

对KVM虚拟机进行基本管理

当虚拟机安装完成之后,就要对虚拟机进行管理,管理 KVM 虚拟机有两种方法,一种是图形管理工具 virt-manager,另一种就是命令行工具 virsh 命令,它们都是通过调用 libvirt API 来实现虚拟化管理。本章将具体讲解对虚拟机的基本管理,包括启动、关闭、暂停虚拟机,以及对虚拟机的网卡、硬盘、CPU、内存等进行管理。

- ▶ 学习目标:
- · 掌握使用 virsh 命令来对虚拟机进行基本管理。
- 掌握虚拟机的一些进阶管理。



3.1 使用 virsh 对虚拟机进行基本的管理

virsh 命令提供了两种模式,一种是交互式,一种是非交互式。交互模式是在命令行 提示符下输入 virsh 命令,按 Enter 键后就可以在交互模式下进行操作,另一种是没有交 互的,直接在 Linux 提示符下进行操作,本书基本上采用的是非交互式。下面主要讲解对 虚拟机进行基本管理的命令。

(1) 查看虚拟机。如图 3-1 所示, virsh list 只能显示运行的虚拟机,要查看所有虚拟 机就需加一个--all 选项。

[root@node1 ~]# virsh list Id Name	State
[root@node1 ~]# [root@node1 ~]# virsh listall Id Name	State
- vm01	shut off
[root@node1 ~]#	

图 3-1 查看虚拟机

(2) 启动虚拟机。如图 3-2 所示, start 命令用于启动虚拟机, 后面接虚拟机的名字。

[root@node1 ~]# virsh start vm01 Domain vm01 started [root@node1 ~]# [root@node1 ~]# virsh list Id Name State 3 vm01 running [root@node1 ~]# 图 3-2 启动虚拟机

(3)关闭虚拟机。如图 3-3 所示, shutdown 命令后面加上虚拟机的名字是一种正常 关闭操作系统的方法。

图 3-3 关闭虚拟机

(4)关闭虚拟机。如图 3-4 所示, destroy 命令则是直接拔掉虚拟机电源进行关闭。

[root@node1 ~]# virsh destroy vm01 Domain vm01 destroyed [root@node1 ~]# [root@node1 ~]# virsh list Id Name State

[root@node1 ~]#

图 3-4 强制关闭虚拟机

(5) 挂起虚拟机。如图 3-5 所示,使用 suspend 命令后面加上虚拟机的名字,挂起(也 是暂停)虚拟机。

[root@node1 ~]# virsh list Id Name State - - - -5 vm01 running [root@node1 ~]# virsh suspend vm01 Domain vm01 suspended [root@node1 ~]# virsh list Ιd Name State 5 vm01 paused [root@node1 ~]# 图 3-5 挂起虚拟机

(6)恢复虚拟机。如图 3-6 所示, resume 命令后面加上虚拟机的名字,恢复挂起的状态, 也是唤醒虚拟机。

[root@node1 ~]# virsh resume vm01 Domain vm01 resumed [root@node1 ~]# virsh list Id Name State 5 vm01 running [root@node1 ~]#

图 3-6 恢复虚拟机

(7)设置自启动虚拟机。如图 3-7 所示, autostart 命令后面加上要自启动的虚拟机的名字, 此时虚拟机随物理机启动而启动。

[root@node1 ~]# virsh autostart vm01-clone Domain vm01-clone marked as autostarted

```
[root@node1 ~]#
```

图 3-7 自启动虚拟机

(8) 查看虚拟机的信息。如图 3-8 所示, dominfo 命令后面加上虚拟机的名字, 就可以查看此虚拟机的相关信息, 如 CPU 的名字、UUID、状态、CPU 个数、内存、是否自启动等信息。

[root@node1 ~]#	virsh dominfo vm01
Id:	12
Name:	vm01
UUID:	8a8952da-c9bb-4d1b-a916-5e8bec53a17c
OS Type:	hvm
State:	running
CPU(s):	1
CPU time:	240.6s
Max memory:	1024000 KiB
Used memory:	1024000 KiB
Persistent:	yes
Autostart:	disable
Managed save:	no
Security model:	selinux
Security DOI:	Θ
Security label:	<pre>system_u:system_r:svirt_t:s0:c664,c756 (permissive)</pre>

图 3-8 查看 CPU 相关信息

(9) 删除虚拟机。如图 3-9 所示, undefine 命令后面接虚拟机的名字, 此时虚拟机在 虚拟机管理器里面查不到了。

[root@node1 ~]# virsh undefine vm01
Domain vm01 has been undefined

[root@node1 ~]#

图 3-9 删除虚拟机

(10) 删除虚拟机,并删除磁盘镜像文件。如图 3-10 所示,在图 3-9 的基础上,加上 --storage 选项,将删除虚拟的磁盘镜像文件。

[root@nodel ~]# virsh undefine vm10 --storage /var/lib/libvirt/images/vm10.qcow2 Domain vm10 has been undefined Volume 'vda'(/var/lib/libvirt/images/vm10.qcow2) removed.

[root@node1 ~]#

图 3-10 删除虚拟机并删除磁盘镜像文件

(11) 删除虚拟机,并删除所有磁盘文件。如图 3-11 所示,在图 3-9 的基础上加上 --remove-all-storage 选项,将删除虚拟机的同时,删除虚拟机的所有磁盘文件。

> [root@node1 ~]# virsh undefine vm02 --remove-all-storage Domain vm02 has been undefined Volume 'vda'(/var/lib/libvirt/images/vm02.qcow2) removed.

[root@node1 ~]#

图 3-11 删除虚拟机并删除所有磁盘文件

(12)禁止虚拟机的自启动。如图 3-12 所示,在自启动配置的基础上加上--disable 选项,此虚拟机随着主机启动,不启动此虚拟机。

[root@node1 ~]# virsh autostart vm01-clone --disable Domain vm01-clone unmarked as autostarted

[root@node1 ~]# _

图 3-12 禁止虚拟机的自启动

(13)显示虚拟机当前的配置文件。如图 3-13 所示,在 dumpxml 命令后加上虚拟机的名字,将显示此虚拟机的配置文件。

图 3-13 显示虚拟机的配置文件

3.2 对虚拟机进行进阶管理

3.2.1 给虚拟机添加网卡

通过 virsh 命令也同样可以给虚拟机添加网卡,其步骤如下:



(1) 在虚拟机里面查看当前虚拟机的 IP 地址,发现只有一块网卡 eth0,如图 3-14 所示。

```
[root@localhost ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:35:c6:51 brd ff:ff:ff:ff:ff
    inet 192.168.122.118/24 brd 192.168.122.255 scope global dynamic eth0
       valid_lft 3583sec preferred_lft 3583sec
    inet6 fe80::5054:ff:fe35:c651/64 scope link
       valid_lft forever preferred_lft forever
[root@localhost ~]#
```

图 3-14 查看虚拟机的网卡

(2) 在宿主机上查看网卡信息,如图 3-15 所示,domiflist 后面接上虚拟机的名字,发现也只有一块网卡 vnet0。

[root@nodel ~]# virsh domiflist vm01 Interface Type Source Model MAC vnet0 network default virtio 52:54:00:35:c6:51 [root@nodel ~]# 图 3-15 查看虚拟机 vm01的网卡

(3)使用 virsh 命令添加一块网卡,如图 3-16 所示, attach-interface 后面接上虚拟机的名字, network 是网络类型, default 是此虚拟机的网络源, default 网络默认是 NAT 连网方式, --persistent 选项则是永久的意思, 也就是意味着重启虚拟机后此网卡还会生效。

[root@nodel ~]# virsh attach-interface vm01 network default --persistent Interface attached successfully

[root@node1	l ~]#			
[root@node]	l ∼]# virsh	domiflist	vm01	
Interface	Туре	Source	Model	MAC
vnet0	network	default	virtio	52:54:00:35:c6:51
vnet2	network	default	rtl8139	52:54:00:5f:ab:81

[root@node1 ~]#

46

图 3-16 给虚拟机 vm01 添加网卡

(4) 在虚拟机里查看网卡,如图 3-17 所示,发现此时多了一块网卡 ens10。

3.2.2 给 vm01 添加磁盘

当虚拟机的存储容量不够时,就可以通过下面的方法来进行添加磁盘,具体操作步骤 如下:

(1) 在宿主机上查看虚拟机的磁盘信息,如图 3-18 所示,发现虚拟机 vm01 只有一块 磁盘 vda,磁盘对应的文件是 vm01. qcow2。

[root@localhost ~]# ip addr 1: lo: <LOOPBACK,UP,LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN glen 1 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lo valid lft forever preferred lft forever inet6 ::1/128 scope host valid lft forever preferred lft forever 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000 link/ether 52:54:00:35:c6:51 brd ff:ff:ff:ff:ff:ff inet 192.168.122.118/24 brd 192.168.122.255 scope global dynamic eth0 valid lft 3364sec preferred lft 3364sec inet6 fe80::5054:ff:fe35:c651/64 scope link valid lft forever preferred lft forever 3: ens10: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 gdisc pfifo fast state UP glen 1000 link/ether 52:54:00:5f:ab:81 brd ff:ff:ff:ff:ff:ff 图 3-17 杳看网卡 [root@node1 ~]# virsh domblklist vm01

Target Source vda /var/lib/libvirt/images/vm01.qcow2

[root@node1 ~]#

图 3-18 查看虚拟机磁盘情况

(2) 通过 dd 命令生成大小为 500MB 的文件来作为虚拟机的磁盘文件,也可以通过 qemu-img 命令来生成磁盘文件,如图 3-19 所示,并使用 attach-disk 命令将新生成的磁盘 添加到虚拟机 vm01 中,作为虚拟机的 vdb,vd 为 virtual disk,b 代表第二块磁盘。

```
[root@node1 ~]# dd if=/dev/zero of=/var/lib/libvirt/images/vm01_1.img bs=1M count=500
500+0 records in
500+0 records out
524288000 bytes (524 MB) copied, 10.28 s, 51.0 MB/s
[root@node1 ~]#
[root@node1 ~]# virsh attach-disk vm01 /var/lib/libvirt/images/vm01_1.img vdb
Disk attached successfully
[root@node1 ~]# virsh domblklist vm01
Target Source
```

vda /var/lib/libvirt/images/vm01.qcow2 vdb /var/lib/libvirt/images/vm01 1.img

[root@node1 ~]#

图 3-19 给虚拟机 vm01 添加新的磁盘

(3) 在虚拟机里也可以通过 lsblk 命令查看新的磁盘,如图 3-20 所示。

[root@localhost ~]# lsblk NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT vda 252:0 9G 0 disk 0 —vda1 252:1 0 1G 0 part /boot -vda2 252:2 0 8G 0 part 0 lvm / -rhel-root 253:0 0 7.1G 924M -rhel-swap 253:1 0 0 lvm [SWAP] vdb 500M 252:16 0 0 disk [root@localhost ~]#

图 3-20 查看磁盘

3.2.3 修改 CPU 与内存

48

调整 CPU 与内存的大小有三种方法,包括配置文件、图形、命令的方式,下面具体来进行介绍。

方法一:通过修改配置文件的方式来调整 CPU 与内存的大小,具体步骤如下:

(1) 使用 lscpu 命令查看 CPU, 如图 3-21 所示, 发现当前只有一个 CPU。

<pre>[root@localhost ~]# lso</pre>	opu
Architecture:	x86_64
CPU op-mode(s):	32-bit, 64-bit
Byte Order:	Little Endian
CPU(s):	1
On-line CPU(s) list:	Θ
Thread(s) per core:	1
Core(s) per socket:	1
Socket(s):	1
NUMA node(s):	1
Vendor ID:	GenuineIntel
CPU family:	6
Model:	61
Model name:	Intel Core Processor (Broadwell)
Stepping:	2
CPU MHz:	3599.998
BogoMIPS:	7199.99
Hypervisor vendor:	KVM
Virtualization type:	full
L1d cache:	32K
Lli cache:	32K
L2 cache:	4096K
NUMA node0 CPU(s):	0

图 3-21 查看 CPU 的个数

(2) 通过 free 命令查看内存的大小,如图 3-22 所示,发现内存为 992MB。

[root@local	host ~]# fre	e-m				
	total	used	free	shared	buff/cache	available
Mem:	992	102	746	6	143	733
Swap:	923	Θ	923			
[root@local	host ~]#					

图 3-22 查看内存大小

(3) 通过 virsh edit 命令修改配置文件,调整 CPU 与内存大小,如图 3-23 所示。

[root@node1 ~]# virsh edit vm01
Domain vm01 XML configuration edited.

[root@node1 ~]#

图 3-23 修改虚拟机的配置文件

(4) 通过图 3-23 的命令打开虚拟机的配置文件后,修改内存与 CPU 的大小,如图 3-24 所示,将内存修改为 2048MB,将 CPU 修改为 2 个,此时需要注意的是 CPU 与内存的大小是不能超过宿主机的。

(5) 重启虚拟机,让其生效,如图 3-25 所示,但是需要注意的是一定要先关机,再启

```
<name>vm01</name>
<uuid>8a8952da-c9bb-4d1b-a916-5e8bec53a17c</uuid>
<memory unit='KiB'>2048576</memory>
<currentMemory unit='KiB'>2048576</currentMemory>
<vcpu placement='static'>2</vcpu>
```

图 3-24 修改 CPU 与内存的大小

动才会生效,直接重启是不会生效的。

[root@nodel ~]# virsh shutdown vm01
Domain vm01 is being shutdown

[root@node1 ~]# virsh start vm01
Domain vm01 started

[root@node1 ~]#

图 3-25 重启虚拟机 vm01

(6) 重启后查看 CPU 与内存,如图 3-26 所示,发现 CPU 现在是 2 个,内存为 1953MB(有点误差,是正常范围可接受的)。

[root@localhost ~]# lscpu |grep '^CPU(s)' CPU(s): 2 [root@localhost ~]# [root@localhost ~]# free -m total used free shared buff/cache available Mem: 1953 107 1709 8 136 1687 923 0 923 Swap: [root@localhost ~]#

图 3-26 查看 CPU 与内存的大小

方法二:通过图形的方式来调整 CPU 与内存的大小,具体步骤如下:

(1) 打开虚拟机 vm01 图形界面中的详情页,如图 3-27 所示,选中 CPUs 面板,可以 查看当前 CPU 与最大 CPU 的配置,也可以修改其值来调整 CPU 的个数。

		vmO1 on QEMU/KVM		
File	Virtual Machine View	Send Key		
ē		•		
2	Overview	CPUs		
-1e-1e	Performance	Logical host CPUs: 2		
-	CPUs	Current allocation: 1 - +		
	Memory			
33	Boot Options	Maximum allocation:		
	VirtIO Disk 1	Configuration		
P	NIC :35:c6:51	Copy host CPU configuration		
P	NIC :5f:ab:81	Model: Broadwell		
	Tablet			
ð	Mouse	▶ Topology		

图 3-27 调整 CPU 参数

(2) 选择详情页的 Memory 面板,如图 3-28 所示,也可以调整当前分配内存大小与 最大分配内存的大小。

		vmO1 on QEMU/KVM
<u>F</u> ile	Virtual <u>M</u> achine <u>V</u> iew	Send <u>K</u> ey
ē		
2	Overview	Memory
44	Performance	Total host memory: 8191 MiB
	CPUs	Current a <u>l</u> location: 1000 — + MiB
	Memory	
60	Boot Options	
	VirtIO Disk 1	
ġ.	NIC :35:c6:51	

图 3-28 调整内存参数

方法三:通过命令的方式来调整 CPU 与内存的大小,具体步骤如下: (1) 调整 CPU 大小,如图 3-29 所示,查看到只有一个 CPU。

[root@node1 ~]# virsh dominfo vm01-clone

Ta.

IU.	-
Name:	vm01-clone
UUID:	f56bbb73-1eef-445e-b7e1-6083e65c1bd0
OS Type:	hvm
State:	shut off
CPU(s):	1
Max memory:	1048576 KiB
Used memory:	1048576 KiB
Persistent:	yes
Autostart:	disable
Managed save:	no
Security model:	selinux
Security DOI:	0
[root@node1 ~]#	
[root@node1 ~]#	virsh start vm01-clone
Domain vm01-clor	ne started

图 3-29 查看 CPU 个数

(2) 设置 CPU 的个数,如图 3-30 所示,将最大允许分配的 CPU 个数设置为 2 个, --config 是修改配置文件,也就意味着重启还会生效。

(3) 调整 CPU 的个数,如图 3-31 所示,给虚拟机的 CPU 在线调整至 2 个,也就意味着立即生效,需要注意的是在线调整 CPU 个数是不能超过最大的 CPU 个数。减少 CPU 必须重启生效,并且不能在线减少 CPU。

(4) 设置内存,如图 3-32 所示,设置最大内存为 4096MB,-- config 选项同样意味着 将写入配置文件,重启之后还将生效。

(5) 查看虚拟机的信息,如图 3-33 所示,发现现在最大内存约为 4GB,当前内存还是 约为 1GB。

```
[root@node1 ~]# virsh setvcpus vm01-clone 2 --maximum --config
[root@node1 ~]#
[root@node1 ~]# virsh dominfo vm01-clone
Id:
                13
Name:
                vm01-clone
UUID:
                f56bbb73-1eef-445e-b7e1-6083e65c1bd0
OS Type:
                hvm
State:
                running
CPU(s):
                1
CPU time:
                67.0s
                1048576 KiB
Max memory:
Used memory:
                1048576 KiB
Persistent:
               yes
Autostart:
                disable
Managed save:
               no
Security model: selinux
Security DOI:
                0
Security label: system u:system r:svirt t:s0:c494,c867 (permissive)
[root@node1 ~]#
                 图 3-30 设置允许分配的 CPU 的个数
[root@node1 ~]# virsh setvcpus vm01-clone 2 --live
[root@node1 ~]#
[root@node1 ~]# virsh dominfo vm01-clone
Id:
                14
Name:
                vm01-clone
UUID:
                f56bbb73-1eef-445e-b7e1-6083e65c1bd0
OS Type:
               hvm
State:
                running
CPU(s):
                2
CPU time:
                62.0s
Max memory:
                1048576 KiB
Used memory:
                1048576 KiB
Persistent:
                yes
Autostart:
                disable
Managed save:
                no
Security model: selinux
Security DOI:
                0
Security label: system_u:system_r:svirt_t:s0:c734,c971 (permissive)
                      图 3-31 调整 CPU 的个数
```

[root@node1 ~]# virsh setmaxmem vm01-clone 4096M --config

[root@nodel ~]#

图 3-32 调整最大允许分配给虚拟机的内存大小

(6)调整当前内存的大小,如图 3-34 所示,将当前内存调整到 2048MB,其中 --current选项意味着当前的意思,调整后,通过 dominfo 进行查看,发现当前内存已调整 到 2GB 了。

```
[root@node1 ~]# virsh dominfo vm01-clone
Td:
                15
Name:
                vm01-clone
UUID:
                f56bbb73-1eef-445e-b7e1-6083e65c1bd0
OS Type:
                hvm
State:
                running
CPU(s):
                1
CPU time:
                69.8s
                4194304 KiB
Max memory:
Used memory:
                1048576 KiB
Persistent.
                Ves
Autostart:
                disable
Managed save:
                no
Security model: selinux
Security DOI:
                ſ٩
Security label: system_u:system_r:svirt_t:s0:c7,c299 (permissive)
                      图 3-33 查看内存的大小
[root@node1 ~]# virsh setmem vm01-clone 2048M --current
[root@node1 ~]#
[root@node1 ~]# virsh dominfo vm01-clone
Id:
                15
Name:
                vm01-clone
                f56bbb73-1eef-445e-b7e1-6083e65c1bd0
UUID:
OS Type:
                hvm
State:
                running
CPU(s):
                1
CPU time:
                81.5s
Max memory:
                4194304 KiB
Used memory:
                2097152 KiB
Persistent:
                yes
                disable
Autostart:
Managed save:
                no
Security model: selinux
Security DOI:
                Θ
Security label: system_u:system_r:svirt_t:s0:c7,c299 (permissive)
```

图 3-34 调整当前内存大小

3.3 本章实验

52

3.3.1 实验目的

▶ 掌握使用 virsh 命令来对 KVM 虚拟机进行基本管理。
 ▶ 掌握 KVM 虚拟机的一些进阶管理,包括磁盘、网络、CPU 与内存等。

3.3.2 实验环境

在安装好 KVM 的宿主机 nodel 上去管理虚拟机 VM01。

3.3.3 实验拓扑

实验拓扑图如图 3-35 所示。

GuestOS(RHEL7) GuestOS(RHEL7)			
vm01 vm01			
RHEL7(KVM)	RHEL7(KVM)	RHEL7	
node1	node2	storage	
VMware Workstation			
Windows 10			
Hardware			

图 3-35 实验拓扑图

3.3.4 实验内容

如图 3-35 所示,在安装好 KVM 的宿主机 nodel 上去管理虚拟机 VM01。一是使用 virsh 命令对虚拟机 VM01 执行基本操作,包括查看虚拟机、开机、关机、暂停、唤醒、删除 等。二是使用 virsh 命令对虚拟机 VM01 执行进阶操作,包括添加一块磁盘、一块网卡, 以及调整 VM01 的 CPU 与内存大小。

说一说:

在调整 CPU 的三种方法中,你认为哪一种方法最好用?为什么?