**Morning：**

1、虚拟化概念

**-如何理解虚拟化与集群的区别？**

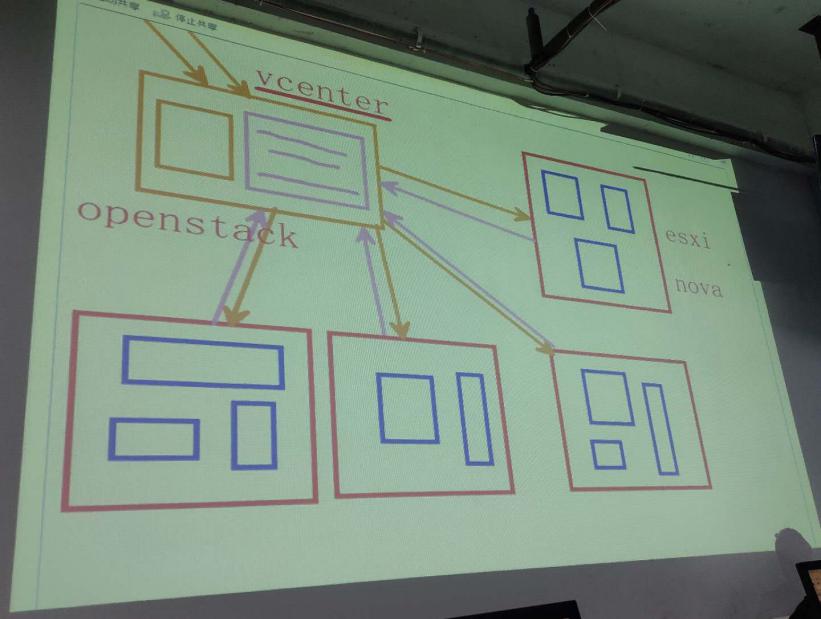
虚拟化：x个物理资源 --> y个逻辑资源（x<=y）

集群：多个物理资源 --> 1个逻辑资源

**-虚拟化主要产品**

VMware、KVM等

VMware的架构：



vcenter：安装在Windows上的管理软件(使用到Sql Server等)

esxi：相当于Linux（有一套专用的管理命令）

**-KVM虚拟机的组成**

1. KVM是Linux内核的模块(内核虚拟化模块)
2. QEMU是虚拟化的仿真工具、与kvm内核进行交互，完成对硬件的虚拟化支持(系统设备仿真)
3. Libvirt是虚拟化管理的接口与工具，提供用户端命令(用户接口管理)
4. xml文件(虚拟机的配置文件) /etc/libvirt/qemu/
5. 磁盘镜像文件(虚拟机的硬盘文件) /var/lib/libvirt/images/

**-虚拟化平台的安装**

yum -y install qemu-kvm \ #为kvm提供底层仿真支持

> libvirt-daemon \ #libvirtd守护进程

> libvirt-client \ #提供用户端管理命令

> libvirt-daemon-driver-qemu #libvirtd连接qemu的驱动

[root@room12pc2 ~]# systemctl start libvirtd

**Afternoon：**

1、管理KVM平台

**-virsh命令**

virsh nodeinfo #查看真机信息

virsh list [--all] #列出虚拟机[--all包括关闭状态]

virsh dominfo 虚机名 #查看虚拟机信息

virsh start|reboot|shutdown 虚机名 #运行/重启/关闭

virsh destroy 虚机名 #强制关机

virsh autostart 虚机名 #设为开机自启

virsh console 虚机名 #串口连接虚拟机（Ctrl+]退出）

**-qemu-img命令**

qemu-img是虚拟机的磁盘管理命令，支持很多磁盘格式(raw、qcow2、vdi、vmdk等等)

qemu-img create -f 格式 镜像文件名 大小 #创建新的镜像盘文件

qemu-img info 镜像文件名 #查询镜像盘文件的信息

qemu-img create -b 后端文件 -f qcow2 镜像文件名

#-b使用后端模板文件、不写默认使用后端盘大小(必须>=后端盘)

**-COW技术原理**

Copy On Write，写时复制。

用户执行读操作时，直接使用后端盘的数据内容；

用户执行写操作时，先拷贝一份到前端盘，再对前端盘内容进行修改。

**前端应用盘1**

**前端应用盘3**

**前端应用盘2**

**后端模板盘**

qcow2节省空间原因：

1. 不会按照最大存储空间占用系统存储，而是按照磁盘真实存储空间占用系统存储。
2. 后端复用技术，将一些公有资源只存在一个虚拟磁盘，所有其他虚拟盘都可以读这个盘的内容（如操作系统）。

2、virsh虚拟机管理

**-客户端工具**

virsh -c qemu:///system #连接本地

virsh -c qemu+ssh://user@xx.xx:port/system #连接远程

**-虚拟交换机管理**

1）创建虚拟交换机

[root@room12pc2 ~]# vim /etc/libvirt/qemu/networks/vbr.xml

<network>

<name>vbr</name> #虚拟网络名称

<forward mode='nat'/> #支持网络地址伪装

<bridge name='vbr' stp='on' delay='0'/> #网卡名、生成树协议

<ip address='192...' netmask='255...'> #默认路由IP地址

<dhcp>

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

</dhcp>

</ip>

</network>

2）virsh管理虚拟网络

virsh net-list [--all] #查看虚拟网络

virsh net-define vbr.xml #创建

virsh net-undefine vbr #删除

virsh net-start vbr #启动

virsh net-destroy vbr #停止

virsh net-edit vbr #修改vbr网络的配置

virsh net-autostart [--disable] vbr #开机自启

3、xml管理

**-xml配置文件**

定义虚拟机名称、CPU、内存、虚拟磁盘、网卡等

详情查看：/root/文档/NSD1808\_阶段4/node\_module.xml

**-xml管理命令**

virsh define 虚拟机名.xml #导入虚拟机

virsh dumpxml 虚拟机名 #打印xml配置文件

virsh edit 虚拟机名 #编辑虚拟机

virsh undefine 虚拟机名 #删除虚拟机

注意：完成xml的配置后，virsh define会自动在xml生成uuid、总线地址、网卡等信息。

4、自定义虚拟机安装

**-准备工作**

1. 创建虚拟网络vbr(192.168.1.254，支持nat，dhcp)

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

[root@room ~]# virsh net-start vbr

[root@room ~]# virsh net-autostart vbr

1. 创建2G的后端磁盘文件

[root@room ~]# qemu-img create -f qcow2 \

> /var/lib/libvirt/images/node.img 2G

3）配置网络安装源

[root@room ~]# mount -t iso9660 -o loop,ro \

> /root/CentOS7-1708.iso /var/ftp/centos7/

**-virt-manager安装虚拟机**

1）创建虚拟机

新建虚拟机 -> 网络安装 -> URL:ftp://176.121.212.154/centos7

-> 双核2G内存 -> 自定义存储node.img -> 名称node，网络vbr

2）初始化系统

英文版 -> 最小安装 -> 手动分区 -> Standard Partition

-> 添加挂载点：根分区 -> 设定root密码

**-virt-install安装虚拟机**

[root@room09pc143 ~]# virt-install \

> --name node \

> --ram 2048 \

> --vcpus 2 \

> --location=ftp://192.168.1.254/centos \

> --disk /var/lib/libvirt/images/node.img \

> --network bridge=vbr

参考：<https://www.cnblogs.com/nulige/p/8319838.html>

5、虚拟机模板制作

1）禁用Selinux

2）卸载防火墙、NetworkManager

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

1. 配置yum源

baseurl=ftp://192.168.1.254/centos7

4）导入公钥，安装常用命令

[root@localhost ~]# lftp 192.168.1.254

lftp 192.168.4.254:~> cd centos

lftp 192.168.4.254:/centos> get RPM-GPG-KEY-CentOs-7

lftp 192.168.4.254:/centos> exit

[root@localhost ~]# rpm --import RPM-GPG-KEY-CentOs-7

#导入公钥，提高安全性

[root@localhost ~]# yum -y install net-tools vim-enhanced \

> bridge-utils psmisc lftp bash-completion

#net-tools提供ifconfig、vim-enhanced提供vim

#bridge-utils用于管理虚拟交换机

#psmisc提供killall、pstree等进程管理命令

#bash-completion，systemctl自动补全命令插件

5）配置网卡

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

DEVICE="eth0"

ONBOOT="yes"

NM\_CONTROLLED="no" #禁止NetworkManager的控制(nmcli)

IPV6INIT="no"

TYPE="Ethernet"

BOOTPROTO ="dhcp"

[root@localhost ~]# systemctl restart network

6）禁用空路由

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

NOZEROCONF="yes" #使用ip r s可以查看空路由被禁用

7）添加console配置(使真机支持virsh console到虚机)

[root@localhost ~]# vim /etc/default/grub #定义传给内核的参数

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=ttyS0,115200n8"

GRUB\_DISABLE\_LINUX\_UUID="true"

GRUB\_ENABLE\_LINUX\_LABEL="true"

GRUB\_DISABLE\_RECOVERY="true"

8）重新生成grub.cfg

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

#利用/etc/default/grub文件重新生成系统引导文件

#真机验证virsh console node

1. 设置公有配置与服务

修改/etc/fstab的UUID为/dev/vda1

卸载postfix

设置时间同步为192.168.1.254(真机allow 0/0) chronyc sources -v

1. 安装拓展分区软件

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

12）关闭虚拟机，执行信息清理工作(真机)

[root@room12pc2 ~]# yum -y install libguestfs-tools-c

[root@room12pc2 ~]# virt-sysprep -d node

Tips：若因libvirt版本不兼容导致报错，按如下方式修复：

rpm -qa |grep libvirt |xargs -i yum remove -y {}

yum install -y qemu-kvm libvirt-daemon libvirt-client libvirt-daemon-driver-qemu libguestfs-tools-c virt-manager

systemctl start libvirtd

1. 保证虚拟机模板只读

[root@room12pc2 ~]# virsh undefine node

14）第一次开机自动扩容（前端机操作）

[root@localhost ~]# chmod 755 /etc/rc.local

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

#将vda第一个分区扩展成最大容量(注意保证LANG=en\_US.UTF-8)

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

#扩展文件系统

1. [脚本]克隆虚拟机

[root@room12pc2 ~]# vim /root/clone.sh

#!/bin/bash

[ -z $1 ] && echo 'error [lack parameter]' && exit 1

cd /var/lib/libvirt/images/

[ -f $1.img ] && echo 'error [VM exists]' && exit 2

qemu-img create -f qcow2 -b node.img $1.img 20G

cd /etc/libvirt/qemu/

cp .module.xml $1.xml

sed -i -r "/<name>/s/(<name>)(.\*)(<.\*)/\1$1\3/" $1.xml

sed -i -r "/source file/s#(.\*images/)(.\*)(.img.\*)#\1$1\3#" $1.xml

virsh define $1.xml

virsh start $1

6、离线访问虚拟机

一般用于拯救虚拟机里的数据

[root@room9pc01 ~]# mkdir /mnt/kdisk

[root@room9pc01 ~]# guestmount -a node.qcow2 \

> -i /mnt/kdisk

[root@room9pc01 ~]# umount /mnt/kdisk