rsync（remote synchronize ）是一个远程文件同步工具，支持多个操作系统，用于在多台服务器之间同步目录和文件。rsync采用了“rsync算法”，这个算法只传输文件的不同部分，而不是每次都全部传输，效率比较高。

rsync有以下特点：

1）支持目录和文件的上传和下载功能；

2）可以镜像保存整个目录树和文件系统；

3）传输效率高，只传输新增和修改过的文件。

# 一、安装软件包

rsync的客户端和服务器软件的安装包都是rsync。

yum -y install rsync

# 二、修改系统配置

## 1、关闭SELINUX

修改/etc/selinux/config文件，把SELINUX参数的值改为disabled。

SELINUX =disabled

重启linux系统或执行 setenforce 0 使修改马上生效。

## 2、开通防火墙端口

rsync缺省的端口是873，您可以修改配置文件中的端口。

1）防火墙开通873端口。

firewall-cmd --zone=public --add-port=873/tcp --permanent

2）重启防火墙。

systemctl restart firewalld

## 3、启用rsyncd服务

Samba的服务名是rsyncd。

1）启动rsyncd服务。

systemctl start rsyncd

2）把rsyncd服务设置为开机自启动。

systemctl enable rsyncd

# 三、配置rsync

接下来我用示例来显示文件同步的配置和使用，需求如下：

1）我只用一台服务器来测试，IP地址是192.168.1.129，既是服务器，也是客户端；

2）服务端的目录是/tmp/docs；

3）我将创建两个客户端用户：client1和client2；

4）客户端client1的目录是/tmp/docs1；

5）客户端client2的目录是/tmp/docs2；

6）客户端client1把/tmp/docs1目录中的文件发送给服务端；

7）客户端client2从服务端下载文件，存放在/tmp/docs2目录中。

rsync的服务器和客户端，这是一个逻辑的概念，并不是物理的，如果您有三个服务器，就可以用三台服务器来测试，原理是一样的。

## 1、创建操作系统用户

操作系统用户可以是普通的用户，也可以是简单的、无需登录的、没有HOME目录的用户，如下：

useradd -M -s /sbin/nologin rsync -g bin # 创建rsync用户，指定组为bin（其它组也行）。

注意，这个用户是在服务器上创建的，不是客户端。

## 2、创建测试目录和文件

执行以下脚本，创建/tmp/docs目录，生成测试和文件，并指定用户和组。

rm -rf /tmp/docs /tmp/docs1 /tmp/docs2 # 删除测试目录。

mkdir /tmp/docs /tmp/docs1 /tmp/docs2 # 创建三个测试目录。

ls /usr > /tmp/docs1/usr.txt # 把ls /usr的结果输出到/tmp/docs1/usr.txt文件。

ls /etc > /tmp/docs1/etc.txt # 把ls /etc的结果输出到/tmp/docs1/etc.txt文件。

chown -R rsync:rsync /tmp/docs # 修改/tmp/docs目录及文件用户和组。

## 3、创建rsnyc登录用户密码文件

在rsync服务器上创建登录用户/密码文件/etc/rsyncd.passwd，用于客户端的身份认证，内容如下：

client1:pwd1

client2:pwd2

以上文件包括了两个用户（用户名/密码分别是client/pwd1和client2/pwd2）。

**把/etc/rsyncd.passwd文件的权限设置为600，如果不这么做，客户端登录会失败。**

chmod 600 /etc/rsync.passwd

## 4、配置rsync服务器参数

rsync服务器的配置文件是/etc/rsyncd.conf，内容如下：

# /etc/rsyncd: configuration file for rsync daemon mode

# See rsyncd.conf man page for more options.

# rsyncd全局参数。

uid = rsync

gid = rsync

port = 873

fake super = yes

use chroot = no

max connections = 200

timeout = 600

ignore errors

read only = false

list = false

auth users = client1,client2

secrets file = /etc/rsync.passwd

log file = /var/log/rsyncd.log

# 同步模块配置。

[docs]

comment = welcome to docs!

path = /tmp/docs

**注意，不要在参数后面加#和说明文字，是非法的。**

全局参数说明：

1）uid = rsync，rsync服务端操作系统的用户，即上面第1点中创建的操作系统用户，您可以创建新的操作系统用户，也可以用现有的用户。

2）gid = rsync，rsync服务端操作系统的用户的组，即uid用户的组。

3）port = 873，用于通信的TCP端口，缺省是873。

4）fake super = yes，rsync服务端操作系统的用户可以不用root。

5）use chroot = no，关闭假根功能。

6）max connections = 200，客户端最大连接数。

7）timeout = 600，超时时间。

8）ignore errors，忽略错误信息。

9）read only = false，是否为只读方式。

10）list = false，不允许查看模块信息。

11）auth users = client1,client2，指定允许登录的客户端认证用户清单，用逗号分隔，必须是/etc/rsync.passwd文件中配置的用户。

12）secrets file = /etc/rsync.passwd，定义rsync客户端用户认证的密码文件。

13）log file = /var/log/rsyncd.log，rsync服务运行日志文件，注意，日志文件日积月累，必须保证有足够的磁盘空间。

同步参数说明：

1）[docs]，模块名称，自定义的名称，不一定要与同步目录相同。

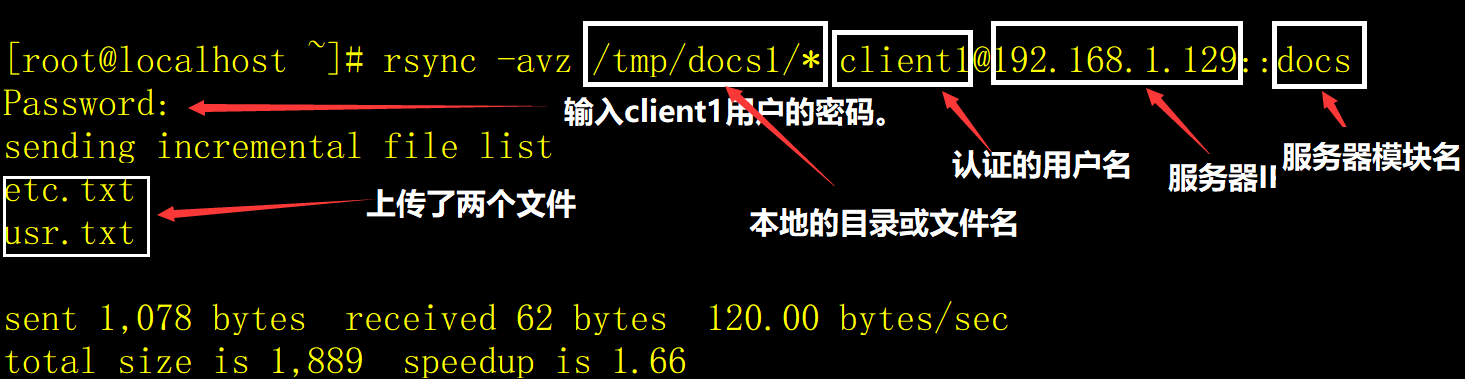
2）comment = welcome to docs!，模块说明文字。

3）path = /tmp/docs，同步的目录名，必须是uid参数指定的用户和gid参数指定的组。

## 5、把客户端的文件同步上传到服务器

1）采用client1用户，把客户端/tmp/docs1目录下的文件同步到服务器，命令如下：

rsync -avz /tmp/docs1/\* client1@192.168.1.129::docs



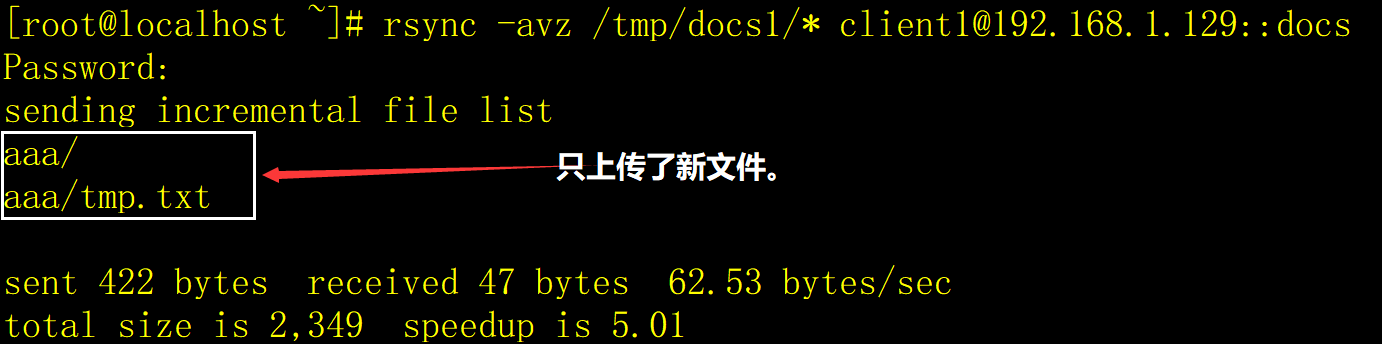
2）检查服务端的/tmp/docs目录和客户端的/tmp/docs目录下的文件是否相同。

3）再生成一些测试文件：创建/tmp/docs1/aaa/tmp.txt文件。

mkdir /tmp/docs1/aaa

ls /tmp > /tmp/docs1/aaa/tmp.txt

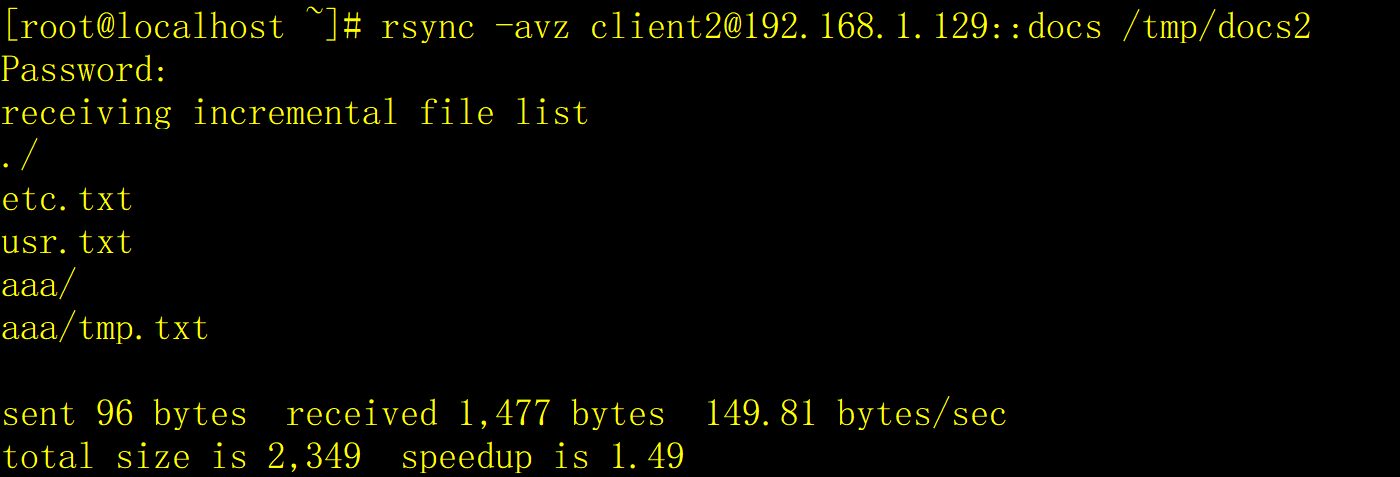
4）再执行一次同步。



## 6、从服务器同步下载文件到客户端

1）采用client2用户，把服务器的文件下载到客户端/tmp/docs2目录命令如下：

rsync -avz client2@192.168.1.129::docs /tmp/docs2



注意，我用client1用户上传文件，用client2下载文件，其目的是为了演示多个客户端帐号的配置和使用方法，您也可以只用一个帐号上传和下载文件。

## 7、客户端的密码配置

以上演示客户端同步文件的时候，需要手工的输入密码，但是在实际应用中，命令可能在后台运行，不希望手工输入密码，这个需求有两种解决方法：

1）设置客户端的密码文件。

例如client1用户，密码文件是/etc/client1.passwd，内容如下：

echo pwd1 > /etc/client1.passwd

chmod 600 /etc/client1.passwd

**注意，客户端的密码文件权限一定要是600，否则认证会失败。**

同步上传的命令如下：

rsync -avz /tmp/docs1/\* client1@192.168.1.129::docs --password-file=/etc/client1.passwd

2）设置客户端的密码环境变量。

export RSYNC\_PASSWORD=pwd1

同步上传的命令如下：

rsync -avz /tmp/docs1/\* client1@192.168.1.129::docs

# 四、应用经验

## 1、小心有坑

rsrync的配置有两个坑：1）配置文件/etc/rsyncd.conf中，参数后面不要用#注释；2）服务端和客户端密码文件的权限一定要是600，否则认证失败。

## 2、客户端权限问题

客户端可以用任何用户来执行，只要该用户对本地目录有足够的权限就可以了。

## 3、日志文件的问题

小心服务端的日志文件（log file）越积越大。

## 4、效率问题

rsync同步文件采用的是增量同步的方法，本质上就是在传输文件之前，先判断客户端与服务器目录的文件变量情况，如果待同步目录下的文件太多，这个判断过程很费时间。

## 5、rsync+sersync架构

上面提到的rsync存在效率问题，最终的解决方法是采用rsync+sersync架构。

1）sersync可以记录被监听目录中发生变化的（增，删，改）具体某个文件或目录的名字；

2）rsync在同步时，只同步发生变化的文件或目录（每次发生变化的数据相对整个同步目录数据来说很小，rsync在遍历查找对比文件时，速度很快），因此效率很高。

rsync+sersync架构在本文就不介绍了，各位真的有需求时再研究它。