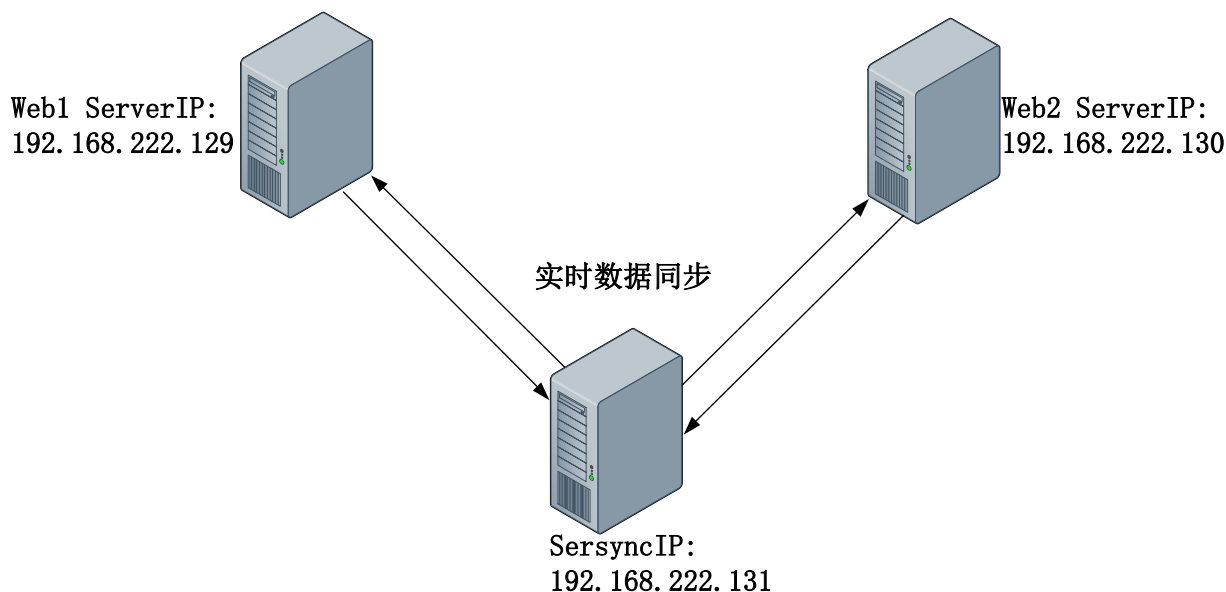


服务器和存储机文件实时同步方案——rsync+sersync

服务器和存储机的规划拓扑图如下：（可不再同一个网段）



注：在 Web1 和 Web2 上的配置是相同的，故以下在 Web1 上进行手动配置，Web2 上脚本自动化配置。

可配置本地 yum 源或者网络 yum 源，其中网络 yum 源如下：

```
cat /etc/yum.repos.d/alyun.repo
[base]
name=CentOS-$releasever - Base - mirrors.aliyun.com
failovermethod=priority
baseurl=http://mirrors.aliyun.com/centos/6/os/$basearch/
enabled=1
gpgcheck=0
gpgkey=http://mirrors.aliyun.com/centos/RPM-GPG-KEY-CentOS-6
```

在 Web1 的 rsync 服务端上的配置：

1、mkdir /data/{web,download}

2、修改 httpd 根目录为 /data/web, 配置文件 (/etc/httpd/conf/httpd.conf) ——>>> DocumentRoot “/data/web”

3、安装并建立 rsync 的配置文件如下：

```
yum install rsync -y
cat /etc/rsyncd.conf
#Global Settings##定义全局量，如果子块没有定义则会继承全局量
uid = root                #定义连接用户
gid = root                #定义连接的用户组
use chroot = no           #安全操作
max connections = 2000     #定义连接数
timeout = 600             #定义 600 秒超时
pid file = /var/run/rsyncd.pid #定义进程文件的位置
log file = /var/log/rsyncd.log  #定义日志文件的位置
lock file = /var/run/rsync.lock #定义锁文件的位置
ignore errors             #忽略错误
strict modes = yes        #是否在严格模式下工作
list = false              #是否允许用户列出文件列表
read only = no            #客户端是否可以推送
write only = no           #只允许写数据不允许拉取数据
hosts allow = 192.168.222.0/24 #定义允许哪些主机可以访问
```

4、提供 rsync 的认证用户密码文件:

5、设置开机自启动: `chkconfig rsync on`

（注：rsync 服务是由超级守护进程 xinetd 管理，因此不能使用 `service rsyncd start`，在开启该服务之前，为保证可以连接服务端，最好关闭防火墙或者开放 873 端口以及关闭 SELinux）

注：以上就是 Web1 的全部配置过程，Web2 同上。rsync 的重启过程是：`killall rsync` 和 `rsync --daemon`
在 Sersync 上配置 rsync 客户端：

2、测试在客户端是否可以实现数据同步;

3、检测 Web1 的 /data/web 目录下是否有 index.html 文件，如果以上的测试成功则可以进行一下的 sersync 服务的安装：

```
cp conf/confxml.xml.bak conf/confxml.xml
```

</localpath>

```
#####
修改 31~34 行，修改如下：
#####
<commonParamsparams="-aruz"/>
```

```
    <auth start="true"users="rsync_bak"passwordfile="/etc/rsync.password"/> #验证文件位置
#####
```

注：以上配置文件，只仅仅针对了 Web1 和 Web2 的 /data/web 目录下的文件同步，要实现 /data/download 目录的文件同步，还需要根据 confxml.xml 的文件重新建立一个文件 confxml_download.xml 并且仿照 confxml.xml 文件修改的位置修改对应地方的信息即可。

5、开启 sersync 守护进程进行数据的同步：

```
echo "export PATH=$PATH:/usr/local/sersync/bin" >> /etc/profile
source /etc/profile
which sersync          #检测系统是否有该项命令
sersync -r -d -o /usr/local/sersync/conf/confxml.xml          #启动/data/web 目录下的自动同步进程
sersync -r -d -o /usr/local/sersync/confxml_download.xml
ps -ef | grep sersync   #开启后查看 sersync 的进程信息
cat >> /etc/rc.d/rc.local << EOF
#sync data to 192.168.222.129 192.168.222.130
sersync -d -o -r /usr/local/sersync/conf/confxml.xml
sersync -r -d -o /usr/local/sersync/confxml_download.xml
EOF
```

6、测试 sersync 服务是否正常：

```
for i in {1..10000};do echo "$i-sync" >/data/web/$i.html &> /dev/null;done
#在/data/web 目录下创建 10000 个文件
ps -ef | grep rsync|wc -l
#统计 rsync 进程个数，经测试该数据是变动的
```

至此，在上述拓扑框架下的实时文件同步搭建完成，以下是关于 sersync 命令的选项说明：

- r #在开启实时监控之前对主服务器和远程目标机之间进行一次完整的数据同步，将本地和远程整体同步
- o #没有-o 程序会默认寻找 confxml.xml 配置文件，加了之后则会选择指定的配置文件，从而实现多进程多线程的数据同步。
- n #指定默认的线程池的线程总数
- d #后台启动服务，在第一次执行时应该加上该参数

实例：sersync -n 16 -o config.xml -r -d #表示设置线程池工作线程为 16 个，指定 sun.xml 作为配置文件，子实时监控前做 一次整体同步，以守护进程方式在后台运行。