

云计算之路

虚拟化环境搭建及 虚拟机创建

.....
成/都/虫/洞/奇/迹/科/技/有/限/公/司

2017/11

版权声明

版权所有 © 虫洞奇迹科技有限公司 2017。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



和其他成都虫洞奇迹科技有限公司商标均为成都虫洞奇迹科技有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受成都虫洞奇迹科技有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，成都虫洞奇迹科技有限公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

成都虫洞奇迹科技有限公司
电话：400-090-2980
邮箱：contact@lingyuecloud.com
网址：www.lingyuecloud.com

目录

版权声明	2
1. 前言	4
2. 虚拟化环境安装	5
2.1 宿主机操作系统安装	5
2.1.1 QEMU/KVM 安装	6
2.1.2 virt-viewer 安装	7
3. 创建虚拟机	7
3.1 镜像准备	7
3.2 启动虚拟机	7
3.2.1 使用 QEMU 命令启动虚拟机	7
3.2.2 使用 libvirt 启动虚拟机	9
3.2.3 使用 virt-manager 启动虚拟机	12
4. 总结	13

1. 前言

在计算机技术中，虚拟化(Virtualization)是一种资源管理技术，它将计算机相关的各种资源（CPU、内存、磁盘、网络适配器等）进行抽象、转换后重新分配使用，大大增加了使用的灵活性。虚拟化有很多类别，包括硬件虚拟化、操作系统级虚拟化、应用虚拟化、服务虚拟化等。本文中所讨论的虚拟化环境指硬件虚拟化环境，即通过使用虚拟机监视器（Virtual Machine Monitor），隐藏底层硬件信息，虚拟出通用的计算环境（即虚拟机）。虚拟机中运行的操作系统称为客户机操作系统（Guest OS），虚拟机监视器(Virtual Machine Monitor)所在的操作系统成为宿主机操作系统(Host OS)。

当前硬件虚拟化技术有很多实现，包括 VMware ESXi、XEN、QEMU/KVM、Microsoft Hyper-V、VirtualBox 等。其中，QEMU/KVM 由于 KVM 代码简洁，复用了 Linux Kernel 中的 CPU 和内存调度等功能，进入了 Linux Kernel 主干，因此最受欢迎。在笔者撰写此文时，AWS 刚刚宣布放弃 XEN，改用 KVM 作为其虚拟化引擎。本文在此也主要介绍 QEMU/KVM 虚拟化环境的搭建。

QEMU(Quick Emulator)是最早由 Fabrice Bellard 所编写的硬件虚拟化软件，通过对指令进行动态翻译来实现虚拟化；同时，还支持模拟多种设备。尽管 QEMU 的动态指令翻译已经做得很好了，然而还是不可避免的存在性能损失，而 KVM 和硬件辅助虚拟化技术解决了该问题。KVM（Kernel-based Virtual Machine）是 Linux Kernel 中的一个模块，它借助 CPU 的虚拟化特性，实现客户机和宿主机运行状态的切换。有了 KVM，QEMU 只需要负责模拟其他硬件，CPU 和内存的虚拟化借助 KVM 即可完成，大大提高了虚拟化的性能。

2. 虚拟化环境安装

2.1 宿主机操作系统安装

灵跃桌面云选用 CentOS 7 X86_64 位版本作为宿主机操作系统进行安装和部署。从 [CentOS 官方网站](#) 下载其发行版的 ISO 文件（初学者建议下载 Everything ISO，其中包含了所需的所有软件包，省去了安装时联网下载过程）。宿主机操作系统建议安装在硬件（PC 或者服务器）之上，若没有硬件环境，可在 VMware Workstation 中进行安装。在安装过程中“软件选择”时，请勾选“虚拟化主机”中的“虚拟化平台”、“GNOME 桌面”中的“GNOME 应用程序”与“传统 X Windows 系统的兼容性”，如图所示：



图 1 宿主机操作系统安装（1）



图 2 宿主机操作系统安装（2）

若在安装过程中没有创建新的系统用户，则在首次进入 GNOME 图形界面后会要求创建，待创建用户之后，即可开始使用。

2.1.1 QEMU/KVM 安装

由于 KVM 是 Linux Kernel 的一个模块，因此在系统安装好后，KVM 也安装完成；同时，对于 KVM 的版本号，由于 KVM 随着 Linux Kernel 一起发布，所谓 KVM 的版本就是 Linux Kernel 的版本。

因为前文在系统安装时选择了“虚拟化平台”选项，因此 QEMU 也已经自动被安装。若读者需要在已有的 CentOS 系统中安装 QEMU，灵跃桌面云建议使用以下命令手动安装（其他不同发行版的 Linux 操作系统，安装命令会有所不同）：

```
[lingyuecloud@lingyuecloud ~]$ sudo yum install qemu-kvm
```

要检验是否已经安装 QEMU，可以使用 rpm 命令查询，如下图所示：

```
[lingyuecloud@lingyuecloud ~]$ sudo rpm -qa | grep qemu
ipxe-roms-qemu-20170123-1.git4e85b27.el7.noarch
```

```
libvirt-daemon-driver-qemu-3.2.0-14.el7.x86_64
qemu-img-1.5.3-141.el7.x86_64
qemu-kvm-1.5.3-141.el7.x86_64
qemu-kvm-common-1.5.3-141.el7.x86_64
qemu-guest-agent-2.8.0-2.el7.x86_64
```

2.1.2 virt-viewer 安装

在启动虚拟机之后，可以使用远程桌面连接软件来交付虚拟机桌面。由于 QEMU 支持基于 VNC 和 Spice 的交付方式，这里使用支持上述两种交付模式的 virt-viewer 包中的 remote-viewer 程序作为连接的客户端。

使用如下命令安装 virt-viewer 即可。

```
[lingyuecloud@lingyuecloud ~]$ sudo yum install virt-viewer
```

3. 创建虚拟机

3.1 镜像准备

灵跃桌面云建议初学者使用现有镜像来创建虚拟机，待熟悉之后再手动制作镜像。现有镜像可以从 [OpenStack 官方网站](#) 上下载，推荐下载 [CirrOS 镜像](#)，其是体积非常小的 Linux 镜像，非常适合用于测试场景。

3.2 启动虚拟机

3.2.1 使用 QEMU 命令启动虚拟机

通过 QEMU 启动虚拟机的命令如下：

```
[lingyuecloud@lingyuecloud ~]$ /usr/libexec/qemu-kvm -hda cirros-0.3.5-x86_64-disk.img -m 256 -boot c -vnc :0
Could not access KVM kernel module: No such file or directory
failed to initialize KVM: No such file or directory
Back to tcg accelerator.
```

其中，后面三行打印日志是说当前环境下的 KVM 支持未启用，将采用指令翻译模式运行。针对该问题，若要启用 KVM 支持，则需要结合宿主主机操作系统所在的环境进行考虑：（1）若宿主机操作系统直接安装在物理硬件之上，则需要 BIOS 中设置开启虚拟化支持；（2）若宿主机操作系统安装在 VMware Workstation 中，需要在 VMware 环境设置的“处理器设置”选项中勾选“虚拟化 Intel VT-x/EPT 或 AMD-V/RVI(V)”选项。

启动命令中各参数含义如下：

- “-usr/libexec/qemu-kvm”：QEMU 二进制可执行文件完整路径；
- “-hda cirros-0.3.5-x86_64-disk.img”：使用下载的 cirros 镜像作为第 1 块虚拟硬盘；
- “-m 256”：内存设置为 256MB；
- “-boot c”：设置为首先从硬盘启动；
- “-vnc :0”：在端口 0（连接的时候使用 5900）上开启 vnc 服务；

虚拟机运行之后，即可使用 remote-viewer 进行连接。在本地 Linux 环境下在 shell 中运行“remote-viewer”命令，然后输入连接地址“vnc://127.0.0.1:5900”即可连接到虚拟机，如图所示：

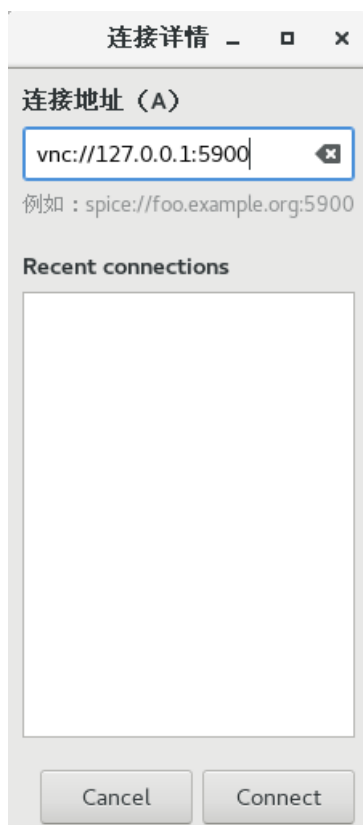


图 3 使用 remote-viewer 连接虚拟机 (1)

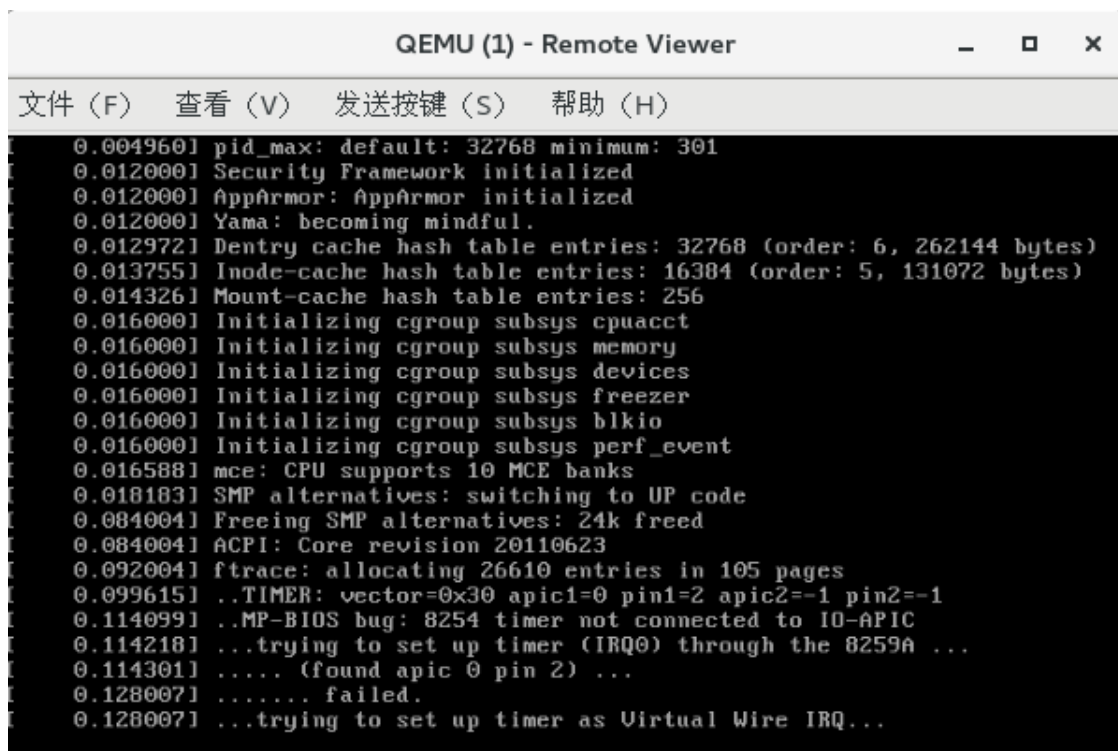


图 4 使用 remote-viewer 连接虚拟机 (2)

若要关闭虚拟机，可在虚拟机窗口中执行关机命令，或在执行 QEMU 命令的 Shell 窗口运行“Ctrl+C”命令（注意：此操作结束 QEMU 命令相当于对 PC 进行断电操作，会丢失 Guest OS 中未及时保存的数据）即可。

3.2.2 使用 libvirt 启动虚拟机

libvirt 是为了更方便地管理虚拟化平台而设计的开源应用程序接口、守护进程和管理工具，不仅提供了对虚拟化客户机的管理，也提供了对虚拟化网络和存储的管理。libvirt 支持多种底层虚拟化技术，屏蔽了底层细节，使得对虚拟机的管理更加方便。

默认情况下，CentOS 并未安装 libvirt，读者可使用如下命令进行安装：

```
[lingyuecloud@lingyuecloud ~]$ sudo yum install libvirt
```

安装完成后，可使用如下命令查看 libvirt 版本。本文撰写时，CentOS 官方

libvirt 已经更新到 3.2.0 版本。

```
[lingyuecloud@lingyuecloud ~]$ virsh --version
3.2.0
```

libvirt 通过 xml 文件来对虚拟机的配置进行管理，在启动虚拟机之前，我们首先需要创建一个 xml 文件，内容如下所示。其中红色的注释，简单解释了 xml 各个字段的含义。

```
<domain type='kvm'> <!-- 这里的'kvm'表示使用 kvm 加速，也可以填'qemu'，则虚拟机是 qemu 全模拟的 -->
<name>lingyuecloud</name> <!-- 虚拟机名字，用户自己定义 -->
<os>
<type arch='x86_64' machine='pc'>hvm</type> <!-- 虚拟机架构以及类型，如果要创建 32 位虚拟机，这里可以将 x86_64 改成 i686 -->
<boot dev='hd'> <!-- 此项表示首先从硬盘启动，如果需要首先从 CDROM 启动，用户可以在前面加上<boot dev='cdrom'> -->
</os>
<features> <!-- 定义一些需要传给虚拟机的 VCPU 的特性 -->
<acpi/>
<apic/>
</features>
<memory unit='KiB'>262144</memory> <!-- 虚拟机内存规格 -->
<vcpu>1</vcpu> <!-- 虚拟机 VCPU 数量 -->
<clock offset='utc'> <!-- 设置虚拟机的时钟，不同时钟之间的区别可以参考内核文档 Timekeeping Virtualization for X86-Based Architectures -->
<timer name='pit' tickpolicy='delay'>
<timer name='rtc' tickpolicy='catchup'>
<timer name='hpet' present='no'>
</clock>
<on_poweroff>destroy</on_poweroff> <!-- 设置虚拟机内部 poweroff 的时候，libvirt 的动作为 destroy -->
<on_reboot>restart</on_reboot> <!-- 设置虚拟机内部 reboot 的时候，libvirt 的动作为 restart -->
<on_crash>destroy</on_crash> <!-- 设置虚拟机内部 crash 的时候，libvirt 的动作为 destroy；此项对 debug 很有用，debug 时设置此项为 pause，
可以在虚拟机内部 crash 的时候，libvirt 使得虚拟机暂停，此时开发人员可以将虚拟机内存 dump 出来进行分析 -->
<devices> <!-- device 标签下都是定义要呈现给虚拟机的各种设备 -->
<emulator>/usr/libexec/qemu-kvm</emulator> <!-- QEMU 二进制文件的完整路径 -->
<disk type='file' device='disk'> <!-- 定义一块磁盘 -->
<driver name='qemu' type='qcow2' cache='none'> <!-- 磁盘类型以及 cache 设置 -->
<source file='/home/lingyuecloud/cirros-0.3.5-x86_64-disk.img'>
<target dev='vda' bus='virtio'> <!-- vda 表示硬盘设备名，或者说盘符，每个硬盘一个，不能重复，后续依次为 vdb、vdc 等，bus 表示设备类
型，如果是 virtio，系统必须有 virtio 驱动；若系统没有 virtio 驱动则可以改成 ide 类型： -->
</disk>
<controller type='ide' index='0'>
<controller type='usb' index='0'>
<input type='tablet' bus='usb'> <!-- 键盘，此处为 USB 类型 -->
<input type='mouse' bus='ps2'> <!-- 鼠标，此处为 PS2 类型 -->
<graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0' keymap='en-us'> <!-- vnc 设置，通常 port 设置为-1，libvirt 会自动分配 port -->
```

```
<video> <!-- 显示设置 -->
    <model type='cirrus' vram='32768' heads='1'/>
</video>
</devices>
</domain>
```

利用 libvirt xml 文件，通过 virsh 命令即可实现对虚拟机的生命周期管理。其中，virsh 有两种使用方式：（1）在 virsh 命令后直接跟相关子命令和参数；（2）在输入 virsh 命令并敲击回车进入 virsh 环境后，再输入相关子命令和参数。本文使用 virsh 后紧跟子命令的用法。

利用 virsh 创建虚拟机的步骤包括 define 和 start 两个步骤。（1）依据创建的虚拟机配置文件，使用“virsh define 虚拟机配置文件.xml”命令定义后的该虚拟机就在 libvirt 的管理之下了；即使 libvirt 服务重启或者宿主机系统重启，define 过的虚拟机依然存在。define 之后虚拟机不会启动；使用“virsh list”命令查看虚拟机列表时，该虚拟机默认不显示，需要加“--all”参数。（2）虚拟机 define 之后，使用“virsh start 虚拟机名”命令来启动，上述过程如下所示：

```
[lingyuecloud@lingyuecloud ~]$ virsh define cirros-0.3.5-x86_64-disk.xml
```

```
定义域 lingyuecloud (从 cirros-0.3.5-x86_64-disk.xml)
```

```
[lingyuecloud@lingyuecloud ~]$ virsh list
```

Id	名称	状态

```
[lingyuecloud@lingyuecloud ~]$ virsh list --all
```

Id	名称	状态

-	lingyuecloud	关闭
---	--------------	----

```
[lingyuecloud@lingyuecloud ~]$ virsh start lingyuecloud
```

```
域 lingyuecloud 已开始
```

```
[lingyuecloud@lingyuecloud ~]$ virsh list
```

Id	名称	状态

1	lingyuecloud	running
---	--------------	---------

```
[lingyuecloud@lingyuecloud ~]$ virsh domdisplay lingyuecloud
```

```
vnc://localhost:1
```

之后，可使用“`virsh domdisplay`”命令来查看虚拟机 vnc 的连接端口，根据返回的端口号，可使用 `remote-viewer` 连接地址“`vnc://127.0.0.1:5901`”即可连接到该虚拟机。

若要关闭虚拟机，可执行“`virsh shutdown 虚拟机名`”命令（发送关机信号到 Guest OS 中）或者“`virsh destroy 虚拟机名`”命令（强制关闭虚拟机，可能造成数据丢失）。其他 libvirt 命令读者可执行“`virsh help`”进行查阅。

3.2.3 使用 virt-manager 启动虚拟机

`virt-manager` 是一个图形化的虚拟机管理工具，相比命令行，`virt-manager` 操作更加直观且不容易出错。CentOS 系统默认没有安装 `virt-manager`，用户需要使用以下命令来安装：

```
[lingyuecloud@lingyuecloud ~]$ sudo yum install virt-manager
```

安装完成后，在 shell 中运行命令“`sudo virt-manager`”来打开 `virt-manager` 的图形界面，然后点击“文件”->“新建虚拟机”即可创建新的虚拟机。创建之前，先将下载的 CirrOS 镜像复制一份，创建过程中，选择“导入现有磁盘映像(E)”，然后浏览选择到复制的 CirrOS 镜像，其他选项默认即可，如图所示：

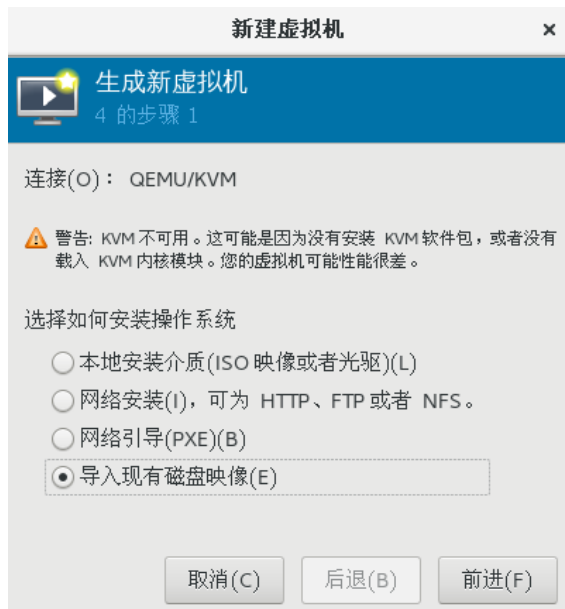


图 5 新建虚拟机（1）



图 6 新建虚拟机（2）

虚拟机运行后，`virt-manager` 会自动连接到虚拟机的控制台，用户可以看到

虚拟机启动过程并进行操作。在控制台界面上，点击“查看”->“详情”还可查看和编辑虚拟机的详细配置（部分配置需要关机后才能修改、部分配置编辑后需要重启生效），如图所示：

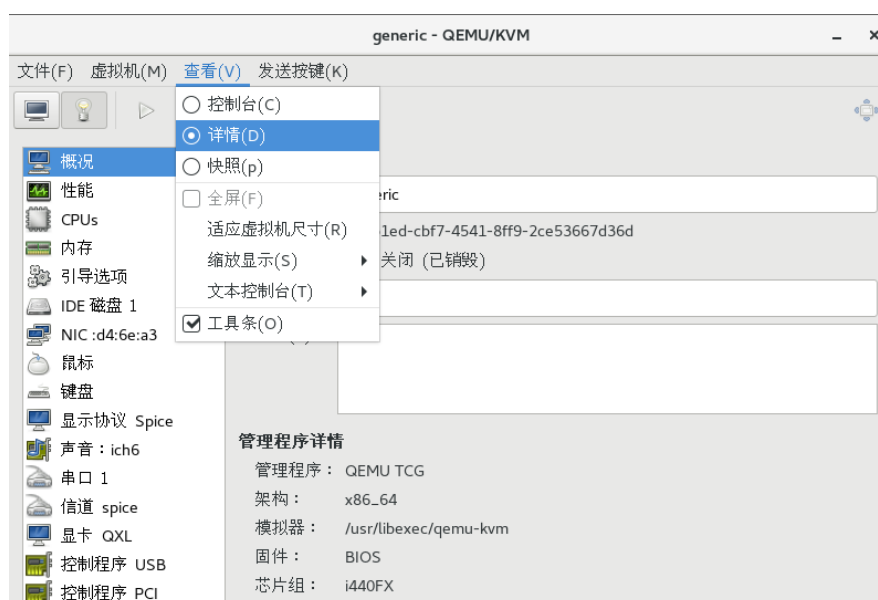


图 7 编辑虚拟机配置

4. 总结

本文介绍了简易的虚拟化环境的安装搭建以及虚拟机的创建过程。总体上讲，不论是 QEMU 命令操作、libvirt 命令操作还是利用 virt-manager 图形化管理工具，对虚拟机的管理都略显复杂，尤其涉及到更多功能操作时如网络虚拟化、存储虚拟化、虚拟机连接认证等更是难以管理。针对上述问题，灵跃桌面云建议采用云计算平台管理工具进行统一管理，其更加高效、简洁，相关讨论将持续更新，敬请关注。