# 第一阶段系统&服务管理进阶

## 构建新的教学环境

1.构建两台虚拟机,操作系统为CentOS7.5

[student@room9pc01 ~]$ clone-vm7

Enter VM number: 8

Creating Virtual Machine disk image...... [OK]

Defining new virtual machine...... [OK]

[student@room9pc01 ~]$ clone-vm7

Enter VM number: 9

Creating Virtual Machine disk image...... [OK]

Defining new virtual machine...... [OK]

#修改两台虚拟机名为A与B

2.利用root进入两台虚拟机,密码为:123456

3.配置永久主机名

虚拟机A:svr7.tedu.cn

虚拟机B:pc207.tedu.cn

4.配置IP地址

虚拟机A:为eth0配置 192.168.4.7/24

虚拟机B:为eth0配置 192.168.4.207/24

5.利用真机进行远程管理两台虚拟机

[student@room9pc01 ~]$ ping 192.168.4.7

[student@room9pc01 ~]$ ping 192.168.4.207

[student@room9pc01 ~]$ ssh -X root@192.168.4.7

[student@room9pc01 ~]$ ssh -X root@192.168.4.207

6.为真机上设置永久别名

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

alias goa='ssh -X root@192.168.4.7'

alias gob='ssh -X root@192.168.4.207'

新开一个终端验证

7.构建Yum仓库

服务端:真机构建FTP服务器,共享光盘所有内容

1)真机检测软件包vsftpd的安装

[student@room9pc01 ~]$ rpm -q vsftpd

vsftpd-3.0.2-22.el7.x86\_64

[student@room9pc01 ~]$ systemctl status vsftpd

2)共享光盘所有内容

默认共享的路径:/var/ftp

]$ mkdir /var/ftp/centos-1804/

]$ mount /iso/CentOS7-1804.iso /var/ftp/centos-1804/

]$ ls /var/ftp/centos-1804/

]$ 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

客户端:两台虚拟机

[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 -y install httpd

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

虚拟机C

1.配置主机名为:svr10.tedu.cn

2.配置eth0的IP地址:192.168.4.10/24

3.配置Yum仓库,指向服务端为真机的FTP服务

虚拟机D

1.配置主机名为:pc20.tedu.cn

2.配置eth0的IP地址:192.168.4.20/24

3.配置Yum仓库,指向服务端为真机的FTP服务

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

扩展的几个应用

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

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

drwxrwxrwt. 8 root root 172 7月 15 10:34 /tmp

## 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 750 /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

## 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 #显示历史命令列表

[root@svr7 ~]# history -c #清空所有的历史命令

[root@svr7 ~]# history

1 history

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

[root@svr7 ~]# ls /root/

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

[root@svr7 ~]# history

[root@svr7 ~]# !cat #执行临时命令中最近以cat开头的命令

[root@svr7 ~]# !ls

## 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 -sh /root

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

[root@svr7 ~]# du -sh /

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

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

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

### 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

2019-07-15

[root@svr7 ~]# date +%Y #显示 年

2019

[root@svr7 ~]# date +%m #显示 月

07

[root@svr7 ~]# date +%d #显示 日期

15

[root@svr7 ~]# date +%H #显示 时

11

[root@svr7 ~]# date +%M #显示 分

42

[root@svr7 ~]# date +%S #显示 秒

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

制作快捷方式 制作连接(链接)文件

### 制作软连接

格式: ln -s /路径/源文件 /路径/快捷方式名 #必须时绝对路径

]# cat /etc/sysconfig/network-scripts/ifcfg-eth0

]# ln -s /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/net0

]# ls /etc/net0

]# cat /etc/net0

]# ls -l /etc/net0

]# ln -s /etc/passwd /opt/

]# ls /opt/

]# ls -l /opt/passwd

#### 软连接特点:

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

#### 硬连接特点:

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

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

[root@svr7 ~]# echo 123456 > /opt/A

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

[root@svr7 ~]# ln -s /opt/A /opt/B #制作软连接

[root@svr7 ~]# ls /opt/

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

[root@svr7 ~]# ln /opt/A /opt/C #制作硬连接

[root@svr7 ~]# ls /opt/

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

[root@svr7 ~]# ls /opt/

[root@svr7 ~]# cat /opt/B #软连接不可用

[root@svr7 ~]# cat /opt/C #硬连接可以使用

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

获取命令帮助

[root@svr7 ~]# man 5 passwd #5代表配置文件类型帮助信息

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

## zip归档工具,跨平台的压缩格式

### • 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 /root/ /etc/passwd

[root@svr7 ~]# ls /opt

[root@svr7 ~]# zip -r /opt/nsd.zip /boot/

[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 ~]# unzip /opt/nsd.zip -d /mnt/

[root@svr7 ~]# ls /mnt/

[root@svr7 ~]# ls /mnt/boot/

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

## 自定义Yum源

1.具备从互联网下载一些软件包

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

2.将tools.tar.gz传递到虚拟机A

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

3.在虚拟机A查看

]# ls /root

]# tar -xf /root/tools.tar.gz -C /

]# ls /

]# ls /tools/

]# ls /tools/other/

4.在虚拟机A生成仓库数据文件

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

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

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

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

5.书写客户端配置文件

[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

[myrpm]

name=other rpm

baseurl=file:///tools/other #指定为本机为Yum服务端

enabled=1

gpgcheck=0

[root@svr7 ~]# yum repolist

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

[root@svr7 ~]# cmatrix

[root@svr7 ~]# cmatrix -C red

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

[root@svr7 ~]# sl

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

[root@svr7 ~]# oneko

[root@svr7 ~]# oneko & #将程序放入后台

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

## vim编辑技巧

命令模式操作

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

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

PgUp 键、PgDn 键 向上翻页、向下翻页

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、3yy 复制光标处的一行、3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,进入插入模式

### 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(大写) 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

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

### 末行模式操作

:r /etc/filesystems 读入其他文件内容

[root@svr7 ~]# echo 123456 > /opt/2.txt

[root@svr7 ~]# echo hahaxixi > /opt/3.txt

[root@svr7 ~]# vim /opt/3.txt

:r /opt/2.txt #读入/opt/2.txt到当前文件

:r /etc/passwd #读入/etc/passwd到当前文件

### 字符串替换

:s/root/admin 替换当前行第一个“root”

:s/root/admin/g 替换当前行所有的“root”

:1,10 s/root/admin/g 替换第1-10行所有的“root”

:% s/root/admin/g 替换文件内所有的“root

[root@svr7 ~]# cat /etc/passwd > /opt/a.txt

[root@svr7 ~]# vim /opt/a.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 -y install

源码包----gcc make---->可以执行的程序-------->运行安装

### • 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

步骤一:安装gcc与make开发工具

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

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

步骤二:进行tar解包

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

]# ls /

]# cd /inotify-tools-3.13/

]# ls

步骤三:运行configure脚本

作用1:检测是否安装gcc

作用2:可以选择软件的功能,也可以指定软件安装的路径

--prefix=指定安装的位置

]# cd /inotify-tools-3.13/

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

常见提示: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.

步骤四:make编译生成可以执行的程序

]# cd /inotify-tools-3.13/

]# make

步骤五:make install将可以执行的程序运行安装

]# cd /inotify-tools-3.13/

]# make install

]# ls /opt/

]# ls /opt/myrpm/

]# ls /opt/myrpm/bin/

一 手动图形安装虚拟机名为nsd01

利用光盘镜像安装

内存2048M

虚拟机名为:nsd01

网络类型选择为:private1

软件包:带GUI的服务器

分区:自动分区

设置root的密码为1,建立lisi用户设置密码1

## 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支持虚拟化所有的指令集(vmx)

[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 虚拟机nsd01进行配置

1.配置永久主机名:kvm.tedu.cn

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

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

2.配置永久IP地址:192.168.4.30/24

[root@localhost ~]# nmcli connection modify eth0 ipv4.method manual ipv4.addresses 192.168.4.30/24 connection.autoconnect yes

[root@localhost ~]# nmcli connection up eth0

[root@localhost ~]# ifconfig

3.修改防火墙默认区域与SELinux

[root@kvm ~]# firewall-cmd --set-default-zone=trusted

[root@kvm ~]# setenforce 0

[root@kvm ~]# getenforce

Permissive

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

SELINUX=permissive

4.构建Yum仓库

]# cd /etc/yum.repos.d/

]# mkdir repo

]# ls

]# mv \*.repo repo

]# ls

]# vim dvd.repo

[dvd]

name=centos7

baseurl=ftp://192.168.4.254/centos-1804

enabled=1

gpgcheck=0

]# yum repolist

]# yum -y install xeyes

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

## 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

]# yum -y install qemu-kvm

]# yum -y install libvirt-daemon

]# yum -y install libvirt-client

]# yum -y install libvirt-daemon-driver-qemu

]# yum -y install virt-manager

### 虚拟化服务: libvirtd

[root@kvm /]# systemctl status libvirtd #查看服务的状态

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

## 三 通过命令管理虚拟机

### 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 控制指令 [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] #重点

#### 列出虚拟网络

]$ virsh net-list [--all]

#### 查看指定虚拟机的信息

]$ virsh dominfo 虚拟机名称 #重点

]$ virsh autostart nsd01 #设置虚拟机开机自启动

]$ virsh dominfo nsd01

]$ virsh autostart --disable nsd01 #禁止虚拟机开机自启动

]$ virsh dominfo nsd01

Virsh console 虚拟机名称 真机连接虚拟机(在虚拟机没有的情况) #重点

#### 开关机操作

]$ virsh start或reboot或shutdown 虚拟机名称 #重点

• 强制关闭指定的虚拟机

]$ virsh destroy 虚拟机名称 #重点

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

## • 一台KVM虚拟机的组成

– xml配置文件(虚拟机描述文件):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

[student@room9pc01 ~]$ ls /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

[student@room9pc01 ~]$ ls /var/lib/libvirt/images/

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

### 查看虚拟机xml文件内容

[student@room9pc01 /]$ virsh dumpxml nsd01

虚拟机名称

虚拟机的UUID唯一标识

虚拟机磁盘文件路径

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

虚拟机网卡信息

<interface type='network'>

<mac address='52:54:00:d4:fe:59'/>

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

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

## 在命令行手动制作一台虚拟机

### 1.磁盘镜像文件

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

]$ du -sh .node\_base.qcow2

]$ cp .node\_base.qcow2 nsd02.qcow2

]$ du -sh nsd02.qcow2

### 2.建立新的xml配置文件, uuidgen可以生成随机的UUID值

]$ cd /etc/libvirt/qemu/

]$ virsh dumpxml nsd01 > /etc/libvirt/qemu/nsd02.xml

]$ vim /etc/libvirt/qemu/nsd02.xml

1)虚拟机名称 <name>nsd02</name>

2)虚拟机的UUID唯一标识 删除整行内容 UUID

3)虚拟机磁盘文件路径 <source file='/var/lib/libvirt/images/nsd02.qcow2'/>

4)虚拟机网卡信息

<mac address='52:54:00:d4:fe:59'/> 删除整行内容

### 3.导入虚拟机配置信息

]$ virsh define /etc/libvirt/qemu/nsd02.xml

]$ virsh list --all

]$ virsh start nsd02

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

命令行删除虚拟机

]$ virsh list --all

]$ virsh destroy nsd02 #强制关闭虚拟机

域 nsd02 被删除

]$ virsh list --all

]$ virsh undefine nsd02 #删除虚拟机xml配置文件

域 nsd02 已经被取消定义

]$ virsh list --all

]$ rm -rf /var/lib/libvirt/images/nsd02.qcow2 #手动删除

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

## 在命令行手动创建虚拟机

### 1.建立新的磁盘文件

]$ cp /var/lib/libvirt/images/.node\_base.qcow2 /tmp/nsd03.qcow2

]$ du -sh /tmp/nsd03.qcow2

### 2.建立xml配置文件

]$ virsh dumpxml nsd01 > /home/student/nsd03.xml

]$ ls /home/student/nsd03.xml

]$ vim /home/student/nsd03.xml

1)虚拟机名称 <name>nsd03</name>

2)虚拟机的UUID唯一标识 删除整行内容 UUID

3)虚拟机磁盘文件路径 <source file='/tmp/nsd03.qcow2'/>

4)虚拟机网卡信息

<mac address='52:54:00:d4:fe:59'/> 删除整行内容

### 3.导入虚拟机配置信息

]$ ls /etc/libvirt/qemu

]$ virsh define /home/student/nsd03.xml #导入虚拟机信息

]$ ls /etc/libvirt/qemu

]$ virsh start nsd03

]$ virsh list

三合一:导出 修改 导入

• 对虚拟机的配置进行调整

编辑:virsh edit 虚拟机名

### 1.建立磁盘文件

]$ cp /var/lib/libvirt/images/.node\_base.qcow2 /var/lib/libvirt/images/nsd04.qcow2

]$ du -sh /var/lib/libvirt/images/nsd04.qcow2

### 2.建立xml配置文件

[student@room9pc01 /]$ virsh edit nsd01

1)虚拟机名称 <name>nsd04</name>

2)虚拟机的UUID唯一标识 删除整行内容 UUID

3)虚拟机磁盘文件路径 <source file='/var/libvirt/images/nsd04.qcow2'/>

4)虚拟机网卡信息

<mac address='52:54:00:d4:fe:59'/> 删除整行内容

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

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

## 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/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

### 查看磁盘镜像文件信息

]$ qemu-img info /var/lib/libvirt/images/nsd04.qcow2

virtual size: 2.0G (2147483648 bytes) #虚拟大小

disk size: 633M #实际占用空间

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

## 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,在修改之前自动将旧数据存入前端盘

– 对前端盘的修改不回写到原始盘

– 原始盘的数据不能做修改

## 快速创建qcow前端盘

• qemu-img 通过 -b 选项复用指定原始盘

– qemu-img create -f qcow2 -b 原始盘 前端盘 大小

### 1.命令手动创建虚拟机磁盘文件

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

]$ qemu-img create -f qcow2 -b .node\_base.qcow2 nsd05.qcow2 5G #通过原始盘产生前端盘

]$ qemu-img info nsd05.qcow2 #查看虚拟磁盘文件信息

virtual size: 5.0G (5368709120 bytes) #虚拟大小

disk size: 196K #实际占用空间大小

backing file: .node\_base.qcow2 #原始盘

### 2.建立xml文件

]$ virsh dumpxml nsd01 > /tmp/nsd05.xml #导出虚拟机信息

]$ vim /tmp/nsd05.xml #修改 虚拟机名 UUID 磁盘路径 网卡MAC

]$ virsh define /tmp/nsd05.xml

## 离线访问虚拟机

• 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况下,直接修改磁盘中的文档

– 方便对虚拟机定制、修复、脚本维护

]$ virsh destroy nsd01

]$ guestmount -a /var/lib/libvirt/images/nsd01.qcow2 -i /mnt/

## 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

### 域名结构: 树型结构

所有的域名都必须以点结尾

www.qq.com. www.baidu.com.

根域名: .

一级域名: .cn .us .kr .tw .hk .jp .......

二级域名: .com.cn .tedu.cn .net.cn .org.cn ......

三级域名: nb.com.cn dc.com.cn tc.com.cn ........

### 完全合格的域名(FQDN)=主机头部+注册的域名

Full Qualified Domain Name,完全合格主机名

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

– 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 #本机负责解析的域名

地址库文件存放路径:/var/named/ #域名的解析结果

## 构建基本DNS服务

虚拟机A:

### 1.安装软件包

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

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

### 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"; #指定地址库文件位置

};

zone "tedu.cn" IN { #设置本机负责的域名

type master; #设置本机为权威主DNS服务器

file "tedu.cn.zone"; #设置地址库文件名称

};

### 3.建立地址库文件tedu.cn.zone

]# cd /var/named/

]# cp -p named.localhost tedu.cn.zone #-p保持权限不变

]# ls -l tedu.cn.zone

]# vim tedu.cn.zone

所有的域名在地址库文件必须以点结尾

当没有以点结尾时,自动补全地址库负责的域名

tedu.cn. NS svr7 #声明区域DNS服务器

svr7 A 192.168.4.7 #声明DNS服务器IP地址

www A 1.1.1.1 #常规DNS正向解析

ftp A 2.2.2.2

[root@svr7 /]# systemctl restart named

虚拟机B:客户端验证

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

]# cat /etc/resolv.conf

]# nslookup www.tedu.cn

]# nslookup ftp.tedu.cn

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

多域名的DNS服务器,负责解析qq.com域名

虚拟机A:

### 1.修改主配置文件

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

options {

directory "/var/named";

};

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

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

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

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

qq.com. NS svr7

svr7 A 192.168.4.7

www A 3.3.3.3

[root@svr7 named]# systemctl restart named

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

### 特殊解析记录

#### 1.基于DNS解析记录负载均衡

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

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.10

www A 192.168.4.20

www A 192.168.4.30

ftp A 2.2.2.2

[root@svr7 /]# systemctl restart named

#### 2.泛域名解析

虚拟机A

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

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.10

ftp A 2.2.2.2

\* A 1.2.3.4

[root@svr7 /]# systemctl restart named

虚拟机B:测试 nslookup haha.tedu.cn

#### 3.解析记录的别名

虚拟机A:

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

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.10

ftp A 2.2.2.2

\* A 1.2.3.4

mail CNAME ftp

[root@svr7 /]# systemctl restart named

虚拟机B测试:

[root@pc207 ~]# nslookup mail.tedu.cn

#### 4.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 制造连续范围的数字

虚拟机A

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

$GENERATE 1-50 pc$ A 192.168.10.$

[root@svr7 ~]# !sys

systemctl restart named

[root@svr7 ~]#

虚拟机B测试:

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

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

[root@pc207 ~]# nslookup pc3.tedu.cn

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

## DNS资源解析记录有哪些?

正向解析记录(A记录)

DNS声明记录(NS记录)

解析记录的别名(CNAME记录)

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

DNS子域授权

父域:qq.com 由虚拟机A进行解析

子域:bj.qq.com 由虚拟机B进行解析

虚拟机B,构建DNS服务器,负责解析bj.qq.com

[root@pc207 ~]# yum -y install bind bind-chroot

[root@pc207 ~]# cp /etc/named.conf /etc/named.bak

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

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 50.60.70.80

[root@pc207 /]# systemctl restart named

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

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

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

子域授权: 让父域的DNS服务器知晓子域DNS服务器

让父域的DNS服务器可以解析子域的域名

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

[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: 50.60.70.80

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

## 递归解析:

由客户端发出请求,首选DNS服务器接受请求,将解析结果带回过程

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

options {

directory "/var/named";

recursion no; #禁止递归解析:

};

迭代解析:首选DNS服务器与其他DNS服务器交互过程

专业测试DNS工具

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

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

主机名映射文件:/etc/hosts

没有DNS服务器前提下,解析域名,只能为本机做域名解析

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

192.168.4.110 www.nb.com

[root@svr7 ~]# ping www.nb.com

域名解析过程中最高优先级为/etc/hosts 匹配即停止

然后在读取/etc/resolv.conf

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

### 缓存DNS,缓存解析结果,加快解析过程

三个角色:客户端 缓存DNS服务 真DNS服务器

虚拟机A:真DNS服务器

虚拟机B:缓存DNS服务

虚拟机C:客户端

1.建立虚拟机C

配置IP地址192.168.4.10/24,主机名为svr10.tedu.cn

虚拟机B:缓存DNS服务

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

options {

directory "/var/named";

forwarders { 192.168.4.7; }; #转发给虚拟机A的IP地址

};

[root@pc207 /]# systemctl restart named

虚拟机C上验证:

[root@svr10 ~]# nslookup www.tedu.cn 192.168.4.207

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

虚拟机A:构建DNS服务器,实现www.sina.com的解析结果为10.20.30.40

1.修改主配置文件

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

options {

directory "/var/named";

recursion no;

};

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

zone "sina.com" IN {

type master;

file "sina.com.zone";

};

2.建立地址库文件

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

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

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

sina.com. NS svr7

svr7 A 192.168.4.7

www A 10.20.30.40

[root@svr7 named]# systemclt restart named

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

## Split分离解析(视图解析)

什么是分离解析

• 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

– 不同客户端解析同一个域名,解析结果不同

– 作用:为客户端就近提供服务器

### BIND的view视图

• 根据源地址集合将客户机分类

由上到下匹配及停止,所有的客户端都要找到自己的分类

所有的zone都必须在view中

view "nsd" {

match-clients { 192.168.1.1; }; #匹配客户机来源地址

zone "12306.cn" IN {

...... 地址库文件12306.zone;

}; };

view "abc" {

match-clients { 192.168.2.1; };

zone "12306.cn" IN {

...... 地址库文件12306.abc;

}; };

view "other" {

match-clients { any; };

zone "12306.cn" IN {

...... 地址库文件12306.other;

}; };

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

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-----> 192.168.4.100

其他地址 -----> 1.2.3.4

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

options {

directory "/var/named";

};

view "nsd" { #分类名称

match-clients { 192.168.4.207; }; #匹配客户机来源地址

zone "tedu.cn" IN {

type master;

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

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

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

};

};

建立地址库文件tedu.cn.zone与tedu.cn.other写入不同的解析结果

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

多个域名的分离解析,每一个view中zone个数保持一致

– A记录分离解析 —— 以 www.tedu.cn 为例

– A记录分离解析 —— 以 www.sina.com 为例

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

view "nsd" {

match-clients { 192.168.4.207; };

zone "tedu.cn" IN {

type master;

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

};

zone "sina.com" IN {

type master;

file "sina.com.zone"; -------> 解析结果为192.168.4.100

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

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

};

zone "sina.com" IN {

type master;

file "sina.com.other"; -------> 解析结果为1.2.3.4

};

};

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

扩展:

– A记录分离解析 —— 以 www.tedu.cn 为例

– A记录分离解析 —— 以 www.sina.com 为例

客户机来自 解析结果

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

192.168.4.7---www.sina.com--> 192.168.4.200

其他地址 --www.tedu.cn---> 1.2.3.4

其他地址 --www.sina.com---> 1.2.3.4

view "nsd" {

match-clients { 192.168.4.207; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "sina.com" IN {

type master;

file "sina.com.other";

};

};

view "abc" {

match-clients { 192.168.4.7; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";

};

zone "sina.com" IN {

type master;

file "sina.com.zone";

};

};

view "other" {

match-clients { any; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.other";

};

zone "sina.com" IN {

type master;

file "sina.com.other";

};

};

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

### acl地址列表,类似于变量作用

acl "test" { 192.168.4.207; 192.168.1.1; 192.168.2.1; 192.168.3.1; 192.168.7.0/24; };

view "nsd" {

match-clients { test; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "sina.com" IN {

type master;

file "sina.com.other";

};

};

view "abc" {

match-clients { test; };

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "sina.com" IN {

type master;

file "sina.com.other";

};

};

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

## RAID磁盘阵列

• 廉价冗余磁盘阵列

– 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

– 至少需要两块磁盘组成

• 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

– 至少需要两块磁盘组成

• 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

– 至少需要三块磁盘组成

• 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

– 至少需要四块磁盘组成

• 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

– 至少需要四块磁盘组成

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

## 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/子进程 树型结构 僵尸进程 孤儿进程

进程唯一标识: PID 进程的编号

systemd:上帝进程,所有进程的父进程

### 查看进程树

• pstree — Processes Tree 15:15上课

– 格式:pstree [选项] [PID或用户名]

• 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

[root@svr7 /]# useradd lisi

[root@svr7 /]# pstree lisi

bash───vim

[root@svr7 /]# pstree -a lisi

bash

└─vim a.txt

[root@svr7 /]# pstree -ap lisi

• ps aux 操作,信息全面

– 列出正在运行的所有进程

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,有进程的父进程的PID值

– 列出正在运行的所有进程

PPID:父进程的PID号

统计正在运行的进程有多少?

[root@svr7 /]# ps aux

[root@svr7 /]# ps aux | wc -l

[root@svr7 /]# ps -elf | 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

按P(大写)可以进行CPU的排序

按M(大写)可以进行内存的排序

load average: 0.00, 0.01, 0.05 #CPU负载量 1分 5分钟 15 分钟

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 #检索所有进程名带a的

[root@svr7 /]# pgrep -lU lisi #检索lisi的所有进程

[root@svr7 /]# pstree -ap lisi #检索lisi的所有进程

进程的前后台调度

• 后台启动,正在运行放入后台

– 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 9000 & #正在运行放入后台

[root@svr7 /]# jobs #查看后台运行的进程

[root@svr7 /]# sleep 8000

^Z #输入Ctrl+z 暂停放入后台

[2]+ 已停止 sleep 8000

[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

sleep 9000

^C #输入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

sleep 8000

^C #输入Ctrl+C 结束

[root@svr7 /]# jobs

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

### 杀死进程

• 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 1000 &

[root@svr7 /]# jobs -l

[root@svr7 /]# kill 4013 #按照PID进行杀死

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# jobs -l

[root@svr7 /]# kill -9 4015 #按照PID强制杀死

[root@svr7 /]# jobs -l

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# sleep 1000 &

[root@svr7 /]# jobs

[root@svr7 /]# killall sleep #杀死所有sleep进程

杀死一个用户开启的所有进程(强制踢出一个用户)

[root@svr7 /]# killall -9 -u 用户名

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

## 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

– 日志消息采用文本格式

– 主要记录事件发生的时间、主机、进程、内容

### • 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

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

用户登录分析

• users、who、w 命令

– 查看已登录的用户信息,详细度不同

pts:图形命令行终端

• last、lastb 命令

– 查看最近登录成功/失败的用户信息

[root@svr7 /]# last -2 #最近登录的2条成功信息

[root@svr7 /]# lastb -2 #最近登录的2条失败信息

[root@svr7 /]# users #查看已登录的用户信息

[root@svr7 /]# who #查看已登录的用户信息

[root@svr7 /]# w #查看已登录的用户信息

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

### 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"

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

systemctl控制

• Linux系统和服务管理器

– systemd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

systemctl is-enabled 服务名 #查看是否为开机自启

### **RHEL6 运行级别**

不同级别,开启的服务不同

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

]# systemctl isolate multi-user.target = init 3

当前直接切换到图形模式

]# systemctl isolate graphical.target = init 5

查看每次开机默认进入模式

[root@svr7 /]# systemctl get-default

multi-user.target

设置永久策略，每次开机自动进入graphical.target

# systemctl set-default graphical.target

# reboot

### DNS分离解析与Web结合

虚拟机C:构建Web服务

[root@svr10 ~]# yum -y install httpd

[root@svr10 ~]# cat /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 ~]#

[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

[root@svr10 ~]# systemctl restart httpd

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

虚拟机D构建Web服务

[root@pc20 ~]# yum -y install httpd

[root@pc20 ~]# cat /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@pc20 ~]# mkdir /var/www/qq /var/www/163

[root@pc20 ~]# echo '<h1>Web2 QQ' > /var/www/qq/index.html

[root@pc20 ~]# echo '<h1>Web2 163' > /var/www/163/index.html

[root@pc20 ~]# systemctl restart httpd

[root@pc20 ~]#

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

虚拟机A实现DNS分离解析

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

[root@svr7 /]# cat /etc/named.conf

options {

directory "/var/named";

};

view "nsd" {

match-clients { 192.168.4.7; };

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

zone "163.com" IN {

type master;

file "163.com.zone";

};

};

view "other" {

match-clients { any; };

zone "qq.com" IN {

type master;

file "qq.com.other";

};

zone "163.com" IN {

type master;

file "163.com.other";

};

};

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

qq.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.10

[root@svr7 /]# cat /var/named/163.com.zone

163.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.10

[root@svr7 /]# cat /var/named/qq.com.other

qq.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.20

[root@svr7 /]# cat /var/named/163.com.other

163.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.20

[root@svr7 /]# systemctl restart named

为四台虚拟机指定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,以广播进行,先到先得

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

### 一个网络中,只能有一台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.安装软件包dhcp

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

2.修改配置文件

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

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

subnet 192.168.4.0 netmask 255.255.255.0 { #分配网段

range 192.168.4.100 192.168.4.200; #分配IP地址范围

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

option routers 192.168.4.254; #网关地址

default-lease-time 600;

max-lease-time 7200;

}

3.重启服务

[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

### PXE组件及过程分析

• 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(或FTP/NFS),提供yum安装源

• 客户机应具备的条件

– 网卡芯片必须支持PXE协议

– 主板支持从网卡启动

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

### 一 构建DHCP服务,指定tftp服务器地址以及网卡引导文件名称

[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tftp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(网络装机说明书) 二进制文件

安装一个软件即可获得该文件

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

### 二 构建tftp服务,提供众多的引导文件

tftp:简单的文件传输协议 默认端口 69

默认共享路径:/var/lib/tftpboot/

#### 1.安装软件tftp-server

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

[root@svr7 /]# systemctl restart tftp

#### 2.部署pxelinux.0文件

]# yum provides \*/pxelinux.0 #查询那个软件包产生该文件

]# yum -y install syslinux

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

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

]# ls /var/lib/tftpboot/

pxelinux.0-->读取菜单文件

/var/lib/tftpboot/pxelinux.cfg/default

#### 3.部署菜单文件

]# poweroff #关闭机器,图形添加光驱设备,放入光盘文件

]# mount /dev/cdrom /mnt/

]# ls /mnt/

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

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

]# ls /var/lib/tftpboot/

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

#### 4.部署图形模块与背景图片

vesamenu.c32(图形模块)

splash.png(背景图片)

[root@svr7 ~]# cp /mnt/isolinux/vesamenu.c32 /mnt/isolinux/splash.png /var/lib/tftpboot/

[root@svr7 ~]# ls /var/lib/tftpboot/

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

#### 5.部署启动内核与驱动程序

vmlinuz(启动内核)

initrd.img(驱动程序)

[root@svr7 ~]# cp /mnt/isolinux/vmlinuz /mnt/isolinux/initrd.img /var/lib/tftpboot/

[root@svr7 ~]# ls /var/lib/tftpboot/

initrd.img pxelinux.cfg vesamenu.c32

pxelinux.0 splash.png vmlinuz

#### 6.修改菜单文件

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

末行模式 开启行号 :set nu

1 default vesamenu.c32 #默认加载图形的模块

2 timeout 600 #读秒时间 1/10秒

10 menu background splash.png #背景图片

11 menu title NSD1906 PXE Server #标题

61 label linux

62 menu label ^Install CentOS 7 #菜单的显示内容

63 menu default #读秒结束后默认选择

64 kernel vmlinuz #加载启动内核

65 append initrd=initrd.img #加载驱动程序

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

#### 总结:

1.DHCP---> IP地址 next-server filename

2.tftp---> pxelinux.0

3.pxelinux.0---> 访问菜单文件

/var/lib/tftpboot/pxelinux.cfg/default

4.default--->图形的模块 背景图片 启动内核 驱动程序

测试验证:

[root@svr7 ~]# systemctl restart dhcpd

[root@svr7 ~]# systemctl restart tftp

创建一台新的虚拟机进行测试验证

1.安装方式: 网络引导安装(PXE)

2.内存需要: 2048M

3.网络类型: private1

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

### 构建Web服务器,共享众多的软件包

#### 1.安装软件包

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

#### 2.重启httpd服务

[root@svr7 ~]# systemctl restart httpd

#### 3.挂载光驱设备

[root@svr7 ~]# mkdir /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 192.168.4.7/centos

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

### 实现无人值守安装,生成应答文件

#### 1.安装可以生成应答文件的软件包

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

#### 2.运行system-config-kickstart

]# LANG=en system-config-kickstart #英文进行运行

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服务器将ks.cfg进行共享

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

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

#### 4.修改菜单文件,指定应答文件位置

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

label linux

menu label ^Install CentOS 7

menu default

kernel vmlinuz

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

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

## Rsync

客户端:开机启动,引导项

1.本机磁盘 2.光驱设备 3.U盘 4.网卡进行网络安装

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

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 ~]# rsync -a /boot /todir //同步整个文件夹

[root@svr7 ~]# rsync -a /boot/ /todir/ //只同步目录下的数据

[root@svr7 ~]# mkdir /nsd19

[root@svr7 ~]# mkdir /todir

[root@svr7 ~]# cp /etc/passwd /nsd19

[root@svr7 ~]# touch /nsd19/1.txt

[root@svr7 ~]# ls /nsd19/

[root@svr7 ~]# rsync -av /nsd19 /todir/ #同步目录本身

[root@svr7 ~]# ls /todir/

[root@svr7 ~]# rsync -av /nsd19/ /todir/ #同步目录内容

[root@svr7 ~]# ls /todir/

[root@svr7 ~]# echo haha >> /nsd19/1.txt

[root@svr7 ~]# cat /nsd19/1.txt

[root@svr7 ~]# rsync -av /nsd19/ /todir/

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

[root@svr7 ~]# rsync -av --delete /nsd19/ /todir/

[root@svr7 ~]# ls /todir/

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

[root@svr7 ~]# ls /todir/

[root@svr7 ~]# rsync -av --delete /nsd19/ /todir/

[root@svr7 ~]# ls /todir/

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

[root@svr7 ~]# ls /todir/

[root@svr7 ~]# touch /nsd19/abc{1..5}.txt

[root@svr7 ~]# ls /nsd19/

[root@svr7 ~]# rsync -av --delete /nsd19/ /todir/

[root@svr7 ~]# ls /todir/

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

### 远程同步(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的/nsd19/目录与虚拟机B的/opt/进行同步

虚拟机A:

]# rsync -av --delete /nsd19/ root@192.168.4.207:/opt/

虚拟机B:

]# ls /opt

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

实时远程同步

#### 一 虚拟机A通过公钥(id\_rsa.pub) 私钥(id\_rsa) ,实现ssh无密码验证

[root@svr7 ~]# ssh-keygen #一路回车

[root@svr7 ~]# ssh-keygen -f /root/.ssh/id\_rsa -N ''　#使用shell脚本是用的非交互用法

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

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

]# rsync -av --delete /nsd19/ root@192.168.4.207:/opt/

拥有私钥的机器,可以无密码登录拥有公钥的机器

#### 二 监控/nsd19目录变化

• Linux内核的 inotify 机制

– 提供事件响应式的文件系统通知机制

– 安装 inotify-tools 控制工具可调用此机制实现监控

真机上:

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

1.虚拟机A找到该软件

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

[root@svr7 ~]# ls /tools/

inotify-tools-3.13.tar.gz other

2.安装gcc与make

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

3.解包

[root@svr7 ~]# tar -xf /tools/inotify-tools-3.13.tar.gz -C /opt

4.运行configure脚本进行配置检测

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

]# ./configure

5.运行make进行编译

]# make

6,进行make install 进行安装

]# make install

]# ls /usr/local/bin/inotifywait #验证是否有运行程序

/usr/local/bin/inotifywait

• 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

书写同步的脚本rsync.sh

循环:for循环适合写有次数的循环

死循环:while循环适合无限次数的循环

格式:

while [条件]

do

重复的代码

done

虚拟机A:

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

#!/bin/bash

while inotifywait -rqq /nsd19/

do

rsync -a --delete /nsd19/ root@192.168.4.207:/opt/

done

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

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

[1] 8297

[root@svr7 /]# jobs -l

[root@svr7 /]#

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