

Rsync 服务器搭建

write by Kevinzou

kissingwolf@gmail.com

版权声明：

本文遵循“署名-非商业性使用-相同方式共享 2.5 中国大陆”协议

您可以自由复制、发行、展览、表演、放映、广播或通过信息网络传播本作品

您可以根据本作品演绎自己的作品

您必须按照作者或者许可人指定的方式对作品进行署名。

您不得将本作品用于商业目的。

如果您改变、转换本作品或者以本作品为基础进行创作，您只能采用与本协议相同的许可协议发布基于本作品的演绎作品。

对任何再使用或者发行，您都必须向他人清楚地展示本作品使用的许可协议条款。

如果得到著作权人的许可，您可以不受任何这些条件的限制。

Kevinzou (kissingwolf@gmail.com)

本文目录

第1节 什么是 rsync	1
第2节 安装设置 rsync.....	3
第2.1节安装 rsync RPM 包.....	3
第2.2节在设置 Rsync 服务端.....	3
第2.3节在 Rsync 客户端测试.....	7
第3节 课后练习.....	12

第1节 什么是 rsync

rsync 是一个 Unix 系统下的文件同步和传输工具。它本来是用以取代 rcp 的工具，随着它的功能的不断完善，渐渐的 rsync 成为独树一帜的备份同步工具，目前它是由 rsync.samba.org 团队维护的。Rsync 使用所谓的 "rsync 算法"，提供一个非常快速的档案传输，使本地和远端二部主机之间的文件达到同步，它主要是传送二个文件的异动部份，而非每次都整份传送，因此速度相当地快。

它具备以下特性：

1. 能更新整个目录和树和文件系统
2. 有选择性的保持符号链链、硬链接、文件属于、权限、设备以及时间等
3. 对于安装来说，无任何特殊权限要求
4. 对于多个文件来说，内部流水线减少文件等待的延时
5. 能用 rsh、ssh 或直接端口做为传输入端口
6. 支持匿名 rsync 同步文件，是理想的镜像工具；

rsync 服务器架设比较简单，可能我们安装好 rsync 后，并没有发现配置文件，以及 rsync 服务器启动程序，因为每个管理员可能对 rsync 用途不一样，所以一般的发行版只是安装好软件就完事了，让管理员来根据自己的用途和方向来自己架设 rsync 服务器；因为这个 rsync 应用比较广，能在同一台主机进行备份工作，还能在不同主机之间进行工作。在不同主机之间的进行备份，是必须架设 rsync 服务器。

对于重量级服务器来说，应该有网络备份服务器来说，只有本地备份还是不够的，必须异地备份。只有这样数据的安全才有保证。毕竟数据放在服务器本地上还是不太安全，比如磁盘坏掉、被骇客攻击删除数据，甚至 911 事件。其

实服务器本身价值并不大,重要的是数据的价值。另外对于大量文件从一台服务器上迁移到另一台服务器上,rsync 也是一个不错的选择。

rsync 有很多不同的用法,我们在此文档中只介绍 rsync 的 daemon 模式。

首先介绍两个概念:Rsync 服务器和 Rsync 客户端。Rsync 服务器是指运行 rsync 服务 daemon 的服务器,而 Rsync 客户端实发起 rsync 连接的服务器。这两个名称我们后面的试验中会用到。

Rsync 服务器会使用 xinetd 服务方式开启 873 端口监听,等待 Rsync 客户端连接。Rsync 客户端发起连接后,Rsync 服务器会检查 Rsync 客户端提交 Rsync 服务器内建的户名和密码是否正确,如果通过认证检测,则开始文件传输,传输的过程是按要求先比对文件的大小、属性、权限、MD5 值等信息,如果两端文件信息不一致,则按要求同步文件的区别块。

第2节 安装设置 *rsync*

第2.1节 安装 *rsync* RPM包

```
#mount 192.168.0.254:/var/ftp/pub /mnt
#cd /mnt/{redhat_file_tree}/{rpm_files_dir}
#rpm -ivh xinetd*.rpm
#rpm -ivh rsync*.rpm
```

根据你系统的不同请选择合适的 rpm 包安装，因为 *rsync daemon* 实基于 *xinetd* 的，所以请首先确认 *xinetd* 被正确的安装。

Rsync 服务端需要打开 *rsync daemon* 和启动 *xinetd* 服务，而 *Rsync* 客户端不需要。

```
#chkconfig xinetd on
#chkconfig rsync on
```

第2.2节 在设置 *Rsync* 服务端

Rsync 服务端需要设置如下四项：

1. 规划建立备份目录
2. 设置开启 *rsync daemon* 的 *xinetd* 服务
(`/etc/xinetd.d/rsync`)
3. 创建和设置 *rsync daemon* 的配置文件(`/etc/rsyncd.conf`)
4. 设置 *rsync* 内建用户和密码

首先确定备份目录，我们使用 `/home` 目录作为备份根目录。

然后我们设置 *rsync daemon* 的 *xinetd* 服务，因为之前我们使用

chkconfig 命令已经打开了 xinetd 和 rsync 的启动项，所以这里我们只是验证服务配置文件是否正确配置。

```
#vi /etc/xinetd.d/rsync
```

其文件内容如下：

```
# default: off
# description: The rsync server is a good addition
to an ftp server, as it \
#         allows crc checksumming etc.
service rsync
{
    disable = no

    socket_type      = stream

    wait            = no

    user            = root

    server          = /usr/bin/rsync

    server_args     = --daemon

    log_on_failure += USERID
}
```

接下来，我们创建 rsync 服务配置文件。

首先是 rsync 全局配置文件 rsyncd.conf

```
#vi /etc/rsyncd.conf
```


其文件内容如下：

```
uid = nobody
gid = nobody
use chroot = no
max connections = 4
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsync.lock
log file = /var/log/rsyncd.log
motd file = /etc/rsyncd.motd
hosts allow = 127.0.0.1 192.168.0.0/255.255.255.0
auth users = BackupUser
secrets file = /etc/rsyncd.secrets
[home]
path = /home
comment = ServerHome
read only = true
list = false
```

uid 和 gid 是系统内建用户，必须存在于系统 /etc/passwd 文件中。这里设置的了最小权限，如果 /home 目录下的文件和目录 nobody 用户和组没有读取权限和进入权限，rsync 就会报错并不操作权限不够的文件或目录。

如果使用 chroot，在传输文件之前，服务器守护程序在将 chroot 到文件系统中的目录中，这样做的好处是可能保护系统被安装漏洞侵袭的可能。缺点是需要超级用户 root 权限。另外对符号链接文件，将会排除在外。也就是说，你在 rsync 服务器上，如果有符号链接，你在备份服务器上运行客户端的同步

数据时, 只会把符号链接名同步下来, 并不会同步符号链接的内容。所以我们在这里不使用 `chroot`。

`max connections` 是客户端同时连接数, 为了限制多线程同步的。

`motd file` 是定义服务器信息的, 要自己写 `rsyncd.motd` 文件内容。

`auth users` 是没必要实服务器上存在的真实的系统用户, 它们是 `rsync` 服务端内建的用户, 如果你想用多个用户, 那就以“, ”号隔开, 比如

```
auth users = kevin , mandy
```

`secrets file` 定义密码文件: `/etc/rsyncd.secrets`, 后面我们会创建这个文件。

`[home]` 是设置备份目录区域, 配置项到下一个 `[]` 组合截止。 `path` 定义其在系统中的真实路径, `comment` 定义显示说明, `read only` 定义访问方式, `list` 定义了 `Rsync` 客户端是否可以列出目录结构。

设置好 `rsyncd.conf` 文件后, 我们要创建另外两个 `rsync` 需要的配置文件, 它们是 `rsyncd.motd` 和 `rsyncd.secrets`。

```
#cat >/etc/rsyncd.motd <<END
This is a rsyncserver for stuXX
END
```

将 `stuXX` 换为你的设备名, 以和其他人的机器作以区分。

```
#cat > /etc/rsyncd.secrets <<END
BackupUser:bfa994f3dab8ec2eae10362d592b9725
END
```

`secrets` 文件的格式是:

内建用户名：密码

某些系统不支持长密码，另外这个密码不适加密存放的，之所以你看起来长，是因为这个密码字符串是一个文件的 MD5 值。

创建好这些文件后需要设置其正确权限，否则 `rsync daemon` 实不会正常工作的。

```
#chmod 600 /etc/rsyncd.conf
#chmod 600 /etc/rsyncd.motd
#chmod 600 /etc/rsyncd.secrets
```

然后重启 `xinetd` 服务，`rsync` 服务器端就开始工作了。

```
#service xinetd restart
```

第2.3节 在 Rsync 客户端测试

首先创建密码文件 `secrets`，放在 `/root` 目录下

```
#cat > /root/secrets <<END
bfa994f3dab8ec2eae10362d592b9725
END
```

密码文件中填入 `Rsync` 服务端定义的用户密码，用户名不在这里出现。

使用如下命令测试 `Rsync` 客户端是否可以同步 `Rsync` 服务端

```
#rsync -Rav --delete --password-file=/root/secrets
BackupUser@192.168.0.xx::home /tmp
```

将 `192.168.0.xx` 换为 `Rsync` 服务端 IP 地址。

`Rsync` 参数参考如下：

-v, --verbose 详细模式输出

-q, --quiet 精简输出模式

-c, --checksum 打开校验开关，强制对文件传输进行校验

-a, --archive 归档模式，表示以递归方式传输文件，并保持所有文件属性，等于-rlptgoD

-r, --recursive 对子目录以递归模式处理

-R, --relative 使用相对路径信息

```
rsync foo/bar/foo.c remote:/tmp/
```

则在/tmp 目录下创建 foo.c 文件，而如果使用-R 参数：

```
rsync -R foo/bar/foo.c remote:/tmp/
```

则会创建文件/tmp/foo/bar/foo.c，也就是会保持完全路径信息。

-b, --backup 创建备份，也就是对于目的已经存在有同样的文件名时，将老的文件重新命名为~filename。可以使用--suffix 选项来指定不同的备份文件前缀。

--backup-dir 将备份文件(如~filename)存放在在目录下。

-suffix=SUFFIX 定义备份文件前缀

-u, --update 仅仅进行更新，也就是跳过所有已经存在于 DST，并且文件时间晚于要备份的文件。（不覆盖更新的文件）

-l, --links 保留软链结

-L, --copy-links 想对待常规文件一样处理软链结

--copy-unsafe-links 仅仅拷贝指向 SRC 路径目录树以外的链结

--safe-links 忽略指向 SRC 路径目录树以外的链结

-H, --hard-links 保留硬链结

-p, --perms 保持文件权限

-o, --owner 保持文件属主信息

-g, --group 保持文件属组信息

-D, --devices 保持设备文件信息

-t, --times 保持文件时间信息

-S, --sparse 对稀疏文件进行特殊处理以节省 DST 的空间

-n, --dry-run 现实哪些文件将被传输

-W, --whole-file 拷贝文件，不进行增量检测

-x, --one-file-system 不要跨越文件系统边界

-B, --block-size=SIZE 检验算法使用的块尺寸，默认是 700 字节

-e, --rsh=COMMAND 指定替代 rsh 的 shell 程序

`--rsync-path=PATH` 指定远程服务器上的 `rsync` 命令所在路径信息

`-C, --cvs-exclude` 使用和 `CVS` 一样的方法自动忽略文件，用来排除那些不希望传输的文件

`--existing` 仅仅更新那些已经存在于 `DST` 的文件，而不备份那些新创建的文件

`--delete` 删除那些 `DST` 中 `SRC` 没有的文件

`--delete-excluded` 同样删除接收端那些被该选项指定排除的文件

`--delete-after` 传输结束以后再删除

`--ignore-errors` 及时出现 `IO` 错误也进行删除

`--max-delete=NUM` 最多删除 `NUM` 个文件

`--partial` 保留那些因故没有完全传输的文件，以是加快随后的再次传输

`--force` 强制删除目录，即使不为空

`--numeric-ids` 不将数字的用户和组 `ID` 匹配为用户名和组名

`--timeout=TIME` `IP` 超时时间，单位为秒

`-I, --ignore-times` 不跳过那些有同样的时间和长度的文件

`--size-only` 当决定是否要备份文件时，仅仅察看文件大小而不考虑文件时间

`--modify-window=NUM` 决定文件是否时间相同时使用的时间戳窗口，默认为 0

-T --temp-dir=DIR 在 DIR 中创建临时文件

--compare-dest=DIR 同样比较 DIR 中的文件来决定是否需要备份

-P 等同于 --partial

--progress 显示备份过程

-z, --compress 对备份的文件在传输时进行压缩处理

--exclude=PATTERN 指定排除不需要传输的文件模式

--include=PATTERN 指定不排除而需要传输的文件模式

--exclude-from=FILE 排除 FILE 中指定模式的文件

--include-from=FILE 不排除 FILE 指定模式匹配的文件

--version 打印版本信息

--address 绑定到特定的地址

--config=FILE 指定其他的配置文件，不使用默认的 rsyncd.conf 文件

--port=PORT 指定其他的 rsync 服务端口

--blocking-io 对远程 shell 使用阻塞 IO

-stats 给出某些文件的传输状态

--progress 在传输时现实传输过程

--log-format=FORMAT 指定日志文件格式

```
--password-file=FILE 从FILE中得到密码  
--bwlimit=KBPS 限制 I/O 带宽 , KBytes per second  
-h, --help 显示帮助信息
```

第3节 课后练习

请编写脚本保证 Rsync 客户端每 5 分钟同步一次 Rsync 服务端
/var/www/html 目录下的所有文件。