

# NSD CLOUD DAY01

1. [案例1：virsh基本管理操作](#)
2. [案例2：qemu-img基本操作管理](#)
3. [案例3：创建一个虚拟网络](#)
4. [案例4：xml管理](#)
5. [案例5：安装虚拟机](#)
6. [案例6：离线访问虚拟机问题](#)

## 1 案例1：virsh基本管理操作

### 1.1 问题

本案例要求熟悉virsh的基本操作，可以熟练运用：

- 列出当前正在运行的虚拟机
- 查看虚拟机的信息
- 管理虚拟机
- 设置虚拟机开机自动运行

### 1.2 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：virsh基本操作

1) 列出当前正在运行的虚拟机

```
01. [root@room9pc01 ~]# virsh list
02. Id   Name                               State
03. ----
04. 1    node1                             running
```

2) 查看虚拟机的信息

```
01. [root@room9pc01 ~]# virsh dominfo node1           //查看node1的信息
02. Id: 1
03. Name: node1
04. UUID: 20e15d2f-ea30-4aa3-96dc-91aab6283b10
05. OS Type: hvm
06. State: running
07. CPU(s): 2
08. CPU time: 92.8s
09. Max memory: 2048000 KiB
```

[Top](#)

10. Used memory : 2048000 KiB
11. Persistent: yes
12. Autostart: disable
13. Managed save: no
14. Security model: none
15. Security DOI: 0

## 步骤二：管理虚拟机

### 1) 启动虚拟机

```
01 [root@room9pc01 ~]# virsh start node1
```

### 2) 重启虚拟机

```
01 [root@room9pc01 ~]# virsh reboot node1
```

### 3) 强制关闭虚拟机

```
01 [root@room9pc01 ~]# virsh destroy node1
```

### 4) 设置虚拟机开机自动运行

```
01 [root@room9pc01 ~]# virsh autostart node1
```

## 2 案例2：qemu-img基本操作管理

### 2.1 问题

本案例要求：

- 创建一个新的镜像盘文件
- 使用后端模板文件创建一个新的镜像盘文件
- 查看镜像盘文件的信息

### 2.2 步骤

实现此案例需要按照如下步骤进行。

[Top](#)

#### 步骤一：创建一个新的镜像盘文件

qemu-img命令格式：qemu-img 命令 参数 块文件名称 大小

01. [root@room9pc01 ~]# qemu-img create -f qcow2 disk.img 50G //qcow2为创建的格式
02. Formatting 'disk.img', fmt=qcow2 size=53687091200 encryption=off cluster\_size=65536 laz

## 2) 使用后端模板文件创建一个新的镜像盘文件

备注：-b使用后端模板文件

01. [root@room9pc01 ~]# qemu-img create -b disk.img -f qcow2 disk1.img
02. Formatting 'disk1.img', fmt=qcow2 size=53687091200 backing\_file='disk.img' encryption=of

## 3) 使用后端模板文件创建一个16G的镜像盘文件

01. [root@room9pc01 ~]# qemu-img create -b disk.img -f qcow2 disk2.img 16G
02. Formatting 'disk1.img', fmt=qcow2 size=53687091200 backing\_file='disk.img' encryption=of

## 步骤二：查看镜像文件的信息

01. [root@room9pc01 ~]# qemu-img info disk1.img
02. image: disk.img
03. file format: qcow2
04. virtual size: 50G ( 53687091200 bytes)
05. disk size: 196K
06. cluster\_size: 65536
07. Format specific information:
08. compat: 1.1
09. lazy refcounts: false | help\_topic |
10. | innodb\_index\_stats |
11. | innodb\_table\_stats |
12. | ndb\_binlog\_index |
13. | plugin |
14. | proc |
15. | procs\_priv |
16. | proxies\_priv |
17. | server\_cost |

[Top](#)

- 18. | servers |
- 19. | slave\_master\_info |

### 3 案例3：创建一个虚拟网络

#### 3.1 问题

创建一个虚拟网络，为之后的自定义安装虚拟机做准备：

- 创建一个名为vbr的虚拟网络
- 设置vbr的ip为192.168.1.254
- 配置vbr虚拟网络的dhcp分配地址范围100-200
- 启动vbr虚拟网络并用ifconfig验证
- 设置vbr虚拟网络开机自启动

#### 3.2 步骤

实现此案例需要按照如下步骤进行。

##### 步骤一：创建一个名为vbr的虚拟网络

```

01. [ root@room9pc01 ~] # vim /etc/libvirt/qemu/networks/vbr.xml
02. <network>
03.   <name>vbr</name>           //vbr为虚拟网络的名字
04.   <bridge name="vbr"/>
05.   <forward mode="nat"/>
06.   <ip address="192.168.1.254" netmask="255.255.255.0"> //ip为192.168.1.254
07.     <dhcp>
08.       <range start="192.168.1.100" end="192.168.1.200"/> //ip范围是100-200
09.     </dhcp>
10.   </ip>
11. </network>

```

##### 步骤二：启动vbr虚拟网络并用ifconfig验证

```

01. [ root@room9pc01 ~] # virsh net-define vbr //定义vbr虚拟网络
02. [ root@room9pc01 ~] # virsh net-start vbr //启动vbr虚拟网络
03. [ root@room9pc01 ~] # ifconfig //ifconfig验证
04. vbr: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
05.   inet 192.168.1.254 netmask 255.255.255.0 broadcast 192.168.1.255
06.   ether 52:54:00:b7:1c:10 txqueuelen 1000 (Ethernet)
07.   RX packets 2460 bytes 176958 (172.8 KiB)
08.   RX errors 0 dropped 0 overruns 0 frame 0

```

[Top](#)

09. TX packets 1948 bytes 532542 ( 520.0 KiB)
10. TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

### 步骤三：设置vbr虚拟网络开机自启动

```
01. [ root@room9pc01 ~] # virsh net- autostart vbr
```

## 4 案例4：xml管理

### 4.1 问题

熟悉xml文件，并对虚拟机的配置进行调整：

- 导出一个虚拟机的xml配置文件
- 编辑xml文件
- 重新定义虚拟机
- 删除此虚拟机

### 4.2 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：导出虚拟机xml的配置文件

##### 1) 查看xml配置文件

```
01. [ root@room9pc01 ~] # cd /etc/libvirt/qemu/
02. [ root@room9pc01 qemu] # virsh dumpxml node1
03. [ root@room9pc01 qemu] # virsh dumpxml node1 > node.xml
04. //导出虚拟机node1的配置文件为node.xml
05. [ root@room9pc01 qemu] # ls
06. node.xml
```

virsh命令：virsh edit 虚拟机名

备注：可以修改name，memory，disk、network等字段

```
01. [ root@room9pc01 qemu] # virsh edit node1
02. <domain type='kvm'>
03.   <name>node1</name> //node1为虚拟机的名称，可以随意修改
04.   <uuid>76d5dc2c- 5eef- 4e30- 8b6c- e58851814f84</uuid> //uuid可以去掉
05.   <memory unit='KiB'>2048000</memory> //内存大小可以调整
06.   <currentMemory unit='KiB'>2048000</currentMemory>
```

[Top](#)

07. `<disk type='file' device='disk'>`
08. `<source file='/var/lib/libvirt/images/rhel7.2.qcow2' />` //后端盘路径
09. `...`
10. `<interface type='network'>`
11. `<mac address='52:54:00:91:52:e4' />`
12. //mac地址和带有address字样的全部去掉
13. `...`

## 步骤二：重新定义虚拟机

### 1) 重新定义虚拟机

```
01. [root@room9pc01 qemu] # virsh define node1.xml
```

### 2) 取消定义的虚拟机

```
01. [root@room9pc01 qemu] # virsh undefine node1
```

## 5 案例5：安装虚拟机

### 5.1 问题

本案例要求可以成功安装一个自定义虚拟机：

- 配置一个网络yum，并安装一个虚拟机
- 制作一个虚拟机模板，包括配置yum，网卡等
- 

### 5.2 步骤

实现此案例需要按照如下步骤进行。

#### 步骤一：配置网络yum源

01. [root@room9pc01 ~] # yum -y install vsftpd
02. [root@room9pc01 ~] # vim /etc/vsftpd/vsftpd.conf
03. listen=YES
04. listen\_ipv6=NO
05. [root@room9pc01 ~] # systemctl restart vsftpd
06. [root@room9pc01 ~] # mkdir /var/ftp/centos
07. [root@room9pc01 ~] # mount /iso/CentOS-7-x86\_64-DVD-1708.iso /var/ftp/centos/
08. mount: /dev/loop1 写保护，将以只读方式挂载

[Top](#)

```

09. [ root@room9pc01 ~] # vim /etc/yum.repos.d/dvd.repo
10. [ dvd]
11. name=dvd
12. baseurl=ftp://192.168.1.254/centos
13. enabled=1
14. gpgcheck=0
15. [ root@room9pc01 ~] # yum clean all
16. [ root@room9pc01 ~] # yum repolist
17. [ root@room9pc01 ~] # cd /var/lib/libvirt/images/
18. [ root@room9pc01 images] # qemu-img create -f qcow2 node.qcow2 16G
19. Formatting 'node.qcow2', fmt=qcow2 size=17179869184 encryption=off cluster_size=65536

```

## 步骤二：安装一个虚拟机

```
01. [ root@room9pc01 images] # virt-manager //真机操作，启动virt-manager
```

使用virt-manager软件选择新建虚拟机如图-1所示：

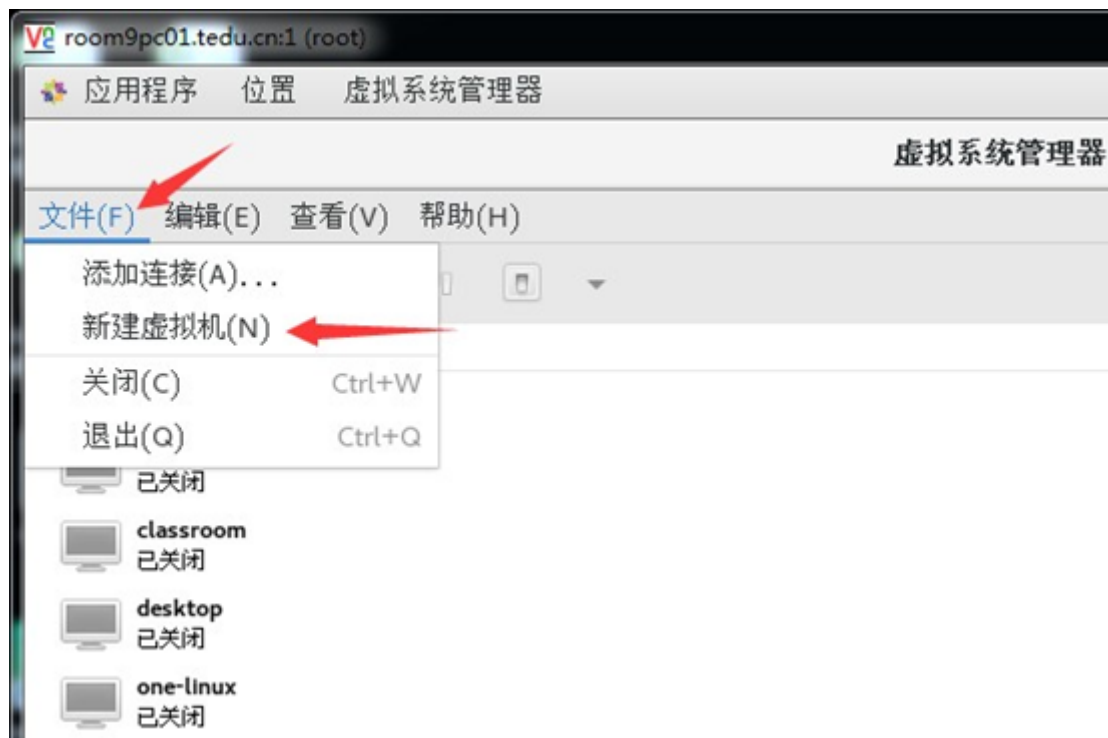


图-1

选择安装方式如图-2所示：

[Top](#)



图-2

选择内存，cpu和自定义存储如图-3所示：



图-3

选择虚拟机名称和网络如图-4所示：

[Top](#)





图-4

选择分区和KDUMP如图-5所示：

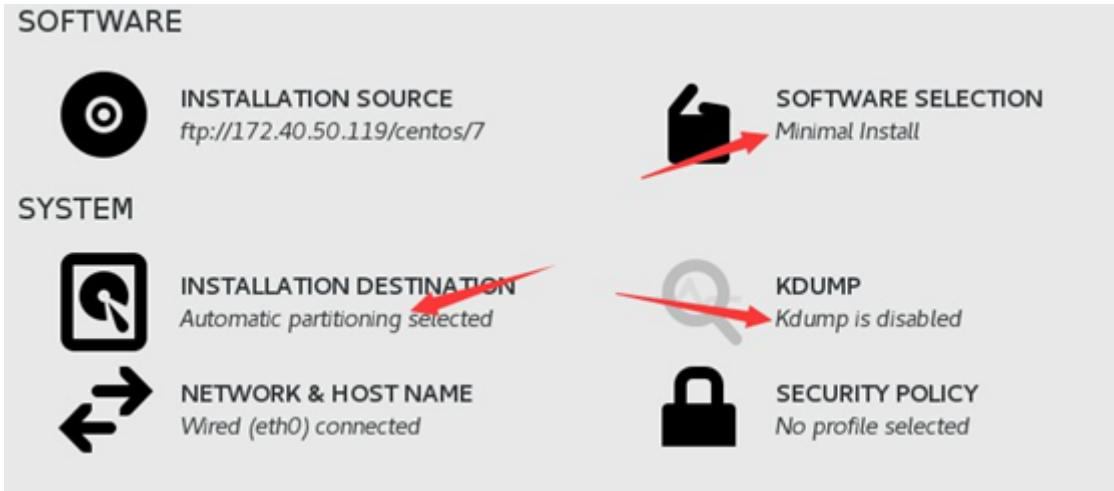


图-5

选择创建分区如图-6所示：

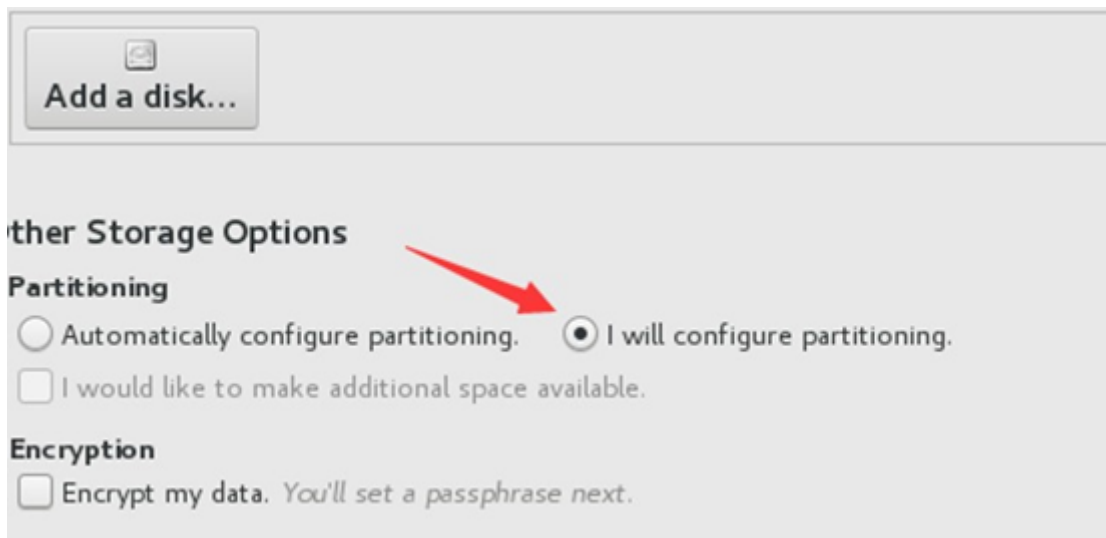


图-6

选择standard Partition如图-7所示：

图-7

创建一个根分区如图-8所示：

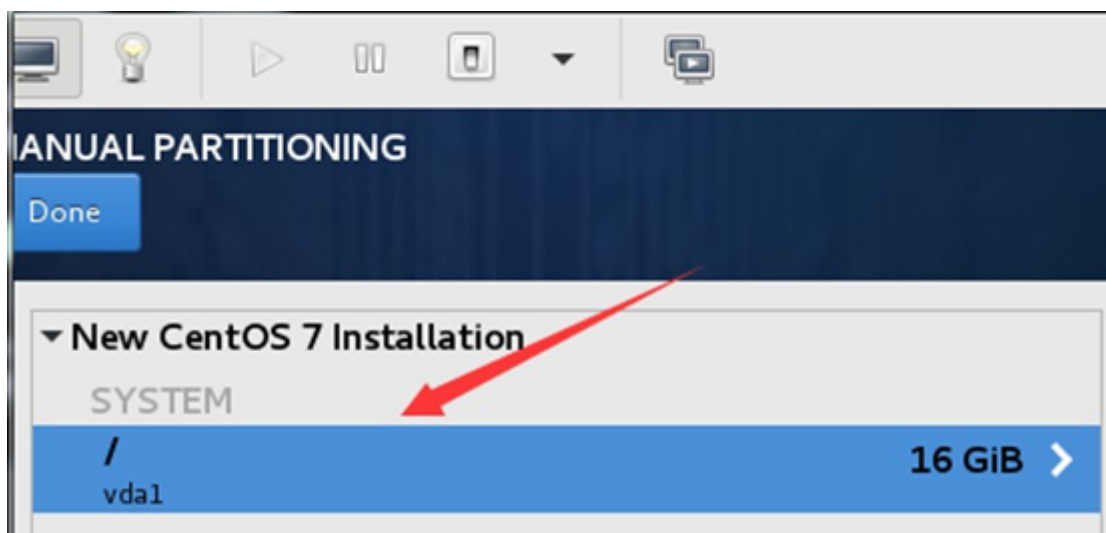


图-8

### 步骤三：制作一个虚拟机模板

#### 1) 禁用selinux

01. [root@localhost ~] # vim /etc/selinux/config
02. SELINUX=disabled

#### 2) 卸载防火墙与NetworkManager

01. [root@localhost ~] # yum -y remove NetworkManager- \* firewall- \* python- firewall

[Top](#)

### 3 ) 配置yum源

```
01. [ root@localhost ~] # vim /etc/yum.repos.d/dvd.repo
02. [ dvd]
03. name=dvd
04. baseurl=ftp://192.168.1.254/centos
05. enabled=1
06. gpgcheck=0
07. [ root@localhost ~] # yum clean all
08. [ root@localhost ~] # yum repolist
```

### 4 ) 导入公钥

注意：把/etc/yum.repos.d/dvd.repo的gpgcheck=0改成gpgcheck=1

```
01. [ root@localhost ~] # lftp 192.168.1.254
02. lftp 192.168.4.254: ~> cd centos
03. lftp 192.168.4.254: /centos> get RPM-GPG-KEY-CentOs-7
04. lftp 192.168.4.254: /centos> exit
05. [ root@localhost ~] # rpm --import RPM-GPG-KEY-CentOs-7
06. [ root@localhost ~] # yum -y install net-tools vim-enhanced bridge-utils psmisc
```

### 5 ) 配置网卡

```
01. [ root@localhost ~] # vim /etc/sysconfig/network-scripts/ifcfg-eth0
02. DEVICE="eth0"
03. ONBOOT="yes"
04. IPV6INIT="no"
05. TYPE="Ethernet"
06. BOOTPROTO="dhcp"
07. [ root@localhost ~] # systemctl restart network
```

### 6 ) 禁用空路由

```
01. [ root@localhost ~] # vim /etc/sysconfig/network
02. NOZEROCONF="yes"
```

[Top](#)

## 7) 添加console配置

01. [root@localhost ~] # vim /etc/default/grub
02. GRUB\_CMDLINE\_LINUX="biosdev name=0 net.if names=0 console=tty S0,115200n8"
03. GRUB\_DISABLE\_LINUX\_UUID="true"
04. GRUB\_ENABLE\_LINUX\_LABEL="true"

## 8) 重新生成grub.cfg

01. [root@localhost ~] # grub2-mkconfig -o /boot/grub2/grub.cfg

## 9) 安装扩展分区软件

01. [root@localhost ~] # yum install -y cloud-utils-growpart

## 10) 第一次开机自动扩容

01. [root@localhost ~] # /usr/bin/growpart /dev/vda 1
02. [root@localhost ~] # /usr/sbin/xfs\_growfs /

## 11) 关闭虚拟机后执行信息清理工作

01. [root@room9pc01 ~] # virt-sysprep -d centos7.0
02. //真机上面操作, centos7.0为虚拟机名称

# 6 案例6：离线访问虚拟机问题

本案例要求可以离线访问虚拟机：

- 利用xml文件生成一个新的虚拟机
- 利用guestmount实现离线访问虚拟机

## 6.1 步骤

实现此案例需要按照如下步骤进行。

**步骤一：用xml生成一个新的虚拟机**

[Top](#)

```

01. [ root@room9pc01 ~] # cd /etc/libvirt/qemu/
02. [ root@room9pc01 qemu] # cp centos7.0.xml local.xml
03. [ root@room9pc01 qemu] # vim local.xml
04. <domain type='kvm'>
05.   <name>local</name>           //修改名字
06.   <memory unit='KiB'>2048000</memory>      //修改内存
07.   <currentMemory unit='KiB'>2048000</currentMemory>
08.   ...
09.   <source file='/var/lib/libvirt/images/local.img' />    //img文件位置

```

注意：除这些外还要把mac地址删掉，带address字样的全部删除

```

01. [ root@room9pc01 ~] # cd /var/lib/libvirt/images/
02. [ root@room9pc01 images] # qemu-img create -b node.qcow2 -f qcow2 local.img
03. [ root@room9pc01 images] # virsh define /etc/libvirt/qemu/local.xml
04. [ root@room9pc01 images] # virsh start local
05. [ root@room9pc01 images] # virsh console local

```

## 步骤二：guestmount实现离线访问

基本用法：guestmount -a 虚拟机磁盘路径 -i /挂载点

-a：指定虚拟磁盘

-i：挂载点

```

01. [ root@room9pc01 ~] # mkdir /mnt/kdisk
02. [ root@room9pc01 ~] # guestmount -a node1.qcow2 -i /mnt/kdisk
03. [ root@room9pc01 ~] # cd /mnt/kdisk
04. [ root@room9pc01 kdisk] # ls
05. bin    home    media    opt      sbin     tmp
06. boot   lib     misc     proc     selinux  usr

```

## 6.2

[Top](#)