**云计算**

**故障排查：**

**重启------->重装----->换机器**

**独立的排错技能：**

**1.排错思路**

**出现报错----》找关键词----》尝试进行排错------》百度关键词-----》官方文档**

**或看错误日志/log/ 系统日志var/log/message**

**2.软件开发的体系思路**

**##############################**

**提问的方式方法**

**问题描述清晰，做到什么程度，遇到什么问题**

**积累学习的知识，经验。一定是要写《技术博客》，换成自己的理解**

**#######################**

**Windows---->ssh------>服务器**

**Xshell**

**secureCRT**

**Putty**

**Yum首选，批量性的包，编译一个rpm批量安装，放入企业的yum源**

企业内部yum机

外网阿里yum

服务器

服务器

**云主机：小型企业**

**云主机提供商：自带提供yum**

**tput cols显示当前的列**

**++++++++++++++++++++++++++++++**

**Yum 源**

**安装FTP yum源**

**vim /etc/vsftpd/vsftpd.conf**

**listen=YES**

**listen\_ipv6=NO**

**yum -y install lftp**

**Cd ~ftp 进入ftp目录下**

**Lftp**

**Lftp ip.XX.XX(默认连接ftp服务)**

**Lftp sftp:192.168.74.1 -u root 默认连接ssh服务(密码口令是要连接机的ssh密码)**

**----------------------------------------------------------------**

**Lftp兼具ssh与sftp的功能,**

**Get 下载可以将对方的内容get到ftp上**

**Put 上传**

**作用：验证ftp是否搭建成功，可以看到共享中的文件类型内容**

**创建一个自定义的yum**

**Mkdir -p /var/ftp/yum**

**Cp xx>rpm /var/ftp/yum**

**Createrepo . 创建新的**

**Createrepo --update . 更新**

**客户端的配置不需要说明了**

**Dns安装配置**

**yum -y install bind-chroot**

**Vim /**

**COW（copy on write）**

后端盘必须只读，所有前端盘在接受访问时，先查询是否在本地有，否则在后端上寻找

若本地访问到了就直接返回，写入时也只写入前端，不写入后端

在设置时一定要写。前端一定不小于后端的存储大小

**访问**

（本机有的内容直接返回）

后端只读

（空时复制后端内容）

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

对前端盘的修改不会写到原始盘，而是在本地

后端盘（原始盘）

虚拟机的远程管理：

Virsh -c qemu+ssh://user@ip/system 远程连接虚拟机 输入密码

Virt-manager:

生成密钥：Ssh Keygen -t rsa -b 2048 -N ''

Qemu-kvm /libvirt-daemon /libvirt-client /

创建虚拟交换机

Virsh net edit---编辑虚拟网络

如何让用网络安装yum

############################################

**虚拟化：**

KVM /QEMU /LIBVIRTD

虚拟机的组成

– 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 不内核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 不用户交互

– 一个 XML 文件(虚拟机配置声明文件) – 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/

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 的驱劢

PC/服务器版代表

VMware VMware Workstation、vSphere

Microsoft VirtualPC、Hyper-V

RedHat KVM、RHEV

Citrix Xen

Oracle Oracle VM VirtualBox

**管理KVM平台**

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]

**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 [--disable] 虚拟机名称

**qemu-img 是虚拟机的磁盘管理命令**

• qemu-img 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

• -b 使用后端模板文件

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

**连接本地/远程KVM**

• 使用 virsh 客户端工具

– 连接本地

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

– 连接远程

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

**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(在里面编写)

<network>

<name>vbr</name> //创建一个虚拟网络 vbr

<bridge name="vbr"/>

<forward mode="nat"/> //使其能够上网

<ip address="192.168.1.254" netmask="255.255.255.0"> //设置 vbr 的 ip 为 192.168.1.254

<dhcp>

<range start="192.168.1.100" end="192.168.1.200"/> //配置 vbr 虚拟网络的 dhcp 分配地址范围 100-200

</dhcp>

</ip>

</network>

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 启劢虚拟网络

– 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 虚拟网络开机自启劢

**xml管理**

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

• 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

• 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,可自劢保

存为新虚拟机配置

[root@kvmsvr ~]# virsh edit rhel-207

<domain type='kvm'>

<name>rhel-207</name>

<uuid>76d5dc2c-5eef-4e30-8b6c-e58851814f84</uuid>

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

<source file='/var/lib/libvirt/images/rhel7.2.qcow2'/>

.. ..

<interface type='network'>

<mac address='52:54:00:91:52:e4'/>