NSD SERVICES DAY01

1 案例1：安装一个KVM服务器

1.1 问题

本例要求准备一台 RHEL7.2 服务器，将其搭建为KVM平台，主要完成下列操作：

1. 关闭本机的SELinux保护、防火墙服务
2. 挂载RHEL7光盘到 /mnt/dvd，将其配置为本机YUM源（baseurl = file:///mnt/dvd）
3. 安装KVM相关包组，确保已启用 libvirtd 服务

1.2 方案

RHEL7中的虚拟化服务软件组：

- 虚拟化平台 —— "Virtualization Platform"
- 虚拟化主机 —— "Virtualization Host"
- 虚拟化客户端 —— "Virtualization Client"

另外需要注意，yum命令的软件组管理操作与RHEL6有些小变化：

- yum groups list [hidden] [组名]...
- yum groups info [组名]...
- yum groups install [组名]...
- yum groups remove [组名]...

Top

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置本地yum仓库

在光驱中插入RHEL7.2的系统光盘（若是虚拟机则连接相应ISO文件），然后在系统中将光盘挂载到/mnt目录。

```
[root@kvmsvr ~]# mkdir  /mnt/dvd                    //创建挂载点
[root@kvmsvr ~]# vim  /etc/fstab                    //添加开机挂载配置
.. ..
/dev/cdrom    /mnt/dvd     iso9660      ro      0 0
[root@kvmsvr ~]# mount  -a                          //根据fstab配置挂载光盘
mount:  /dev/sr0 is write-protected, mounting read-only
```

2）确认/mnt/dvd访问点

```
[root@kvmsvr ~]# ls  /mnt/dvd/Packages/*.rpm        //确认软件包位置
.. ..
/mnt/Packages/zsh-5.0.2-14.el7.x86_64.rpm
/mnt/Packages/zziplib-0.13.62-5.el7.i686.rpm
/mnt/Packages/zziplib-0.13.62-5.el7.x86_64.rpm
```

3）将本地目录/mnt/dvd配置为本机的yum源

Top

```
[root@kvmsvr ~]# yum-config-manager  --add-repo  file:///mnt/dvd     //添加源
.. ..
```

```
[root@kvmsvr ~]# vim /etc/yum.repos.d/mnt_dvd.repo
[mnt_dvd]
name=added from: file:///mnt/dvd
baseurl=file:///mnt/dvd
enabled=1
gpgcheck=0                                  //禁止GPG检查

[root@kvmsvr ~]# yum repolist                       //列出可用仓库
mnt_dvd                          | 4.1 kB     00:00
(1/2): mnt_dvd/group_gz          | 136 kB     00:00
(2/2): mnt_dvd/primary_db        | 3.6 MB     00:00
repo id          repo name                      status
mnt_dvd          added from: file:///mnt/dvd    4620
repolist: 4620
```

步骤二：确认RHEL7中的虚拟化软件组

1）安装兼容组信息

```
[root@kvmsvr ~]# yum groups mark convert
... ...
There is no installed groups file.
Maybe run: yum groups mark convert (see man yum)
Converted old style groups to objects.
```

Top

2）查看全部软件组，过滤出与虚拟化相关的软件组

```
[root@kvmsvr ~]# yum groups list hidden| grep -i virt
   Virtualization Host
   Virtualization Client
   Virtualization Hypervisor
   Virtualization Tools
   Virtualization Platform
```

其中主要的KVM软件组包括Virtualization Host、Virtualization Client、Virtualization Platform，其他两个组会由于依赖关系自动被关联。

步骤三：安装KVM虚拟化

1）安装主要KVM软件组

```
[root@pc207 ~]# yum -y groups install "Virtualization Host" "Virtualization Client" "Virtualization Platform"
... ...
```

2）确保libvirtd服务可用

```
[root@kvmsvr ~]# systemctl restart libvirtd
[root@kvmsvr ~]# systemctl enable libvirtd
```

3)确保虚拟系统管理器（virt-manager）可用

在KVM服务器的桌面环境中，可以通过“应用程序”菜单组找到“系统工具” --> “虚拟系统管理器”快捷方式（如图-1所示）。

Top

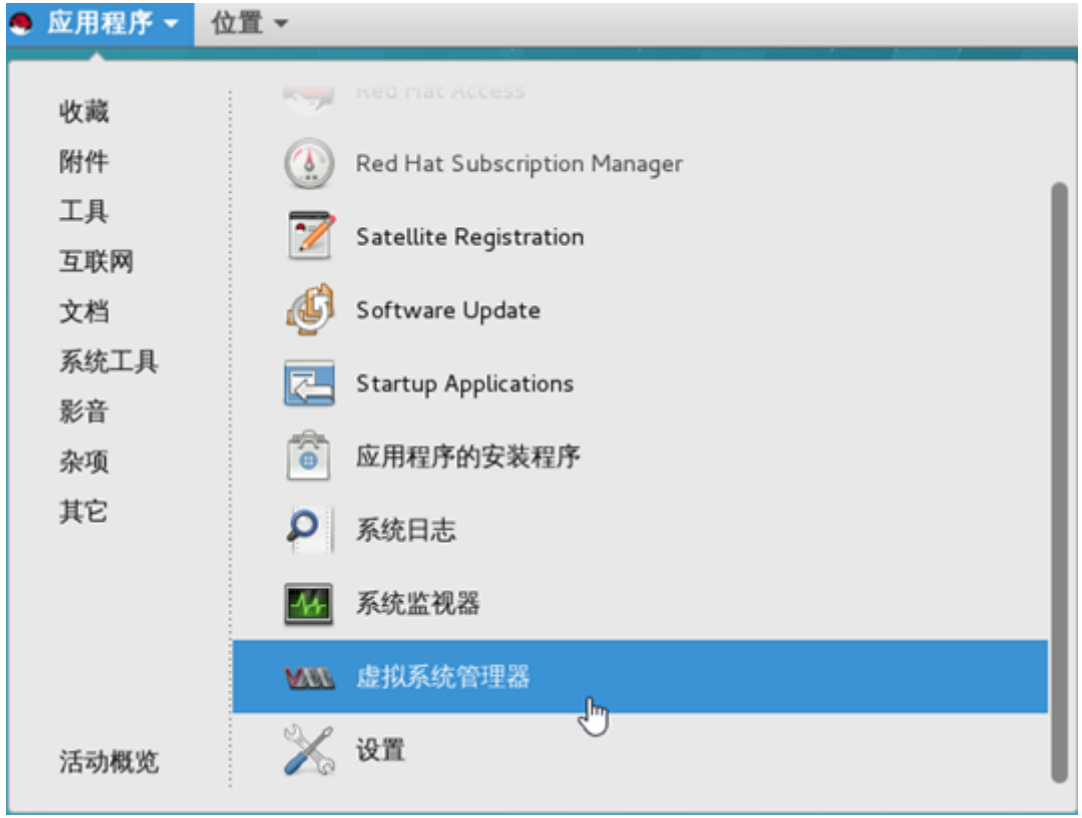

图-1

单击可以成功开启KVM管理工具（如图-2所示），可以看到还没有任何虚拟机。

Top

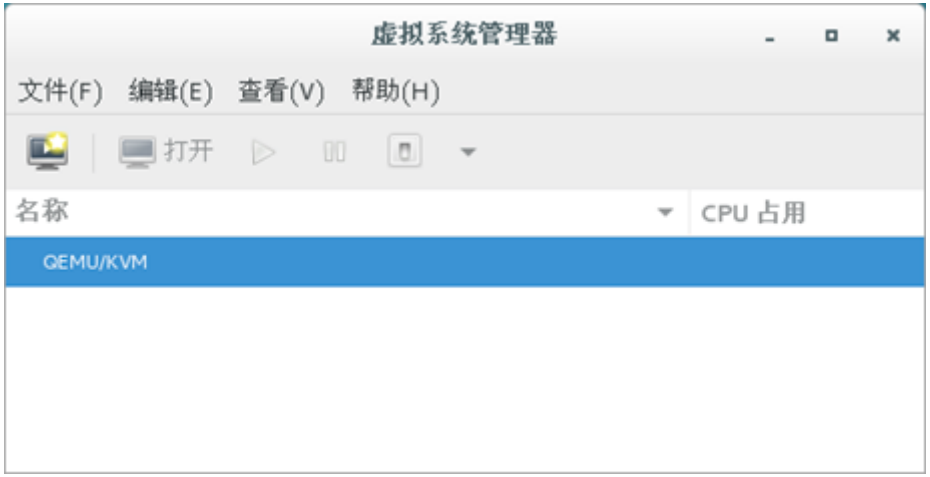

图-2

2 案例2：KVM平台构建及简单管理

2.1 问题

本例要求在真实KVM服务器上完成以下任务：

1. 新建一个名为 rhel7.2 的虚拟机，并为其安装好操作系统（注意禁用SELinux机制、禁用防火墙）
2. 将虚拟机 rhel7.2 克隆为 rhel7-c1
3. 开启虚拟机 rhel7-c1 ，以 root 用户登入到系统
4. 彻底删除虚拟机 rhel7-c1

2.2 方案

使用KVM提供的virt-manager图形化管理程序来操作。

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：新建名为rhel7.2的虚拟机

Top

1）在“虚拟系统管理器”中单击左上方“创建新虚拟机”按钮（如图-3所示）。

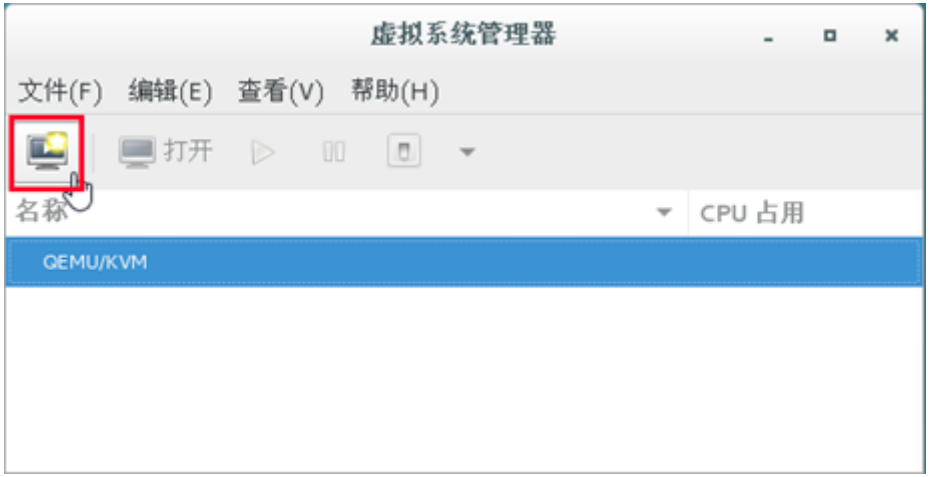

图-3

2）弹出“新建虚拟机”向导，选择“本地安装介质”（如图-4所示），单击“前进”。

图-4

Top

3）接下来“定位安装介质”，请正确指定RHEL7系统的ISO光盘镜像文件位置（如图-5所示），确认自动识别到操作系统类型，单击“前进”。

图-5

4）选择内存和CPU设置，建议为虚拟机分配内存不小于1024MB（如图-6所示），单击“前进”。

Top

图-6

5）为虚拟机启用存储，例如分配一个40GiB的磁盘（如图-7所示），单击“前进”。

Top

图-7

6）为虚拟机命名，设为rhel7.2（如图-8所示），单击“完成”。

Top

图-8

步骤二：为虚拟机rhel7.2安装操作系统

1）确认新虚拟机从光盘引导

上述设置全部完成后，将会自动开启新虚拟机，并进入安装过程（如图-9所示）。

Top

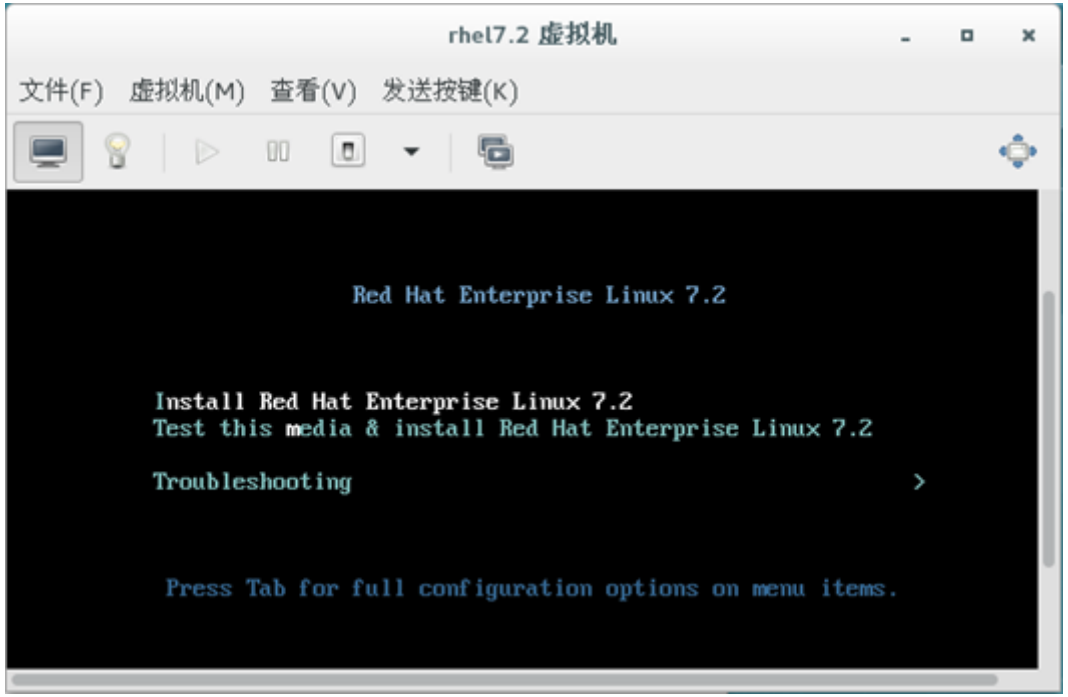

图-9

2）完成后续手动安装过程

具体过程与普通安装相同，注意禁用防火墙、禁用SELinux机制。

3）确认安装结果

新装的虚拟机rhel7.2可以正常启动、登录。

在“虚拟系统管理器”中也能够看到此虚拟机（如图-10所示）。

Top

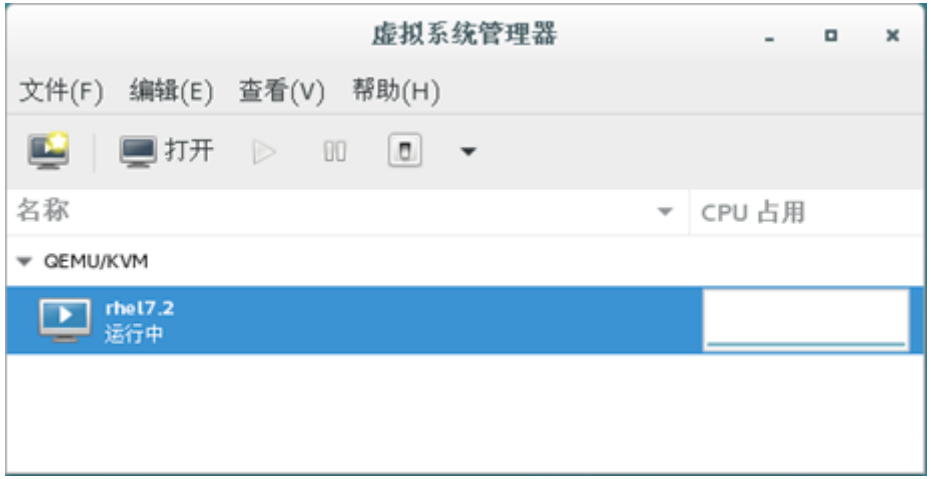

图-10

步骤三：克隆虚拟机并验证

1）将被克隆的虚拟机rhel7.2关机

右击选中的虚拟机rhel7.2，选择“关机”-->“强制关机”（如图-11所示），根据提示确认即可。

图-11

Top

2）执行克隆操作

右击已关闭的虚拟机rhel7.2，选择“克隆”，在弹出对话框中指定名称，确认默认设置（如图-12所示），然后单击右下角的“克隆”按钮完成操作。

图-12

3）确认克隆结果

新克隆的虚拟机rhel7.2-c1可以正常启动、登录，可以独立运行。

在“虚拟系统管理器”中也能够看到此虚拟机（图-13所示）。

Top

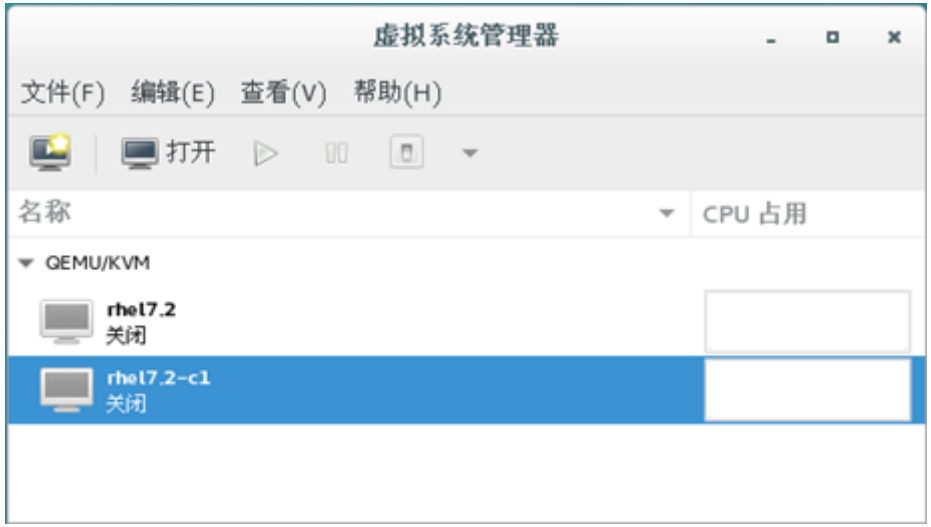

图-13

步骤四：删除指定的虚拟机

1）删除虚拟机rhel7.2-c1

将虚拟机rhel7.2-c1关机，右击选择“删除”，在弹出窗口中根据需要确认是否删除虚拟机的磁盘，单击右下角的“删除”按钮即可（如图-14所示）。

Top

图-14

2）确认删除结果

在“虚拟系统管理器”界面中，虚拟机rhel7.2-c1已经没有了（如图-15所示）。

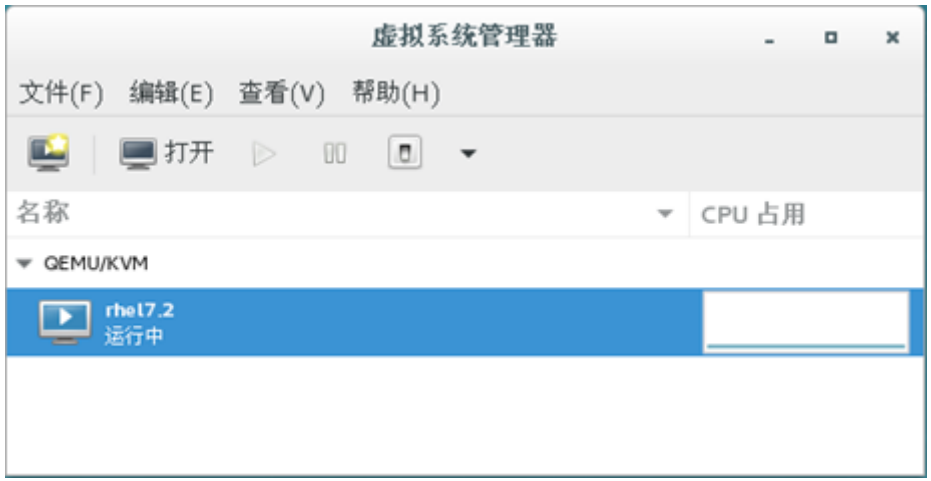

图-15

Top

3 案例3：virsh基本管理操作

3.1 问题

本例要求学会使用virsh命令管理KVM虚拟机，主要完成下列任务：

1. 查看当前KVM服务器的内存/CPU
2. 列出有哪些虚拟机、查看各虚拟机的状态
3. 启动/重启/关机/强制关机操作
4. 设置虚拟机开机自动运行

3.2 方案

virsh命令提供了用来管理各虚拟机的命令接口，支持交互模式，可以实现对虚拟机的查看/创建/停止/关闭等各种操作。

用法参考：

```
virsh 控制指令 [虚拟机名称] [参数]
```

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：查看KVM服务器及虚拟机信息

1）查看KVM服务器节点

```
[root@kvmsvr ~]# virsh nodeinfo
CPU 型号：          x86_64
CPU：               4
CPU 频率：          2600 MHz
CPU socket：        1
每个 socket 的内核数： 4
```

Top

```
每个内核的线程数： 1
NUMA 单元：       1
内存大小：       16230564 KiB
```

2）列出有哪些虚拟机（包括未开启的）及各自的状态

```
[root@kvmsvr ~]# virsh list --all
 Id    Name                           State
----------------------------------------------------
 -     rhel7.2                        shut off
```

3）查看指定虚拟机rhel7.2的配置摘要信息

```
[root@kvmsvr ~]# virsh dominfo rhel7.2
Id:             -
Name:           rhel7.2
UUID:           207a2b25-fd0f-436e-81ae-ad0fa8861315
OS Type:        hvm
State:          shut off
CPU(s):         1
Max memory:     1000448 KiB
Used memory:    0 KiB
Persistent:     yes
Autostart:      disable
Managed save:   no
Security model: selinux
```

Top

```
14.     Security DOI:  0
```

步骤二：虚拟机开关机操作

1）将虚拟机rhel7.2开启

```
01.  [root@kvmsvr ~]# virsh  start  rhel7.2                    //开机
02.  Domain rhel7.2 started
03.
04.  [root@kvmsvr ~]# virsh  list                             //检查结果
05.   Id    Name                           State
06.  ----------------------------------------------------
07.   5     rhel7.2                        running
```

2）将虚拟机rhel7.2关机

```
01.  [root@kvmsvr ~]# virsh  shutdown  rhel7.2                 //关机
02.  Domain rhel7.2 is being shutdown
03.  .. .. //稍等片刻
04.  [root@kvmsvr ~]# virsh  list  --all                      //检查结果
05.   Id    Name                           State
06.  ----------------------------------------------------
07.   -     rhel7.2                        shut off
```

Top

3）将虚拟机rhel7.2强制关机（shutdown无效时适用）

```
[root@kvmsvr ~]# virsh destroy rhel7.2                    //强制关机
Domain rhel7.2 destroyed

[root@kvmsvr ~]# virsh list --all                         //检查结果
 Id    Name                           State
----------------------------------------------------
 -     rhel7.2                        shut off
```

步骤三：虚拟机自启设置

1）将虚拟机rel7.2设为自动启动

```
[root@kvmsvr ~]# virsh autostart rhel7.2                  //设置自启动
Domain rhel7.2 marked as autostarted

[root@kvmsvr ~]# virsh dominfo rhel7.2                    //确认结果
Id:             -
Name:           rhel7.2
...
Autostart:      enable
...
```

2）将虚拟机rel7.2取消自动启动

```
[root@kvmsvr ~]# virsh autostart --disable rhel7.2        //取消自启动
Domain rhel7.2 unmarked as autostarted
```

Top

```
03.
04.     [root@kvmsvr ~]# virsh  dominfo  rhel7.2                    //确认结果
05.     Id:             -
06.     Name:           rhel7.2
07.     .. ..
08.     Autostart:      disable
09.     .. ..
```

4 案例4：xml配置文件的应用

4.1 问题

本例要求在KVM服务器上通过使用xml文件完成下列任务：

1. 将虚拟机 rhel7.2 改名为 rhel-207
2. 将虚拟机 rhel-207 复制为 rhel-7
3. 上述虚拟机的CPU/内存/网络类型保持不变
4. 但这2个虚拟机有可能会同时运行，不应出现冲突

4.2 方案

KVM虚拟机的xml配置文件也就是通常所说的虚拟机的描述文件，主要用来定义一个虚拟机的名称、UUID、CPU、内存、虚拟磁盘、网卡等各种参数设置。

KVM虚拟机的xml配置文件默认位于： /etc/libvirt/qemu/虚拟机名.xml 。

修改虚拟机配置的基本思路：

1. 编辑虚拟机配置：virsh edit 虚拟机名
2. 根据需要修改，保存配置结果

4.3 步骤

Top

实现此案例需要按照如下步骤进行。

步骤一：将虚拟机rhel7.2改名为rhel-207

1）生成新的UUID字串，并复制备用

```
[root@kvmsvr ~]# uuidgen
76d5dc2c-5eef-4e30-8b6c-e58851814f84
```

2）编辑虚拟机rhel7.2的配置

调整name、uuid值，保存修改：

```
[root@kvmsvr ~]# virsh edit rhel7.2
<domain type='kvm'>
  <name>rhel-207</name>                                  //新名称
  <uuid>76d5dc2c-5eef-4e30-8b6c-e58851814f84</uuid>      //新UUID值
.. ..
```

3）确认已自动导入的新配置

```
[root@kvmsvr ~]# virsh list --all
 Id    Name                           State
----------------------------------------------------
 -     rhel-207                       shut off        //新名称的虚拟机
 -     rhel7.2                        shut off
```

Top

4）删除旧名称的虚拟机配置

```
[root@kvmsvr ~]# virsh undefine rhel7.2                    //取消定义虚拟机
Domain rhel7.2 has been undefined

[root@kvmsvr ~]# virsh list --all                          //确认结果
 Id    Name                           State
----------------------------------------------------
 -     rhel-207                       shut off
```

步骤二：将虚拟机rhel-207复制为rhel-7

1）生成新UUID，并复制备用

```
[root@kvmsvr ~]# uuidgen
90908905-bde4-4c4a-90b0-8a8f5bba1e25
```

2）修改导出后的配置（调整名称、UUID、磁盘路径、网卡MAC）

修改导出的xml配置文件，调整name、uuid、disk路径、mac地址值：

```
[root@kvmsvr ~]# virsh edit rhel-207
<domain type='kvm'>
  <name>rhel-7</name>                                      //新名称
  <uuid>90908905-bde4-4c4a-90b0-8a8f5bba1e25</uuid>        //新UUID值
  ...
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2'/>
      <source file='/var/lib/libvirt/images/rhel-7.qcow2'/>    //新磁盘路径
```

Top

```
09.          .. ..
10.        </disk>
11.    .. ..
12.        <interface type='network'>
13.          <mac address='52:54:00:91:50:07' />                    //新MAC地址
14.          <source network='default' />
15.          <model type='virtio' />
16.          <address type='pci' domain='0x0000' bus='0x00' slot='0x03
17.    ' function='0x0' />
18.        </interface>
19.    .. ..
```

3）确认已自动导入的新配置

```
01.    [root@kvmsvr ~]# virsh  list  --all                    //确认结果
02.     Id    Name                           State
03.    ----------------------------------------------------
04.     -     rhel-207                       shut off
05.     -     rhel-7                         shut off              //新虚拟机
```

4）复制虚拟机磁盘文件

为新虚拟机提供一份独立的磁盘文件：

```
01.    [root@kvmsvr ~]# cd  /var/lib/libvirt/images/
02.    [root@kvmsvr images]# cp  rhel7.2.qcow2  rhel-7.qcow2
```

Top

```
[root@kvmsvr images]# ls                    //确认结果
rhel-7.qcow2  rhel7.2.qcow2
```

5 案例5：为虚拟机制作快照备份

5.1 问题

本例要求使用 qemu-img 及必要的工具，完成下列任务：

1. 将虚拟机 rhel-7 关机
2. 为虚拟机 rhel-7 的磁盘制作名为 snap1 的快照
3. 开启并登入虚拟机 rhel-7，在桌面上新建文件 1.txt
4. 再次关闭虚拟机 rhel-7，还原到快照 snap1
5. 重新开启并登入虚拟机 rhel-7，检查 1.txt 文件

5.2 方案

KVM虚拟机的快照：通过在虚拟机磁盘镜像内保存不同时间点的状态数据实现备份，在必要时可将虚拟机恢复到指定的快照状态。

qemu-img快照管理基本操作：

- 创建快照：qemu-img snapshot -c 快照名 qcow2磁盘
- 列出快照：qemu-img snapshot -l qcow2磁盘
- 恢复快照：qemu-img snapshot -a 快照名 qcow2磁盘
- 删除快照：qemu-img snapshot -d 快照名 qcow2磁盘

5.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：为虚拟机制作快照

1）确保虚拟机rhel-7已经关机

Top

```
[root@kvmsvr ~]# virsh list --all | grep rhel-7
```

```
02.      -      rhel-7                    shut off
```

2）为虚拟机rhel-7的磁盘制作快照

```
01.  [root@kvmsvr ~]# cd  /var/lib/libvirt/images/
02.  [root@kvmsvr images]# qemu-img  snapshot  -c snap1  rhel-7.qcow2    //制作快照
03.  [root@kvmsvr images]# qemu-img  snapshot  -l  rhel-7.qcow2          //列出快照
04.  Snapshot list:
05.  ID       TAG              VM SIZE              DATE      VM CLOCK
06.  1        snap1                0 2017-01-05 15:44:25   00:00:00.000
```

步骤二：正常使用/更新虚拟机

1）启动虚拟机rhel-7

```
01.  [root@kvmsvr ~]# virsh start rhel-7
02.  Domain rhel-7 started
```

2）正常登入虚拟机rhel-7，在桌面建立文件1.txt

过程略。

步骤三：还原快照并检查恢复结果

1）关闭虚拟机rhel-7

Top

```
01.  [root@kvmsvr ~]# virsh  destroy  rhel-7
```

```
02.  Domain rhel-7 destroyed
```

2）将虚拟机rhel-7的磁盘还原到快照snap1

```
01.  [root@kvmsvr images]# qemu-img snapshot -a snap1 rhel-7.qcow2
```

3）重新开启虚拟机rhel-7

```
01.  [root@kvmsvr ~]# virsh start rhel-7
02.  Domain rhel-7 started
```

4）正常登入虚拟机rhel-7，检查桌面的文件1.txt

因为此文件是在建快照之后才建立的，所以还原快照以后就没有了。

6 案例6：快建新虚拟机

6.1 问题

本例要求利用qcow2磁盘特性快建2台新的KVM虚拟机，配置要求如下：

1. svr7：svr7.tedu.cn，192.168.4.7/24
2. pc207： pc207.tedu.cn，192.168.4.207/24
3. 为上述虚拟机配好网络，确认yum源可用
4. 从CentOS真机可ssh远程访问这两台虚拟机

6.2 方案

Top

快建新虚拟机的基本思路：

1. 提前准备好一台模板虚拟机（镜像磁盘+xml配置文件）
2. 基于qcow2磁盘复用技术快建新虚拟机的磁盘
3. 通过调整模板机的配置快建新虚拟机的xml配置文件
4. 导入新虚拟机

Copy On Write，写时复制技术原理：

- 直接映射原始盘的数据内容
- 当原始盘有修改时，在修改之前将旧数据存入前端盘
- 对前端盘的修改不会回写到原始盘

6.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：准备模板虚拟机

找一台已经装好RHEL7系统、配置好本地yum源、关闭SELinux的虚拟机，提取模板虚拟机磁盘、xml配置文件备用。

1）准备磁盘目录、模板虚拟机磁盘文件

```
[root@kvmsvr ~]# qemu-img info  /data/images/rhel7_muban.qcow2
image: /data/images/rhel7_muban.qcow2
file format: qcow2
virtual size: 300G (322122547200 bytes)                    //虚拟机磁盘容量
disk size: 3.2G                                            //在KVM服务器占用容量
```

2）准备模板虚拟机配置文件

```
[root@kvmsvr ~]# cat  /data/images/rhel7_muban.xml
<domain type='kvm'>
  <name>rhel7.2</name>
```

Top

```
    <uuid>a1992150-5cc7-e19d-20df-cd5cea7d8aa2</uuid>
    <memory unit='KiB'>2097152</memory>
  ...
```

步骤二：快建虚拟机svr7

1）为虚拟机svr7快速建立前端盘（复用模板机的磁盘数据）

```
[root@kvmsvr ~]# qemu-img create -f qcow2 -b /data/images/rhel7_muban.qcow2 /data/images/svr7.qcow2
Formatting '/data/images/svr7.qcow2', fmt=qcow2 size=322122547200 backing_file='/data/images/rhel7_muban.qcow2' encryption=off cluster_s
```

2）为虚拟机svr7准备xml配置

```
[root@kvmsvr ~]# cp /data/images/rhel7_muban.xml /tmp/svr7.xml    //拷贝配置
[root@kvmsvr ~]# vim /tmp/svr7.xml                                //修改配置
<domain type='kvm'>
  <name>svr7</name>                                               //新名称
  <uuid>b20a1a1c-a2de-4b2f-bb03-91a3e36257c7</uuid>               //新UUID值
...
    <channel type='unix'>
      <source mode='bind' path='/var/lib/libvirt/qemu/channel/t
arget/domain-rhel-7/org.qemu.guest_agent.0'/>                     //改套接字路径
      ...
    </channel>

    <disk type='file' device='disk'>
```

Top

```
14.          <driver name='qemu' type='qcow2'/>
15.          <source file='/var/lib/libvirt/images/svr7.qcow2'/>        //新磁盘路径
16.          .. ..
17.        </disk>
18.    .. ..
19.        <interface type='network'>
20.          <mac address='52:54:00:11:00:07'/>                        //新MAC地址
21.          <source network='default'/>
22.          <model type='virtio'/>
23.          <address type='pci' domain='0x0000' bus='0x00' slot='0x03
24.    ' function='0x0'/>
25.        </interface>
26.    .. ..
```

3）定义新虚拟机svr7

```
01.    [root@room9pc00 ~]# virsh define /tmp/svr7.xml
02.    定义域 svr7（从 /tmp/svr7.xml）
```

4）为虚拟机svr7配置主机名、IP地址

开启虚拟机svr7，使用root用户登入系统。

```
01.    [root@svr7 ~]# vim /etc/hostname                    //配置主机名
02.    svr7.tedu.cn
03.    [root@svr7 ~]# nmcli connection modify eth0 ipv4.method manual ipv4.addresses 192.168.4.7/24 connection.autoconnect yes      //配置
```

Top

```
[root@svr7 ~]# nmcli connection up eth0          //激活连接
成功激活的连接（D-Bus 激活路径：/org/freedesktop/NetworkManager/ActiveConnection/8）

[root@svr7 ~]# ifconfig eth0                     //确认配置结果
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.4.7  netmask 255.255.255.0  broadcast 192.168.4.255
        inet6 fe80::20c:29ff:fe5e:f686  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:11:00:07  txqueuelen 1000  (Ethernet)
        RX packets 112143  bytes 9388455 (8.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 507844  bytes 771354289 (735.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

步骤三：快建虚拟机pc207

操作与步骤二类似，此处略。

步骤四：快建虚拟机的运行测试

1）启动两台虚拟机svr7、pc207，均可正常运行

2）从svr7可ping通pc207

```
[root@svr7 ~]# ping 192.168.4.207
PING 192.168.4.207 (192.168.4.207) 56(84) bytes of data.
64 bytes from 192.168.4.207: icmp_seq=1 ttl=64 time=0.392 ms
64 bytes from 192.168.4.207: icmp_seq=2 ttl=64 time=0.369 ms
...
```

Top

3）从CentOS真机可ssh远程访问这两台虚拟机

```
[root@room9pc00 ~]# ssh -X root@192.168.4.7
root@192.168.4.7's password:                    //验证svr7的root密码
Last login: Sun Mar 26 11:30:35 2017 from 192.168.4.254
[root@svr7 ~]# exit
[root@room9pc00 ~]# ssh -X root@192.168.4.207
root@192.168.4.207's password:                  //验证pc207的root密码
Last login: Sun Mar 26 11:32:35 2017 from 192.168.4.254
[root@pc207 ~]# exit
[root@room9pc00 ~]#
```

Top