CLOUD-day01

云平台部署与管理

# 搭建KVM服务器

## 虚拟化概念

virtualization 资源管理

x个物理资源 --> y个逻辑资源

实现程度:完全、部分、硬件辅劣(CPU)

**虚拟化主要厂商及产品**

|  |  |
| --- | --- |
| **系 列** | **PC/服务器版代表** |
| **VMware** | VMware Workstation、vSphere |
| **Microsoft** | VirtualPC、Hyper-V |
| **RedHat** | KVM、RHEV |
| **Citrix** | Xen |
| **Oracle** | Oracle VM VirtualBox |

## 安装虚拟化服务器平台

### KVM/QEMU/LIBVIRTD

- kvm是linux内核的模块,它需要CPU的支持，采用硬件辅助虚拟化技术Intel-VT,AMD-V,内存的相关如Intel的EPT和AMD的RVI技术

- QEMU是一个虚拟化的仿真工具，通过ioctl与内核kvm交互完成对硬件的虚拟化支持

- Libvirt是一个对虚拟化管理的接口和工具，提供用户端程序virsh,virt-install,

virt-manager,virt-view与用户交互

### 必备软件

qemu-kvm

为kvm提供底层仿真支持

libvirt-daemon

libvirtd守护进程，管理虚拟机

libvirt-client

用户端软件，提供客户端管理命令

libvirt-daemon-driver-qemu

libvirtd连接qemu的驱动

### 可选功能

virt-install //系统安装工具

virt-manager //图形管理工具

virt-v2v //虚拟机迁移工具

virt-p2v //物理机迁移工具

### 虚拟化平台的安装

# yum install -y qemu-kvm libvirt-daemon libvirt-client 、

libvirt-daemon-driver-qemu

# systemctl start libvirtd

### 虚拟机的组成

内核虚拟化模块（KVM）

系统设备仿真（QEMU）

虚拟机管理程序（LIBVIRT）

**一个XML文件（虚拟机配置生命文件）**

位置/etc/libvirt/qemu/

**一个磁盘镜像文件(虚拟机的硬盘)**

位置/var/lib/libvirt/images/

# 管理KVM平台

## virsh命令工具介绍

### 提供管理各虚拟机的命令接口

– 支持交互模式,查看/创建/停止/关闭 .. ..

– 格式:virsh 控制挃令 [虚拟机名称] [参数]

### [root@room9pc01 ~]# virsh

### 欢迎使用 virsh，虚拟化的交互式终端。

### 输入：'help' 来获得命令的帮助信息

### 'quit' 退出

### virsh #

### 查看KVM节点(服务器)信息

– virsh nodeinfo

### 列出虚拟机

– virsh list [--all]

### 列出虚拟网络

– virsh net-list [--all]

### 查看挃定虚拟机的信息

– virsh dominfo 虚拟机名称

### 运行|重启|关闭挃定的虚拟机

– virsh start|reboot|shutdown 虚拟机名称

### 强制关闭挃定的虚拟机

– virsh destroy 虚拟机名称

### 将挃定的虚拟机设为开机自劢运行

– virsh autostart 虚拟机名称

## 常用镜像盘类型

### 虚拟机的磁盘镜像文件格式

|  |  |  |
| --- | --- | --- |
| **特点/类型** | **RAW** | **QCOW2** |
| **KVM默认** | **否** | **是** |
| **I/0效率** | **高** | **较高** |
| **占用空间** | **大** | **小** |
| **压缩** | **不支持** | **支持** |
| **后端盘复用** | **不支持** | **支持** |
| **快照** | **不支持** | **支持** |

**LANG=en\_US.UTF-8 //中文切换为英文**

**LANG=zh\_CN.UTF-8 //英文切换为中文**

## qemu-img命令

### qemu-img 是虚拟机的磁盘管理命令,支持非常多的磁盘格式,例如raw、qcow2、vdi、vmdk等等

### qemu-img 命令格式

– qemu-img 命令 参数 块文件名称 大小

### 常用的命令有

– create 创建一个磁盘

– convert 转换磁盘格式

– info 查看磁盘信息

– snapshot 管理磁盘快照

### 创建新的镜像盘文件

– qemu-img create -f 格式 磁盘路径 大小

– qemu-img create -f qcow2 disk.img 50G

### 查询镜像盘文件的信息

– qemu-img info 磁盘路径

– qemu-img info disk.img

### [root@room9pc01 ~]# cd /var/lib/libvirt/images

### //存放镜像文件的目录

### [root@room9pc01 images]# qemu-img info rh7\_node1.img

### 

镜像盘当前大小8.2G；最大支持20G

### -b 使用后端模板文件

**– qemu-img create -b disk.img -f qcow2 disk1.img**

## COW技术原理

### Copy On Write,写时复制

**– 直接映射原始盘的数据内容**

**– 当原始盘的旧数据有修改时,在修改之前自劢将旧数据存入前端盘**

**– 对前端盘的修改丌回写到原始盘**



**连接本地/远程KVM**

**• 使用virsh客户端工具**

**– 连接本地**

– virsh

– virsh# connect qemu:///system (默认选项)

**– 连接远程**

– virsh# connect

qemu+ssh://user@ip.xx.xx.xx:port/system

[root@room9pc01 ~]# virsh -c qemu+ssh://root@176.4.12.147/system

或

[root@room9pc01 ~]# virsh

virsh # connect qemu+ssh://root@176.4.12.147/system

## 虚拟机远程管理

### 使用virt-manager客户端工具

– virt-manager 也可以通过 add connection 管理其它机器上的虚拟机,一般通过"remote tunnel over ssh"就可以了

– 需要注意的是 virt-manager 需要使用 ssh 免密码登录如果没有免密码登录需要单独安装 python 的 ssh 相关模块,这里我们可以使用部署 key 的方法解决

### 网络管理

**• virsh管理虚拟网络**

**– net-list**

### 查看虚拟网络

**– net-define vbr.xml 创建虚拟网络**

**– net-undefine vbr 初除虚拟网络**

**– net-start vbr 启动虚拟网络**

**– net-destroy vbr 停止虚拟网络**

**– net-edit vbr 修改 vbr 网络的配置**

**– net-autostart vbr 设置 vbr 虚拟网络开机自启动**

## 创建虚拟交换机

### libvirtd 网络接口

**– 原理:**调用 dnsmasq 提供 DNS、DHCP 等功能

**– 创建配置文件** /etc/libvirt/qemu/networks/vbr.xml

[root@room9pc01～]# vim /etc/libvirt/qemu/networks/vbr.xml

<network>

<name>vbr</name>

<bridge name="vbr"/>

<forward mode="nat"/>

<ip address="192.168.1.254" netmask="255.255.255.0">

<dhcp>

<range start="192.168.1.100" end="192.168.1.200"/>

</dhcp>

</ip>

</network>

[root@room9pc01～]# virsh net-define /etc/libvirt/qemu/networks/vbr.xml

[root@room9pc01 networks]# virsh net-start vbr

[root@room9pc01 networks]# virsh net-autostart vbr

# xml管理

## 导出虚拟机

### • xml 配置文件

– 定义了一个虚拟机的名称、CPU、内存、虚拟磁盘、网卡等各种参数设置

– 默认位于/etc/libvirt/qemu/虚拟机名.xml

[root@room9pc01 ~]# vim /etc/libvirt/qemu/node.xml

<domain type='kvm'>

<name>node</name>

<memory unit='KB'>2097152</memory> //内存大小

<currentMemory unit='KB'>2097152</currentMemory>

<vcpu placement='static'>2</vcpu> //CPU个数

<os>

<type arch='x86\_64' machine='pc'>hvm</type>

<boot dev='hd'/> //启动时从硬盘引导

<bootmenu enable='yes'/>

<bios useserial='yes'/>

</os>

<features> //电源管理，除非手动关机，否则永远不断电

<acpi/>

<apic/>

</features>

<cpu mode='host-passthrough'>

</cpu>

<clock offset='localtime'/>

<on\_poweroff>destroy</on\_poweroff> //关机命令

<on\_reboot>restart</on\_reboot> //重启命令

<on\_crash>restart</on\_crash> //强制重启命令

<devices>

<emulator>/usr/libexec/qemu-kvm</emulator>

<disk type='file' device='disk'>

<driver name='qemu' type='qcow2'/> //指定硬件仿真程序

<source file='/var/lib/libvirt/images/node.img'/>

<target dev='vda' bus='virtio'/>

</disk>

<interface type='bridge'>

<source bridge='vbr'/> //网卡

<model type='virtio'/>

</interface>

<channel type='unix'>

<target type='virtio' name='org.qemu.guest\_agent.0'/>

</channel>

<serial type='pty'></serial>

<console type='pty'> //console串口，用于管理虚拟机

<target type='serial'/>

</console>

<memballoon model='virtio'></memballoon>

</devices>

</domain>

### • 导出 xml 配置文件

– 查看:virsh dumpxml 虚拟机名

– 备份:virsh dumpxml 虚拟机名 > 虚拟机名.xml

## 编辑虚拟机设置

### • 对虚拟机的配置进行调整

– 编辑:virsh edit 虚拟机名

– 若修改name、memory、disk、network,可自动保存为新虚拟机配置

### 导入虚拟机

• 根据修改后的独立 xml 文件定义新虚拟机virsh define XML描述文件

### 删除虚拟机

**• 必要时可去除多余的 xml 配置**

**– 比如虚拟机改名的情冴**

**– 避免出现多个虚拟机的磁盘戒 MAC 地址冲突**

**- virsh undefine 虚拟机名**

# 自定义虚拟机安装

## 网络 yum 源的安装和配置

## • 快速配置网络 yum 源

### – 配置ftp # yum install vsftp

### – 修改配置文件 /etc/vsftpd/vsftpd.conf

# vim /etc/vsftpd/vsftpd.conf

– listen=YES

– listen\_ipv6=NO

# systemctl enable vsftpd

# systemctl start vsftpd

**– vsftp 默认根目录为 /var/ftp**

## 关闭防火墙

[root@room9pc01 ~]# systemctl stop firewalld

[root@room9pc01 ~]# systemctl mask firewalld

## 快速配置网络YUM源

### 在ftp根目录创建文件夹CentOS7

[root@room9pc01 ~]# mkdir /var/ftp/CentOS7

### 将CentOS7的光盘挂载到刚刚创建的目录上

[root@room9pc01 ~]# vim /etc/fstab

/root/桌面/CLOUD/ISO/CentOS7-1708.iso /var/ftp/CentOS-1807/ iso9660 defaults 0 0

[root@room9pc01 ~]# mount -a

### 在客户机里面配置/etc/yum.repos.d/CenOS.repo

[root@room9pc01 ~]# vim /etc/yum.repos.d/CentOS7.repo

[CentOS7]

name=CentOS7-1708

baseurl=ftp://176.4.12.147/CentOS7

enabled=1

gpgcheck=0

[root@room9pc01 ~]# yum clean all

[root@room9pc01 ~]# yum repolist

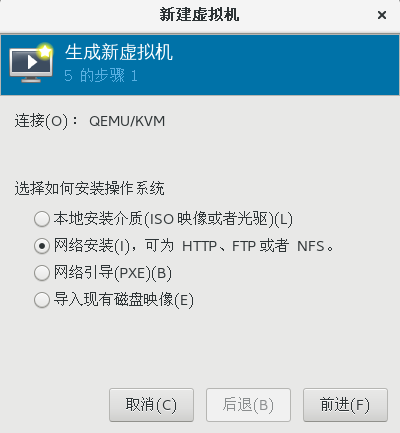
### 创建后端模板文件

[root@room9pc01 ~]# cd /var/lib/libvirt/images/

[root@room9pc01 images]# qemu-img create -f qcow2 node.qcow2 2G

## Virt-manager安装虚拟机

启动virt-manager软件，选择新建虚拟机,选择网络安装源



RAM：2048M，CPU：2

选择或创建自定义存储

/var/lib/libvirt/images/node.qcow2

网络选择自定义的vbr



## 安装完成之后的配置：

1.语言必须选英文

2.时区：Asia/上海

3.软件选择minimal，关闭KDUMP

4.选择手工分区

I will configure partitioning

使用标准分区格式，只分一个根分区

选取一个根分区，开启安装系统

5.设置root用户密码

# 虚拟机模板制作

## 软件包安装及YUM源配置

### 配置YUM源

[root@localhost ~]# mkdir /etc/yum.repos.d/repo

[root@localhost ~]# mv /etc/yum.repos.d/CentOS-\* /etc/yum.repos.d/repo

[root@localhost ~]# vim /etc/yum.repos.d/CentOS7.repo

[CentOS7]

name=CentOS7-1708

baseurl=ftp://192.168.1.254/CentOS7

enabled=1

gpgcheck=0

[root@localhost ~]# yum repolist

### 安装常用工具包

[root@localhost ~]# yum -y install vim git net-tools bridge-utils psmisc lftp lrzsz bash-completion.noarch

### 禁用selinux

[root@localhost ~]# vim /etc/selinux/config

SELINUX=disable

### 卸载firewalld-\*

[root@localhost ~]# yum remove firewalld-\*

[root@localhost ~]# reboot //重启，使selinux更改

### 安装软件

[root@localhost ~]# yum -y install lftp

### 连接ftp

[root@localhost ~]# lftp 192.168.1.254

lftp 192.168.1.254:~> cd /CentOS-1807/ //进入光盘文件的根目录

lftp 192.168.1.254:/CentOS-1807> get RPM-GPG-KEY-CentOS-7

//下载gpg key到本机

[root@localhost ~]# ll

-rw-r--r-- 1 root root 1690 12月 10 2015 RPM-GPG-KEY-CentOS-7

[root@localhost ~]# vim /etc/yum.repos.d/CentOS-1807.repo

**...**

**gpgcheck=1 //进行签名认证**

### 测试

[root@localhost ~]# yum clean all

[root@localhost ~]# yum repolist

[root@localhost ~]# yum -y install gcc //可正常装包即可

## 网卡及配置文件设置

### 删除网络配置里的个性化信息

[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE="eth0"

ONBOOT="yes"

NM\_CONTROLLED="no"

TYPE="Ethernet"

BOOTPROTO="dhcp"

### 删除NetworkManager

[root@localhost ~]# systemctl stop NetworkManager

[root@localhost ~]# yum remove -y NetworkManager-\*

[root@localhost ~]# reboot

### 禁用空路由

[root@localhost ~]# vim /etc/sysconfig/network

NOZEROCONF="yes"

[root@localhost ~]# systemctl restart network

## Console及磁盘分区配置

### 添加Console配置及删除磁盘分区里的个性化信息

[root@localhost ~]# vim /etc/default/grub

GRUB\_TIMEOUT=5

GRUB\_DISTRIBUTOR="$(sed 's, release .\*$,,g' /etc/system-release)"

GRUB\_DEFAULT=saved

GRUB\_DISABLE\_SUBMENU=true

GRUB\_TERMINAL="serial console"

GRUB\_SERIAL\_COMMAND="serial --unit=1 --speed=115200"

GRUB\_CMDLINE\_LINUX="biosdevname=0 net.ifnames=0 console=tty0 console=ttyS0,115200"

GRUB\_DISABLE\_LINUX\_UUID="true"

GRUB\_ENABLE\_LINUX\_LABEL="true"

GRUB\_DISABLE\_RECOVERY="true"

### 重新生成内核引导文件grub.cfg

[root@localhost ~]# grub2-mkconfig -o /boot/grub2/grub.cfg

[root@localhost ~]# reboot

[root@room9pc01 ~]# virsh console centos7.0

**– /etc/fstab 文件中到 UUID 手工修改成系统设备**

**– blkid 查看 uuid 对应的磁盘设备,修改 fstab 文件**

[root@localhost ~]# blkid

/dev/vda1: UUID="017ffffb-6bf3-430b-8357-cab530baf339" TYPE="xfs"

[root@localhost ~]# vim /etc/fstab

**/dev/vda1 / xfs defaults 0 0**

### 安装分区扩展软件

[root@localhost ~]# yum -y install cloud-utils-growpart

## 清理日志信息

### 关闭虚拟机

[root@localhost ~]# poweroff

### 真机执行

[root@room9pc01 ~]# yum -y install libguestfs-tools-c-1.36.3-6.el7.x86\_64

//提供virt-sysprep命令

[root@room9pc01 ~]# virt-sysprep -d centos7.0(为虚拟机名称)

## 以上部分为后端盘（模版盘）的制作流程，到此结束

# 自定义虚拟机

### 使用后端模版创建（前端）磁盘

[root@room9pc01 images]# qemu-img create -b node.qcow2 -f qcow2 node1.img 16G

**node.qcow2————后端模版创建的模版文件**

**node1.img————虚拟机的配置文件名**

[root@room9pc01 ~]# cd /root/nsd1807/ //进入到存放模版文件的目录

[root@room9pc01 nsd1807]# sed 's,node,node1,' node.xml > /etc/libvirt/qemu/node1.xml //对模版文件修改并拷贝成新的虚拟机文件

### 利用虚拟机文件快速创建虚拟机

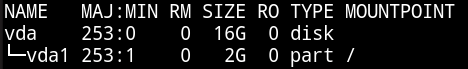
[root@room9pc01 nsd1807]# virsh define /etc/libvirt/qemu/node1.xml

[root@room9pc01 nsd1807]# virsh start node1

[root@room9pc01 ~]# virsh console node1

## 根分区扩容

[root@localhost ~]# lsblk

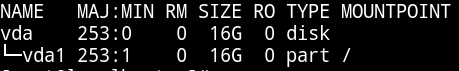


**分区扩容**

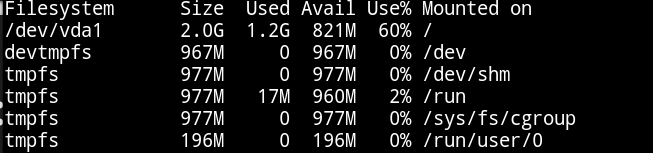
[root@localhost ~]# /usr/bin/growpart /dev/vda 1

CHANGED: partition=1 start=2048 old: size=4192256 end=4194304 new: size=33552351,end=33554399

[root@localhost ~]# lsblk



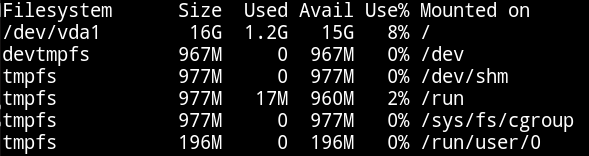
[root@localhost ~]# df -h



**文件系统扩容**

[root@localhost ~]# /usr/sbin/xfs\_growfs /

[root@localhost ~]# df -h



自动扩容的脚本 ecp.sh

自动配置静态IP的脚本 eip.sh

克隆虚拟机的脚本 clone-auto.sh

## 离线访问虚拟机

前提是得准确知道后端盘的名字

• 使用guestmount工具

– 支持离线挂载 raw、qcow2 格式虚拟机磁盘

– 可以在虚拟机关机的情冴下,直接修改磁盘中的文档

– 方便对虚拟机定制、修复、脚本维护

!!! 需要注意 SELinux 机制的影响

• 基本用法

guestmount -a 虚拟机磁盘路径 -i /挂载点

[root@kvmsvr ~]# mkdir /mnt/kdisk

[root@kvmsvr ~]# guestmount -a **node1.qcow2** -i /mnt/kdisk

[root@kvmsvr ~]# ls /mnt/kdisk

bin home media opt sbin tmp boot lib misc

proc selinux usr