# 系统&服务管理进阶

# 一. 搭建教学环境

## 1. 真机建立全新两台虚拟机

clone-vm7

## 2. 配置虚拟机A：

a. 利用virsh console管理虚拟机进行配置

virsh conesole A

b. 配置eth0的IP地址。

nmcli connection modify eth0

ipv4.method manual

ipv4.addresses "192.168.4.7/24"

connection.autoconnect yes

c. 配置主机名。

hostnamectl set-hostname svr7.tedu.cn

说明：

-- 真机管理本机虚拟机，前提关闭所有虚拟机图形窗口

-- Ctrl + ] : 退出管理

-- vi是Linux最基本的编辑器

## 3. ssh远程管理

ssh -X [root@192.168.4.7](mailto:root@192.168.4.7)

## 4. 搭建yum仓库

1）服务端

构建Web服务或FTP服务，共享光盘所有内容

镜像文件路径 /var/ftp/centos-1894 iso9660 defaults 0 0

2）客户端

修改yum仓库路径

baseurl=ftp://服务端IP地址/centos-1804

# 二. 目录结构

## 1. 认识（查看）Linux目录层次

man hier

## 2. 常见一级目录

|  |
| --- |
| **/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** |

# 三. 扩展应用

## 1. 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

e.g. :

chmod 700 /Julian

chmod 1755 /Julian

## 2. 历史命令

1）history命令

功能：查看历史命令列表

格式：history [选项]

选项：

-c ：清空历史命令

2）调用曾经执行过的命令

!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

3）调整历史命令的数量

vim /etc/profile

HISTSIZE=1000 #默认记录1000条

## 3. 实用小工具

### 1）du

功能：统计文件的占用空间

格式：du [选项]... [目录或文件]...

选项：

-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)

### 2）bc

功能：交互式计算器

运算符号：

+ - \* / %(取余数运算)

### 3）date

功能：查看/调整系统日期时间

格式：

date +%F

date +%R

date +"%Y-%m-%d %H:%M:%S"

date -s "yyyy-mm-dd HH:MM:SS"

e.g. :

~]# date +%F # 显示年月日

~]# date +%R # 显示时分

~]# date +%Y # 显示年

~]# date +%m # 显示月

~]# date +%d # 显示日期

~]# date +%H # 显示时

~]# date +%M # 显示分

~]# date +%S # 显示秒

### 4）ln

功能：制作快捷方式（连接文件 符号文件）

格式：

ln -s 源文件 生成快捷方式名称 # 制作软连接

ln 源文件 生成快捷方式名称 # 制作硬链接

说明：

-- 若源文件被删除，软连接文件将失效，硬链接文件仍可用

-- 软连接可以存放在不同的分区、文件系统，也可以针对目录

-- 硬连接与源文件必须在同一分区、文件系统，必须是文件

### 5）zip

功能：归档、压缩工具（跨平台压缩工具）

格式：zip [选项] 备份文件.zip 被归档的文档...

选项：

-r 递归选项

### 6）unzip

功能：释放归档、解压

格式：unzip 备份文件.zip [-d 目标文件夹]

# 四. 发布自定义yum仓库

把互联网下载的软件包，采用yum的机制进行管理。

1. 具备从互联网上下载的软件包

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/

e.g. :

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

4. 虚拟机A解包

5. 生成仓库清单(仓库数据文件)

yum –y install createrepo

createrepo /tools/other/

6. 修改配置文件指定新的Yum服务端

[my\_repo]

name=my\_repo

baseurl="file:///tools/other"

enabled=1

gpgcheck=0

# 五. vim编辑技巧

## 1. 命令模式

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 |
| **Ctr l + r** | 取消前一次撤销操作 |
| **ZZ（大写）** | 保存修改并退出 |

## 2. 末行模式

读入其他文件内容

:r /opt/b.txt # 读入/opt/b.txt内容到当前文件

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” |

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

# 六. 源码编译安装

源码包 --> gcc与make --> 可以执行的文件 --> 运行安装

## 1. 步骤

第一步：安装依赖关系包（安装开发工具）

yum -y install gcc make

第二步：tar解包，释放源代码至指定目录

第三步：./configure配置，指安装目录、功能模块等选项

格式：./configure --prefix=指定安装位置

作用：

-- 检测本机是否安装gcc

-- 指定安装目录、功能模块等选项

e.g. :

./configure --prefix=/mnt/myrpm

第四步：make编译，生成可执行的二进制程序文件

make

第五步：make install安装，将编译好的文件复制到安装目录

make install

# 七. KVM构建

## 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）

e.g. : 查看CPU是否支持虚拟化

lscpu | grep vmx

虚拟化主要产商及产品

|  |  |
| --- | --- |
| **VMware** | **VMware Workstation、vSphere** |
| **Microsoft** | VirtualPC、Hyper-V |
| **RedHat** | KVM、RHEV |
| **Citrix** | Xen |

## 2. 手动安装一台带有图形化界面的虚拟机

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

## 3. 针对虚拟机进行配置

1）关闭SELinux

setenforce 0

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

4）主机名：kvm.tedu.cn

hostnamectl set-hostname kvm.tedu.cn

5）构建Yum仓库，使用真机FTP作为服务

## 4. 安装虚拟化服务平台

### 1）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

e.g. : 虚拟化服务

systemctl restart libvirtd

### 2）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

• 删除虚拟机

virsh undefine 虚拟机名

### 3）XML配置文件（虚拟机描述文件）

默认存放路径：

/etc/libvirt/qemu

输出虚拟机xml文件内容：

virsh dumpxml nsd01 | less

e.g. : 虚拟机名称

<name>nsd01</name>

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

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

e.g. : 网卡MAC地址

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

## 5. 一台KVM虚拟机的组成

1）xml配置文件：

定义虚拟机的名称、UUID、CPU、内存、虚拟磁盘、网卡等。

默认路径：/etc/libvirt/qemu/

2）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/

## 6. 手动创建一台虚拟机virtual01

第一步：创建虚拟机磁盘镜像文件

cd /var/lib/libvirt/images/

cp .node\_tedu.qcow2 virtual01.qcow2

第二步：创建虚拟机xml配置文件

virsh dumpxml virtual01 > /etc/libvirt/qemu/virtual01.xml

vim /etc/libvirt/qemu/virtual01.xml

<name>virtual01</name>

# 虚拟机名称

<uuid>

# 删除整行内容，如果删多了按u进行撤销

<source file='/var/lib/libvirt/images/virtual01.qcow2'/>

# 镜像文件路径

<mac address='52:54:00:20:37:74'/>

# 删除整行内容

第三步：导入虚拟机信息

virsh define /etc/libvirt/qemu/virtual01.xml

第四步：测试

virsh list --all

virsh start virtual01

virsh list

## 7. 三合一的命令

导出xml配置文件、修改、导入

1）创一个虚拟机磁盘文件

cd /var/lib/libvirt/images/

cp .node\_tedu.qcow2 virtual02.qcow2

2）三合一命令

virsh enit virtual01

<name>virtual02</name>

# 虚拟机名称

<uuid>

# 删除整行内容，如果删多了按u进行撤销

<source file='/var/lib/libvirt/images/virtual02.qcow2'/>

# 镜像文件路径

<mac address='52:54:00:20:37:74'/>

# 删除整行内容

# 八. COW技术

Copy On Write机制；写时复制

## 1. 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 |

说明：

-- 直接映射原始的数据内容

-- 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

-- 前端盘的大小最好比原始盘大或者相等

## 2. 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

e.g. : 查看磁盘文件信息

qemu-img info test.qcow2

## 3. 修改虚拟机xml文件内容

virsh edit virtual01

......

e.g. : 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/

# 九. DNS服务器基础

## 1. 基本概念

1）作用

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

2）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

说明：

-- 所有的域名都必须以点作为结尾

-- 完全合格的主机名：主机头部 + 域名

## 2. BIND域名服务

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

主要执行程序: /usr/sbin/named

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

地址库文件: /var/named/.. ..

## 3. 构建DNS服务

指定本机DNS服务器配置文件：/etc/resolv.conf

nameserver DNS服务IP地址

### 单区域DNS（虚拟机A）

第一步：安装软件

bind # 域名服务包

bind-chroot # 提供虚拟根支持，牢笼政策

第二步：修改主配置文件

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 "www.tedu.cn"; # 地址库文件名称

};

第四步：建立地址库文件

cd /var/named

cp -p named.localhost tedu.cn.zone

# -p 保证源文件属性不变

vim tedu.cn.zone

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.207

ftp A 192.168.4.7

第五步：重启服务、测试

systemctl restart named

nslookup [www.tedu.cn](http://www.tedu.cn)

nslookup [ftp.tedu.cn](ftp://ftp.tedu.cn)

### 多区域DNS

第一步：修改主配置文件

......

zone "tedu.cn" IN {

type master;

file "tedu.cn.zone";

};

zone "qq.com" IN {

type master;

file "qq.com.zone";

};

第二步：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

第三步：重启服务、测试

### 虚拟机B

options {

directory "/var/named";

};

zone "bj.qq.com" IN {

type master;

file "bj.qq.com.zone";

};

cd /var/named/

cp -p named.localhost bj.qq.com.zone

vim bj.qq.com.zone

bj.qq.com. NS pc207

pc207 A 192.168.4.207

www A 11.12.13.14

systemctl restart named

## 4. 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

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

\* A 1.2.3.4

tedu.cn. A 10.20.30.40

### 3）有规律的泛域名解析

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

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

# ..........

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

$GENERATE 1-50 pc$ A 192.168.10.$

内置函数：$GENERATE

功能：制造连续范围数字

4）解析记录的别名

......

tts cname ftp

e.g. : 主机名映射文件：/etc/hosts

追加一行：192.168.4.110 [www.360.com](http://www.360.com)

ping [www.360.com](http://www.360.com)

## 5. 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

nslookup [www.qq.com](http://www.qq.com) 192.168.4.7

nslookup [www.bj.qq.com](http://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

1）修改虚拟机A；DNS主配置文件

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

2）重启服务

systemctl restart named

3）测试

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

## 6. 递归、迭代

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，最终将解析结果返回给客户端。

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。

## 7. 缓存DNS

作用：缓存DNS服务器,缓存解析记录加快解析

基本环境分配：

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

需求：

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

1）准备虚拟机C

配置IP地址为192.168.4.10/24

2）配置虚拟机B

vim /etc/named.conf

options {

directory "/var/named";

forwarders { 192.168.4.7; }; # 转发给192.168.4.7

};

3）重启、测试

systemctl restart named

nslookup www.tedu.cn 192.168.4.207

## 总结：

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

yum -y install httpd

2. 建立调用配置文件

vim /etc/httpd/conf.d/virtual.conf

<VirtualHost \*:80>

ServerName www.tedu.cn

DocumentRoot /var/www/tedu

</VirtualHost>

<VirtualHost \*:80>

ServerName www.qq.com

DocumentRoot /var/www/qq

</VirtualHost>

3. 编写测试网页

mkdir /var/www/tedu /var/www/qq

echo '<h1>tedu' > /var/www/tedu/index.html

echo '<h1>QQ' > /var/www/qq/index.html

4. 重启服务

systemctl restart httpd

虚拟机A

1.修改地址库文件记录

vim /var/named/tedu.cn.zone

tedu.cn. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.207

vim /var/named/qq.com.zone

qq.com. NS svr7

svr7 A 192.168.4.7

www A 192.168.4.207

2. 重启服务

systemctl restart named

虚拟机C

1. 指定DNS服务器地址

echo nameserver 192.168.4.7 > /etc/resolv.conf

2. 测试DNS解析

nslookup www.qq.com

nslookup www.tedu.cn

3. 测试访问Web页面

firefox www.qq.com

firefox [www.tedu.cn](http://www.tedu.cn)

# 十. Split分离解析

## 1. 什么是分离解析（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

## 2. 分离解析配置

第一步：修改配置文件

opteions {

directory "/var/named";

};

view "nsd" {

match-clients { 192.168.4.7; };

zone "tedu.cn" IN {

type master;

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

};

zone "qq.cn" IN {

type master;

file "qq.cn.zone";

};

};

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

};

zone "qq.cn" IN {

type master;

file "qq.cn.other";

};

};

第二步：建立不同的地址库文件

## 3. acl地址列表

acl test { 192.168.4.207; 192.168.4.1; .... };

view "nsd" {

match-clients { test; };

zone "tedu.cn" IN {

type master;

firle "tedu.cn.zone";

};

};

## 4. 基本规则

-- 客户端匹配原则：由上到下，匹配即停止

-- 每一个客户端都必须找到自己的分类

-- 所有的zone都必须在view中

# 十一. RAID磁盘阵列

## 1. 概述

廉价冗余磁盘阵列

Redundant Arrays of Inexpensive Disks

1）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

2）阵列的价值:提升I/O效率、硬件级别的数据冗余

3）不同RAID级别的功能、特性各不相同

## 2. RAID磁盘类型

• RAID 0,条带模式

-- 至少由两块磁盘组成

-- 同一个文档分散存放在不同磁盘

-- 并行写入以提高效率

• RAID 1,镜像模式

-- 至少由两块磁盘组成

-- 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

• RAID 5,高性价比模式

-- 至少由三块磁盘组成

-- 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

• RAID 6,高性价比/可靠模式

-- 至少由四块磁盘组成

-- 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

• RAID 0+1 、RAID 1+0

-- 至少由四块磁盘组成

-- 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

## 3. RAID磁盘各级别特点对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **对比项** | **RAID 0** | **RAID 1** | **RAID 10** | **RAID 5** | **RAID6** |
| **磁盘数** | ≥2 | ≥2 | ≥4 | ≥3 | ≥4 |
| **存储利用率** | 100% | ≤50% | ≤50% | n-1/n | n-2/n |
| **校验盘** | 无 | 无 | 无 | 1 | 2 |
| **容错性** | 无 | 有 | 有 | 有 | 有 |
| **IO性能** | 高 | 低 | 中 | 较高 | 较高 |

## 4. RAID整列实现方式

硬RAID : 由RAID控制卡管理阵列

主板 --> 阵列卡 --> 磁盘 --> 操作系统 --> 数据

软RAID : 由操作系统来管理阵列

主板 --> 磁盘 --> 操作系统 --> RAID软件 --> 数据

# 十二. 进程管理

## 1. 查看进程

### 1）pstree -- Processes Tree

功能：查看进程树

格式：pstree [选项] [PID或者用户名]

选项：

-a 显示完整的命令行

-p 列出对应PID编号

e.g. : 列出正在运行的所有进程、信息非常全面

ps aux

e.g. : 列出正在运行的所有进程、可以显示该进程的父进程PID

ps -elf

e.g. : 计算当前系统中运行的进程个数

ps aux | wc -l

USER PID %CPU %MEM VSZ RSS

用户 进程ID %CPU %内存 虚拟内存 固定内存

TTY STAT START TIME COMMAND

终端 状态 起始时间 CPU时间 程序指令

### 2）top命令

功能：进程动态排名；交互工具

格式：top [-d 刷新秒数] [-U 用户名]

交互选项：

P 进行CPU排序

M 进行内存排序

### 3）pgrep -- Process Grep

功能：检索进程

格式：pgrep [选项]... 查询条件

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

## 2. 控制进程

### 1）jobs命令

功能：查看后台任务列表

### 2）fg命令

功能：将后台任务恢复到前台运行

### 3）bg命令

功能：激活后台被挂起的任务

### 4）杀死进程的不同方法

Ctrlt + c 组合键，终端当前命令程序

kill [-9] PID... 、kill [-9] %后台任务编号

killall [-9] 进程名...

pkill 查找条件

e.g. ：强制踢出一个用户（杀死一个用户开启的所有进程）

killall -9 -u lisi

### 5）进程的后台启动

-- 在命令行末尾添加“&”符号，不占用当前终端

-- Ctrlt + z组合键；挂起当前进程（暂停并转入后台）

# 十三. 日志管理

## 1.日志概述

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

## 2. 常见的日志文件

|  |  |
| --- | --- |
| **日志文件** | **主要用途** |
| **/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 |

用户日志

-- 由登录程序负责记录/管理

-- 日志消息采用二进制格式

-- 记录登录用户的时间、来源、执行的命令等信息

|  |  |
| --- | --- |
| **日志文件** | **主要用途** |
| **/var/log/lastlog** | 记录最近的用户登录事件 |
| **/var/log/wtmp** | 记录成功的用户登录/注销事件 |
| **log/btmp** | 记录失败的用户登录事件 |
| **/var/run/utmp** | 记录当前登录的每个用户的相关信息 |

## 3. 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

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

## 4. 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 |

## 5. journalctl工具

功能：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

常用格式：

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"

# 十四. systemctl控制

## 1. 作用

-- 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

## 2. 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

## 3. 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

## 4. 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

## 5. RHEL7 运行模式

字符模式：multi-user.target

图形模式：graphical.target

e.g. ：当前直接切换到字符模式

systemctl isolate multi-user.target

e.g. ：当前直接切换到图形模式

systemctl isolate graphical.target

e.g. ：查看每次开机默认进入模式

systemctl get-default

e.g. ：设置永久策略，每次开机自动进入graphical.target

systemctl set-default multi-user.target

# 十五. 批量装机环境

## 1. 部署DHCP服务器

Dynamic Host Configuration Protocol

概念：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

作用：

-- 主要分配IP地址/子网掩码/广播地址

-- 默认网关地址、DNS服务 器地址

-- PXE引I导设置( TFTP服务器地址、引导文件名)

原理：

DHCP地址分配的四次会话

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

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

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

## 2. 基础条件

装机条件准备

1）准备CentOS7安装源( HTTP方式YUM库)

2）启用DHCP服务

PXE引导配置

3）启用TFTP服务,提供装机用的内核、初始化文件

4.）提供PXE引导程序、配置启动菜单

## 3. 什么是PXE

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：

1）PXE client集成在网卡的启动芯片中

2）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

## 4. 网络装机

### 1）配置DHCP服务器

第一步：安装dhcp软件包

第二步：修改配置文件

vim /etc/dhcp/dhcp.conf

末行模式：

：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;

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

}

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

第三步：重启dhcp服务

systemctl restart dhcpd

### 2）构建TFTP服务

简单的文本传输协议

默认端口：69

默认共享位置：/var/lib/tftpboot

第一步：安装软件包tftp-server

第二步：重启服务

systemctl restart tftp

### 3）部署pxelinux.0引导文件

e.g. ：查询那个包产生的该文件

yum provides \*/pxelinux.0

e.g. ：查询软件包安装清单

rpm –ql syslinux | grep pxelinux.0

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

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

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

第一步：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

mount /dev/cdrom /mnt/

mkdir /var/lib/tftpboot/pxelinux.cfg

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

第二步：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

第三步：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

第四步：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

第五步：重启服务

systemctl restart dhcpd

systemctl restart tftp

第六步：初步测试

新建一台全新的虚拟机

-- 安装方式选择：PXE网络引导安装

-- 内存必须2G

-- 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

### 4）构建web服务

将光盘内容共享给客户端

第一步：安装软件包http

第二步：将镜像文件挂载到web

mkdir /var/www/html/centos

mount /dev/cdrom /var/www/html/centos

systemctl restart httpd

第三步：测试

firefox <http://192.168.4.7/centos>

### 5）部署无人值守安装

第一步：安装system-config-kickstart图形化软件

yum -y install system-config-kickstart

第二步：运行图形化工具

e.g. ：以英文语言进行运行

LANG=en system-config-kickstart

说明：

-- 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]

第三步：生成应答文件

ls /root/ks.cfg

第四步：利用Web服务器共享应答文件

cp /root/ks.cfg /var/www/html/

firefox http://192.168.4.7/ks.cfg

第五步：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 ... 指定获取软件包方式

获取软件包方式 --> url=<http://192.168.4.7/centos>

# 十六. 同步操作

## 1. rsync同步操作

作用：Remote Sync远程同步

格式：rsync [选项...] 源目录 目标目录

选项：

-n 测试同步过程,不做实际修改

-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

--delete 删除目标文件夹内多余的文档

说明：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

e.g. ：同步目录本身

rsync -avz /dir01 /abc/

e.g. ：同步目录内容

rsync -avz /dir01/ /abc/

## 2. rsync + SSH同步

作用：与远程ssh目录保持同步

格式：

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

e.g. ：将文件dir01下内容，同步到IP为192.168.4.207主机，opt目录下

rsync -avz --delete /dir01/ [root@](mailto:root@192.168.4.207:/opt/)[192.168.4.207:/opt/](mailto:root@192.168.4.207:/opt/)

## 3. inotifywait监控

格式：inotifywait [选项] 目标文件夹

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

## 4. 实时同步

第一步：生成公钥与私钥,实现ssh无密码验证

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

ssh-copy-id [root@192.168.4.207](mailto:root@192.168.4.207) # 将公钥传给192.168.4.207

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

第二步：监控目录内容的变化

1）安装inotify-tools工具

让虚拟机A具备inotify-tools-3.13.tar.gz

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

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

2）编写同步脚本

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

while inotifywait -rqq /dir01

do

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

done

e.g. ：执行脚本

chmod +x /opt/rsync.sh

/opt/rsync.sh

# 十七. Cobbler装机平台

## 1. 基本概念

a. Cobbler是一款快速的网络系统部署工具

b. 集中管理所需服务，如DHCP、DNS、TFTP、Web

c. Cobbler内部集成了一个镜像版本仓库。

d. Cobbler内部集成了一个ks应答文件仓库。

e. Cobbler还提供了包括yum源管理、Web界面管理、API接口、电源管理等功能。

## 2. Cobbler平台部署

### 1）基本环境

第一步：具备一个CentOS虚拟机

第二步：虚拟机设置

设置防火墙为trusted

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

当前及永久设置SELinux状态为permissive

setenforce 0 # 当前临时关闭

vim /etc/selinux/config

SELINUX=permissive

第三步：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

解压cobbler.zip包

unzip /root/cobbler.zip -d /

[root@svr7 ~]# ls /cobbler/

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

unzip /cobbler/cobbler\_rpm.zip -d /opt/

第四步：安装cobbler主程序、工具包等

yum -y install dhcp httpd mod\_ssl

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

### 2）cobbler网络装机部署

第一步：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

第二步：配置cobbler

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

e.g. ：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

第三步：配置cobbler的dhcp

vim /etc/cobbler/dhcp.template

:%s /192.168.1/192.168.4/g # 全文替换

第四步：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

第五步：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

第六步：同步刷新cobbler配置

cobbler sync # 检测所有配置是否正确

......

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

第七步：测试网页

firefox https://192.168.4.7/cobbler\_web

用户名：cobbler

密码：cobbler

## 3）cobbler应用

第一步：导入安装镜像数据

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

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

第二步：将光盘换成RHEL7

umount /dvd/

mount /dev/cdrom /dvd

安装完成机器默认：root 密码：cobbler

第三步：自定义应答文件：开头注释行删除

system-config-kickstart # 生成ks文件

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

cobbler list

第四步：修改kickstart文件

cobbler profile edit --name=CentOS7 \

--kickstart=/var/lib/cobbler/kickstarts/自定义.cfg

cobbler profile report

cobbler sync # 同步配置

e.g. ：查看有哪些系统

cobbler list

e.g. ：删除

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