

系统&服务管理进阶

NSD SERVICES

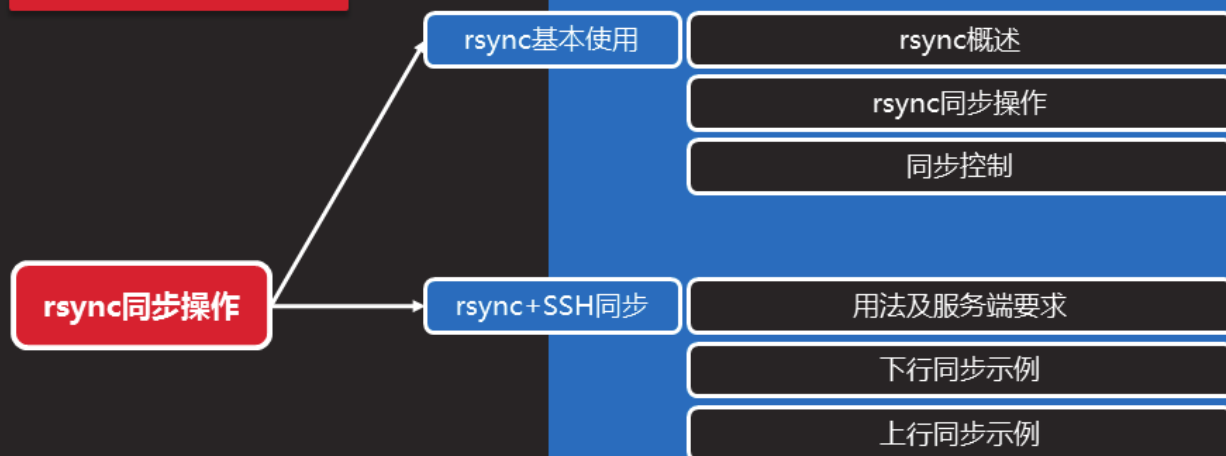
DAY06

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	rsync同步操作
	10:30 ~ 11:20	inotify实时同步
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	Cobbler装机平台
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



rsync同步操作



rsync基本使用

rsync概述

- Remote Sync , 远程同步
 - 支持本地复制 , 或其他SSH、rsync主机同步
 - 官方网站 : <http://rsync.samba.org/>

知识讲解

The logo for rsync, featuring the word "rsync" in a stylized, metallic font. The letters are white with a green-to-purple gradient background behind them.

rsync同步操作

知识讲解

- 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



rsync同步操作（续1）

知识讲解

- 本地同步
 - `rsync [选项...] 本地目录1 本地目录2`
 - `rsync [选项...] 本地目录1/ 本地目录2`
- ```
[root@svr7 ~]# mkdir /todir
[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 ~]# ls /todir/
boot/ grub/ config-2.6.18-348.el5
.. ..
```



# 同步控制

## 知识讲解

- 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

-r : 递归，包括目录/子目录及所有文件  
-l : 保留符号链接文件  
-p、-t : 保留文件的权限、时间标记  
-o、-g : 保留文件的属主/属组标记  
-D : 保留设备文件及其他特殊文件



## 案例1：rsync基本用法

使用 rsync 命令完成以下任务

- 1) 将目录 /boot 同步到目录 /todir 下
- 2) 将目录 /boot 下的文档同步到目录 /todir 下
- 3) 在目录 /boot 下新增文件 a.txt，删除 /todir 下的子目录 grub2，再次同步使 /todir 与 /boot 一致
- 4) 验证 -a、-v、-n、--delete 选项的含义

## 课堂练习



# rsync+SSH同步

## 用法及服务端要求

知识讲解

- 列出 SSH 服务端资源
  - `rsync user@host:远程目录/`
- 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`
- 服务端要求
  - 开启sshd服务，并提供授权的用户、密码



## 下行同步示例

- 将远程主机的 /boot/ 目录备份到本地

```
[root@pc207~]# rsync root@192.168.4.7:/boot/
.. .. //浏览远程目录
drwxr-xr-x 1024 2014/08/12 09:44:12 efi
drwxr-xr-x 1024 2014/08/12 09:47:18 grub
```

```
[root@pc207~]# rsync -az root@192.168.4.7:/boot/ /fromssh/
root@192.168.4.7's password: //通过验证后执行同步
```

```
[root@pc207~]# ls -l /fromssh/ //确认同步结果
.. ..
drwxr-xr-x. 3 root root 4096 8月 12 09:44 efi
drwxr-xr-x. 2 root root 4096 8月 12 09:47 grub
```

知识讲解



## 上行同步示例

- 将本地的 /etc 目录备份到远程主机

```
[root@pc207~]# du -sh /etc/ //确认本地目录
39M /etc/
```

```
[root@pc207~]# rsync -az /etc root@192.168.4.7:/opt/
root@192.168.4.7's password: //通过验证后执行同步
```

```
[root@svr7 ~]# du -sh /opt/etc/ //在服务器上确认结果
39M /opt/etc/
```

知识讲解



## 案例2：rsync+SSH同步

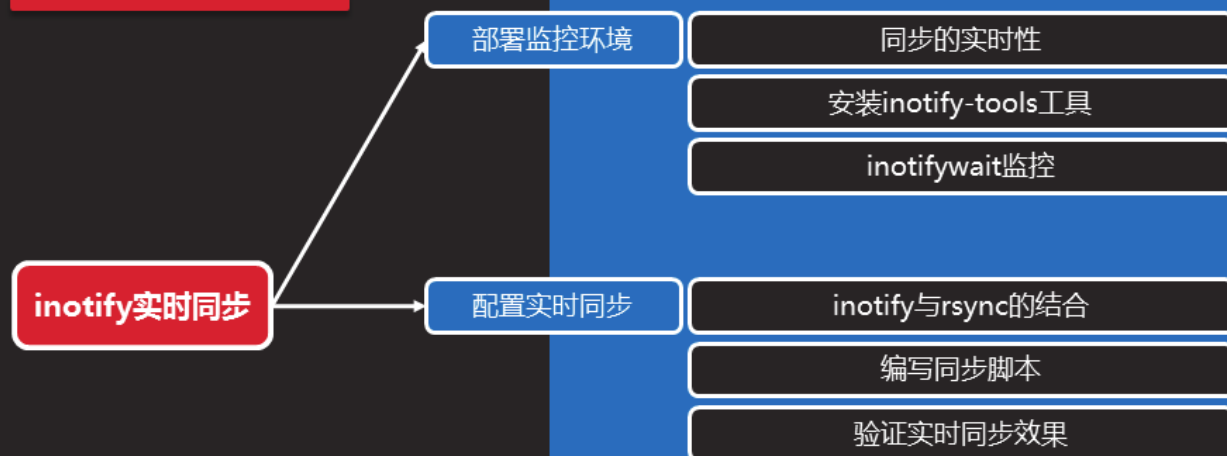
使用 rsync 访问SSH远程主机 svr7

- 1) 查看远程主机的 / 目录下有哪些子目录
- 2) 从远程主机下载 /etc/passwd 文件到当前目录
- 3) 将远程主机的 /boot/ 目录同步为本地的 /fromssh
- 4) 将本机的 /etc 目录同步到远程主机的 /opt/下

课堂练习



### inotify实时同步





# 部署监控环境

## 同步的实时性

- 按照固定周期定期同步
  - 时间间隔不好固定，同步不及时或资源浪费
  - 实时性较差
- 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



# 安装inotify-tools工具

知识讲解

- 参考下载地址：
  - <http://download.sf.net/inotify-tools/>
  - 标准的源码、编译安装即可

```
[root@svr7 ~]# tar xf inotify-tools-3.13.tar.gz
[root@svr7 ~]# cd inotify-tools-3.13
[root@svr7 inotify-tools-3.13]# ./configure
.. ..
[root@svr7 ~]# make && make install
```



# inotifywait监控

知识讲解

- 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



## inotifywait监控 (续1)

知识讲解

- 持续跟踪指定文件夹的变化
  - 目标文件夹： /opt
  - 当文档出现监视的事件时，会立即给出相应提示

```
[root@svr7 ~]# inotifywait -mrq -e modify,create,delete /opt
.. ..
/opt/ CREATE,ISDIR tdir1
/opt/ CREATE file1.txt
/opt/ MODIFY file1.txt
/opt/ DELETE file1.txt
.. ..
```



## 案例3：使用inotifywait工具

针对文件夹 /opt 启用 inotifywait 监控

- 1) 当此目录下出现删除、改名、新建、修改文件、更改权限等事件时能给出提示
- 2) 验证上述监控事件的效果

课堂练习



# 配置实时同步

## inotify与rsync的结合

- 基本思路
  - 利用while循环来反复检查单次监控结果

while inotifywait监控操作

do

需要执行的rsync同步操作

done



## 编写同步脚本

知识讲解

- 实现从本地到远程的Web目录推送

- 源与目标：/var/www/html

```
[root@svr7 ~]# vim /root/isync.sh
#!/bin/bash
FROM_DIR="/var/www/html/"
RSYNC_CMD="rsync -az --delete $FROM_DIR
root@192.168.4.207:/var/www/html"
while inotifywait -rqq -e modify,move,create,delete,attrib $FROM_DIR
do
 $RSYNC_CMD
done &
[root@svr7 ~]# chmod +x /root/isync.sh
```



## 验证实时同步效果

知识讲解

- 部署环境并测试
  - 1) 为从 svr7 上 SSH 远程到 pc207 实现免密码登录  
`ssh-keygen`、`ssh-copy-id root@远程机地址`
  - 2) 为 svr7 和 pc207 准备目录/var/www/html/
  - 3) 运行同步脚本 /root/isync.sh
  - 4) 修改 svr7 上 /var/www/html/ 目录下的内容
  - 5) 观察 pc207 上对应目录的实时变化



## 案例4：配置Web镜像同步

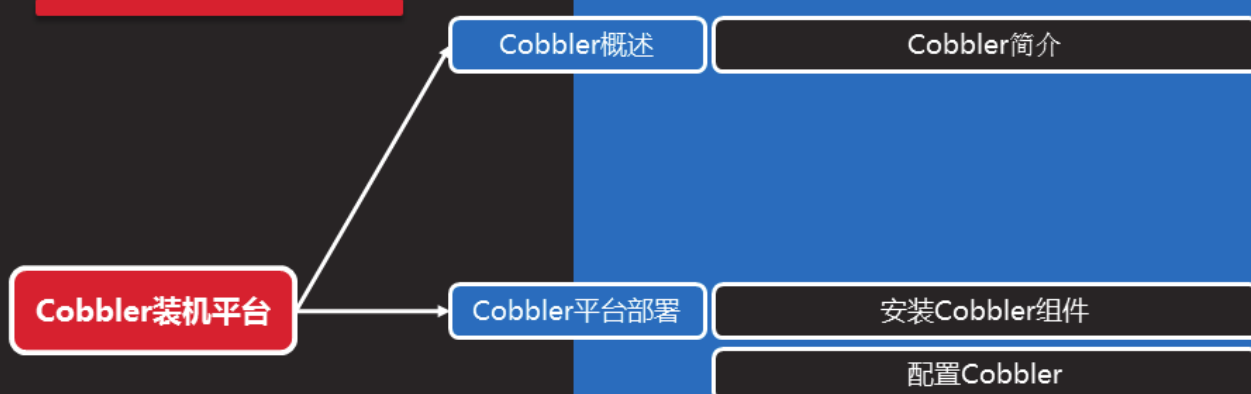
实现 svr7 --> pc207 的网站镜像同步

- 1) 以 svr7 为发起方，原始目录为 /var/www/html/
- 2) 以 pc207 为同步目标，基于SSH免密验证
- 3) 编写 inotify+rsync 同步脚本，验证实时同步效果

课堂练习



### Cobbler装机平台



# Cobbler概述

## Cobbler简介

- 基本概念
  - Cobbler是一款快速的网络系统部署工具，
  - 集中管理所需服务，如DHCP、DNS、TFTP、Web
  - Cobbler内部集成了一个镜像版本仓库
  - Cobbler内部集成了一个ks应答文件仓库
  - 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



# Cobbler简介 (续1)

知识讲解

- 基本概念
  - DHCP、TFTP、PXE
  - pxelinux.0、vmlinuz、initrd.img
- PXE引导配置
  - /TFTP目录/pxelinux.cfg/default
  - 背景图片、多系统支持
- kickstart自动应答
  - 软件源指定、 %POST安装后脚本
  - system-config-kickstart工具



## Cobbler平台部署



## 安装Cobbler组件

- 安装软件包

```
[root@svr7 ~]# yum -y install cobbler cobbler-web dhcp tftp-server
pykickstart httpd xinetd
```

知识讲解

- 开启所需服务

```
[root@svr7 ~]# systemctl start xinetd.service
[root@svr7 ~]# systemctl enable xinetd.service
[root@svr7 ~]# systemctl start httpd
[root@svr7 ~]# systemctl enable httpd
[root@svr7 ~]# systemctl start cobblerd.service
[root@svr7 ~]# systemctl enable cobblerd.service
```



## 配置Cobbler

- 修改配置文件 `/etc/cobbler/settings`
- 部署cobbler相关的包
- 配置DHCP服务
- 导入系统镜像
- kickstart文件生成

知识讲解



## 总结和答疑

总结和答疑

rsync+inotify同步

问题现象

故障分析及排除

**Tedu.cn**  
达内教育

# rsync+inotify同步

## 问题现象

知识讲解

- 实时同步无法实现
  - 问题1：编译安装 inotify-tools 时，报错：... no acceptable C compiler found in \$PATH
  - 问题2：测试 isync.sh 脚本时，多个 rsync 进程相互干扰，影响同步效果



## 故障分析及排除

知识讲解

- 原因分析
  - 问题1：没有可用的C语言编译器，缺少开发工具
  - 问题2：直接杀脚本同名进程，可以终止对应的子进程
- 解决办法
  - 问题1：yum -y install gcc gcc-c++ make
  - 问题2：pkill isync.sh



