh-4.2# cp /mnt/sdb1/EFI/centos/grubx64.efi /mnt/sdb1/EFI/BOOT/grubx64.efi
sh-4.2# cp /mnt/sdb1/EFI/centos/grubx64.efi /mnt/sdb1/EFI/BOOT/grub.efi
sh-4.2#
```

差不多了，可以关闭 centos7 安装光盘的救援模式了，

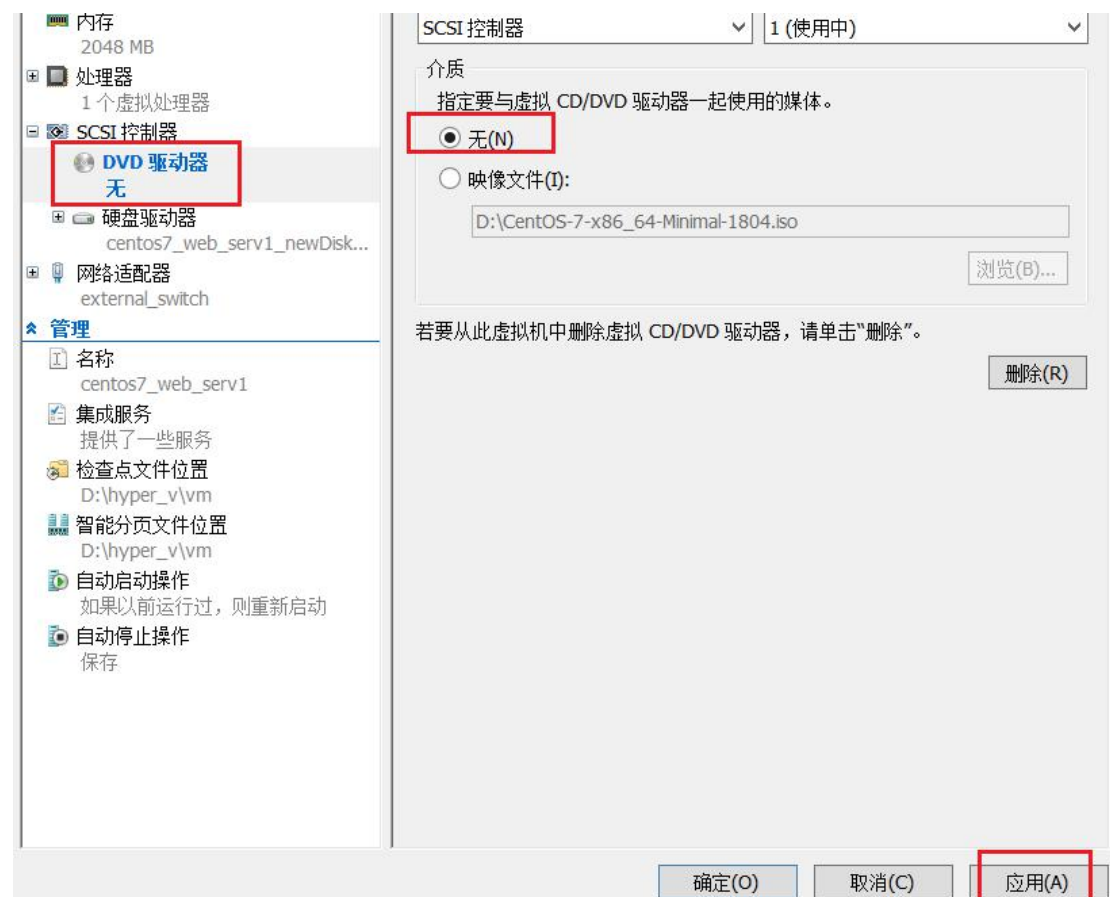
```
# init 0
```

```
sh-4.2#
sh-4.2# init 0
```

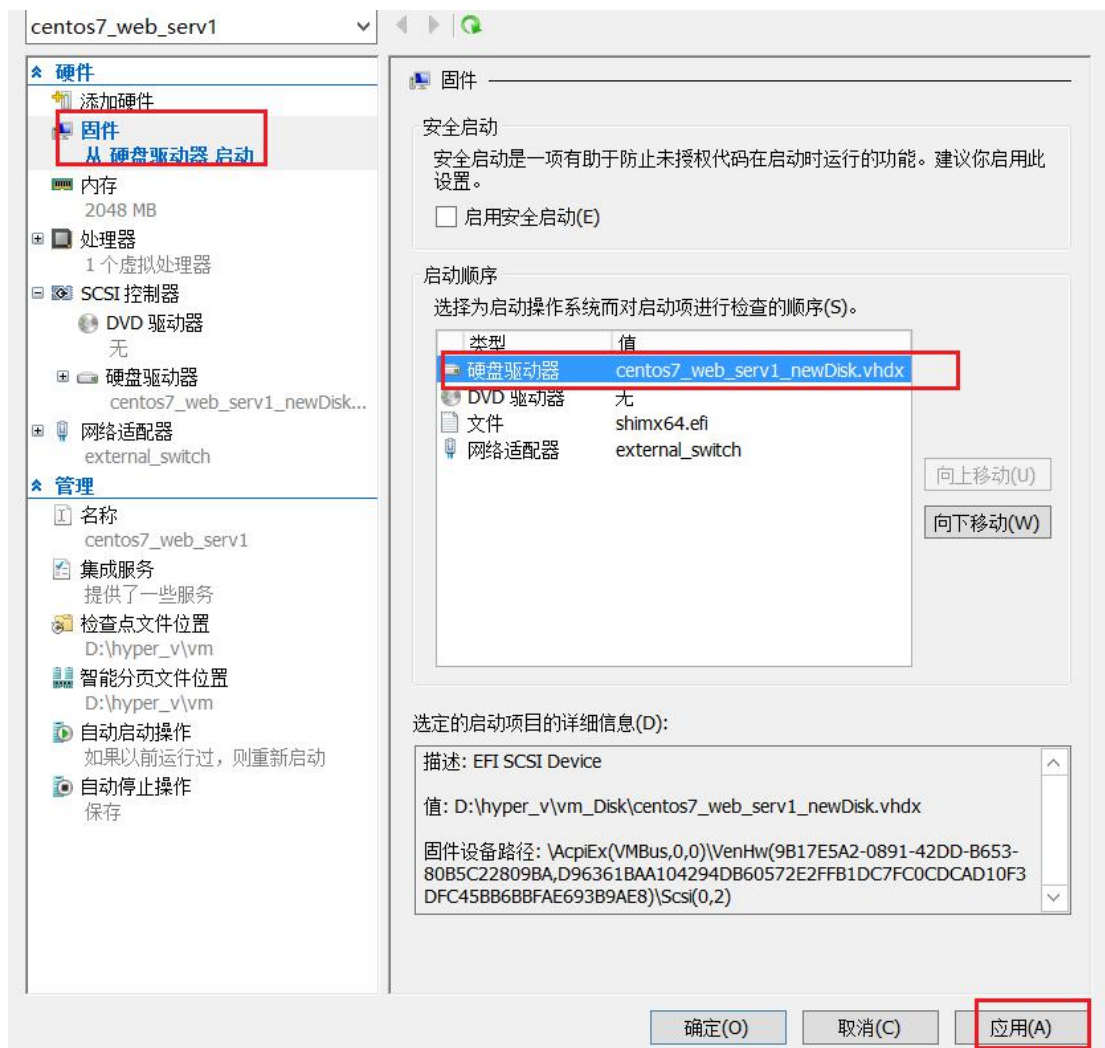
8.删除目标虚拟机旧磁盘



在目标虚拟机的配置界面，删除旧的 100GB 磁盘，这个操作只是卸载磁盘，并不会把虚拟磁盘文件从物理机上删除。

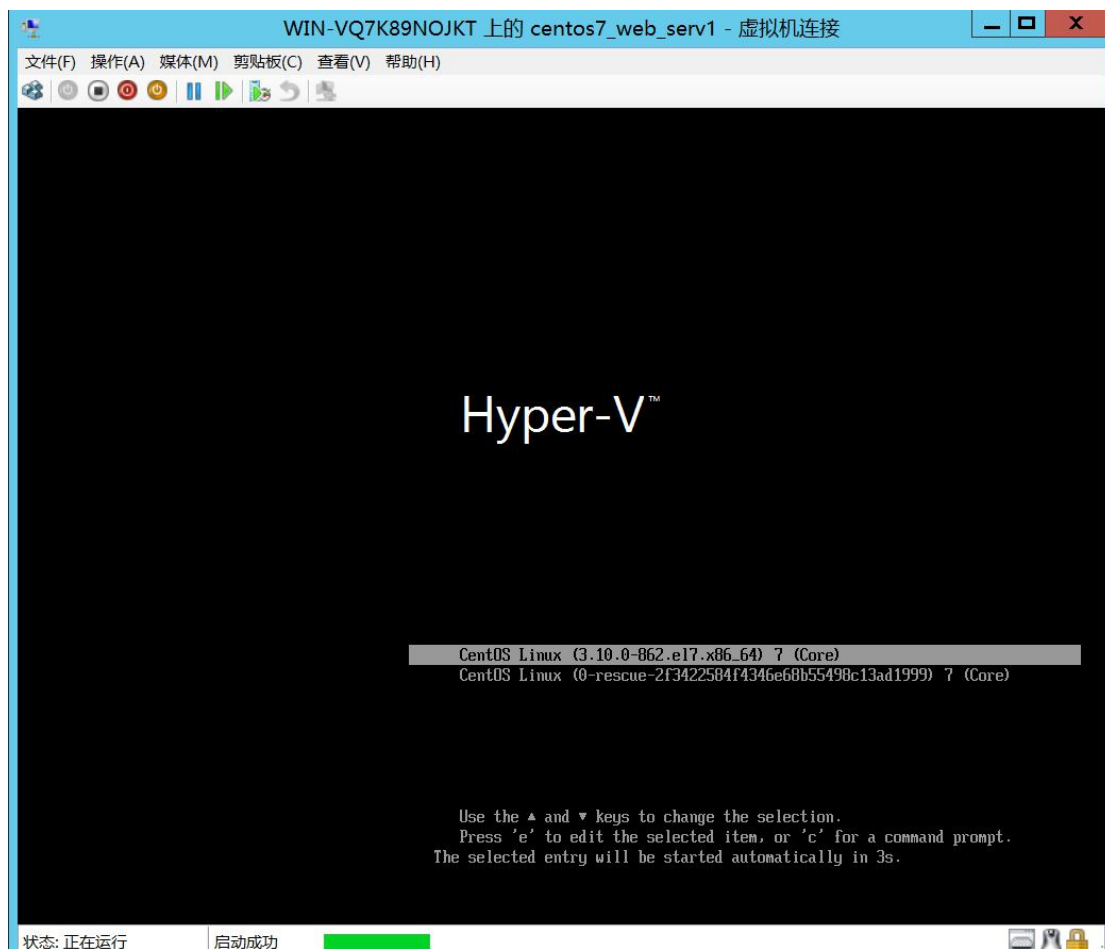


再把虚拟光驱的 iso 文件也卸载了，点击“应用”

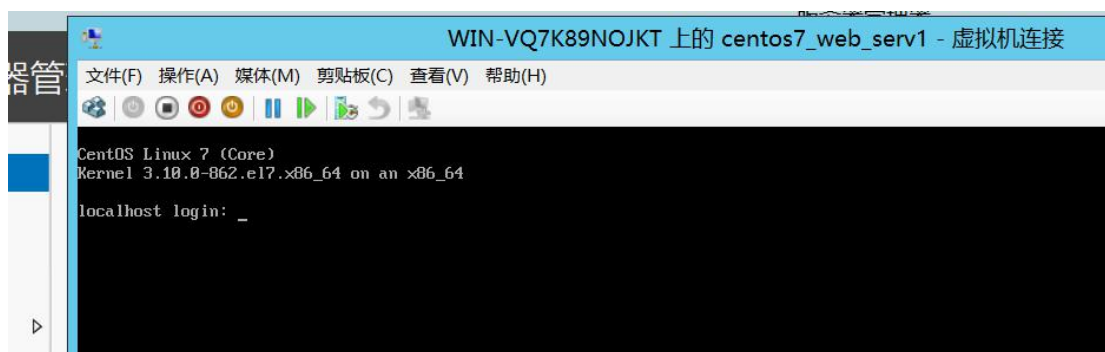


再点击左侧的“固件”，把新磁盘移到启动顺序的顶端，点击“应用”，确定

最后，双击目标虚拟机，点击开机按键，激动人心的时刻到了：



出现上图这个启动菜单项，就说明从新磁盘启动了，回车，顺利的话，它会在将要进入系统时停留一会儿，再重新启动（停留一会实际是已经启动了，只是 selinux 要重新给文件打标签，再重启）



输入用户名和密码，

```

CentOS Linux 7 (Core)
Kernel 3.10.0-862.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Sat Oct 17 12:36:01 on tty1
[root@localhost ~]#
[root@localhost ~]# df -Th
Filesystem                Type      Size  Used Avail Use% Mounted on
/dev/mapper/centos2-root  xfs       7.0G  1015M  6.0G   15% /
devtmpfs                  devtmpfs  903M    0   903M    0% /dev
tmpfs                     tmpfs     914M    0   914M    0% /dev/shm
tmpfs                     tmpfs     914M  8.6M   906M    1% /run
tmpfs                     tmpfs     914M    0   914M    0% /sys/fs/cgroup
/dev/sda2                 xfs       494M  127M  368M   26% /boot
/dev/sda1                 vfat      200M   12M  189M    6% /boot/efi
/dev/mapper/centos2-home  xfs       2.0G   33M   2.0G    2% /home
tmpfs                     tmpfs     183M    0   183M    0% /run/user/0
[root@localhost ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   10G  0 disk
├─sda1       8:1    0   200M  0 part /boot/efi
├─sda2       8:2    0   500M  0 part /boot
├─sda3       8:3    0   9.3G  0 part
├─centos2-root 253:0    0    7G  0 lvm  /
└─centos2-home 253:1    0    2G  0 lvm  /home
sr0         11:0    1 1024M  0 rom

```

great，成功进入系统，确定目前是磁盘是新的 10G 磁盘。

然后，可以再关机，再回到目标虚拟机的设置界面，把固件的启动顺序改回原来的从文件 shimx64.efi 启动。因为刚刚系统成功启动了，所以这个 hyper-v 又智能地识别到了新的磁盘的信息，又能从文件启动了。

不改启动顺序也行，就从新磁盘启动。都可以

启动系统后，再启动相关的服务就行了

常见问题：

如果无法进入启动界面，是因为 grub2 的配置文件写错了，重新挂载 iso 文件进入救援模式，修改新磁盘的 grub 配置文件，要和之前的截图对比一下，确保不写错 uuid 和 Lvm 卷组名和逻辑卷名。还有系统挂载配置文件/etc/fstab 文件也一样要检查一下。

如果进入启动界面，却一直要求重新输入用户名和密码，一定是没创建根目录下的 .autorelabel 文件，或者文件名搞错了，也是进入救援模式，重新创建该文件。

如果以下 2 步都检查了，确定没问题，那就是自己的基础薄弱了，可以学习一下 grub2 引导和它的配置命令以及 uefi 的相关知识。

作者：Cof-Lee

日期：2020-10-18