一.dns转发+时间同步

物理机配置:

1.转发dns安装配置

# yum install -y bind-chroot

# vim /etc/named.conf

listen-on port 53 { 192.168.1.10; };

//listen-on-v6 port 53 { ::1; };

...

allow-query { any; };

...

forwarders { 202.106.196.115; };

dnssec-enable no;

dnssec-validation no;

# systemctl strat named

# nslookup

> server 192.168.1.10

defalut server: 192.168.1.10

address: 192.168.1.10#53

> set q=a

> [www.baidu.com](http://www.baidu.com)

# ss -ltun

1. 时间同步服务器,server,client

# yum install chrony -y

server:配置文件

# vim /etc/chrony.conf

server s2b.time.edu.cn iburst //设定上一级时间服务器

bindacqaddress 0.0.0.0

allow 0/0

cmdallow 127.0.0.1

# systemctl restart chronyd

# chronyc -sources -v

client:配置文件

1. shell命令 exec重定向
2. KVM虚拟机管理
3. 虚拟化

KVM 直接集成在内核里面

Linux自身 xen

家用: VMW

商用: vsphere -- ESXI

云计算 -- 虚拟化

容器 -- docker (k8s) -- go

openstack

启动的实例 (虚拟机-->KVM)

国外:AWS,Google cloud,微软,阿里

国内:阿里,华为

华为云 --> openstack --> python

· 虚拟化kvm核心三个组件

kvm:linux内核模块,需要cpu支持

bios选项可以禁用

云主机上一般不支持

qemu:虚拟化硬件(cpu,ram,disk,总线)

libvirt:虚拟化管理接口/管理工具

virt-manager

virsh

· 虚拟机构成

虚拟机配置文件 xx.xml

/etc/libvirt/qemu

虚拟机镜像文件 xx.qcows2 xx.img xx

/var/lib/libvirt/images/

ovl前端盘 qcow后端盘

cow技术: copy on write 写时复制

快照

后端盘提供基础数据

有改变时写入前端盘

虚拟机和真机通信 (kvm)

网络通信

区分虚拟网卡和物理网卡,两个配置文件不在一起

虚拟机A --> eth0 --> 只能使用192.168.4.0/24网段的ip

傻瓜式交换机: 无法配置,接电启用

# ls /etc/libvirt/qemu/network

# vim /etc/libvirt/qemu/network/private1.xml

Linux查看路由转发是否开启

# cat /proc/sys/net/ipv4/ip\_forward

# vim /etc/sysctl.conf //可以调整所有的内核参数

net.ipv4.ip\_forward = 0

建立一台虚拟机

1. 建立后端磁盘

· 真正的后端盘文件

· 以后新建的虚拟机都是通过此后端盘文件生成

# qemu-img create -f qcow2 test.qcow2 30G

· 用命令行安装虚拟机

# virt-install --virt-type=kvm --name=test --vcpus=2 --memory=2048 --location=/ISO/CentOS-7-x86\_64-Everything-1708.iso --disk path=test.qcow2,format=qcow2 --network bridge=private1 --graphics none --extra-args='console=ttyS0' --force

· 根据提示安装

系统自己分区,会自动建立swap分区

· 用console连接虚拟机

# virsh console 虚拟机名

退出快捷键:ctrl+]

· virsh命令

# virsh start server 开机

# virsh shutdown server 正常关闭系统

# virsh destory server 强制关闭(拔电源)

# virsh autostart classroom 开机自动开启虚拟机

# virsh edit server 修改虚拟机配置文件

# virsh define server.xml 生成虚拟机配置文件

# virsh undefine server 删除虚拟机配置文件

虚拟机网络配置

# virsh net-start private1 开启网络

# virsh net-destory private1 关闭网络

# virsh net-autostart private1 自动开启虚拟网络

# virsh net-autostart --disable private1 关闭自动开启网络

qcow2 支持压缩,影响做openstack上传速度

下午:

后端盘虚拟机初始化

1. 控制端口进入虚拟机
2. 做yum源,导入证书

# rpm --import ftp://192.168.4.254/Centos7.4/RPM-GPG-KEY-CentOS-7

1. 配ip

# ip address add 192.168.4.50/24 dev eth0 //临时配置ip

1. ssh连接,开始装软件

· 安装vim,net-tools,bash-completion(vim编辑器,网络管理工具,Tab补全)

1. 关闭selinux,防火墙

# vim /etc/selinux/config

# rpm -qa | grep firewall

firewalld-filesystem-0.4.4.4-6.el7.noarch

python-firewall-0.4.4.4-6.el7.noarch

firewalld-0.4.4.4-6.el7.noarch

# yum remove -y firewalld firewalld-filesystem python-firewall

# yum -y remove NetworkManager //NetworkManager与openstack冲突

卸载后用network管理ip

1. 禁止空路由

# vim /etc/sysconfig/network

NOZEROCONF="yes"

1. 做一个ip配置脚本

# vim eip.sh

# cat > /etc/sysconfig/network-script/ifcfg-eth0 << EOF

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=none

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=no

NAME=eth0

DEVICE=eth0

ONBOOT=yes

IPADDR=176.19.7.$1

PREFIX=24

GATEWAY=176.19.7.1

DNS1=176.19.0.26

IPV6INIT=no

EOF

8.安装扩展分区软件

9.清除虚拟机硬件信息

# virt-sysprep -d test

1. 修改对应的xml文件

所有uuid删除

所有MAC地址删除

所有usb删除

根据后端盘文件,生成新的虚拟机

# /etc/libvirt/qemu

# cp test.xml test1.xml

# vim test1.xml

# cd /var/lib/libvirt/images

# qemu-img create -b test.qcow2 -f qcow2 test1.img

快速生成虚拟机 ---> mysql

快速删除

for i in ...

undefine

快速修改虚拟机配置文件 -->

快速管理虚拟机

任务

新建8台虚拟机

# vim /usr/local/bin/clone-vm7

if [ $# -ne 0 ];then

VMNUM=$1

else

read -p "Enter VM number: " VMNUM

fi

# for i in {50..57}; do clone-vm7 $i; done

配置好ip

# for i in rh7\_node{50..57}

do

virsh console $i

done

# vim /etc/hosts

做好真机对虚拟机的免密

配置好yum

做好虚拟机之间两两免密

tmux分屏工具

# tmux

ctrl+b," 上下分屏

ctrl+b,% 左右分屏

ctrl+b,x 关闭一个分屏