教学部 COOKBOOK

(NSD 制作模板虚拟机)

版本编号 1.0

2019-02

达内杭州西溪云计算中心

制作模板虚拟机

0. 真机环境

真机硬件 cpu 四代及以上，内存 16G 及以上，硬盘固态 256G，没有用机械问题也不大，无显卡、主板要求。
安装最新版的 centos7.6，光盘镜像为 CentOS-7-x86_64-DVD-1810.iso
语言推荐 English，时区选择 Asia/ShangHai
安装选择“GNOME Desktop”，最小安装你们 hold 不住
分区格式选择 standard partition（标准分区），只分一个/分区
生产环境推荐选择 LVM 分区
如果主板开了 fast boot，使用了 UEFI 启动项，分区需要新增两个启动分区，/boot 推荐 500M 和/boot/efi 推荐 100M
内存 16G 及以上不需要安装 swap 分区
关闭 KDUMP
开启网络，修改用户名为 GYP-HOME(自己写自己名字，命名不要随便)

- **真机环境初始化**

1) 禁用 selinux

```
[root@GYP-HOME ~]# vim /etc/selinux/config
SELINUX=disabled
```

2) 设置防火墙

```
[root@GYP-HOME ~]# firewall-cmd --set-default-zone=trusted
```

3) 下载阿里网络 yum 源

```
[root@GYP-HOME ~]# mkdir /etc/yum.repos.d/bak
[root@GYP-HOME ~]# mv /etc/yum.repos.d/*.repo /etc/yum.repos.d/bak
[root@GYP-HOME ~]# curl -o /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
[root@GYP-HOME ~]# yum repolist
[root@GYP-HOME ~]# yum update
```

阿里开源共享网址为：https://opsx.alibaba.com/mirror

找到对应链接，直接复制，粘贴。

CentOS 7

```
wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
```

注意：生产环境不推荐，换 yum 源执行 yum update 会直接升级当前系统版本，比如 7.2 升级到 7.6，会造成部分软件的依赖关系出错。

4) 安装虚拟化平台

a)

```
[root@GYP-HOME ~]# yum group list hidden
[root@GYP-HOME ~]# yum -y group install “Virtualization Client” “Virtualization Hypervisor” “Virtualization Platform” “Virtualization Tools”
```

b)

```
[root@GYP-HOME ~]# yum -y install virt-manager qemu-kvm libvirt-daemon libvirt-client libvirt-daemon-driver-qemu libguestfs-tools
```

两种安装方式，a 是通过组安装，不需要记忆软件包名，都可以查出。b 是对虚拟化十分熟悉，非常明白自己要装那几个安装包。

5) 安装 ftp 服务做真机对虚拟机网络 yum 源共享

```
[root@GYP-HOME ~]# yum -y install vsftpd
[root@GYP-HOME ~]# systemctl start vsftpd
[root@GYP-HOME ~]# systemctl enable vsftpd
[root@GYP-HOME ~]# mkdir /iso          //放入 CentOS-7.4-x86_64-Everything-1708.iso 光盘
[root@GYP-HOME ~]# mkdir /var/ftp/centos7.4
[root@GYP-HOME ~]# vim /etc/fstab
/iso/CentOS-7.4-x86_64-Everything-1708.iso /var/ftp/centos7.4 iso9660 loop,ro 0 0
[root@GYP-HOME ~]# mount -a
```

6) 关闭 ssh 的 hostkey 认证，即第一次远程登陆时不需要输入(yes/no)

```
[root@GYP-HOME ~]# vim /etc/ssh/ssh_config
……
58 Host *
59         GSSAPIAuthentication yes
60         StrictHostKeyChecking no
:x
```

1. 创建两个虚拟网络

- **问题**
  - 创建两个虚拟网络，为之后的自定义安装虚拟机做准备
  - 设置 vbr_4 为 192.168.4.254
  - 配置 vbr_4 虚拟网络的 dhcp 分配地址范围 1-253
  - 设置 vbr_2 为 192.168.2.254
  - 配置 vbr_2 虚拟网络的 dhcp 分配地址范围 1-253

- **步骤**

1) 备份系统默认虚拟网络 default，并关闭开机自启

```
[root@GYP-HOME ~]# cd /etc/libvirt/qemu/networks/
[root@GYP-HOME networks]# mv default.xml default.xml.bak
[root@GYP-HOME networks]# cp default.xml.bak vbr_2.xml
[root@GYP-HOME networks]# cp default.xml.bak vbr_4.xml
[root@GYP-HOME networks]# virsh net-destroy default                  //关闭 default 网络
[root@GYP-HOME networks]# virsh net-autostart --disable default      //关闭 default 开机自启
```

2) 创建名为 vbr_2 与 vbr_4 的虚拟网络

```
[root@GYP-HOME networks]# vim vbr_2.xml
<network>
  <name>vbr_2</name>
  <forward mode='nat'/>
  <bridge name='vbr_2' stp='on' delay='0'/>
  <ip address='192.168.2.254' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.2.1' end='192.168.2.253'/>
    </dhcp>
  </ip>
</network>
:x                                          //删除 uuid 以及 mac 这类代表唯一标识的信息,修改标黄区域
[root@GYP-HOME networks]# vim vbr_4.xml
<network>
  <name>vbr_4</name>
  <forward mode='nat'/>
  <bridge name='vbr_4' stp='on' delay='0'/>
  <ip address='192.168.4.254' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.4.1' end='192.168.4.253'/>
    </dhcp>
  </ip>
</network>
:x
```

3) 启动两个虚拟网络并设置开机自启

```
[root@GYP-HOME networks]# virsh net-define vbr_2.xml        //定义生成新的虚拟网络
[root@GYP-HOME networks]# virsh net-define vbr_4.xml
[root@GYP-HOME networks]# virsh net-start vbr_2             //开启 vbr_2 网络
[root@GYP-HOME networks]# virsh net-start vbr_4
[root@GYP-HOME networks]# virsh net-autostart vbr_2         //开机自启 vbr_2 网络
[root@GYP-HOME networks]# virsh net-autostart vbr_4
```

4) 验证:出现以下信息为配置正确

```
[root@GYP-HOME networks]# virsh net-list
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 vbr_2                active     yes           yes
 vbr_4                active     yes           yes
[root@GYP-HOME networks]# ifconfig
vbr_2: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.2.254  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 52:54:00:0c:3e:35  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vbr_4: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.4.254  netmask 255.255.255.0  broadcast 192.168.4.255
        ether 52:54:00:d8:93:f4  txqueuelen 1000  (Ethernet)
        RX packets 5749  bytes 679394 (663.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7945  bytes 51822758 (49.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5) 虚拟网络拓扑详解，参看串讲计划：GYP-虚拟机网络

2. 制作虚拟机模板

- **问题**
  - 通过光盘安装一个虚拟机
  - 制作虚拟机模板，包括配置 yum、网卡、免密、预装软件
- **创建虚拟机**

1) 创建虚拟机后端磁盘文件

```
[root@GYP-HOME images]# qemu-img create -f qcow2 demo.qcow2 10G
Formatting 'demo.qcow2', fmt=qcow2 size=10737418240 encryption=off cluster_size=65536 lazy_refcounts=off
[root@GYP-HOME images]# du -sh demo.qcow2
196K    demo.qcow2
```

注意：如果不先创建后端磁盘文件，直接通过新建虚拟机的方式生成后端盘文件，那么结果是设置多大的磁盘就会使用多大的物理空间,即如果设置 10G 大小,物理磁盘就会直接占用 10G,而不会像直接创建的空磁盘只占用 196k，真实使用多少就用多少。

2) 通过 virt-manager 安装虚拟机

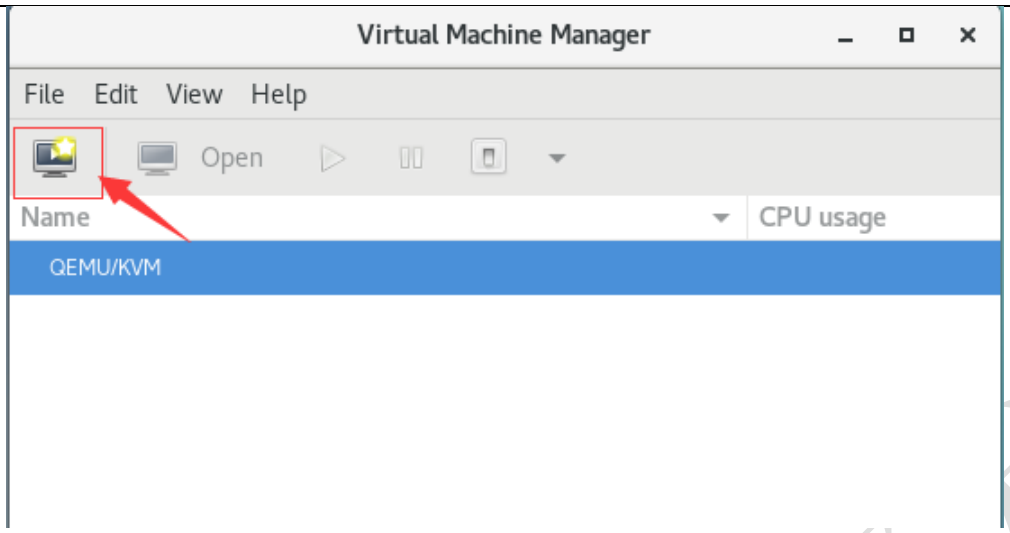


图-1

3) 选择光盘镜像安装

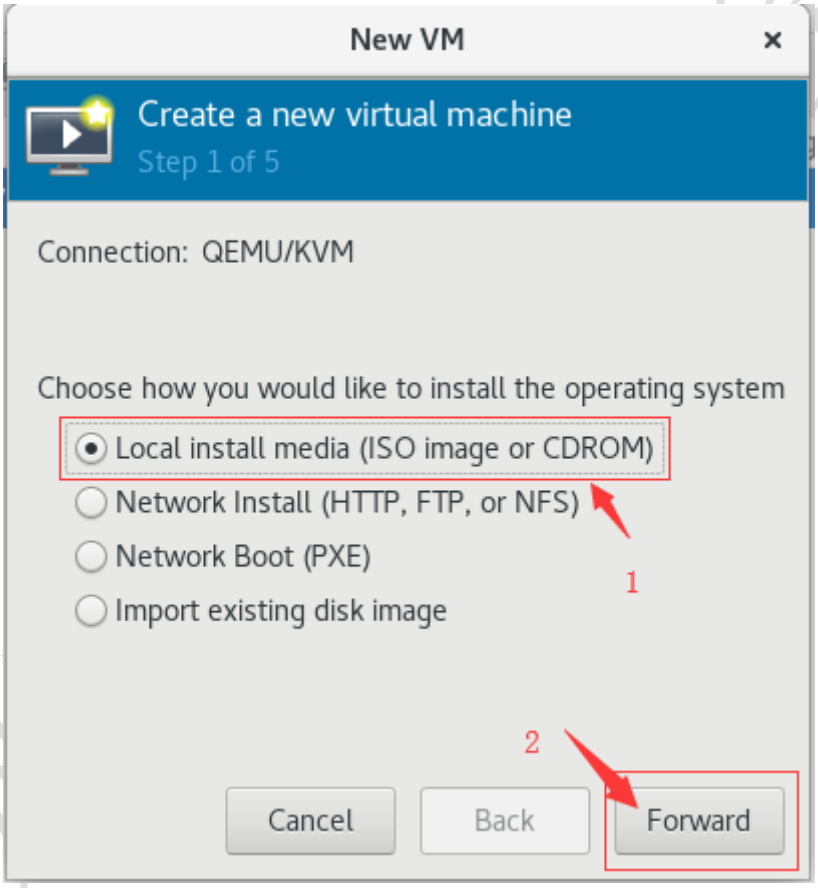


图-2

4) 找到光盘所在路径

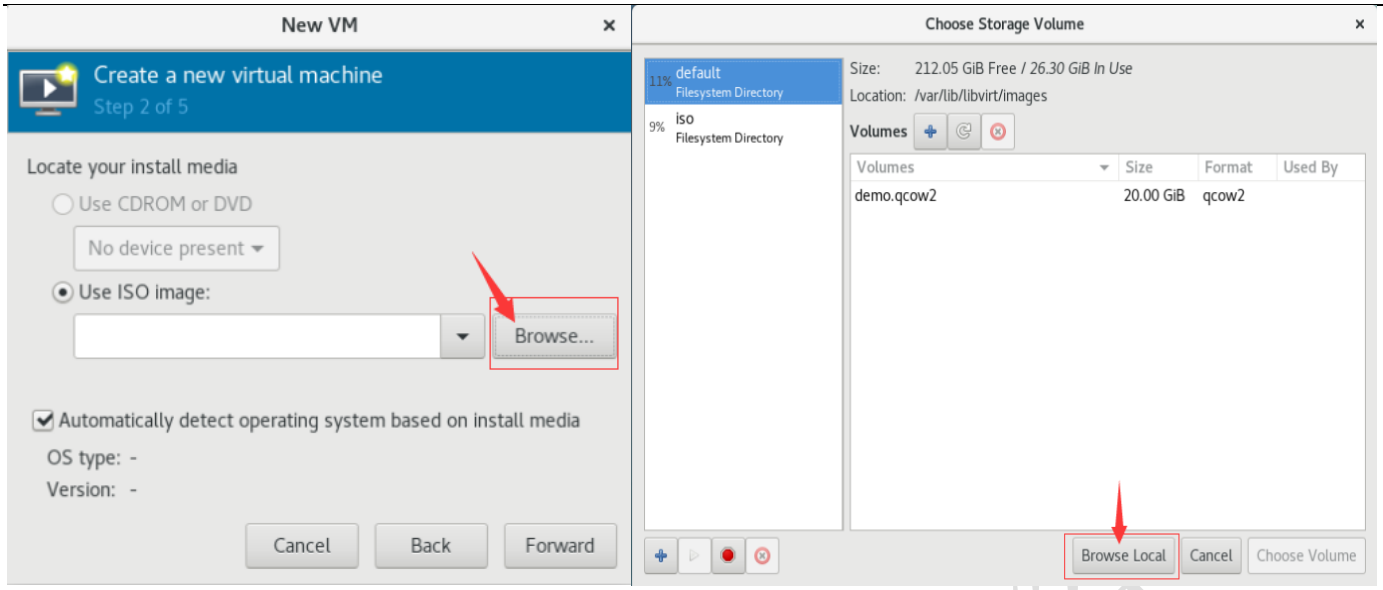


图-3

图-4

选择好后下一步

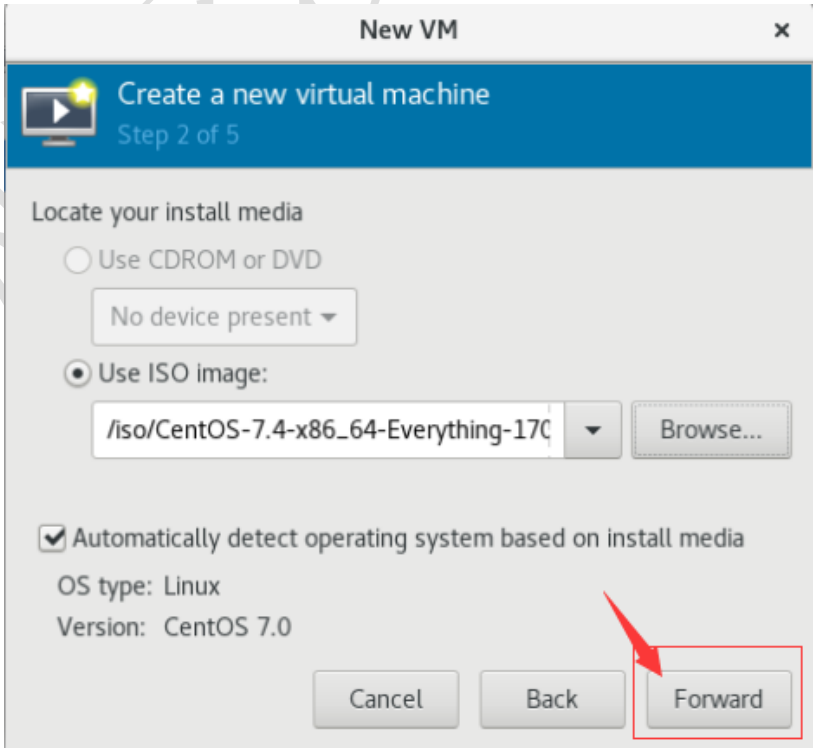


图-5

5) 选择内存，cpu 和自定义存储

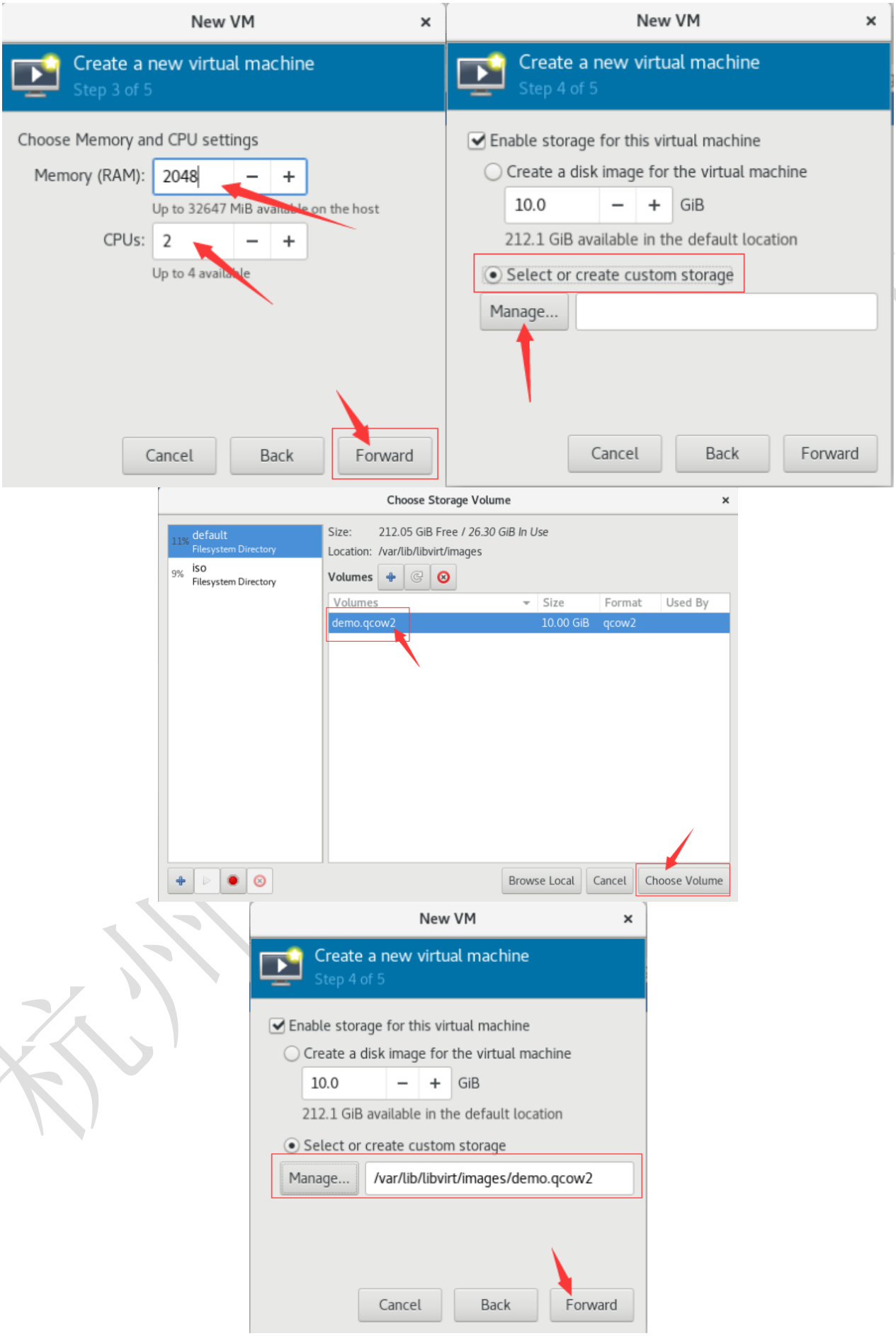


图-6

6) 设置虚拟机名称，选择默认网络

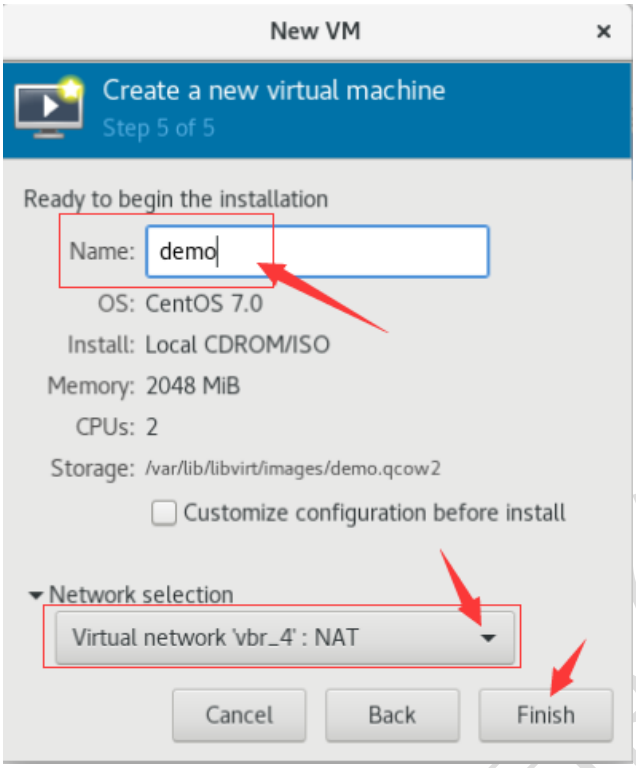


图-7

7) 选择时区，分区，kdump 如图所示

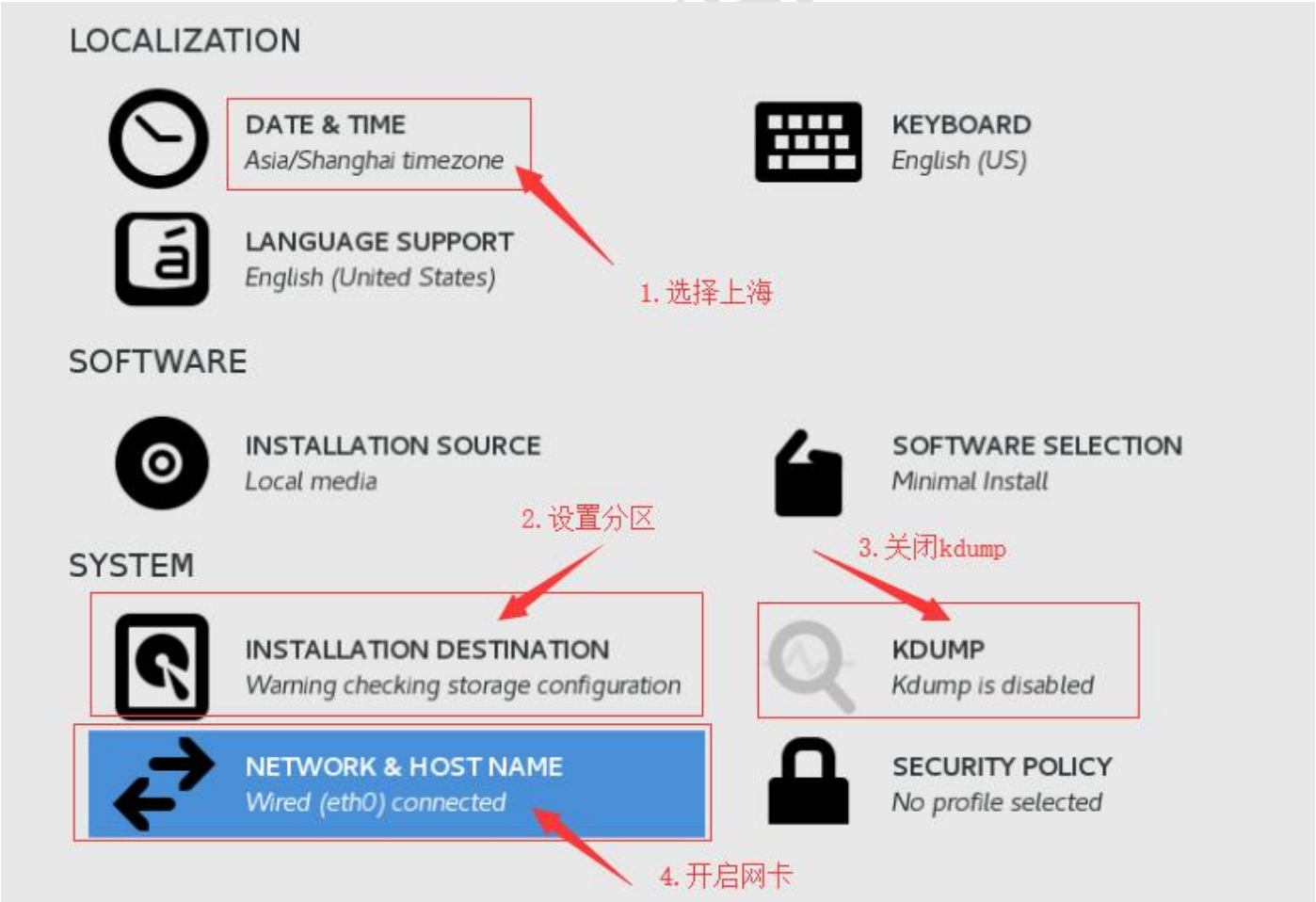
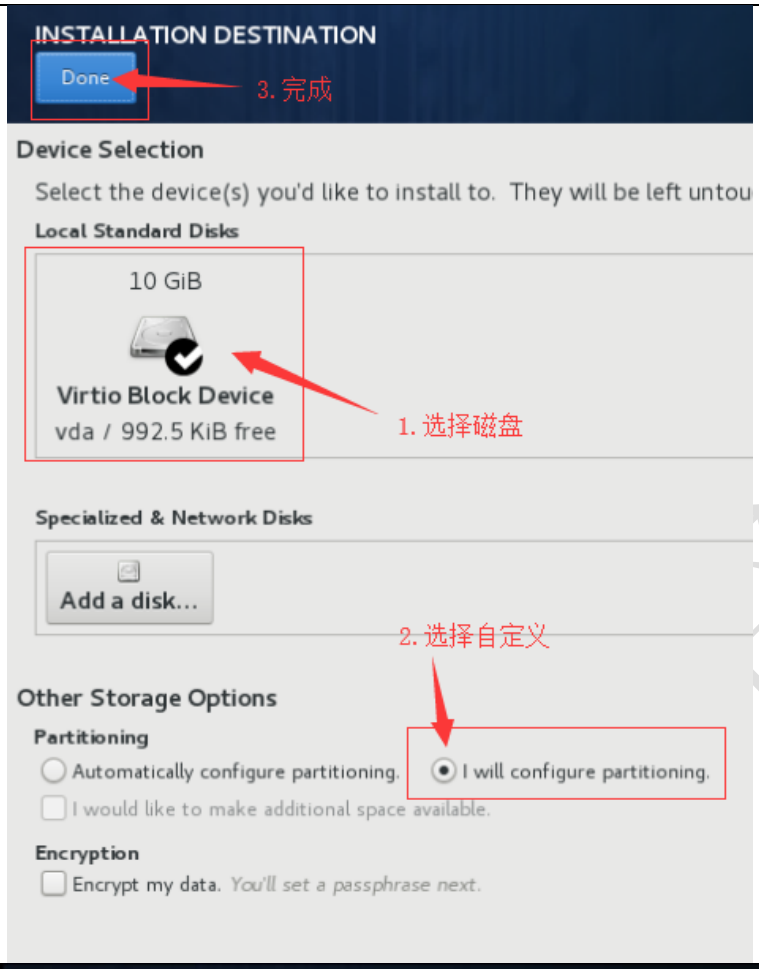
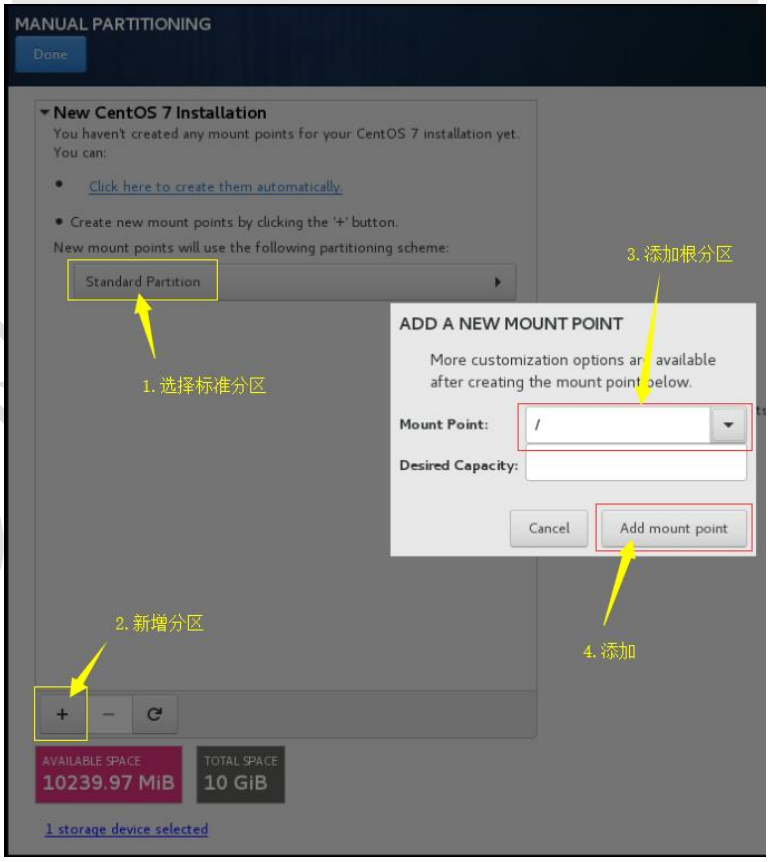


图-8

8) 创建分区如图所示


INSTALLATION DESTINATION
Done
3. 完成
Device Selection
Select the device(s) you'd like to install to. They will be left untou
Local Standard Disks
10 GiB
Virtio Block Device
vda / 992.5 KiB free
1. 选择磁盘
Specialized & Network Disks
Add a disk...
2. 选择自定义
Other Storage Options
Partitioning
Automatically configure partitioning.
I will configure partitioning.
I would like to make additional space available.
Encryption
Encrypt my data. You'll set a passphrase next.


MANUAL PARTITIONING
Done
New CentOS 7 Installation
You haven't created any mount points for your CentOS 7 installation yet.
You can:
Click here to create them automatically.
Create new mount points by clicking the '+' button.
New mount points will use the following partitioning scheme:
Standard Partition
1. 选择标准分区
3. 添加根分区
ADD A NEW MOUNT POINT
More customization options are available after creating the mount point below.
Mount Point:
/
Desired Capacity:
Cancel
Add mount point
4. 添加
2. 新增分区
AVAILABLE SPACE
10239.97 MiB
TOTAL SPACE
10 GiB
1 storage device selected

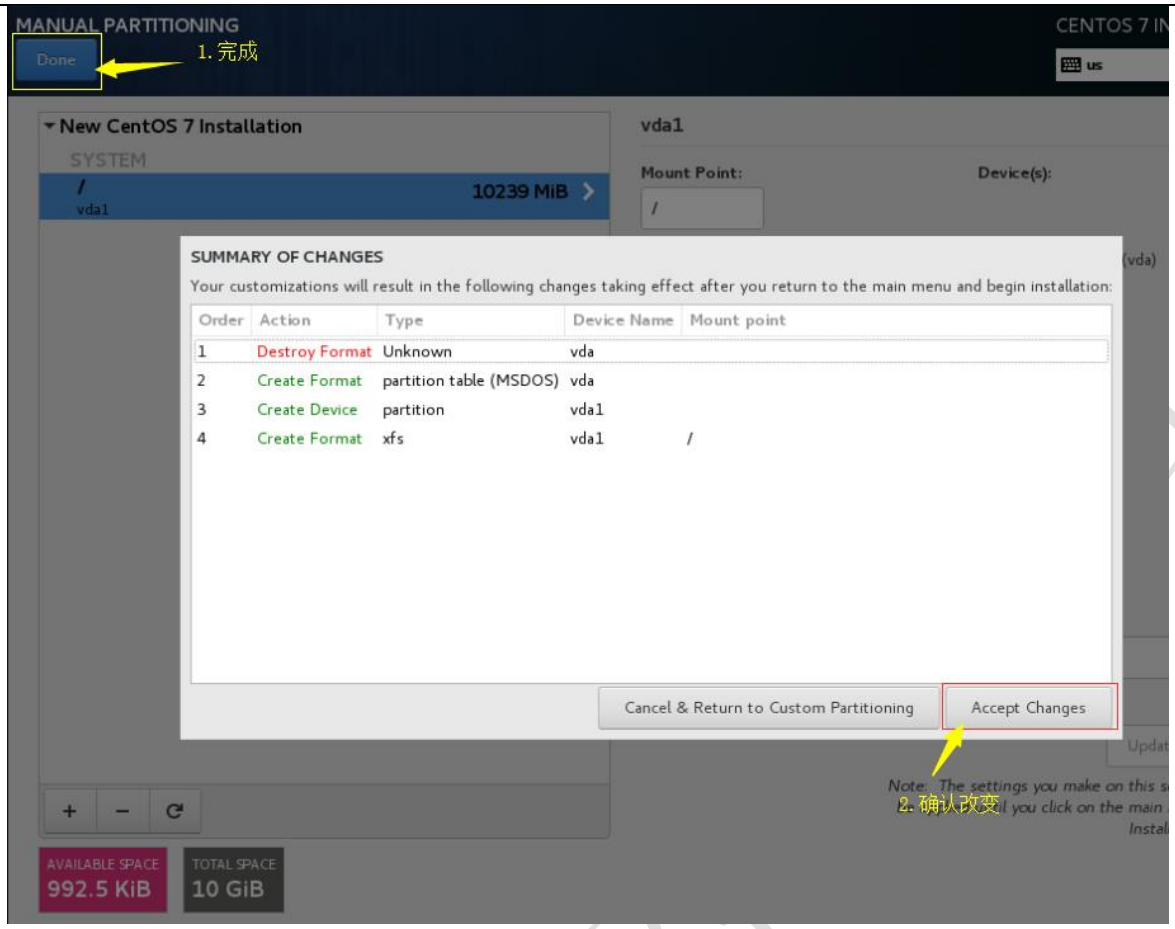


图-9

9) root 密码设置非常简单，这里就不在截图了。

- **制作一个虚拟机模板**

1) 添加第二张网卡

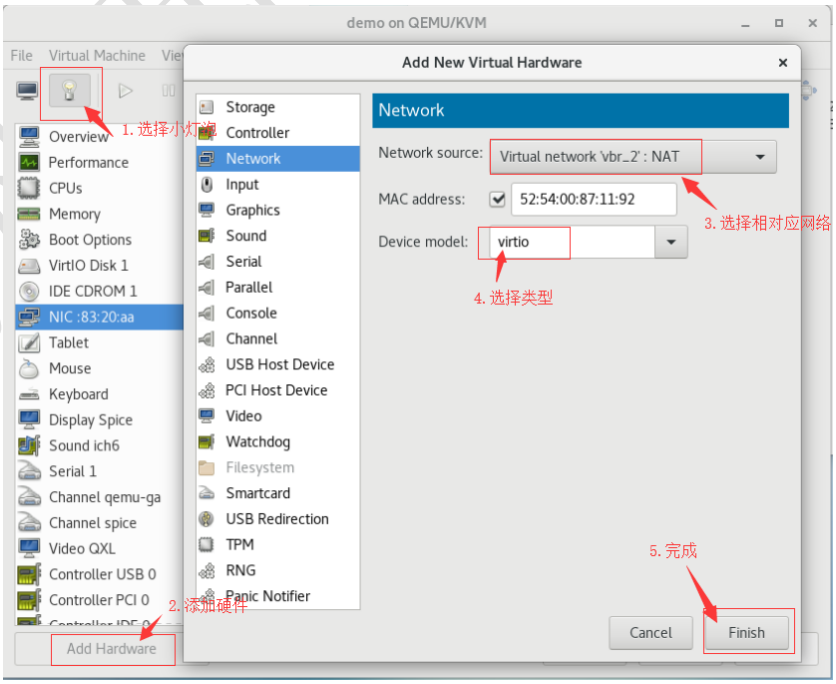


图-10

2) 禁用 selinux

```
[root@localhost ~]# vi /etc/selinux/config
SELINUX=disabled
```

3) 配置 yum 源(在虚拟中操作，注意命令提示符)

```
[root@localhost ~]# rm -rf /etc/yum.repos.d/*
[root@localhost ~]# vi /etc/yum.repos.d/centos.repo
[centos7.4]
name=centos7.4
baseurl=ftp://192.168.4.254/centos7.4
enabled=1
gpgcheck=1
gpgkey=ftp://192.168.4.254/centos7.4/RPM-GPG-KEY-CentOS-7
:x
[root@localhost ~]# yum clean all
[root@localhost ~]# yum repolist
```

4) 卸载防火墙与 NetworkManager (与后期 openstack 实验冲突)

```
[root@localhost ~]# rpm -qa | grep firewall
firewalld-filesystem-0.4.4.4-6.el7.noarch
python-firewall-0.4.4.4-6.el7.noarch
firewalld-0.4.4.4-6.el7.noarch
[root@localhost ~]# yum -y remove firewalld-filesystem python-firewall firewalld
```

5) 安装常用软件

```
[root@localhost ~]# yum -y install vim-enhanced psmisc net-tools bash-completion
//vim, pstree, ifconfig, tab 补全
```

6) 编写配置 ip 脚本

a) 简单 shell 脚本

```
[root@localhost ~]# vim eip
#!/bin/bash
cat > /etc/sysconfig/network-scripts/ifcfg-eth0 <<EOF
TYPE="Ethernet"
BOOTPROTO="none"
IPV6INIT="no"
NAME="eth0"
DEVICE="eth0"
ONBOOT="yes"
IPADDR="192.168.4.$1"
PREFIX=24
GATEWAY="192.168.4.254"
EOF
:x
[root@localhost ~]# chmod +x eip
[root@localhost ~]# mv eip /usr/local/bin/
```

b) 中级 shell 脚本

```
#!/bin/bash
CFG_PATH=/etc/sysconfig/network-scripts/

config_ip(){
echo "TYPE=Ethernet
BOOTPROTO=none
IPV6INIT=no
NAME=$eth
DEVICE=$eth
ONBOOT=yes
IPADDR=$ip
PREFIX=24
GATEWAY=$gw" > $CFG_PATH/ifcfg-$eth
}

check_ip(){
echo $ip | grep -E "^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$" > /dev/null
if [ $(echo $?) != 0 ];then
  echo "IP $ip not available!"
  exit 1
fi
}

menu(){
clear
echo -n "VM has two network devices named "
echo -e "\033[31m eth0 eth1 \033[0m"
echo -ne "eth0 network segment is\t"
echo -e "\033[36m 192.168.4.0/24 \033[0m"
echo -ne "eth1 network segment is\t"
echo -e "\033[36m 192.168.2.0/24 \033[0m"
echo "Please configure the correct values"
echo
read -p "Choice which net device: " eth
read -p "Enter the IP: " ip
read -p "Enter the gateway:" gw
}

menu
check_ip
config_ip
/usr/sbin/ifdown $eth > /dev/null
/usr/sbin/ifup $eth > /dev/null
echo -e "\033[32m $eth is configured \033[0m"
```

效果图

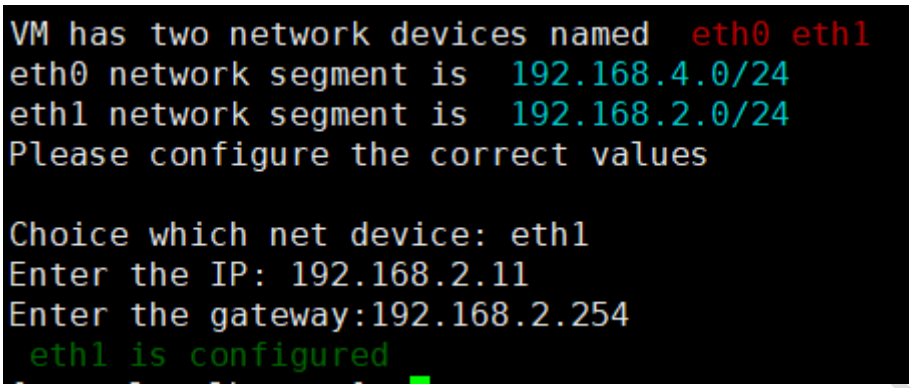


图-11

这个 shell 脚本还可以优化，这就就不再赘述。

c) 高级 python 脚本

张志刚老师 python 百例 https://www.jianshu.com/p/436bad220b5d

推荐大家使用第一个，牺牲一定灵活性带来的是高效

7) 禁用空路由

```
[root@localhost ~]# vim /etc/sysconfig/network
NOZEROCONF="yes"      //本机 dhcp 获取不到 ip 时自动添加 168 开头的 ip
```

8) 添加 console 配置

```
[root@localhost ~]# vim /etc/default/grub
GRUB_CMDLINE_LINUX="biosdevname=0 net.ifnames=0 console=ttyS0,115200n8"
//biosdevname=0 net.ifnames=0 关闭 rhel7 自动根据网卡设备来命名
//console=ttyS0,115200n8 设置虚拟机 console 口连接，可用 virsh console vm-name 链接虚拟机
:x
[root@localhost ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
```

9) 设置开机自动扩容 (新建前端磁盘容量大于后端盘时使用)

```
[root@localhost ~]# yum -y install cloud-utils-growpart.noarch
[root@localhost ~]# vim /etc/rc.d/rc.local
vda_size=`lsblk | awk '/vda / {print $4}'`
vda1_size=`lsblk | awk '/vda1/ {print $4}'`
if [ $vda_size != $vda1_size ];then
  /usr/bin/growpart /dev/vda 1
  /usr/sbin/xfs_growfs /
fi
:x
[root@localhost ~]# chmod +x /etc/rc.d/rc.local
```

10) 设置真机免密登录虚拟机

```
[root@GYP-HOME ~]# ssh-keygen -t RSA -N '' -f /root/.ssh/id_rsa
[root@GYP-HOME ~]# ssh-copy-id 192.168.4.36
```

11) 设置虚拟机之间两两免密

```
[root@GYP-HOME ~]# scp /root/.ssh/id_rsa 192.168.4.36:/root/.ssh/
```

12) ssh 免密登录原理参看串讲计划：GYP-ssh 远程登录原理

13) 关闭虚拟机 ssh 的 hostkey 认证

```
[root@localhost ~]# vim /etc/ssh/ssh_config
……
58 Host *
59        GSSAPIAuthentication yes
60        StrictHostKeyChecking no
:x
```

14) 开启真机次级 NTP 时间同步服务，并让虚拟机的时间同步服务器指向真机

```
[root@GYP-HOME ~]# vim /etc/chrony.conf
server ntp1.aliyun.com iburst
bindacqaddress 0.0.0.0
allow 0/0
cmdallow 127.0.0.1
:x       //注释不用的时间同步服务，添加 3 条配置
[root@GYP-HOME ~]# systemctl restart chronyd

[root@localhost ~]# yum -y install ntpdate chrony
[root@localhost ~]# vim /etc/chrony.conf
server 192.168.4.254 iburst
:x
[root@localhost ~]# ntpdate 192.168.4.254          //验证时间同步是否可用
 5 Feb 14:08:33 ntpdate[2005]: step time server 192.168.4.254 offset -1.182465 sec
```

15) 虚拟机清理工作

```
[root@localhost ~]# > /etc/sysconfig/network-scripts/ifcfg-eth0
[root@localhost ~]# > /etc/sysconfig/network-scripts/ifcfg-eth1
[root@localhost ~]# > /root/.bash_history
[root@localhost ~]# history -c
[root@localhost ~]# poweroff

[root@GYP-HOME ~]# virt-sysprep -d demo
```

3. 通过模板创建新的虚拟机

- **问题**
  - 修改 demo.xml 文件成为模板机的 xml 配置文件
  - 下线 demo 虚拟机，防止误开修改后端盘文件
  - 生成新虚拟机

- **创建新虚拟机**

1) 导出 demo 的 xml 文件

```
[root@GYP-HOME ~]# cd /etc/libvirt/qemu/
[root@GYP-HOME qemu]# virsh dumpxml demo > demo.xml.bak
```

2) 下线 demo 虚拟机(顺序不能错！先导出文件，再下线)

```
[root@GYP-HOME qemu]# virsh undefine demo
```

3) 修改 demo.xml.bak 为通用模板

删除 UUID、MAC、总线地址(address)、别名配置(alias)，usb 相关设备、仿真设备中的 pci，tablet，spicevmc，graphics、video，sound，redirdev。

```
[root@GYP-HOME qemu]# vim demo.xml.bak
<domain type='kvm'>
  <name>demo</name>
  <memory unit='KiB'>2097152</memory>
  <currentMemory unit='KiB'>2097152</currentMemory>
  <vcpu placement='static'>2</vcpu>
  <os>
    <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <cpu mode='custom' match='exact' check='partial'>
    <model fallback='allow'>Westmere-IBRS</model>
  </cpu>
  <clock offset='utc'>
    <timer name='rtc' tickpolicy='catchup'/>
    <timer name='pit' tickpolicy='delay'/>
    <timer name='hpet' present='no'/>
  </clock>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2'/>
      <source file='/var/lib/libvirt/images/demo.qcow2'/>
      <target dev='vda' bus='virtio'/>
    </disk>
    <controller type='ide' index='0'>
    </controller>
```

```
    <controller type='virtio-serial' index='0'>
    </controller>
    <interface type='network'>
      <source network='vbr_4'/>
      <model type='virtio'/>
    </interface>
    <interface type='network'>
      <source network='vbr_2'/>
      <model type='virtio'/>
    </interface>
    <serial type='pty'>
      <target type='isa-serial' port='0'>
        <model name='isa-serial'/>
 </target>
    </serial>
    <console type='pty'>
      <target type='serial' port='0'/>
    </console>
    <channel type='unix'>
      <target type='virtio' name='org.qemu.guest_agent.0'/>
      <address type='virtio-serial' controller='0' bus='0' port='1'/>
    </channel>
    <input type='mouse' bus='ps2'/>
    <input type='keyboard' bus='ps2'/>
    <memballoon model='virtio'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>
    </memballoon>
  </devices>
</domain>
:x
```

标黄地区为新建虚拟机需要修改的地方。每个参数的意义详情请参考 cloud 阶段第一天 ppt。

4) 创建新虚拟机 test1.xml 文件

```
[root@GYP-HOME qemu]# cp demo.xml.bak test1.xml
[root@GYP-HOME qemu]# sed -i '/name/s/demo/test1/;s/demo.qcow2/test1.img/' test1.xml
```

5) 创建新虚拟机磁盘文件

```
[root@GYP-HOME qemu]# cd /var/lib/libvirt/images/
[root@GYP-HOME images]# qemu-img create -b demo.qcow2 -f qcow2 test1.img 20G
//20G 为了验证 2.9 步骤开机是否自动扩容，可不写
```

6) 组成新的虚拟机并启动验证

```
[root@GYP-HOME images]# cd -
[root@GYP-HOME qemu]# virsh define test1.xml
[root@GYP-HOME qemu]# virsh start test1
[root@GYP-HOME qemu]# virsh console test1
```

```
localhost login: root
Password:
[root@localhost ~]# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda    253:0    0  20G  0 disk
└─vda1 253:1    0  20G  0 part /
```

7) 编写快速生成虚拟机脚本

```
#!/bin/bash
check_vmnum(){
expr $VMNUM + 0 &> /dev/null
if [ $? -ne 0 ];then
  echo "You mast input a number."
  exit 4
fi

if [ $VMNUM -lt 1 -o $VMNUM -gt 99  ];then
  echo "Input out of range"
  exit 5
elif [ $VMNUM -le 9 ];then
  VMNUM=0$VMNUM
fi

if [ -e $IMG_DIR/$NEWVMNUM.img ];then
  echo "File exists"
  exit 6
fi
}

IMG_DIR=/var/lib/libvirt/images/
CONF_DIR=/etc/libvirt/qemu/

create(){
qemu-img create -b $IMG_DIR/demo.qcow2 -f qcow2 $IMG_DIR/${NEWVMNUM}.img > /dev/null
cp $CONF_DIR/demo.xml.bak $CONF_DIR/${NEWVMNUM}.xml
sed -i "/name/s/demo/$NEWVMNUM/;s/demo.qcow2/$NEWVMNUM.img/" $CONF_DIR/${NEWVMNUM}.xml
virsh define $CONF_DIR/${NEWVMNUM}.xml > /dev/null
}

read -p "Enter VM number: " VMNUM
check_vmnum
NEWVMNUM=node${VMNUM}
echo -en "Creating Virtual Machine ......\t"
create
echo -e "\033[32m[ok]\033[0m"
```

4. 离线访问虚拟机

- **问题**
  - 通过 guestmount 命令挂载后端磁盘文件，并修改虚拟机 root 密码为 hehe
  - 通过 guestfish 命令给虚拟机安装 vsftpd 并开启 selinux

- **步骤**

1) guestmount 实现离线访问

```
[root@GYP-HOME conf]# ./clone-vm.sh
Enter VM number:7
[root@GYP-HOME conf]# cd /var/lib/libvirt/images/
[root@GYP-HOME images]# guestmount -a node07.img -i /mnt/test
[root@GYP-HOME images]# ls /mnt/test
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
```

2) 修改虚拟机密码

```
[root@GYP-HOME images]# chroot /mnt/test
[root@GYP-HOME /]# echo hehe | passwd --stdin root
Changing password for user root.
passwd: all authentication tokens updated successfully.
[root@GYP-HOME /]# exit
[root@GYP-HOME images]# guestunmount /mnt/test
[root@GYP-HOME images]# virsh start node07
[root@GYP-HOME images]# virsh console node07
[root@GYP-HOME images]# virsh destroy node07
```

3) guestfish 挂载虚拟机磁盘

```
[root@GYP-HOME images]# guestfish -i --network -a node07.img
Welcome to guestfish, the guest filesystem shell for
editing virtual machine filesystems and disk images.

Type: ‘help’ for help on commands
      ‘man’ to read the manual
      ‘quit’ to quit the shell

Operating system: CentOS Linux release 7.4.1708 (Core)
/dev/sda1 mounted on /

><fs> command "yum -y install vsftpd"
><fs> edit /etc/selinux/config
><fs> selinux-relabel /etc/selinux/targeted/contexts/files/file_contexts  /
><fs> exit    //对文件系统中的内容修改实在 guestfish 环境下，不会获得有效的 SElinux 标签，因此需要重打标签。
```