Linux服务器构建与运维管理

第09章：使用KVM实现虚拟化

阮晓龙

13938213680 / ruanxiaolong@hactcm.edu.cn

https://internet.hactcm.edu.cn
http://www.51xueweb.cn

河南中医药大学信息技术学院互联网技术教学团队
河南中医药大学医疗健康信息工程技术研究所

2023.9

提纲

河南中医药大学信息技术学院互联网技术教学团队 / https://internet.hactcm.edu.cn / http://www.51xueweb.cn
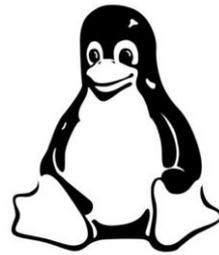
1.虚拟化技术

- 虚拟化技术（Virtualization）是伴随着计算机技术的产生而出现的，在计算机技术的发展历程中一直扮演着重要的角色。
  - 虚拟化，是指通过虚拟化技术将一台计算机虚拟为多台逻辑计算机。在一台计算机上同时运行多个逻辑计算机，每个逻辑计算机可运行不同的操作系统，并且应用程序都可以在相互独立的空间内运行而互不影响，从而显著提高计算机的工作效率。
  - 虚拟化，是一种资源管理技术，是将计算机的各种实体资源，如服务器、网络、内存及存储等，予以抽象、转换后呈现出来，打破实体结构间的不可切割的障碍，使用户可以更好应用资源。
  - 虚拟化，是一个为了简化管理、优化资源分配的解决方案。


1.虚拟化技术

- 虚拟化技术通过把物理资源抽象转换为逻辑上可以管理的资源，达到整合简化物理基础设施架构、提高资源整体利用率、降低运维管理成本等目标，解决物理基础设施之间耦合性强的弊端，实现基于业务运行实际而弹性的自动化分配资源。
- 虚拟化技术通过透明化底层物理硬件达到最大化利用物理硬件的目标，解决高性能的物理硬件产能过剩和老旧硬件产能过低的重组重用。简单来说，就是将底层资源进行分区，并向上层提供特定的、多样化的运算环境。
- 虚拟化技术通过有效管理虚拟资源和物理资源之间的映射关系，达到充分共享物理资源的目标，解决应用系统从资源独占到资源共享的转变，实现业务服务的高可用。
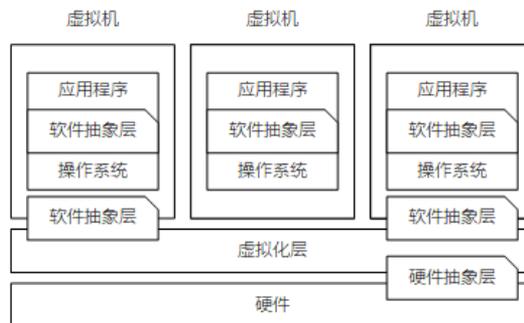
1.虚拟化技术

5

1.虚拟化技术

- 根据实现方式不同，虚拟化技术可以分为全虚拟化、半虚拟化、操作系统级虚拟化等。
  - 全虚拟化
    - 在全虚拟化中，虚拟机（"guest"，客户机）和硬件之间，安装有"Hypervisor（超级管理器）"。Hypervisor是一切硬件资源的管理者，并将其虚拟成各种设备，客户机操作系统无需做任何修改，就能直接对虚拟化的硬件发出请求。客户机操作系统内核执行的任何有特权的指令都需要经过Hypervisor翻译，才能正确地被处理。
    - 全虚拟化是最为安全的一种虚拟化技术，因为客户机操作系统和底层硬件之间已被隔离。客户机操作系统的内核不要求做任何修改，可以在不同底层体系结构之间自由移植客户机操作系统。只要有虚拟化软件，客户机就能在任何体系结构的处理器上运行，但是在翻译CPU指令时会有一定的性能损失。

6
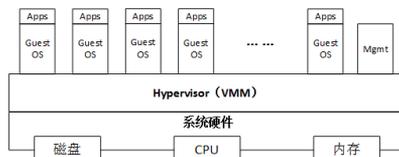
1.虚拟化技术

□ 根据实现方式不同，虚拟化技术可以分为全虚拟化、半虚拟化、操作系统级虚拟化等。

- 全虚拟化

1.虚拟化技术

□ 根据实现方式不同，虚拟化技术可以分为全虚拟化、半虚拟化、操作系统级虚拟化等。

- 半虚拟化
  - 半虚拟化技术也叫做准虚拟化技术，是在全虚拟化的基础上，对客户机操作系统进行修改，增加一个专门的API，使用API将客户机操作系统发出的指令进行最优化处理，不需要Hypervisor耗费一定的资源进行翻译操作，因此Hypervisor的工作负担变得非常小，系统整体的性能有较大提升。
 - 半虚拟化技术的缺点是需要修改操作系统以包含API，不能够实现对通用操作系统的支持。
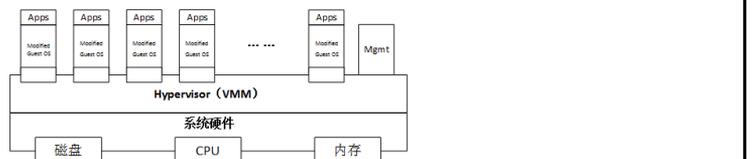
1.虚拟化技术

- 根据实现方式不同，虚拟化技术可以分为全虚拟化、半虚拟化、操作系统级虚拟化等。
    - 半虚拟化

1.虚拟化技术

- 根据实现方式不同，虚拟化技术可以分为全虚拟化、半虚拟化、操作系统级虚拟化等。
    - 操作系统级虚拟化
        - 操作系统级虚拟化并不是在硬件系统里创建多个虚拟机环境，而是让一个操作系统创建多个彼此相互独立的应用环境，这些应用环境访问同一内核。操作系统级的虚拟化可以想象是内核的一种功能，而不是抽象成一层独立的软件。
        - 因为不存在实际的翻译层或者虚拟化层，所以操作系统级的虚拟机开销很小，大多数都能达到原本的性能。该类型不能使用多种操作系统，所有虚拟机需要共享一个内核。

1.虚拟化技术

- 根据实现方式不同，虚拟化技术可以分为全虚拟化、半虚拟化、操作系统级虚拟化等。
  - 操作系统级虚拟化

11
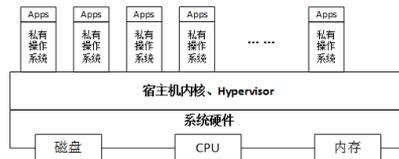
1.虚拟化技术

- 虚拟化产品分为开源虚拟化软件和商业虚拟化软件两大阵营。
  - 典型的代表有Xen、KVM、VMware、Hyper-V、Docker容器等，其中Xen、KVM是开源免费的虚拟化软件，VMware、Hyper-V是付费的虚拟化软件。
  - 虚拟化软件产品有很多，无论是开源还是商业的，每款软件产品有其特点及应用场景，需要根据业务场景选择合适的软件。
  - 最常见的虚拟化软件提供商有Citrix、IBM、VMware、Microsoft等，国产虚拟化平台有云宏CNware等。

12

1.虚拟化技术

□ 虚拟化产品分为开源虚拟化软件和商业虚拟化软件两大阵营。

表 9-0-1 常见虚拟化软件产品对比

名称	开发厂商	虚拟类型	执行效率	GuestOS跨平台	许可证
Xen	Virtual Iron http://www.xensource.com	半虚拟化	高	支持	GPL
OpenVZ	Swsoft http://www.openvz.org	操作系统级虚拟化	高	不支持	GPL
VMware	VMware http://www.vmware.com	全虚拟化	中	支持	私有
QEMU	QEMU http://www.qemu.com	仿真	低	支持	LGPL/GPL
VirtualBox	Oracle http://www.virtualbox.org	桌面虚拟化	低	支持	GPL
KVM	http://kvm.sourceforge.net	全虚拟化	中	支持	GPL
z/VM	IBM http://www.vm.ibm.com	全虚拟化	高	不支持	私有

2.使用KVM实现虚拟化

□ KVM是基于Linux内核的虚拟机软件（Kernel-based Virtual Machine），是第一个整合到Linux内核的虚拟化软件。

- KVM嵌入Linux系统内核，使Linux变成了一个Hypervisor，通过优化内核来使用虚拟技术，使用Linux自身的调度器进行虚拟机管理。
- KVM是内核的一个模块，用户空间通过QEMU模拟硬件提供虚拟机使用，一台虚拟机就是一个普通的Linux进程，通过对这个进程的管理，完成对虚拟机的管理。
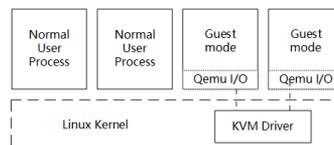
2.使用**KVM**实现虚拟化

- KVM的主要优势：
  - 开源免费
    - KVM是一个开源项目，一直以开放的姿态接受各种新技术，许多虚拟化的新技术都首先在KVM上应用，再到其他虚拟化引擎上推广。因为开源，绝大部分KVM的解决方案都是免费方案。随着KVM的发展，KVM虚拟机越来越稳定，兼容性越来越好，因而得到了广泛应用。
  - 紧密结合Linux
    - KVM是第一个整合进Linux内核的虚拟化技术，和Linux系统紧密结合，因此形成了从底层Linux操作系统，中间层Libvirt管理工具，到云管平台OpenStack的KVM生态链。
  - 性能
    - KVM性能优越，在同样的硬件环境下，能提供更好的虚拟机性能。

2.使用**KVM**实现虚拟化

任务1：安装KVM

任务2：配置宿主机网络

任务3：创建KVM虚拟机

任务4：管理KVM虚拟机

2.使用KVM实现虚拟化

任务1：安装KVM

步骤1：创建虚拟机并完成CentOS的安装

步骤2：完成虚拟机的主机配置、网络配置及通信测试

步骤3：配置宿主机支持虚拟化

步骤4：配置宿主机网络混杂模式

步骤5：检测CPU是否支持虚拟化

17

2.使用KVM实现虚拟化

任务1：安装KVM

步骤6：安装KVM

步骤7：启动libvirtd服务

步骤8：查看KVM运行信息

步骤9：配置libvirtd服务为开机自启动

步骤10：验证KVM安装的正确性：加载+服务自身检测

18

设备名称的明确：
- 本地主机：
  - 安装有VirtualBox软件的Windows 10操作系统，是物理计算机
- 宿主机：
  - 安装有KVM软件的CentOS操作系统，是VirtualBox中创建的虚拟机
- KVM虚拟机：
  - 在宿主机CentOS操作系统中使用KVM创建的虚拟机

19

任务1：安装KVM
- 任务目标：
  - 实现KVM软件的安装配置
  - 实现KVM服务的测试与管理
  - 在CentOS上实现KVM

操作视频 / 现场演示

20

1. #查看/proc/cpuinfo文件确定CPU是否支持VT技术
2. [root@Project-09-Task-01 ~]# cat /proc/cpuinfo | egrep 'vmx|svm'

3. #使用yum工具安装KVM软件及相关管理工具
4. [root@Project-09-Task-01 ~]# yum install -y qemu-kvm virt-manager virt-viewer libvirt virt-install acpid

5. #使用systemctl start命令启动libvirtd服务
6. [root@Project-09-Task-01 ~]# systemctl start libvirtd

7. #使用systemctl status查看libvirtd服务
8. [root@Project-09-Task-01 ~]# systemctl status libvirtd

9. #使用systemctl list-unit-files命令验证libvirtd服务是否已配置为开机自启动
10. [root@Project-03-Task-01 ~]# systemctl list-unit-files | grep libvirtd.service

11. #命令lsmod | grep kvm可检测KVM是否加载成功。
12. [root@Project-09-Task-01 ~]# lsmod | grep kvm

13. #命令virsh list查看KVM虚拟机列表。
14. [root@Project-09-Task-01 ~]# virsh list

命令指南 / 操作引导

21

```
s]
[MIRROR] systemd-container-239-41.el8_3.2.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.163.com/centos/8.3.2
011/BaseOS/x86_64/os/Packages/systemd-container-239-41.el8_3.2.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transferre
d the last 30 seconds]
[MIRROR] nfs-utils-2.3.3-35.el8.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.nju.edu.cn/centos/8.3.2011/Bas
eOS/x86_64/os/Packages/nfs-utils-2.3.3-35.el8.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transferred the last 30 sec
onds]
[MIRROR] systemd-container-239-41.el8_3.2.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.bfsu.edu.cn/centos/8
.3.2011/BaseOS/x86_64/os/Packages/systemd-container-239-41.el8_3.2.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transf
erred the last 30 seconds]
[MIRROR] nfs-utils-2.3.3-35.el8.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.163.com/centos/8.3.2011/BaseOS
/x86_64/os/Packages/nfs-utils-2.3.3-35.el8.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transferred the last 30 second
s]
[MIRROR] systemd-container-239-41.el8_3.2.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.huaweicloud.com/cent
os/8.3.2011/BaseOS/x86_64/os/Packages/systemd-container-239-41.el8_3.2.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec tr
ansferred the last 30 seconds]
[MIRROR] nfs-utils-2.3.3-35.el8.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.bfsu.edu.cn/centos/8.3.2011/Ba
seOS/x86_64/os/Packages/nfs-utils-2.3.3-35.el8.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transferred the last 30 se
conds]
[MIRROR] systemd-container-239-41.el8_3.2.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.neusoft.edu.cn/cento
s/8.3.2011/BaseOS/x86_64/os/Packages/systemd-container-239-41.el8_3.2.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec tra
nsferred the last 30 seconds]
[MIRROR] nfs-utils-2.3.3-35.el8.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.huaweicloud.com/centos/8.3.201
1/BaseOS/x86_64/os/Packages/nfs-utils-2.3.3-35.el8.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transferred the l
0 seconds]
[MIRROR] systemd-container-239-41.el8_3.2.x86_64.rpm: Curl error (28): Timeout was reached for http://mirror.lzu.edu.cn/centos/8.3
.2011/BaseOS/x86_64/os/Packages/systemd-container-239-41.el8_3.2.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transfer
red the last 30 seconds]
[MIRROR] nfs-utils-2.3.3-35.el8.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.neusoft.edu.cn/centos/8.3.2011
/BaseOS/x86_64/os/Packages/nfs-utils-2.3.3-35.el8.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transferred the last 30
seconds]
[MIRROR] systemd-container-239-41.el8_3.2.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.tuna.tsinghua.edu.cn
/centos/8.3.2011/BaseOS/x86_64/os/Packages/systemd-container-239-41.el8_3.2.x86_64.rpm [Operation too slow. Less than 1000 bytes/s
ec transferred the last 30 seconds]
[MIRROR] nfs-utils-2.3.3-35.el8.x86_64.rpm: Curl error (28): Timeout was reached for http://mirror.lzu.edu.cn/centos/8.3.2011/Base
OS/x86_64/os/Packages/nfs-utils-2.3.3-35.el8.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transferred the last 30 seco
nds]
[MIRROR] systemd-container-239-41.el8_3.2.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.cqu.edu.cn/CentOS/8.
3.2011/BaseOS/x86_64/os/Packages/systemd-container-239-41.el8_3.2.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transfe
rred the last 30 seconds]
[MIRROR] nfs-utils-2.3.3-35.el8.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.tuna.tsinghua.edu.cn/centos/8.
3.2011/BaseOS/x86_64/os/Packages/nfs-utils-2.3.3-35.el8.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transferred the l
ast 30 seconds]
[MIRROR] systemd-container-239-41.el8_3.2.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.ustc.edu.cn/centos/8
.3.2011/BaseOS/x86_64/os/Packages/systemd-container-239-41.el8_3.2.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transf
erred the last 30 seconds]
[MIRROR] nfs-utils-2.3.3-35.el8.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.cqu.edu.cn/CentOS/8.3.2011/Bas
eOS/x86_64/os/Packages/nfs-utils-2.3.3-35.el8.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec transferred the last 30 sec
onds]
[MIRROR] systemd-container-239-41.el8_3.2.x86_64.rpm: Curl error (28): Timeout was reached for http://mirrors.njupt.edu.cn/centos/
8.3.2011/BaseOS/x86_64/os/Packages/systemd-container-239-41.el8_3.2.x86_64.rpm [Operation too slow. Less than 1000 bytes/sec trans
ferred the last 30 seconds]
[FAILED] systemd-container-239-41.el8_3.2.x86_64.rpm: No more mirrors to try - All mirrors were already tried without success
(225/225): nfs-utils-2.3.3-35.el8.x86_64.rpm  99% [=====================================-] 148 kB/s | 113 MB   00:03 ETA
下载的软件包保存在缓存中，直到下次成功执行事务。
您可以通过执行‘yum clean packages’删除软件包缓存。
错误：下载软件包出错
    Cannot download Packages/systemd-container-239-41.el8_3.2.x86_64.rpm: All mirrors were tried
[root@Project-09-Task-01 ~]#
```

22

2.使用KVM实现虚拟化

任务2：配置宿主机网络

步骤1：查看宿主机网络情况

步骤2：创建bridge

河南中医药大学信息技术学院互联网技术教学团队 / https://internet.hactcm.edu.cn / http://www.51xueweb.cn

23

✓ 任务2：配置宿主机网络
- 任务目标：
  - 实现宿主机网桥配置
  - 实现宿主机的连通性测试
  - 实现宿主机和KVM虚拟机保持同一网络

操作视频 / 现场演示

24

2.使用KVM实现虚拟化

2.5 任务3

任务3：创建KVM虚拟机

步骤1：创建存储池

步骤2：获取CentOS 7

步骤3：安装CentOS 7

步骤4：使用KVM虚拟机并远程管理

河南中医药大学信息技术学院互联网技术教学团队 / https://internet.hactcm.edu.cn / http://www.51xueweb.cn

25

✓ KVM的存储虚拟化通过存储池（Storage Pool）和卷（Volume）来管理。

✓ 存储池是宿主机可以管理的存储空间，拥有多种类型。卷是在存储池中划分出的一块空间，宿主机将卷分配给虚拟机，卷在虚拟机中就是一块硬盘。

表 9-3-3 存储池类型

存储模式	存储池类型	类型说明
基于文件系统的存储	dir	使用文件系统目录来存储
	fs	使用预格式化分区来存储
	netfs	使用 NFS 等网络共享存储来存储
基于设备的存储	disk	使用物理硬盘来存储
	iscsi	使用网络共享的 ISCSI 存储来存储
	scsi	使用本地 SCSI 存储来存储
虚拟磁盘文件	lvm	取决于 LVM 卷组来存储

26

13
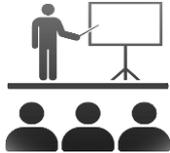
KVM支持多种类型的卷格式。

表 9-3-4 常用卷格式

虚拟磁盘格式	格式说明
raw	KVM 默认的磁盘格式，移植性好，性能好，但大小固定，不能节省磁盘空间
qcow2	KVM 推荐的磁盘格式，支持按需分配磁盘空间，支持快照，支持 zlib 磁盘压缩，支持 AES 磁盘加密
vmdk	VMware 默认使用的磁盘格式，性能与功能较为出色

操作视频／现场演示

任务3：创建KVM虚拟机
- 任务目标：
  - 实现KVM虚拟机的创建
  - 实现KVM虚拟机的操作系统安装
  - 实现KVM虚拟机的远程管理

2.使用KVM实现虚拟化

任务4：管理KVM虚拟机

步骤1：查看KVM虚拟机列表

步骤2：设置KVM虚拟机为自动启动

步骤3：修改KVM虚拟机的硬件配置

步骤4：KVM虚拟机的挂起、开启、关闭

步骤5：KVM虚拟机的克隆、快照

河南中医药大学信息技术学院互联网技术教学团队 / https://internet.hactcm.edu.cn / http://www.51xueweb.cn

29

操作视频 / 现场演示

✓ 任务3：管理KVM虚拟机
- 任务目标：
  □ 掌握virsh管理工具
  □ 通过virsh管理KVM虚拟机

30

3.管理KVM虚拟化平台

- KVM自身实现了虚拟化核心的监视工具，由于KVM开源特性，其管理工具非常丰富，比较典型的管理工具有virsh、virt-manager、ovirt等。
  - virsh：命令行管理工具，功能强大，能完成几乎所有虚拟机管理任务，包括在线迁移，虚拟机快照，创建和转换虚拟机磁盘文件格式等，适合以脚本的形式自动管理虚拟机
  - virt-manager：桌面应用管理工具，提供了方便与性能兼具的高效率管理，支持多节点管理，可以完全一样的方式管理多个节点
  - Ovirt：Web方式管理工具，实现KVM绝大部分的管理，可方便随时访问虚拟机状态，获取虚拟机监视器界面，使虚拟机管理可跨越地域限制，也实现了云计算雏形。

31

32

33

3.管理KVM虚拟化平台

3.1 KVM虚拟化平台管理工具

virt-manager

Apache CloudStack

OpenStack

NoKVM 塔斯云

河南中医药大学信息技术学院互联网技术教学团队 / https://internet.hactcm.edu.cn / http://www.51xueweb.cn

34

3.管理KVM虚拟化平台

- The virt-manager application is a desktop user interface for managing virtual machines through libvirt.
  - It primarily targets KVM VMs, but also manages Xen and LXC (linux containers). It presents a summary view of running domains, their live performance & resource utilization statistics.
  - Wizards enable the creation of new domains, and configuration & adjustment of a domain's resource allocation & virtual hardware.
  - An embedded VNC and SPICE client viewer presents a full graphical console to the guest domain.

3.管理KVM虚拟化平台

- About virt-manager's supporting tools
  - virt-install is a command line tool which provides an easy way to provision operating systems into virtual machines.
  - virt-viewer is a lightweight UI interface for interacting with the graphical display of virtualized guest OS. It can display VNC or SPICE, and uses libvirt to lookup the graphical connection details.
  - virt-clone is a command line tool for cloning existing inactive guests. It copies the disk images, and defines a config with new name, UUID and MAC address pointing to the copied disks.
  - virt-xml is a command line tool for easily editing libvirt domain XML using virt-install's command line options.
  - virt-bootstrap is a command line tool providing an easy way to setup the root file system for libvirt-based containers.

37
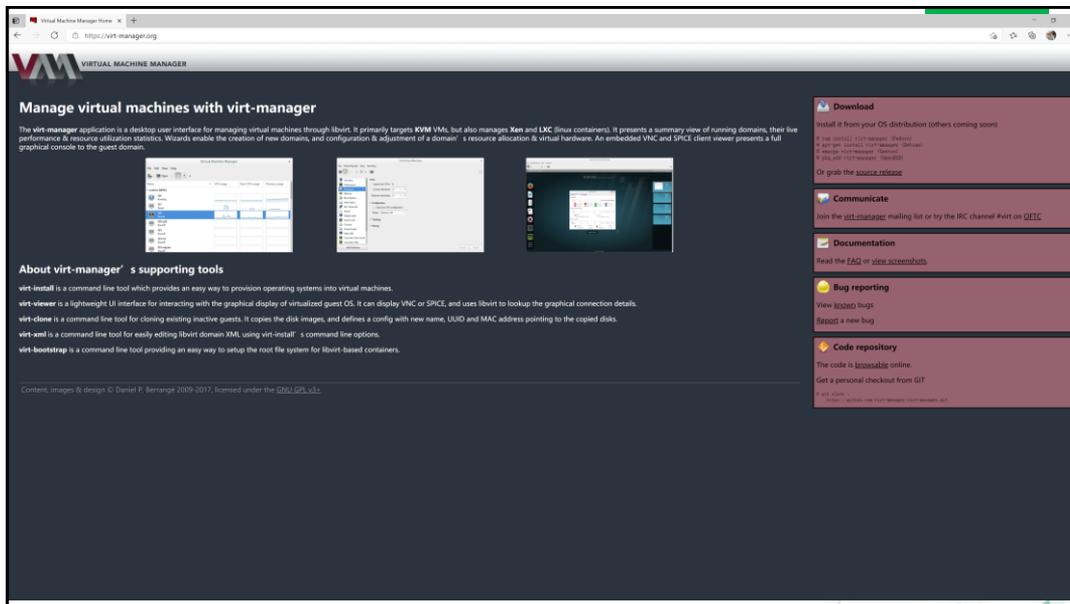
3.管理KVM虚拟化平台

3.3 Apache CloudStack

- Apache CloudStack is open source software designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a Service (IaaS) cloud computing platform.
  - CloudStack is used by a number of service providers to offer public cloud services, and by many companies to provide an on-premises (private) cloud offering, or as part of a hybrid cloud solution.
  - CloudStack is a turnkey solution that includes the entire "stack" of features most organizations want with an IaaS cloud: compute orchestration, Network-as-a-Service, user and account management, a full and open native API, resource accounting, and a first-class User Interface (UI).
  - CloudStack currently supports the most popular hypervisors: VMware, KVM, Citrix XenServer, Xen Cloud Platform (XCP), Oracle VM server and Microsoft Hyper-V.
  - Users can manage their cloud with an easy to use Web interface, command line tools, and/or a full-featured RESTful API. In addition, CloudStack provides an API that's compatible with AWS EC2 and S3 for organizations that wish to deploy hybrid clouds.

河南中医药大学信息技术学院互联网技术教学团队 / https://internet.hactcm.edu.cn / http://www.51xueweb.cn

38

3.管理KVM虚拟化平台

3.4 Apache CloudStack

- OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms.
  - A dashboard is also available, giving administrators control while empowering their users to provision resources through a web interface.
  - Beyond standard infrastructure-as-a-service functionality, additional components provide orchestration, fault management and service management amongst other services to ensure high availability of user applications.

河南中医药大学信息技术学院互联网技术教学团队 / https://internet.hactcm.edu.cn / http://www.51xueweb.cn

39
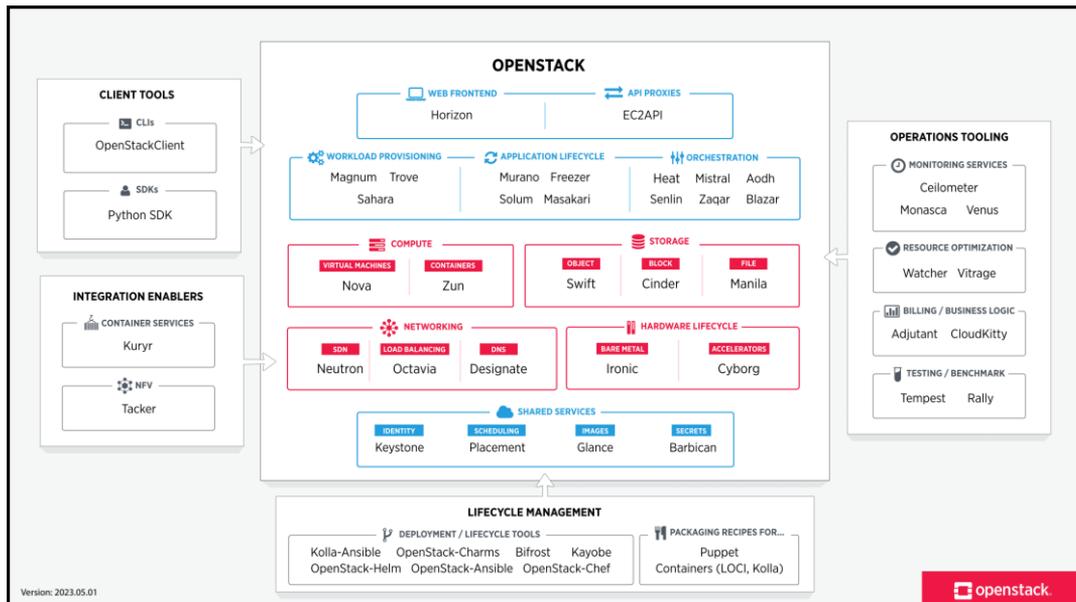
40
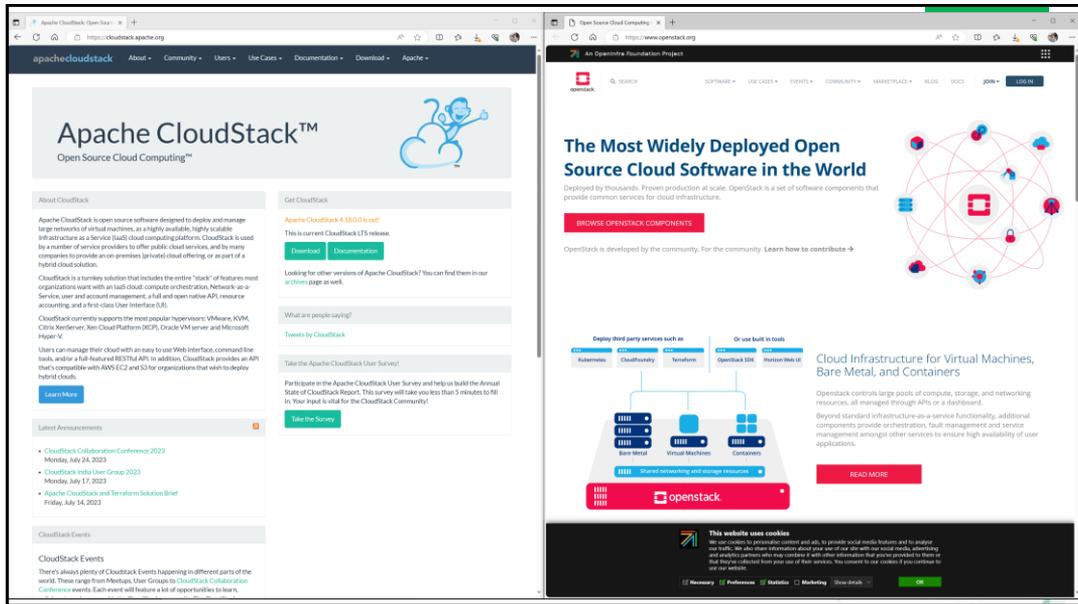
41
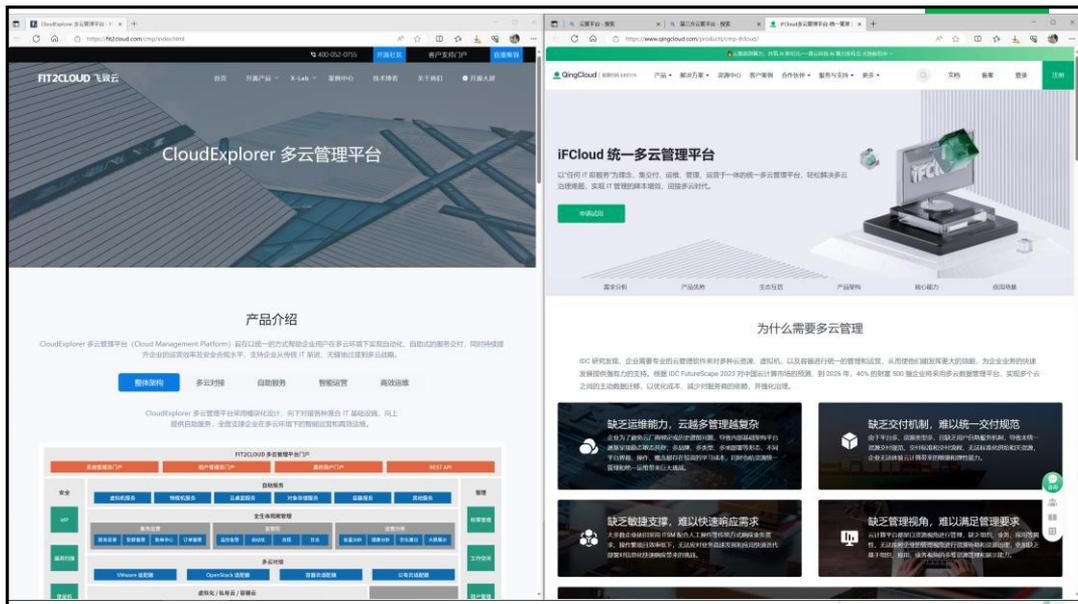
42

网络与信息系统智能运维

请扫码访问课程体系学习平台

43