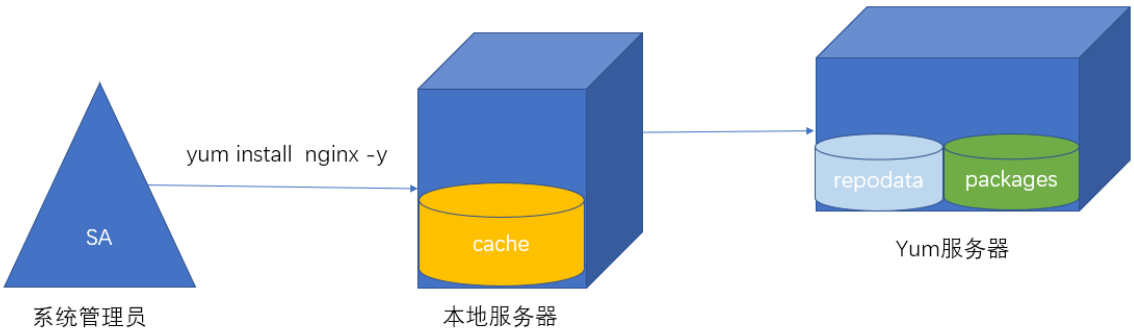
1. YUM管理：

yum命令是在Fedora和RedHat以及SUSE中基于rpm的软件包管理器，它可以使系统管理人员交互和自动化地更细与管理RPM软件包，能够从指定的服务器自动下载RPM包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软体包，无须繁琐地一次次下载、安装。

1.1 yum工作原理：

当我们执行 `yum install nginx -y`，yum会先访问本地缓存，如果有直接安装，如果没有，则通过元数据找到该软件包，通过该软件内部数据库的提示，找到相应的依赖包，然后继续查找元数据中是否有这些依赖包，如果没有会提示依赖包没有镜像提供。如果nginx软件包和依赖包都找到了，就根据配置文件中的url去下载。



1.2 配置网络源：

```
# 安装163的yum源：
wget -O /etc/yum.repos.d/CentOS7-Base-163.repo
http://mirrors.163.com/.help/CentOS7-Base-163.repo
# 安装阿里云的yum源：
wget -O /etc/yum.repos.d/CentOS-Base.repo
http://mirrors.aliyun.com/repo/Centos-7.repo
```

```
# 搜狐没有现成的yum源文件下载，需要自己配置：

[sohu]
name=Centos-$releasever-sohu
baseurl=http://mirrors.sohu.com/centos/$releasever/os/$basearch
gpgcheck=1
gpgkey=http://mirrors.sohu.com/centos/$releasever/os/$basearch/RPM-GPG-KEY-CentOS-$releasever

重新执行: yum makecache
```

1.3 配置本地源：

```
mkdir /mnt/cdrom
mount /dev/cdrom /mnt/cdrom
vim /etc/yum.repos.d/centos-7-local.repo

[local]
name=centos-$releasever-local
baseurl=file:///mnt/cdrom
gpgcheck=1
gpgkey=file:///mnt/cdrom/RPM-GPG-KEY-CentOS-$releasever
```

1.4 自动配置仓库：

```
# 安装yum的扩展包：
yum install yum-utils -y

# 自动配置国内epel仓库：
yum-config-manager --add-repo=https://mirrors.tuna.tsinghua.edu.cn/epel/7/x86_64/
```

1.5 禁用/启用仓库：

```
# epel 是仓库的id [epel]
yum-config-manager --disable epel
yum-config-manager  --enable  epel

# 查看仓库状态：
yum  repolist all
```

1.6 yum常用命令：

```
yum repolist {all|enabled|disabled}          列出所有/已启用/已禁用的yum源
yum list {all|installed|avaliable}           列出所有/已安装/可安装的软件包
yum info package                             显示某一个软件包的信息
yum install package                          安装软件包
yum reinstall package                        重新安装软件包
yum remove|erase package                     卸载软件包
yum provides files                           查询某个文件是哪个软件包生成的
yum  search  file    查询某个文件是哪个软件包生成的
```

2. 同步外网源：

在企业实际应用场景中，仅仅靠光盘里面的RPM软件包是不能满足需要，我们可以把外网的YUM源中的所有软件包同步至本地，可以完善本地YUM源的软件包数量及完整性。

2.1 安装reposync工具：

```
yum  install yum-utils createrepo -y
```

2.2 同步源：

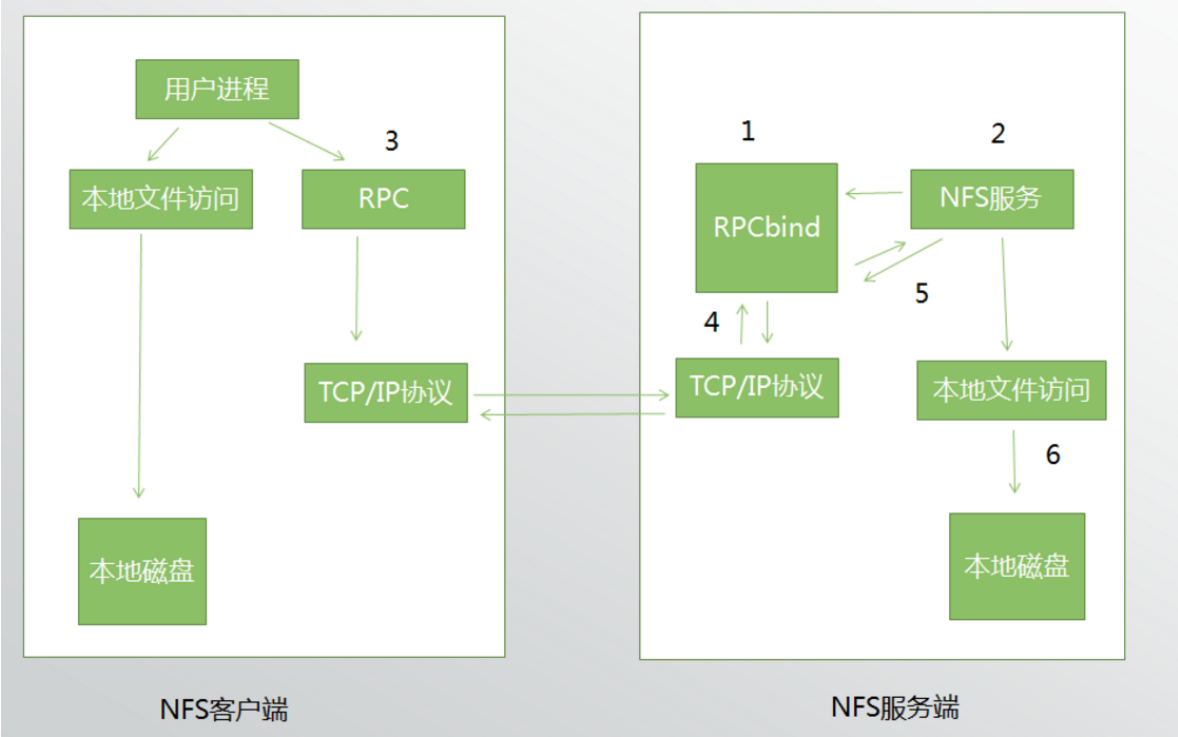
```
# 创建本地目录:
mkdir /data/{centos,epel}
# 同步yum源:
reposync -r base -r updates -p /data/centos/
# 生成元数据:
createrepo /data/centos

# 结合前面所学制作本地源，如果想让其他服务器使用该源，后面可以结合nginx发布。
```

3. NFS文件共享服务器:

NFS是network file sytem的缩写，他最大的特点就是可以通过网络，让不同的机器，不同的系统实现文件共享。NFS客户端可以将NFS服务器共享的目录挂载在本地的文件系统中，访问目录就如同访问自己本地目录一样。

3.1 NFS工作原理:



1. 首先NFS服务器端开启rpcbind;
2. 然后服务端开启NFS服务，这时NFS的各项功能都需要向RPC服务注册，这时rpc会通知portmap模块将可用的端口分配给statd，rquotad等;
3. 然后NFS客户端RPC服务就会通过网络向NFS服务端的RPC服务的111端口发出NFS文件存取功能的询问请求。

4. NFS服务端的RPC服务找到对应的已注册的NFSdaemon端口后，通知NFS客户端的RPC服务。
5. 此时NFS客户端就可获取到nfs服务端各个进程的正确端口，然后通过客户端rpc就直接与NFS服务器的rpc进行存取数据了（rpc知道了nfs的具体端口，就可以实现远程调用，即传输）。

3.2 NFS安装部署：

服务器和客户端都关闭防火墙，装好nfs服务组件：

```
nfs服务端：192.168.75.130
nfs客户端：192.168.75.135

# 关闭防火墙：
systemctl stop firewalld && systemctl disable firewalld
# 临时关闭selinux：
setenforce 0
# 永久关闭selinux：
sed -i 's/=enforcring/=disabled/' /etc/selinux/config
# 安装nfs服务组件：
yum -y intall nfs-utils
```

3.2.1 配置服务端：

- 编辑/etc/exports文件

```
/data/jfedu   192.168.75.0/24(rw,sync)

# 格式：
# /data/jfedu    要共享的目录，需要存在
# 192.168.75.0/24    谁能挂载使用，可以是网段，也可以指定具体ip
# (rw,sync)          挂载的一些参数，rw表示挂载为可读可写，sync表示同步
```

- 导出（广播）编辑的文件，并重启rpc和nfs服务

```
systemctl restart rpcbind
systemctl restart nfs
exportfs -r
```

3.2.2 配置客户端：

- 可用showmount搜索网络中可用的共享文件

```
showmount -e 192.168.75.130
```

- 创建目录，用于挂载

```
mkdir /mnt/nfs
```

- 挂载

```
mount -t nfs  192.168.75.130:/data/jfedu /mnt/nfs

#推荐使用：
mount -t nfs -o soft,timeo=1   192.168.75.130:/data/jfedu
/mnt/nfs

soft：    软挂载，遇到报错会终止挂载，并返回信息，默认是硬挂载，一直尝试挂载。
timeo:   超时时间，如果不设置，一直链接，可以设置小点
```

挂载完成之后，进入目录，可能会发现**无法对目录中的文件进行修改。**

这主要是因为客户端访问服务器时，身份被压缩成nobody，相对服务器文件系统来说，就是其他用户。所以要想编辑，需要在服务端对文件授权或者更改exports文件，设置no_root_squash（不压缩客户端root身份）。

3.3 解读exports文件：

```
[root@localhost ~]# exportfs -v
/data/jfedu
192.168.75.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,
rw,secure,no_root_squash,no_all_squash)

其中:
rw:可读可写
ro:仅可读
sync:是指数据同步写入内存和磁盘
root_squash:如果客户端用root身份访问，则被压缩成nobody,权限也将受到限制。
no_root_squash:也就是不压缩，客户端使用root身份登录，全有所有权限，很危险。
all_squash:不管访问者是什么身份，包括root，全部压缩至匿名用户。
no_all_squash:保留访问用户的身份uid以及gid,一般只能查看，不能修改，权限问题，但是可以强制保存。
```

3.4 报错处理:

3.4.1 卸载时报错:

```
umount.nfs4: /mnt/jfedu: device is busy

umount -l /mnt/jfedu    强行解除挂载

或者使用
fuser -m /mnt/data 将会显示使用这个模块的pid
fuser -mk /mnt/data 将会直接kill那个pid
```

#####