# NSD Service全部笔记

## Day01

搭建新教学环境

一 关闭虚拟机classroom与server与desktop

二 真机建立全新的两台虚拟机

[student@room9pc01 ~]$ clone-vm7

Enter VM number: 4

[student@room9pc01 ~]$ clone-vm7

Enter VM number: 5

vi是Linux最基本的编辑器

虚拟机root的密码:123456

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

三 配置虚拟机A

1.为虚拟机A:配置eth0的IP地址为192.168.4.7/24

真机管理本机虚拟机,前提关闭所有虚拟机图形窗口

[student@room9pc01 ~]$ virsh console A

Ctrl + ]:退出管理

[root@localhost ~]# nmcli connection modify eth0 ipv4.method manual ipv4.address 192.168.4.7/24 connection.autoconnect yes

[root@localhost ~]# nmcli connection up eth0

[root@localhost ~]# ifconfig | head -2

2.为虚拟机A配置主机名:svr7.tedu.cn

[root@localhost ~]# echo svr7.tedu.cn > /etc/hostname

svr7.tedu.cn

[root@localhost ~]# hostname svr7.tedu.cn

[root@localhost ~]# hostname

svr7.tedu.cn

[root@svr7 ~]# cat /etc/motd #每次开机自动显示内容

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

真机利用virsh console管理虚拟机B进行配置

[root@localhost ~]# hostname pc207.tedu.cn

[root@localhost ~]# echo pc207.tedu.cn > /etc/hostname

[root@localhost ~]# nmcli connection modify eth0 ipv4.method manual ipv4.address 192.168.4.207/24 connection.autoconnect yes

[root@localhost ~]# nmcli connection up eth0

[root@localhost ~]# ifconfig | head -2

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

利用真机进行ssh远程管理

真机上设置

[student@room9pc01 ~]$ vim /home/student/.bashrc

alias goa='ssh -X root@192.168.4.7'

alias gob='ssh -X root@192.168.4.207'

开启新的终端才能验证

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

为虚拟机A与虚拟机B构建Yum仓库

一 真机:服务端

构建Web服务或FTP服务,共享光盘所有内容

1.构建FTP服务,共享光盘所有内容

[student@room9pc01 ~]$ rpm -q vsftpd

vsftpd-3.0.2-22.el7.x86\_64

[student@room9pc01 ~]$ ls /var/ftp/centos-1804/

[student@room9pc01 ~]$ systemctl status vsftpd

[student@room9pc01 ~]$ vim /etc/fstab

..........

/var/lib/libvirt/images/iso/CentOS7-1804.iso /var/ftp/centos-1804 iso9660 defaults 0 0

]$ firefox ftp://192.168.4.254/centos-1804

二 客户端:虚拟机A与虚拟机B

[root@svr7 ~]# vim /etc/yum.repos.d/local.repo

[local\_repo]

name=CentOS-$releasever - Base

baseurl=ftp://192.168.4.254/centos-1804

enabled=1

gpgcheck=0

[root@svr7 ~]# yum repolist

[root@svr7 ~]# yum -y install xeyes

[root@svr7 ~]# xeyes

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

总结新建虚拟机:

1.真机利用clone-vm7生成虚拟机,开启虚拟机

2.真机利用virsh console 虚拟机名 设置IP地址

3.真机利用ssh远程管理虚拟机,配置主机名与Yum客户端配置文件

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

虚拟机C

1.配置eth0的IP地址:192.168.4.10/24

2.配置主机名:svr10.tedu.cn

3.配置Yum仓库,以真机提供FTP作为服务端

虚拟机D

1.配置eth0的IP地址:192.168.4.20/24

2.配置主机名:pc20.tedu.cn

3.配置Yum仓库,以真机提供FTP作为服务端

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

目录结构

• 认识Linux的目录层次:

– man hier

• 常见一级目录的用途

/boot 存放系统引导必需的文件,包括内核、启动配置

/bin、/sbin 存放各种命令程序

/dev 存放硬盘、键盘、鼠标、光驱等各种设备文件

/etc 存放Linux系统及各种程序的配置文件

/root、/home/用户名 分别是管理员root、普通用户的默认家目录

/var 存放日志文件、邮箱目录等经常变化的文件

/proc 存放内存中的映射数据,不占用磁盘

/tmp 存放系统运行过程中使用的一些临时文件

权限的数值表示

• 权限的数值化

– 基本权限:r = 4,w = 2,x = 1

– 附加权限:SUID = 4,SGID = 2,Sticky Bit = 1

[root@svr7 ~]# mkdir /nsd01

[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 700 /nsd01

[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 007 /nsd01

[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 755 /nsd01

[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 750 /nsd01

[root@svr7 ~]# ls -ld /nsd01

[root@svr7 ~]# chmod 3755 /nsd01

[root@svr7 ~]# ls -ld /nsd01

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

历史命令

• 管理/调用曾经执行过的命令

– history:查看历史命令列表

– history -c:清空历史命令

– !n:执行命令历史中的第n条命令

– !str:执行最近一次以str开头的历史命令

• 调整历史命令的数量

[root@svr7 ~]# vim /etc/profile

HISTSIZE=1000 #默认记录1000条

[root@svr7 ~]# history #查看历史命令

[root@svr7 ~]# history -c #清空历史命令

[root@svr7 ~]# history

1 history

[root@svr7 ~]# cat /etc/redhat-release

[root@svr7 ~]# ls -l /etc/redhat-release

[root@svr7 ~]# cat /etc/hostname

[root@svr7 ~]# history

[root@svr7 ~]# !cat #执行历史命令中最近一条cat开头

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

实用小命令工具

• du,统计文件的占用空间

– du [选项]... [目录或文件]...

– -s:只统计每个参数所占用的总空间大小

– -h:提供易读容量单位(K、M等)

[root@svr7 ~]# du -s /root/

[root@svr7 ~]# du -sh /root/

[root@svr7 ~]# du -sh /boot/ /etc/

[root@svr7 ~]# du -sh /

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

bc 交互式计算器

[root@svr7 ~]# bc

-bash: bc: 未找到命令

[root@svr7 ~]# yum -y install bc

[root@svr7 ~]# bc #进入交互式模式

+ - \* / %(取余数运算)

公式是: 被除数 ÷ 除数 = 商

余数一定小于除数

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

• date,查看/调整系统日期时间

– date +%F、date +%R

– date +"%Y-%m-%d %H:%M:%S"

– date -s "yyyy-mm-dd HH:MM:SS"

[root@svr7 ~]# date +%F #显示年月日

[root@svr7 ~]# date +%R #显示时分

[root@svr7 ~]# date +%Y #显示年

[root@svr7 ~]# date +%m #显示月

[root@svr7 ~]# date +%d #显示日期

[root@svr7 ~]# date +%H #显示时

[root@svr7 ~]# date +%M #显示分

[root@svr7 ~]# date +%S #显示秒

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

制作快捷方式(连接文件 符号文件)

格式:ln -s /路径/源文件 /路径/生成快捷方式名称 #软连接

[root@svr7 ~]# cat /etc/hostname

[root@svr7 ~]# ln -s /etc/hostname /

[root@svr7 ~]# ls /

[root@svr7 ~]# ls -l /hostname

[root@svr7 ~]# ln -s /etc/hostname /hn

[root@svr7 ~]# ls /

[root@svr7 ~]# ls -l /hn

[root@svr7 ~]# cat /hn

[root@svr7 ~]# cat /hostname

若原始文件或目录被删除,连接文件将失效

软连接可存放在不同分区/文件系统,也可以针对目录

[root@svr7 ~]# echo 123 > /opt/A

[root@svr7 ~]# cat /opt/A

123

[root@svr7 ~]# ln -s /opt/A /opt/B

[root@svr7 ~]# ls /opt/

A B

[root@svr7 ~]# rm -rf /opt/A

[root@svr7 ~]# ls /opt/

B

[root@svr7 ~]# cat /opt/B

ln,创建硬连接

– ln 原始文件 硬连接文件

若原始文件被删除,连接文件仍可用

硬连接与原始文件必须在同一分区/文件系统,必须时文件

[root@svr7 ~]# echo haha > /opt/1.txt

[root@svr7 ~]# ln /opt/1.txt /opt/2.txt

[root@svr7 ~]# ls /opt/

[root@svr7 ~]# cat /opt/2.txt

[root@svr7 ~]# rm -rf /opt/1.txt

[root@svr7 ~]# ls /opt/

[root@svr7 ~]# cat /opt/2.txt

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

获取命令帮助

[root@svr7 ~]# date --help

[root@svr7 ~]# man date

[root@svr7 ~]# man passwd

[root@svr7 ~]# man 5 passwd #配置文件帮助信息

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

zip归档工具(跨平台的压缩工具)

• 归档+压缩操作

– zip [-r] 备份文件.zip 被归档的文档...

[root@svr7 ~]# yum -y install zip

[root@svr7 ~]# zip -r /opt/file.zip /etc/passwd /home

[root@svr7 ~]# ls /opt/

[root@svr7 ~]# zip -r /opt/abc.zip /etc/fstab /etc/hosts

[root@svr7 ~]# ls /opt/

• 释放归档+解压操作

– unzip 备份文件.zip [-d 目标文件夹]

[root@svr7 ~]# yum -y install unzip

[root@svr7 ~]# mkdir /nsd02

[root@svr7 ~]# unzip /opt/file.zip -d /nsd02

[root@svr7 ~]# ls /nsd02

[root@svr7 ~]# ls /nsd02/etc/

[root@svr7 ~]# ls /nsd02/home/

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

发布自定义Yum仓库

把从互联网下载的软件包,采用Yum的机制进行管理

1.具备从互联网下载的软件包

真机上:

[student@room9pc01 ~]$ ls /linux-soft/01

Cobbler.zip tools.tar.gz

2.将真机tools.tar.gz上传到虚拟机A的/root目录下

scp=ssh+cp结合

scp /本地路径/源文件 root@对方IP地址:/目标路径/

scp /linux-soft/01/tools.tar.gz root@192.168.4.7:/root

3.验证在虚拟机A进行查看

[root@svr7 ~]# ls /root/

Desktop tools.tar.gz

[root@svr7 ~]#

4.虚拟机A解包:

[root@svr7 ~]# tar -xf /root/tools.tar.gz -C /

[root@svr7 ~]# ls /

[root@svr7 ~]# ls /tools/

[root@svr7 ~]# ls /tools/other/

5.生成仓库清单(仓库数据文件)

[root@svr7 ~]# yum -y install createrepo

[root@svr7 ~]# createrepo /tools/other/

[root@svr7 ~]# ls /tools/other/

6.修改配置文件指定新的Yum服务端

[root@svr7 ~]# ls /etc/yum.repos.d/local.repo

[local\_repo]

name=CentOS-$releasever - Base

baseurl=ftp://192.168.4.254/centos-1804

enabled=1

gpgcheck=0

[myrpm]

name=myrpm

baseurl=file:///tools/other #指定本机为服务端

enabled=1

gpgcheck=0

[root@svr7 ~]# yum repolist

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

[root@svr7 ~]# yum -y install cmatrix

[root@svr7 ~]# cmatrix

[root@svr7 ~]# yum -y install sl

[root@svr7 ~]# sl

[root@svr7 ~]# yum -y install oneko

[root@svr7 ~]# oneko &

关闭终端,可以结束oneko程序

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

vim编辑技巧

命令模式

[root@svr7 ~]# cp /etc/passwd /opt/pa.txt

[root@svr7 ~]# vim /opt/pa.txt

光标跳转

Home 键 或 ^、数字 0 跳转到行首

End 键 或“$”键 跳转到行尾

1G 或 gg 跳转到文件的首行

G 跳转到文件的末尾行

复制/粘贴/删除

复制 yy、#yy 复制光标处的一行、#行

粘贴 p、P 粘贴到光标处之后、之前

x 或 Delete键 删除光标处的单个字符

dd、#dd 删除光标处的一行、#行

d^ 从光标处之前删除至行首

d$或D(大写) 从光标处删除到行尾

C(大写) 从光标处删除到行尾,并且进入插入模式

u 撤销

查找/撤销/保存

/word 向后查找字符串“word”

n、N 跳至后/前一个结果

u 撤销最近的一次操作

U 撤销对当前行的所有修改

Ctrl + r 取消前一次撤销操作

ZZ(大写) 保存修改并退出

[root@svr7 ~]# cp /etc/passwd /opt/pa.txt

cp：是否覆盖"/opt/pa.txt"？ y

[root@svr7 ~]# vim /opt/pa.txt

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

末行模式操作

读入其他文件内容

[root@svr7 ~]# echo 123 > /opt/a.txt

[root@svr7 ~]# echo abc > /opt/b.txt

[root@svr7 ~]# vim /opt/a.txt

末行模式下 :r /opt/b.txt #读入/opt/b.txt内容到当前文件

末行模式下 :r /etc/hostname

末行模式下 :r /etc/hosts

字符串替换

:s/root/admin 替换光标所在的当前行第一个“root”

:s/root/admin/g 替换光标所在的当前行所有的“root”

:5,10 s/root/admin/g 替换第5-10行所有的“root”

:% s/root/admin/g 替换文件内所有的“root”

[root@svr7 ~]# cp /etc/passwd /opt/p.txt

[root@svr7 ~]# vim /opt/p.txt

开关参数的控制

:set nu或nonu 显示/不显示行号

:set ai或noai 启用/关闭自动缩进

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

源码编译安装

RPM包: rpm -ivh yum

源码包----gcc与make---->可以执行的文件------->运行安装

源码编译安装的优势

• 主要优点

– 获得软件的最新版,及时修复bug

– 软件功能可按需选择/定制,有更多软件可供选择

– 源码包适用各种平台

步骤1:安装依赖关系包(安装开发工具)

[root@svr7 ~]# yum -y install gcc make

[root@svr7 ~]# rpm -q gcc

gcc-4.8.5-28.el7.x86\_64

[root@svr7 ~]# rpm -q make

make-3.82-23.el7.x86\_64

步骤2:tar解包,释放源代码至指定目录

]# tar -xf /tools/inotify-tools-3.13.tar.gz -C /

]# ls /

]# cd /inotify-tools-3.13/

]# ls

步骤3: ./configure 配置,指定安装目录/功能模块等选项

./=当前路径下运行 作用1:检测本机是否安装gcc

作用2:指定安装目录/功能模块等选项

--prefix=指定安装位置

]# cd /inotify-tools-3.13/

]# ./configure --prefix=/mnt/myrpm

常见报错提示:没有安装gcc

checking for gcc... no

checking for cc... no

checking for cl.exe... no

configure: error: no acceptable C compiler found in $PATH

See `config.log' for more details.

步骤4:make 编译,生成可执行的二进制程序文件

]# cd /inotify-tools-3.13/

]# make

步骤5:make install 安装,将编译好的文件复制到安装目录

]# cd /inotify-tools-3.13/

]# make install

]# ls /mnt/

]# ls /mnt/myrpm/ #查看安装生成的目录

]# ls /mnt/myrpm/bin/

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

一 真机远程管理虚拟机A

虚拟化概述

• 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、硬件辅助(CPU)

CPU能够直接识别运算虚拟机指令

虚拟机A:

[root@svr7 ~]# lscpu | grep vmx #查看CPU是否支持虚拟化

真机查看CPU是否支持虚拟化

[student@room9pc01 ~]$ lscpu | grep vmx

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

• 虚拟化主要厂商及产品

VMware VMware Workstation、vSphere

Microsoft VirtualPC、Hyper-V

RedHat KVM、RHEV

Citrix Xen

Oracle Oracle VM VirtualBox

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

二 手动安装一台有图形的虚拟机nsd01

操作系统为:CentOS 7.5

内存为:2048M

硬盘大小:9G

cpu:1

虚拟名为:nsd01

网络类型:private1

软件包选择:带GUI的服务器

分区:自动分区

root密码设置为1 创建lisi用户密码设置为1

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

三 针对虚拟机nsd01进行配置

1.关闭SELinux

[root@localhost ~]# getenforce

[root@localhost ~]# setenforce 0

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

SELINUX=permissive

2.防火墙设置为trusted

3.设置IP地址为:192.168.4.30/24

]# nmcli connection modify eth0

ipv4.method manual ipv4.addresses 192.168.4.30/24

connection.autoconnect yes

]# nmcli connection up eth0

]# ifconfig | head -2

4.主机名:kvm.tedu.cn

5.构建Yum仓库,使用真机FTP最为服务端

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

[root@localhost yum.repos.d]# mv \*.repo repo

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

虚拟机nsd01:安装虚拟化服务器平台

• 主要软件包

– 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

– virt-manager:图形管理工具

[root@kvm ~]# yum -y install qemu-kvm

[root@kvm ~]# yum -y install libvirt-daemon

[root@kvm ~]# yum -y install libvirt-daemon-driver-qemu

[root@kvm ~]# yum -y install libvirt-client

[root@kvm ~]# yum -y install virt-manager

[root@kvm ~]# virt-manager #图形虚拟系统管理器

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

虚拟化服务:

[root@kvm ~]# systemctl status libvirtd

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

真机上进行虚拟机管理:

virsh命令工具介绍

• 查看KVM节点(服务器)信息

– virsh nodeinfo

• 列出虚拟机

– virsh list [--all]

• 查看指定虚拟机的信息

– virsh dominfo 虚拟机名称

• 将指定的虚拟机设为开机自动运行

– virsh autostart [--disable] 虚拟机名称

• 强制关闭指定的虚拟机

– virsh destroy 虚拟机名称

• 运行|重启|关闭指定的虚拟机

– virsh start 虚拟机名称

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

xml配置文件(虚拟机描述文件)

默认存放路径:

[student@room9pc01 ~]$ ls /etc/libvirt/qemu

输出虚拟机xml文件内容

[student@room9pc01 ~]$ virsh dumpxml nsd01 | less

虚拟机名称: <name>nsd01</name>

<uuid>cf4e78cb-65bb-48df-a84b-56741a470fcb</uuid>

<source file='/var/lib/libvirt/images/nsd01.qcow2'/>

网卡MAC地址:网卡唯一编号<mac address='52:54:00:20:37:74'/>

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

一台KVM虚拟机的组成

– xml配置文件:定义虚拟机的名称、UUID、CPU、内

存、虚拟磁盘、网卡等各种参数设置

默认路径:/etc/libvirt/qemu

– 磁盘镜像文件:保存虚拟机的操作系统及文档数据,

镜像路径取决于xml配置文件中的定义

默认路径:/var/lib/libvirt/images/

真机:手动创建一个虚拟机abc01

1.创建虚拟机磁盘镜像文件

]$ cd /var/lib/libvirt/images/

]$ cp .node\_tedu.qcow2 abc01.qcow2

2.创建虚拟机xml配置文件

]$ virsh dumpxml nsd01 > /etc/libvirt/qemu/abc01.xml

]$ ls /etc/libvirt/qemu

]$ ls -l /etc/libvirt/qemu/abc01.xml

]$ vim /etc/libvirt/qemu/abc01.xml

虚拟机名称: <name>abc01</name>

<uuid>删除整行内容,如果删多了按u进行撤销

<source file='/var/lib/libvirt/images/abc01.qcow2'/>

<mac address='52:54:00:20:37:74'/> 删除整行内容

3.导入虚拟机信息

]$ virsh define /etc/libvirt/qemu/abc01.xml

]$ virsh list --all

]$ virsh start abc01

]$ virsh list

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

手动创建虚拟机:

1.创建新的虚拟机磁盘文件

2.导出虚拟机xml配置文件

]$ virsh dumpxml nsd01 > /etc/libvirt/qemu/abc01.xml

]$ vim /etc/libvirt/qemu/abc01.xml

虚拟机名称: <name>abc01</name>

<uuid>删除整行内容,如果删多了按u进行撤销

<source file='/var/lib/libvirt/images/abc01.qcow2'/>

<mac address='52:54:00:20:37:74'/> 删除整行内容

3.导入虚拟机信息

]$ virsh define /etc/libvirt/qemu/abc01.xml

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

三合一的命令:导出xml配置文件 修改 导入

1.创建一个虚拟机磁盘文件

]$ cd /var/lib/libvirt/images/

]$ cp .node\_tedu.qcow2 abc02.qcow2

2.三合一的命令:导出xml配置文件 修改 导入

[student@room9pc01 /]$ virsh edit nsd01

虚拟机名称: <name>abc02</name>

<uuid>删除整行内容,如果删多了按u进行撤销

<source file='/var/lib/libvirt/images/abc02.qcow2'/>

<mac address='52:54:00:20:37:74'/> 删除整行内容

[student@room9pc01 /]$ virsh list --all

[student@room9pc01 /]$ virsh start abc02

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

虚拟机命令行的删除

[student@room9pc01 /]$ virsh list

[student@room9pc01 /]$ virsh destroy abc02 #强制关机

域 abc02 被删除

[student@room9pc01 /]$ virsh list --all

[student@room9pc01 /]$ virsh undefine abc02 #取消定义

域 abc02 已经被取消定义

]$ virsh list --all

]$ cd /var/lib/libvirt/images/

]$ ls abc02.qcow2

]$ rm -rf abc02.qcow2 #删除磁盘文件

]$ ls

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

COW(Copy On Write)机制 写时复制

• 虚拟机的磁盘镜像文件格式

特点\类型 RAW QCOW2

KVM默认 否 是

I/O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

– 直接映射原始盘的数据内容

– 当前端盘的数据有修改时,在修改之前自动将原始盘的旧数据存入前端盘

– 对前端盘的修改不回写到原始盘

前端盘大小最好比原始盘大或相等

1.基于原始盘.node\_tedu.qcow2生成前端盘test.qcow2

]$ cd /var/lib/libvirt/images/

]$ qemu-img create -f qcow2 -b .node\_tedu.qcow2 test.qcow2 10G

]$ qemu-img info test.qcow2 #查看磁盘文件信息

2.修改虚拟机xml文件内容

]$ virsh edit nsd01

虚拟机名称: <name>test</name>

<uuid>删除整行内容,如果删多了按u进行撤销

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

<mac address='52:54:00:20:37:74'/> 删除整行内容

]$ virsh start test

]$ virsh list --all

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

离线访问虚拟机:虚拟机关机状态,直接修改磁盘文件里面的数据

]$ guestmount -a /var/lib/libvirt/images/tedu\_node09.img -i /home/student/nsdfile/

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

• DNS服务器的功能

– 正向解析:根据注册的域名查找其对应的IP地址

– 反向解析:根据IP地址查找对应的注册域名,不常用

所有的域名都必须以点作为结尾

根域：　　　　　　　　．

一级域名：　.cn .us .tw .hk .jp .kr

二级域名：　tedu.cn .com.cn .net.cn .org.cn

三级域名：　nb.com.cn haha.com.cn haxi.com.cn

完全合格的主机名: 主机头部+域名

该网站的FQDN为: www.qq.com

Full Qualified Domain Name(FQDN),完全合格主机名

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

bind-9.9.4-29.el7.x86\_64 //域名服务包

bind-chroot-9.9.4-29.el7.x86\_64 //提供虚拟根支持,牢笼政策

– 系统服务:named

– 默认端口:TCP/UDP 53

– 运行时的虚拟根环境:/var/named/chroot/

• 主配置文件:/etc/named.conf #设置负责解析域名 tedu.cn

• 地址库文件:/var/named/ #记录域名与IP地址对应关系

虚拟机A构建DNS服务

1.安装软件

[root@svr7 ~]# yum -y install bind bind-chroot

2.修改主配置文件

[root@svr7 ~]# cp /etc/named.conf /etc/named.bak

[root@svr7 ~]# vim /etc/named.conf

options {

directory "/var/named"; #地址库文件存放路径

};

zone "tedu.cn" IN { #设置负责解析的域名

type master; #类型为主DNS服务器

file "tedu.cn.zone"; #地址库文件名称

};

3.建立地址库文件

[root@svr7 ~]# cd /var/named/

[root@svr7 named]# cp -p named.localhost tedu.cn.zone

-p:保持源文件属性不变

[root@svr7 named]# ls -l tedu.cn.zone

[root@svr7 named]# vim tedu.cn.zone

所有的域名都必须以点结尾

如果没有以点结尾,那么自动补全本地址库负责的域名

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 1.1.1.1

ftp A 2.2.2.2

[root@svr7 named]# systemctl restart named

虚拟机B测试域名解析

1.指定本机DNS服务器/etc/resolv.conf

]# echo nameserver 192.168.4.7 > /etc/resolv.conf

]# cat /etc/resolv.conf

]# nslookup www.tedu.cn

]# nslookup ftp.tedu.cn

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

多区域的DNS服务器

1.修改主配置文件

options {

directory "/var/named";

};

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

2.建立地址库文件qq.com.zone

qq.com. NS svr7

svr7 A 192.168.4.7

www A 3.3.3.3

ftp A 4.4.4.4

3.重启named服务

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

搭建DNS服务器,实现www.sina.com解析结果为192.168.4.100

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

DNS服务器资源解析记录:

NS解析记录

A正向地址解析记录

CNAME解析记录的别名

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

特殊的解析记录

1.DNS轮询,基于DNS的站点负载均衡

– 一个域名 ---> 多个不同IP地址

[root@svr7 /]# vim /var/named/tedu.cn.zone

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.1

www A 192.168.4.2

www A 192.168.4.3

ftp A 2.2.2.2

2.泛域名解析

虚拟机A

[root@svr7 /]# vim /var/named/tedu.cn.zone

最后追加写入

\* A 1.2.3.4

tedu.cn. A 10.20.30.40

[root@svr7 /]# systemctl restart named

虚拟机B

[root@pc207 /]# nslookup wwww.tedu.cn

[root@pc207 /]# nslookup haxi.tedu.cn

[root@pc207 /]# nslookup tedu.cn

3.有规律的泛域名解析

pc1.tedu.cn ------>192.168.10.1

pc2.tedu.cn ------>192.168.10.2

pc3.tedu.cn ------>192.168.10.3

pc4.tedu.cn ------>192.168.10.4

..........

pc50.tedu.cn ------>192.168.10.50

内置函数: $GENERATE 制造连续范围数字

虚拟机A

[root@svr7 /]# vim /var/named/tedu.cn.zone

最后追加写入

$GENERATE 1-50 pc$ A 192.168.10.$

[root@svr7 /]# systemctl restart named

虚拟机B

[root@pc207 /]# nslookup pc1.tedu.cn

[root@pc207 /]# nslookup pc2.tedu.cn

4.解析记录的别名

[root@svr7 /]# vim /var/named/tedu.cn.zone

最后追加写入

tts CNAME ftp

[root@svr7 /]# systemctl restart named

虚拟机B

[root@pc207 /]# nslookup tts.tedu.cn

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

主机名映射文件:/etc/hosts

1.解析域名过程中,最高优先级

2.只为本机提供解析

虚拟机A:

[root@svr7 /]# vim /etc/hosts

[root@svr7 /]# tail -1 /etc/hosts

最后追加写入

192.168.4.110 www.360.com

[root@svr7 /]# ping www.360.com

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

虚拟机B,构建DNS服务器,负责解析bj.qq.com

1.安装软件bind bind-chroot

2.修改主配置文件

options {

directory "/var/named";

};

zone "bj.qq.com" IN {

type master;

file "bj.qq.com.zone";

};

[root@pc207 ~]# cd /var/named/

[root@pc207 named]# cp -p named.localhost bj.qq.com.zone

[root@pc207 named]# vim bj.qq.com.zone

bj.qq.com. NS pc207

pc207 A 192.168.4.207

www A 11.12.13.14

[root@pc207 named]# systemctl restart named

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

DNS子域授权

父域:www.qq.com

子域:www.bj.qq.com

虚拟机A为DNS服务,负责qq.com域名的解析

虚拟机B为DNS服务,负责bj.qq.com域名的解析

[root@pc207 /]# nslookup www.qq.com 192.168.4.7

[root@pc207 /]# nslookup www.bj.qq.com 192.168.4.207

解析请求www.bj.qq.com询问父域DNS服务器虚拟机A,也能够解析

虚拟机A:

[root@svr7 ~]# vim /var/named/qq.com.zone

qq.com. NS svr7

bj.qq.com. NS pc207

svr7 A 192.168.4.7

pc207 A 192.168.4.207

www A 3.3.3.3

ftp A 4.4.4.4

[root@svr7 ~]# systemctl restart named

[root@pc207 /]# nslookup www.bj.qq.com 192.168.4.7

Server: 192.168.4.7

Address: 192.168.4.7#53

Non-authoritative answer: #非权威解答

Name: www.bj.qq.com

Address: 11.12.13.14

递归查询

客户端请求域名解析,主DNS服务器与其他DNS服务器交互,最终将解析结果带回来

[root@svr7 ~]# vim /etc/named.conf

options {

directory "/var/named";

recursion no; #禁止DNS递归查询

};

迭代查询

主DNS服务器与其他DNS服务器交互

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

缓存DNS服务器,缓存解析记录加快解析

虚拟机A:互联网真DNS服务器

虚拟机B:缓存DNS服务器

虚拟机C:测试客户端

虚拟机C解析请求发送给虚拟机B,虚拟机B将请求发送给虚拟机A

一 准备虚拟机C,配置IP地址为192.168.4.10/24

二 配置虚拟机B

[root@pc207 /]# vim /etc/named.conf

options {

directory "/var/named";

forwarders { 192.168.4.7; }; #转发给192.168.4.7

};

[root@pc207 /]# systemctl restart named

[root@pc207 /]# nslookup www.tedu.cn 192.168.4.207

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

综合实验:

虚拟机A:DNS服务器

虚拟机B:Web服务器,提供www.tedu.cn与www.qq.com

虚拟机C:客户端测试

虚拟机B:

1.安装httpd软件包

[root@pc207 /]# yum -y install httpd

2.建立调用配置文件

[root@pc207 /]# vim /etc/httpd/conf.d/nsd01.conf

<VirtualHost \*:80>

ServerName www.tedu.cn

DocumentRoot /var/www/tedu

</VirtualHost>

<VirtualHost \*:80>

ServerName www.qq.com

DocumentRoot /var/www/qq

</VirtualHost>

[root@pc207 /]# mkdir /var/www/tedu /var/www/qq

[root@pc207 /]# echo '<h1>tedu' > /var/www/tedu/index.html

[root@pc207 /]# echo '<h1>QQ' > /var/www/qq/index.html

[root@pc207 /]# systemctl restart httpd

虚拟机A:

1.修改地址库文件记录

[root@svr7 ~]# vim /var/named/tedu.cn.zone

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.207

[root@svr7 ~]# vim /var/named/qq.com.zone

qq.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.207

[root@svr7 ~]# !sys

systemctl restart named

[root@svr7 ~]#

虚拟机C:测试

1.指定DNS服务器地址

]# echo nameserver 192.168.4.7 > /etc/resolv.conf

]# nslookup www.qq.com

]# nslookup www.tedu.cn

2.测试访问Web页面

]# firefox www.qq.com

]# firefox www.tedu.cn

Split分离解析(视图解析)

• 当收到客户机的DNS查询请求的时候

– 能够区分客户机的来源地址

– 为不同类别的客户机提供不同的解析结果(IP地址)

– 为不同的客户端,提供最近的解析结果

案例需求及要点

• 环境及需求

– 权威DNS:svr7.tedu.cn 192.168.4.7

– 负责区域:tedu.cn

– A记录分离解析 —— 以 www.tedu.cn 为例

客户机来自 解析结果

192.168.4.207 ---www.tedu.cn----> 192.168.4.100

其他地址 ---www.tedu.cn-----> 1.2.3.4

分离解析配置:

1.客户端匹配原则为,由上到下,匹配即停止

2.每一个客户端都必须找到自己的分类

3.所有的zone都必须在view中

虚拟机A:

1.修改主配置文件

options {

directory "/var/named";

};

view "nsd" {

match-clients { 192.168.4.207; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";---->写入解析结果192.168.4.100

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";---->写入解析结果1.2.3.4

};

};

2.建立不同地址库文件

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

多区域的分离解析

客户机来自 解析结果

192.168.4.207 -------> 192.168.4.100

其他地址 --------> 1.2.3.4

1.每一个view中zone的个数,以及zone负责的域名均要一致

view "nsd" {

match-clients { 192.168.4.207; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";

};

zone "qq.com" IN {

type master;

file "qq.com.other";

};

};

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

了解内容:

acl地址列表

acl test { 192.168.4.207; 192.168.4.1; 192.168.7.0/24; 192.168.4.2; 192.168.4.27; };

view "nsd" {

match-clients { test; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

};

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

RAID磁盘阵列

• 廉价冗余磁盘阵列

– Redundant Arrays of Inexpensive Disks

– 通过硬件/软件技术,将多个较小/低速的磁盘整合成一个大磁盘

– 阵列的价值:提升I/O效率、硬件级别的数据冗余

– 不同RAID级别的功能、特性各不相同

• RAID 0,条带模式

– 同一个文档分散存放在不同磁盘

– 并行写入以提高效率

– 至少由两块磁盘组成

• RAID 1,镜像模式

– 一个文档复制成多份,分别写入不同磁盘

– 多份拷贝提高可靠性,效率无提升

– 至少由两块磁盘组成

• RAID5,高性价比模式

– 相当于RAID0和RAID1的折中方案

– 需要至少一块磁盘的容量来存放校验数据

– 至少由三块磁盘组成

• RAID6,高性价比/可靠模式

– 相当于扩展的RAID5阵列,提供2份独立校验方案

– 需要至少两块磁盘的容量来存放校验数据

– 至少由四块磁盘组成

• RAID 0+1/RAID 1+0

– 整合RAID 0、RAID 1的优势

– 并行存取提高效率、镜像写入提高可靠性

– 至少由四块磁盘组成

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

进程管理

程序:静态的代码 占用磁盘空间

进程:动态执行的代码 占用CPU 内存

父进程与子进程 结构:树型结构

僵尸进程 孤儿进程

PID:进程的编号

查看进程树

• pstree — Processes Tree

– 格式:pstree [选项] [PID或用户名]

systemd:上帝进程,所有进程的父进程

[root@svr7 /]# pstree

• 常用命令选项

– -a:显示完整的命令行

– -p:列出对应PID编号

[root@svr7 ~]# pstree lisi #查看lisi用户开启的进程

bash───vim

[root@svr7 ~]# pstree -p lisi

bash(3394)───vim(3419)

[root@svr7 ~]# pstree -ap lisi

bash,3394

└─vim,3419 a.txt

[root@svr7 ~]#

• ps aux 操作

– 列出正在运行的所有进程,信息非常全面

用户 进程ID %CPU %内存 虚拟内存 固定内存 终端 状态 起始时间 CPU时间 程序指令

• ps -elf 操作

– 列出正在运行的所有进程,可以显示该进程的父进程的PID

PPID:父进程的PID号

请计算当前系统中运行的进程个数?

]# wc -l /etc/passwd

]# ps aux | wc -l

]# find /etc/ -name '\*tab' -type f

]# find /etc/ -name '\*tab' -type f | wc -l

]# find /etc/ -name '\*.conf' -type f

]# find /etc/ -name '\*.conf' -type f | wc -l

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

进程动态排名

• top 交互式工具

– 格式:top [-d 刷新秒数] [-U 用户名]

[root@svr7 ~]# top -d 1

P(大写)进行CPU排序

M(大写)进行内存排序

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

检索进程

• pgrep — Process Grep

– 用途:pgrep [选项]... 查询条件

• 常用命令选项

– -l:输出进程名,而不仅仅是 PID

– -U:检索指定用户的进程

– -x:精确匹配完整的进程名

[root@svr7 ~]# pgrep -l a

[root@svr7 ~]# pgrep -l log

[root@svr7 ~]# pgrep -U lisi

[root@svr7 ~]# pgrep -lU lisi

[root@svr7 ~]# pstree -ap lisi

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

进程的前后台调度

• 后台启动

– 在命令行末尾添加“&”符号,不占用当前终端

• Ctrl + z 组合键

– 挂起当前进程(暂停并转入后台)

• jobs 命令

– 查看后台任务列表

• fg 命令

– 将后台任务恢复到前台运行

• bg 命令

– 激活后台被挂起的任务

[root@svr7 ~]# sleep 1000 & #将进程正在运行放入后台

[root@svr7 ~]# jobs #查看后台进程信息

[root@svr7 ~]# sleep 800

^Z #按Ctrl+z 暂停放入后台

[2]+ 已停止 sleep 800

[root@svr7 ~]# jobs

[root@svr7 ~]# bg 2 #将后台编号为2的进程,继续运行

[root@svr7 ~]# jobs

[root@svr7 ~]# fg 1 #将后台编号为1的进程,恢复到前台运行

sleep 1000

^C #按Ctrl+c结束

[root@svr7 ~]# jobs

[root@svr7 ~]# fg 2 #将后台编号为2的进程,恢复到前台运行

sleep 800

^C #按Ctrl+c结束

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

• 干掉进程的不同方法

– Ctrl+c 组合键,中断当前命令程序

– kill [-9] PID... 、kill [-9] %后台任务编号

– killall [-9] 进程名...

– pkill 查找条件

[root@svr7 ~]# sleep 100 &

[root@svr7 ~]# sleep 200 &

[root@svr7 ~]# sleep 300 &

[root@svr7 ~]# jobs -l

[root@svr7 ~]# kill 3573

[root@svr7 ~]# jobs

[root@svr7 ~]# killall sleep

[root@svr7 ~]# jobs

[root@svr7 ~]# sleep 400 &

[root@svr7 ~]# jobs

[root@svr7 ~]# killall -9 sleep #强制杀

[root@svr7 ~]# jobs

强制踢出一个用户(杀死一个用户开启的所有进程)

[root@svr7 ~]# killall -9 -u lisi

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

日志管理

日志的功能

• 系统和程序的“日记本”

– 记录系统、程序运行中发生的各种事件

– 通过查看日志,了解及排除故障

– 信息安全控制的 依据

• 由系统服务rsyslog统一记录/管理

• 常见的日志文件

/var/log/messages 记录内核消息、各种服务的公共消息

/var/log/dmesg 记录系统启动过程的各种消息

/var/log/cron 记录与cron计划任务相关的消息

/var/log/maillog 记录邮件收发相关的消息

/var/log/secure 记录与访问限制相关的安全消息

日志分析

• 通用分析工具

– tail、tailf、less、grep等文本浏览/检索命令

– awk、sed等格式化过滤工具

tailf:实时跟踪日志消息

[root@svr7 ~]# echo 123 > /opt/1.txt

[root@svr7 ~]# cat /opt/1.txt

123

[root@svr7 ~]# tailf /opt/1.txt

123

haha

• users、who、w 命令

– 查看已登录的用户信息,详细度不同

pts/0:图形终端

• last、lastb 命令

– 查看最近登录成功/失败的用户信息

[root@svr7 ~]# users

[root@svr7 ~]# who

[root@svr7 ~]# w

[root@svr7 ~]# last -2 #最近登录成功的用户记录

[root@svr7 ~]# lastb -2 #最近登录失败的用户记录

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

日志消息的优先级

• Linux内核定义的事件紧急程度

– 分为 0~7 共8种优先级别

– 其数值越小,表示对应事件越紧急/重要

0 EMERG（紧急） 会导致主机系统不可用的情况

1 ALERT（警告） 必须马上采取措施解决的问题

2 CRIT（严重） 比较严重的情况

3 ERR（错误） 运行出现错误

4 WARNING（提醒） 可能会影响系统功能的事件

5 NOTICE（注意） 不会影响系统但值得注意

6 INFO（信息） 一般信息

7 DEBUG（调试） 程序或系统调试信息等

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

使用journalctl工具

• 提取由 systemd-journal 服务搜集的日志

– 主要包括内核/系统日志、服务日志

• 常见用法

– journalctl | grep 关键词

– journalctl -u 服务名 [-p 优先级]

– journalctl -n 消息条数

– journalctl --since="yyyy-mm-dd HH:MM:SS" --

until="yyyy-mm-dd HH:MM:SS"

[root@svr7 ~]# journalctl -u httpd

[root@svr7 ~]# journalctl -xe

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

systemctl控制

• Linux系统和服务管理器

– 是内核引导之后加载的第一个初始化进程(PID=1)

– 负责掌控整个Linux的运行/服务资源组合

systemd

• 一个更高效的系统&服务管理器

– 开机服务并行启动,各系统服务间的精确依赖

– 配置目录:/etc/systemd/system/

– 服务目录:/lib/systemd/system/

– 主要管理工具:systemctl

对于服务的管理

systemctl restart 服务名 #重起服务

systemctl start 服务名 #开启服务

systemctl stop 服务名 #停止服务

systemctl status 服务名 #查看服务当前的状态

systemctl enable 服务名 #设置服务开机自启动

systemctl disable 服务名 #设置服务不开机自启动

RHEL6 运行级别

0：关机

1：单用户模式（基本功能的实现，破解Linux密码）

2：多用户字符界面（不支持网络）

3：多用户字符界面（支持网络）服务器默认的运行级别

4：未定义

5：图形界面

6：重起

切换运行级别：init 5

RHEL7 运行模式

字符模式：multi-user.target

图形模式：graphical.target

[student@room9pc01 ~]$ rht-vmctl reset classroom

classroom [OK]

域 classroom 已开始

[student@room9pc01 ~]$ rht-vmctl reset server

server [OK]

域 server 已开始

[root@svr7 /]# ls -l /lib/systemd/system/

当前直接切换到字符模式

[root@svr7 /]# systemctl isolate multi-user.target

当前直接切换到图形模式

[root@svr7 /]# systemctl isolate graphical.target

查看每次开机默认进入模式

[root@svr7 /]# systemctl get-default

graphical.target

设置永久策略，每次开机自动进入graphical.target

# systemctl set-default multi-user.target

# reboot

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

综合实验:

1.构建虚拟机C与虚拟机D的Web服务器

虚拟机C:

[root@svr10 ~]# vim /etc/httpd/conf.d/nsd01.conf

<VirtualHost \*:80>

ServerName www.qq.com

DocumentRoot /var/www/qq

</VirtualHost>

<VirtualHost \*:80>

ServerName www.163.com

DocumentRoot /var/www/163

</VirtualHost>

[root@svr10 ~]# mkdir /var/www/qq /var/www/163

[root@svr10 ~]# echo '<h1>Web1 QQ' > /var/www/qq/index.html

[root@svr10 ~]# echo '<h1>Web1 163' > /var/www/163/index.html

虚拟机D

[root@svr10 ~]# vim /etc/httpd/conf.d/nsd01.conf

<VirtualHost \*:80>

ServerName www.qq.com

DocumentRoot /var/www/qq

</VirtualHost>

<VirtualHost \*:80>

ServerName www.163.com

DocumentRoot /var/www/163

</VirtualHost>

[root@svr10 ~]# mkdir /var/www/qq /var/www/163

[root@svr10 ~]# echo '<h1>Web2 QQ' > /var/www/qq/index.html

[root@svr10 ~]# echo '<h1>Web2 163' > /var/www/163/index.html

2.虚拟机A构建DNS服务器

[root@svr7 ~]# vim /etc/named.conf

options {

directory "/var/named";

};

view "nsd" {

match-clients { 192.168.4.7; };

zone "163.com" IN {

type master;

file "163.com.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

};

view "other" {

match-clients { any; };

zone "163.com" IN {

type master;

file "163.com.other";

};

zone "qq.com" IN {

type master;

file "qq.com.other";

};

};

[root@svr7 ~]# cd /var/named/

[root@svr7 named]# vim qq.com.zone

qq.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.10

[root@svr7 named]# vim qq.com.other

qq.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.20

[root@svr7 named]# cp -p qq.com.zone 163.com.zone

[root@svr7 named]# vim 163.com.zone

163.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.10

[root@svr7 named]# cp -p 163.com.zone 163.com.other

[root@svr7 named]# vim 163.com.other

163.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.20

[root@svr7 /]# systemctl restart named

3.指定DNS服务器地址

[root@svr7 /]# echo 'nameserver 192.168.4.7' > /etc/resolv.conf

[root@pc207 ~]# echo 'nameserver 192.168.4.7' > /etc/resolv.conf

[root@svr10 ~]# echo 'nameserver 192.168.4.7' > /etc/resolv.conf

[root@pc20 ~]# echo 'nameserver 192.168.4.7' > /etc/resolv.conf

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

部署DHCP服务器

• Dynamic Host Configuration Protocol

– 动态主机配置协议,由 IETF(Internet 网络工程师任

务小组)组织制定,用来简化主机地址分配管理

• 主要分配以下入网参数

– IP地址/子网掩码/广播地址

– 默认网关地址、DNS服务器地址

• DHCP地址分配的四次会话(以广播形式进行,先到先得)

– DISCOVERY --> OFFER --> REQUEST -->ACK

在一个网络中,只能有一个DHCP服务器

• 服务端基本概念

– 租期:允许客户机租用IP地址的时间期限,单位为秒

– 作用域:分配给客户机的IP地址所在的网段

– 地址池:用来动态分配的IP地址的范围

虚拟机A:

1.安装dhcp软件包

[root@svr7 ~]# yum -y install dhcp

2.修改配置文件

[root@svr7 ~]# vim /etc/dhcp/dhcpd.conf

末行模式下:

:r /usr/share/doc/dhcp\*/dhcpd.conf.example

subnet 192.168.4.0 netmask 255.255.255.0 { #分配的网段

range 192.168.4.100 192.168.4.200; #分配的IP范围

option domain-name-servers 192.168.4.7; #分配DNS地址

option routers 192.168.4.254; #分配网关地址

default-lease-time 600;

max-lease-time 7200;

}

3.重启dhcpd服务

[root@svr7 ~]# systemctl restart dhcpd

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

网络装机概述

网络装机的优势

• 规模化:同时装配多台主机

• 自动化:装系统、配置各种服务

• 远程实现:不需要光盘、U盘等物理安装介质

什么是PXE网络

• PXE,Pre-boot eXecution Environment

– 预启动执行环境,在操作系统之前运行

– 可用于远程安装

• 工作模式

– PXE client 集成在网卡的启动芯片中

– 当计算机引导时,从网卡芯片中把PXE client调入内存

执行,获取PXE server配置、显示菜单,根据用户选

择将远程引导程序下载到本机运行

•服务端需要哪些服务组件

– DHCP服务,分配IP地址、定位引导程序

– TFTP服务,提供引导程序下载

– HTTP服务,提供yum安装源

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

在虚拟机A构建PXE网络装机服务器

一 配置DHCP服务器,指定下一个服务器地址

[root@svr7 ~]# vim /etc/dhcp/dhcpd.conf

subnet 192.168.4.0 netmask 255.255.255.0 {

range 192.168.4.100 192.168.4.200;

option domain-name-servers 192.168.4.7;

option routers 192.168.4.254;

default-lease-time 600;

max-lease-time 7200;

next-server 192.168.4.7; #指定下一个服务器地址

filename "pxelinux.0"; #指定网卡引导文件名称

}

[root@svr7 ~]# systemctl restart dhcpd

pxelinux.0:网卡引导文件 ,安装说明书,二进制文件

安装一个软件即可获得,默认生成的名字pxelinux.0

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

二 构建tftp服务

tftp:简单的文件传输协议

端口默认为:69

默认共享位置:/var/lib/tftpboot

1.安装软件包为tftp-server

[root@svr7 ~]# yum -y install tftp-server

2.重启服务

[root@svr7 ~]# systemctl restart tftp

3.部署pxelinux.0引导文件

]# yum provides \*/pxelinux.0 #查询那个包产生的该文件

]# yum -y install syslinux

]# rpm -ql syslinux | grep pxelinux.0 #查询软件包安装清单

]# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/

]# ls /var/lib/tftpboot/

客户端----->DHCP---->next-server--->pxelinux.0

pxelinux.0---->/var/lib/tftpboot/pxelinux.cfg/default

4.部署默认菜单文件

首先关闭虚拟机A,图形添加一个光驱设备,放入光盘文件

[root@svr7 ~]# ls /dev/cdrom

[root@svr7 ~]# mount /dev/cdrom /mnt/

[root@svr7 ~]# ls /mnt/

]# mkdir /var/lib/tftpboot/pxelinux.cfg

]# cp /mnt/isolinux/isolinux.cfg /var/lib/tftpboot/pxelinux.cfg/default

]# ls -l /var/lib/tftpboot/pxelinux.cfg/default

5.部署图形模块(vesamenu.c32)和背景图片(splash.png)

]# cp /mnt/isolinux/vesamenu.c32 /mnt/isolinux/splash.png /var/lib/tftpboot/

]# ls /var/lib/tftpboot/

pxelinux.0 pxelinux.cfg splash.png vesamenu.c32

6.部署启动内核(vmlinuz)和驱动程序(initrd.img)

]# cp /mnt/isolinux/vmlinuz /mnt/isolinux/initrd.img /var/lib/tftpboot/

]# ls /var/lib/tftpboot/

initrd.img pxelinux.cfg vesamenu.c32

pxelinux.0 splash.png vmlinuz

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

总结:

dhcp--->IP地址 next-server filename

tftp--->pxelinux.0

pxelinux.0--->读取default菜单文件

default--->vesamenu.c32 splash.png vmlinuz initrd.img

7.修改菜单文件内容

]# vim /var/lib/tftpboot/pxelinux.cfg/default

末行模式 :set nu

1 default vesamenu.c32 #默认加载运行图形的模块

2 timeout 600 #默认读秒时间 1/10秒

10 menu background splash.png #背景图片

11 menu title NSD1904 PXE Server #显示标题信息

61 label linux

62 menu label Install CentOS7.5 #屏幕显示信息

63 menu default #读秒结束后默认选项

63 kernel vmlinuz #指定启动内核

64 append initrd=initrd.img #指定驱动程序

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

[root@svr7 ~]# systemctl restart dhcpd

[root@svr7 ~]# systemctl restart tftp

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

初步测试:

新建一台全新的虚拟机

1.安装方式选择:PXE网络引导安装

2.内存必须2G

3.网络类型选择:private1

总结:

dhcp--->IP地址 next-server filename

tftp--->pxelinux.0

pxelinux.0--->读取default菜单文件

default--->vesamenu.c32 splash.png vmlinuz initrd.img

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

三 构建Web服务器,将光盘内容共享给客户端

1.安装软件包

[root@svr7 ~]# yum -y install httpd

[root@svr7 ~]# mkdir /var/www/html/centos

[root@svr7 ~]# ls /var/www/html/centos

[root@svr7 ~]# mount /dev/cdrom /var/www/html/centos

mount: /dev/sr0 写保护，将以只读方式挂载

[root@svr7 ~]# ls /var/www/html/centos

[root@svr7 ~]# systemctl restart httpd

[root@svr7 ~]# firefox http://192.168.4.7/centos

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

四 部署无人值守安装,生成应答文件

1.安装一个system-config-kickstart图形化软件,生成应答文件

[root@svr7 ~]# yum -y install system-config-kickstart

2.运行图形化工具

以英文语言进行运行

[root@svr7 ~]# LANG=en system-config-kickstart

点击 "软件包选择(Package Selection)" 查看是否可以进行选择

需要Yum仓库的支持:指向光盘的内容仓库,标识必须为 [development]

[root@svr7 ~]# vim /etc/yum.repos.d/local.repo

[development]

name=CentOS-$releasever - Base

baseurl=ftp://192.168.4.254/centos-1804

enabled=1

gpgcheck=0

[root@svr7 ~]# LANG=en system-config-kickstart

点击 "软件包选择(Package Selection)" 查看是否可以进行选择

[root@svr7 ~]# ls /root/ks.cfg

/root/ks.cfg

[root@svr7 ~]# vim /root/ks.cfg

3.利用Web服务器共享应答文件

[root@svr7 ~]# cp /root/ks.cfg /var/www/html/

[root@svr7 ~]# ls /var/www/html/

centos ks.cfg

[root@svr7 ~]# firefox http://192.168.4.7/ks.cfg

4.通过菜单文件,指定ks应答文件获取方式

]# vim /var/lib/tftpboot/pxelinux.cfg/default

label linux

menu label Install CentOS7.5

menu default

kernel vmlinuz

append initrd=initrd.img ks=http://192.168.4.7/ks.cfg

]#

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

总结:

dhcp--->IP地址 next-server filename

tftp--->pxelinux.0

pxelinux.0--->读取default菜单文件

default--->图形模块 背景 内核 驱动 ks应答文件

ks应答文件--->语言 分区 密码 时区 ... 指定获取软件包的方式

获取软件包的方式---> --url="http://192.168.4.7/centos"

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

总结:

dhcp--->IP地址 next-server filename

tftp--->pxelinux.0

pxelinux.0--->读取default菜单文件

default--->图形模块 背景 内核 驱动 ks应答文件

ks应答文件--->语言 分区 密码 时区 ... 指定获取软件包的方式

获取软件包的方式---> --url="http://192.168.4.7/centos"

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

rsync同步操作

rsync同步操作

• 命令用法

– rsync [选项...] 源目录 目标目录

• 同步与复制的差异

– 复制:完全拷贝源到目标

– 同步:增量拷贝,只传输变化过的数据

同步控制

• rsync操作选项

– -n:测试同步过程,不做实际修改

– --delete:删除目标文件夹内多余的文档

– -a:归档模式,相当于-rlptgoD

– -v:显示详细操作信息

– -z:传输过程中启用压缩/解压

[root@svr7 ~]# mkdir /dir01 /abc

[root@svr7 ~]# cp /etc/passwd /etc/fstab /dir01

[root@svr7 ~]# ls /dir01

[root@svr7 ~]# rsync -avz /dir01 /abc/ #同步目录本身

[root@svr7 ~]# ls /abc

[root@svr7 ~]# ls /abc/dir01/

[root@svr7 ~]# rsync -avz /dir01/ /abc/ #同步目录内容

[root@svr7 ~]# ls /abc/

[root@svr7 ~]# touch /dir01/1.txt

[root@svr7 ~]# ls /dir01/

[root@svr7 ~]# rsync -avz /dir01/ /abc/

[root@svr7 ~]# ls /abc/

[root@svr7 ~]# ls /dir01/

[root@svr7 ~]# ls /abc/

[root@svr7 ~]# touch /abc/haha.txt

[root@svr7 ~]# ls /abc/

[root@svr7 ~]# rsync -avz --delete /dir01/ /abc/

[root@svr7 ~]# ls /abc/

[root@svr7 ~]# touch /abc/{1..5}.txt

[root@svr7 ~]# ls /abc/

[root@svr7 ~]# rsync -avz --delete /dir01/ /abc/

[root@svr7 ~]# ls /abc/

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

rsync+SSH同步

• 与远程的 SSH目录保持同步

– 下行:rsync [...] user@host:远程目录 本地目录

– 上行:rsync [...] 本地目录 user@host:远程目录

虚拟机A:

[root@svr7 ~]# rsync -avz --delete /dir01/ root@192.168.4.207:/opt/

虚拟机B:

[root@pc207 ~]# ls /opt/

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

虚拟机A:

[root@svr7 ~]# touch /dir01/xixi.txt

[root@svr7 ~]# rsync -avz --delete /dir01/ root@192.168.4.207:/opt/

虚拟机B:

[root@pc207 ~]# ls /opt

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

实时同步

虚拟机A

一 生成公钥与私钥,实现ssh无密码验证

[root@svr7 ~]# ssh-keygen #一路回车

[root@svr7 ~]# ls /root/.ssh/

authorized\_keys(别的机器传递过来的公钥) id\_rsa(私钥) id\_rsa.pub(公钥)

[root@svr7 ~]# ssh-copy-id root@192.168.4.207

[root@svr7 ~]# rsync -avz --delete /dir01/ root@192.168.4.207:/opt/

二 监控目录内容的变化

安装inotify-tools工具

真机上:

[student@room9pc01 ~]$ ls /linux-soft/01/

Cobbler.zip tools.tar.gz

[student@room9pc01 ~]$ scp /linux-soft/01/tools.tar.gz root@192.168.4.7:/root

1.让虚拟机A具备inotify-tools-3.13.tar.gz

[root@svr7 /]# ls /root/

[root@svr7 /]# mkdir /test

[root@svr7 /]# tar -xf /root/tools.tar.gz -C /test

[root@svr7 /]# ls /test

[root@svr7 /]# ls /test/tools/

步骤1:安装依赖关系包(安装开发工具)

]# yum -y install gcc make

步骤2:tar解包,释放源代码至指定目录

]# tar -xf /test/tools/inotify-tools-3.13.tar.gz -C /opt

]# ls /opt

]# cd /opt/inotify-tools-3.13/

]# ls

步骤3: ./configure 配置,指定安装目录/功能模块等选项

]# cd /opt/inotify-tools-3.13/

]# ./configure

步骤4:make 编译,生成可执行的二进制程序文件

]# make

步骤5:make install 安装,将编译好的文件复制到安装目录

]# make install

]# ls /usr/local/bin/inotifywait

inotifywait监控

• 基本用法

– inotifywait [选项] 目标文件夹

• 常用命令选项

– -m,持续监控(捕获一个事件后不退出)

– -r,递归监控、包括子目录及文件

– -q,减少屏幕输出信息

– -e,指定监视的 modify、move、create、delete、attrib等事件类别

三 书写Shell脚本

for循环:适合有规定次数的循环

while适合死循环:次数无限及无法确定

while [条件]

do

重复执行操作

done

[root@svr7 /]# vim /opt/rsync.sh

#!/bin/bash

while inotifywait -rqq /dir01

do

rsync -az --delete /dir01/ root@192.168.4.207:/opt/

done

[root@svr7 /]# chmod +x /opt/rsync.sh

[root@svr7 /]# /opt/rsync.sh

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

Cobbler装机平台,不同版本的多系统的安装

教学环境虚拟机3.1版本进行扩容

[root@svr7 ~]# df -h

文件系统 容量 已用 可用 已用% 挂载点

/dev/vda1 2.0G 1.8G 256M 88% /

[root@svr7 ~]# LANG=en growpart /dev/vda 1

[root@svr7 ~]# df -h

[root@svr7 ~]# xfs\_growfs /dev/vda1

[root@svr7 ~]# df -h

文件系统 容量 已用 可用 已用% 挂载点

/dev/vda1 20G 1.8G 19G 9% /

一 具备一个CentOS虚拟机

[root@svr7 ~]# cat /etc/redhat-release

CentOS Linux release 7.5.1804 (Core)

二、虚拟机设置

1.设置防火墙为trusted

]# firewall-cmd --set-default-zone=trusted

2.当前及永久设置SELinux状态为permissive

]# setenforce 0 #当前临时关闭

]# getenforce

]# vim /etc/selinux/config

SELINUX=

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

[root@svr7 ~]# LANG=en growpart /dev/vda 1

[root@svr7 ~]# df -h

[root@svr7 ~]# xfs\_growfs /dev/vda1

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

三、利用scp真机传递cobbler.zip包到虚拟机192.168.4.7中

# scp /home/student/桌面/cobbler.zip root@192.168.4.7:/root/

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

四、搭建Cobbler装机平台

Cobbler概述软件，管理dhcp、tftp、Web服务

自由的导入镜像与ks应答文件

1.解压cobbler.zip包

[root@svr7 ~]# yum -y install unzip

[root@svr7 ~]# unzip /root/cobbler.zip -d /

[root@svr7 ~]# ls /cobbler/

cobbler\_boot.tar.gz cobbler\_web.png

cobbler\_rpm.zip

[root@svr7 ~]#

[root@svr7 ~]# unzip /cobbler/cobbler\_rpm.zip -d /opt/

[root@svr7 ~]# ls /opt/cobbler/

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

五、安装cobbler主程序、工具包等

]# yum -y install dhcp httpd mod\_ssl

]# yum -y install /opt/cobbler/\*.rpm

[root@svr7 ~]# rpm -q cobbler

cobbler-2.8.2-1.el7.x86\_64

[root@svr7 ~]#

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

cobbler网络装机部署

1.安装软件 cobbler cobbler-web dhcp tftp-server pykickstart httpd tftp-server

cobbler #cobbler程序包

cobbler-web #cobbler的web服务包

pykickstart #cobbler检查kickstart语法错误

httpd #Apache web服务

dhcp #dhcp服务

tftp-server #tftp服务

2.配置cobbler

[root@svr7 /]# vim /etc/cobbler/settings

next\_server: 192.168.4.7 #设置下一个服务器还为本机

server: 192.168.4.7 #设置本机为cobbler服务器

manage\_dhcp: 1 #设置cobbler管理dhcp服务

pxe\_just\_once: 1 #防止客户端重复安装操作系统

开机启动项： 匹配及停止

1.硬盘启动 2.光驱设备 3.U盘 4.网络引导

3.配置cobbler的dhcp

[root@svr7 /]# vim /etc/cobbler/dhcp.template

:%s /旧/新/g #全文替换

:%s /192.168.1/192.168.4/g #全文替换

5 次替换，共 4 行

4.绝对路径解压cobbler\_boot.tar.gz #众多的引导文件

# tar -tf /cobbler/cobbler\_boot.tar.gz #查看包里面内容

# tar -xPf /cobbler/cobbler\_boot.tar.gz #绝对路径释放

# ls /var/lib/cobbler/loaders/

5.启动相关服务

]# systemctl restart cobblerd

]# systemctl enable cobblerd

]# systemctl restart httpd

]# systemctl enable httpd

]# systemctl restart tftp

]# systemctl enable tftp

]# systemctl restart rsyncd #同步服务

]# systemctl enable rsyncd

6.同步刷新cobbler配置

[root@svr7 /]# cobbler sync #检测所有配置是否正确

......

\*\*\* TASK COMPLETE \*\*\*

]# firefox https://192.168.4.7/cobbler\_web

用户名：cobbler

密码：cobbler

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

cobbler应用

cobbler import --path=挂载点 --name=导入系统命名（随意起）

导入安装镜像数据

# mkdir /dvd

# mount /dev/cdrom /dvd

# ls /dvd/

# cobbler import --path=/dvd --name=CentOS7

cobbler导入的镜像放在：/var/www/cobbler/ks\_mirror

[root@cobbler /]# cobbler list #查看有哪些系统

distros:

CentOS7-x86\_64 #安装客户端至少2G内存

profiles:

CentOS7-x86\_64

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

删除

# cobbler list

# cobbler profile remove --name=CentOS7-x86\_64 #删除菜单信息

# cobbler distro remove --name=CentOS7-x86\_64 #删除镜像信息

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

# umount /dvd/

# mount /dev/cdrom /dvd 将光盘换成RHEL7

# ls /dvd/

安装完成机器 默认root 密码 cobbler

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

自定义应答文件：开头注释行删除

[root@cobbler ~]# system-config-kickstart #生成ks文件

必须默认kickstart文件存放位置：/var/lib/cobbler/kickstarts/

[root@cobbler ~]# cobbler list

修改kickstart文件：

[root@cobbler ~]# cobbler profile edit --name=CentOS7 --kickstart=/var/lib/cobbler/kickstarts/自定义.cfg

[root@cobbler ~]# cobbler profile report

[root@cobbler ~]# cobbler sync #同步配置

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