

CentOS7中安装KVM虚拟机基本相同，本文以最新版CentOS为例介绍KVM虚拟机的安装。

宿主机环境：

CentOS Linux release 7.5.1804 (Core)：最小化安装

虚拟机镜像：

CentOS-7-x86_64-DVD-1804.iso

IP：192.168.1.187

安装KVM虚拟机相关软件

1、验证CPU是否支持KVM：结果中有vmx（intel的CPU）或svm（AMD的CPU），则说明支持

```
egrep '(vmx|svm)' /proc/cpuinfo
```

```
[root@think8848-srv02 ~]# egrep '(vmx|svm)' /proc/cpuinfo
flags                : fpu vme de pse tsc mtr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
sse2 ss ht tm pbe syscall nx rdtscp lm constant_tsc arch_perfmon pebs bts nopl xtopology nonstop
pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 sse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic i
r_xsave avx lahf_lm arat epb pln pts dtherm tpr_shadow vnmi flexpriority ept vpid xsaveopt
flags                : fpu vme de pse tsc mtr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
sse2 ss ht tm pbe syscall nx rdtscp lm constant_tsc arch_perfmon pebs bts nopl xtopology nonstop
pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 sse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic i
r_xsave avx lahf_lm arat epb pln pts dtherm tpr_shadow vnmi flexpriority ept vpid xsaveopt
flags                : fpu vme de pse tsc mtr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
sse2 ss ht tm pbe syscall nx rdtscp lm constant_tsc arch_perfmon pebs bts nopl xtopology nonstop
pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 sse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic i
r_xsave avx lahf_lm arat epb pln pts dtherm tpr_shadow vnmi flexpriority ept vpid xsaveopt
```

2、关闭selinux

```
# 临时关闭(服务器重启失效)
setenforce 0
# 永久关闭
# 修改/etc/selinux/config文件中的SELINUX=enforcing为SELINUX=disabled
```

3、安装KVM虚拟机软件

在CentOS7上安装KVM虚拟机软件没必要编译安装，yum安装即可

```
yum -y install qemu-kvm libvirt virt-install bridge-utils libguestfs-tools
# qemu-kvm: kvm虚拟机核心
# libvirt:管理模块
# virt-install:虚拟机安装工具
# bridge-utils:网卡bridge模块
# libguestfs-tools:访问虚拟机磁盘的工具
```

4、验证kvm模块是否安装

```
lsmod | grep kvm
```

5、启动kvm并设置开机自启动

```
systemctl start libvirtd
systemctl enable libvirtd
```

6、验证启动情况

```
systemctl status libvirtd
systemctl is-enabled libvirtd
```

7、配置网桥

①备份/etc/sysconfig/network-scripts/ifcfg-p8p1（名称可能不同，根据实际情况修改），并修改，如下

```
BOOTPROTO=none
DEVICE=p8p1
NM_CONTROLLED=no
ONBOOT=yes
BRIDGE=br0
```

②添加br0，在/etc/sysconfig/network-scripts/目录下创建ifcfg-br0文件，内容如下

```
TYPE=Bridge
BOOTPROTO=static
DEVICE=br0
NM_CONTROLLED=no
IPADDR=192.168.1.187
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
```

③重启网卡

```
systemctl restart network
```

创建虚拟机

1、上传CentOS镜像文件到目录/usr/local/kvm/iso目录中（新建该目录；镜像文件较大，可以通过下载，或者分包压缩方式等上传）

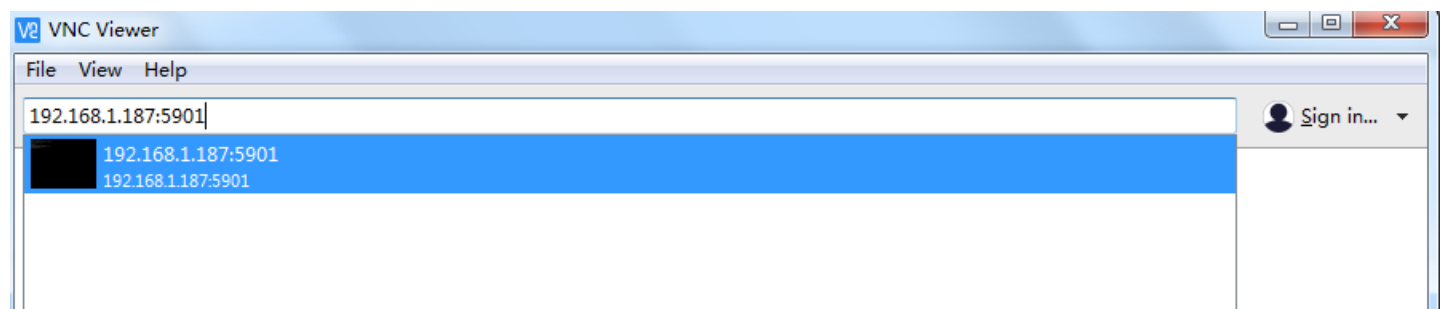
2、创建目录用于存放虚拟机文件

```
mkdir -p /usr/local/kvm/vhost
```

3、创建虚拟机（以名为CentOS7-base的虚拟机为例）

```
virt-install -n CentOS7-base -r 1024 --disk /usr/local/kvm/vhost/CentOS7-base.qcow2,format=qcow2,size=20 --
network bridge=br0 --os-type=linux --os-variant=rhel7 --cdrom=/usr/local/kvm/iso/CentOS-7-x86_64-DVD-
1804.iso --vnc --vncport=5901 --vnclisten=0.0.0.0
# -n CentOS7-base:虚拟机名字为CentOS7-base
# -r 1024:内存大小为1024M (1G)
# --disk /usr/local/kvm/vhost/CentOS7-base.qcow2,format=qcow2,size=20:磁盘数据文件存放
在/usr/local/kvm/vhost/CentOS7-base.qcow2，磁盘格式为qcow2，大小为20G
# --network bridge=br0:通过br0连接网络
# --os-type=linux --os-variant=rhel7:操作系统为rhel7 linux
# --cdrom=/usr/local/kvm/iso/CentOS-7-x86_64-DVD-1804.iso:镜像文件为/usr/local/kvm/iso/CentOS-7-x86_64-DVD-
1804.iso
# --vnc --vncport=5901 --vnclisten=0.0.0.0: vnc连接配置，端口为5901，所有ip都可以连接
```

4、**不要关闭远程连接的软件，也不要终止上述命令的执行**，通过vnc连接该虚拟机，如下图



5、连接成功后，会显示CentOS安装界面，按照正常的步骤即可安装好虚拟机。这时**远程连接软件中创建虚拟机的命令正常结束**。

6、在vnc的连接界面中，配置新创建的虚拟机，配置ip，安装必要的软件等(根据实际需求安装)

```
yum -y install zip unzip ntpdate wget vim
```

7、虚拟机创建完成。

必要的文件存放位置如下：

①虚拟机的配置文件：/etc/libvirt/qemu/CentOS7-base.xml（所有关于磁盘、内存、cpu、网卡等的信息都在该XML文件中）

②虚拟机的磁盘数据：/usr/local/kvm/vhost/CentOS7-base.qcow2

必要的命令：

①查看系统中有哪些虚拟机在启动状态：virsh list

②查看系统中有哪些虚拟机：virsh list --all

③启动虚拟机（以CentOS7-base虚拟机为例）：virsh start CentOS7-base

④停止虚拟机（以CentOS7-base虚拟机为例）：virsh shutdown CentOS7-base

⑤强制停止虚拟机（以CentOS7-base虚拟机为例）：virsh destroy CentOS7-base

⑥修改虚拟机配置文件（xml文件，以CentOS7-base虚拟机为例）：virsh edit CentOS7-base（**打开/etc/libvirt/qemu/CentOS7-base.xml文件并修改，无法使配置生效**）

注意：如果启动与停止虚拟机命令不生效，可能是系统缺少acpi软件（电源管理），在虚拟机（**非宿主机**）中安装acpi即可

```
yum -y install acpi
```

重新加载xml配置文件

重新加载模板文件，一般用于虚拟机迁移，分两种情况：

①新增宿主机，模板虚拟机复用

②用于集群或者其他原因，虚拟机在宿主机中转移等

一般数据文件都很大，内网或者有移动存储设备的话，该方法比较好，不然，不如重新生成个模板。

新宿主机（或另一台无该虚拟机的宿主机）中kvm的配置，要与有该虚拟机的宿主机的kvm的安装配置一样，即对应的目录一致

1、转移数据与xml文件（以内网传输、本文中kvm安装与配置为例）

```
# 传输数据（192.168.1.1.71中/usr/local/kvm/vhost目录已建立）
scp /usr/local/kvm/vhost/CentOS7-base.qcow2 root@192.168.1.71:/usr/local/kvm/vhost
# 传输xml配置文件
scp /etc/libvirt/qemu/CentOS7-base.xml root@192.168.1.72:/etc/libvirt/qemu/
```

2、加载虚拟机

```
virsh define /etc/libvirt/qemu/CentOS7-base.xml
```

克隆虚拟机

kvm克隆虚拟机非常方便、快速，一条命令即可搞定（以用模板创建test虚拟机为例）

1、克隆虚拟机

```
virt-clone -o CentOS7-base -n test -f /usr/local/kvm/vhost/test.qcow2
# -o CentOS7-base: 源（模板）为CentOS7-base
# -n test: 克隆的虚拟机的名字为test
# -f /usr/local/kvm/vhost/test.qcow2: 磁盘数据文件
```

2、配置虚拟机（命令：virsh edit test），修改xml中的vnc相关配置，即可用vnc软件远程连接test虚拟机

```
<graphics type='vnc' port='5902' autoport='no' listen='0.0.0.0'>
# port为宿主机中暂未使用的端口
# autoport改为no，即使用固定端口
# listen='0.0.0.0'即所有的ip都可连接该虚拟机
```

3、与虚拟机CentOS7-base一样，通过vnc连接虚拟机，即可配置网络等信息

扩展虚拟机磁盘（以系统使用LVM为例）

qcow2格式的磁盘，只需要直接扩展虚拟机即可

- 1、关闭虚拟机
- 2、扩展虚拟机磁盘

```
qemu-img resize CentOS7-base.qcow2 +20G
```

- 3、启动虚拟机，并登录
- 4、分区

```
fdisk /dev/vda
# 一次输入: n、p、回车、回车、回车、t、3、8e、w
# 说明: n: 新的分区
#       p: 主分区
#       回车: 这里需要选择分区号, 默认的就好 (本次操作默认是3)
#       回车: 选择分区 (在磁盘中) 开始的位置, 默认就好
#       回车: 选择分区 (在磁盘中) 结束的位置, 如果需要分区几个, 这里填写合适的数值, 本次操作默认就好
#       t: 分区的类型
#       3: 因这里需要对刚产生的分区设置类型, 刚产生的分区号为3, 因此这里填写3, 或者回车
#       8e: lvm的类型代号即为8e
#       w: 把分区信息写入磁盘
```

- 5、重启服务器或者执行partprobe或kpartx
- 6、扩展lvm

```
#扩展lv (ext格式磁盘)
# 创建pv
pvcreate /dev/vda3
# 扩展vg
vgextend centos /dev/vda3
# 扩展lv
lvextend -l +100%FREE /dev/mapper/centos-root
# 重定义lv大小
resize2fs /dev/mapper/centos-root
```

```
# 扩展lv (xfs格式磁盘)
# 创建pv
pvcreate /dev/vda3
# 扩展vg
vgextend centos /dev/vda3
# 扩展lv
lvextend -l +100%FREE /dev/mapper/centos-root
# 重定义lv大小
xfs_growfs /dev/mapper/centos-root
```

- 7、检查磁盘扩容情况

```
df -h
```

附件-克隆虚拟机脚本

```
#!/bin/bash
#
#
TEMP=`getopt -o n:t:m:a:g:h --long hostname:,vmtemp:,help,vmname:,vmip:,vmgw: -n 'createvm.sh' -- "$@"`
if [ $? != 0 ]; then
    echo "Terminating..."
    exit 1
fi

eval set -- "$TEMP"
usage ()
{
    echo "-----usage-----"
    echo -e "-h|--help help\n -t|--vmtemp 要克隆的虚拟机模板名\n -n|--vmname 要创建的虚拟机名\n -m|--hostname 要创建的虚拟机的hostname名字\n -a|--vmip 设置虚拟机的ip\n -g|--vmgw 设置虚拟机的网关\n"
}
while true
```

```

do
case "$1" in
-t | --vmtemp)
case "$2" in
"" ) shift 2 ;;
* ) vmtemp_t=$2; shift 2 ;;
esac;;
-n | --vmname)
case "$2" in
"" ) shift 2 ;;
* ) vmname_n=$2; shift 2 ;;
esac;;
-m | --hostname)
case "$2" in
"" ) shift 2 ;;
* ) hostname_h=$2; shift 2 ;;
esac;;
-a | --vmip)
case "$2" in
"" ) shift 2 ;;
* ) vmip_a=$2; shift 2 ;;
esac;;
-g | --vmgw)
case "$2" in
"" ) shift 2 ;;
* ) vmgw_g=$2; shift 2 ;;
esac;;
-h|--help) usage;exit 1 ;;
-- ) shift; break ;;
* ) echo "internal error"; exit 1 ;;

esac
done
if [ -z $vmtemp_t ] || [ -z $vmname_n ] || [ -z $hostname_h ] || [ -z $vmip_a ] || [ -z $vmgw_g ] ;then
usage
exit 1
fi

echo -e "\033[36;1m -----信息如下-----\033[0m"
echo -e "\033[36;1m 克隆模板名字 -----> ${vmtemp_t} \033[0m"
echo -e "\033[36;1m 虚拟机名字 -----> ${vmname_n} \033[0m"
echo -e "\033[36;1m 虚拟机os的主机名 -----> ${hostname_h} \033[0m"
echo -e "\033[36;1m 虚拟机的ip -----> ${vmip_a} \033[0m"
echo -e "\033[36;1m 虚拟机网关 -----> ${vmgw_g} \033[0m"

echo -n -e "\033[31;1m 是否根据以上配置信息创建虚拟机(y/n): \033[0m"
read answers
echo $answers
if [ $answers = 'y' ] || [ $answers = 'Y' ] ;then

echo -e "\033[34;1m starting to create vm \033[0m"
virt-clone -o ${vmtemp_t} -n ${vmname_n} --auto-clone
[ $? -eq 0 ]&&virt-sysprep -d ${vmname_n} --hostname ${hostname_h} --edit '/etc/sysconfig/network-
scripts/ifcfg-eth0':s/IPADDR=.*IPADDR=${vmip_a}/; s/GATEWAY=.*GATEWAY=${vmgw_g}/'
[ $? -eq 0 ]&& echo -e "\033[34;1m vm is created now !!!!!!!!!!!!!!!!!!!!!!!!!!!!!\n to start ,use command
\033[35;1m virsh start ${vmname_n} \033[0m \033[0m"
else
echo -e "\033[34;1m stop to create vm \033[0m"

fi

```