1. Yum的repo文件:

 repo文件是Fedora中yum源（软件仓库）的配置文件，通常一个repo文件定义了一个或者多个软件仓库的细节内容，例如我们将从哪里下载需要安装或者升级的软件包，repo文件中的设置内容将被yum读取和应用！  
 YUM的工作原理并不复杂，每一个 RPM软件的头（header）里面都会纪录该软件的依赖关系，那么如果可以将该头的内容纪录下来并且进行分析，可以知道每个软件在安装之前需要额外安装 哪些基础软件。也就是说，在服务器上面先以分析工具将所有的RPM档案进行分析，然后将该分析纪录下来，只要在进行安装或升级时先查询该纪录的文件，就可 以知道所有相关联的软件。所以YUM的基本工作流程如下：  
    服务器端：在服务器上面存放了所有的RPM软件包，然后以相关的功能去分析每个RPM文件的依赖性关系，将这些数据记录成文件存放在服务器的某特定目录内。  
    客户端：如果需要安装某个软件时，先下载服务器上面记录的依赖性关系文件(可通过WWW或FTP方式) ，通过对服务器端下载的纪录数据进行分析，然后取得所有相关的软件，一次全部下载下来进行安装。

2./etc/yum.conf配置文件中的参数说明：

[main]  
cachedir=/var/cache/yum  
#cachedir：yum缓存的目录，yum在此存储下载的rpm包和数据库，一般是/var/cache/yum。  
debuglevel=2  
#debuglevel：除错级别，0──10,默认是2 貌似只记录安装和删除记录  
logfile=/var/log/yum.log  
pkgpolicy=newest  
#pkgpolicy： 包的策略。一共有两个选项，newest和last，这个作用是如果你设置了多个repository，而同一软件在不同的repository中同时存 在，yum应该安装哪一个，如果是newest，则yum会安装最新的那个版本。如果是last，则yum会将服务器id以字母表排序，并选择最后的那个 服务器上的软件安装。一般都是选newest。  
distroverpkg=centos-release  
#指定一个软件包，yum会根据这个包判断你的发行版本，默认是redhat-release，也可以是安装的任何针对自己发行版的rpm包。  
tolerant=1  
#tolerent，也有1和0两个选项，表示yum是否容忍命令行发生与软件包有关的错误，比如你要安装1,2,3三个包，而其中3此前已经安装了，如果你设为1,则yum不会出现错误信息。默认是0。  
exactarch=1  
#exactarch，有两个选项1和0,代表是否只升级和你安装软件包cpu体系一致的包，如果设为1，则如你安装了一个i386的rpm，则yum不会用1686的包来升级。  
retries=20  
#retries，网络连接发生错误后的重试次数，如果设为0，则会无限重试。  
obsoletes=1  
gpgcheck=1  
#gpgchkeck= 有1和0两个选择，分别代表是否是否进行gpg校验,也就是gpg完整性测试，如果没有这一项，默认是检查的。  
reposdir=/etc/yy.rm #默认是 /etc/yum.repos.d/ 低下的 xx.repo后缀文件  
#默认都会被include 进来 也就是说 /etc/yum.repos.d/xx.repo 无论配置文件有多少个 每个里面有多少个[name] 最后其实都被整合到 一个里面看就是了 重复的[name]后面的覆盖前面的  
exclude=xxx  
#exclude 排除某些软件在升级名单之外，可以用通配符，列表中各个项目要用空格隔开，这个对于安装了诸如美化包，中文补丁的朋友特别有用。  
keepcache=[1 or 0]  
#　　设置 keepcache=1，yum 在成功安装软件包之后保留缓存的头文件 (headers) 和软件包。默认值为 keepcache=0 不保存

Plugins =1

#用于是否启动yum的plugin插件。如果为1则为启动，为0则禁用所有。这是一个总开关，如果为1则启动，还需要取决各个yum plugin插件在/etc/yum/pluginconf.d/自己的配置中是否启动。如果没0则，全部禁用，就不需要查看个体的plugin启动情况了。

使用yum info yum可以查看yum情况，包括其加载的插件。

3. yum. repo文件的说明  
[serverid]  
#其中serverid是用于区别各个不同的repository，必须有一个独一无二的名称, 这个名称作为repo源的名称。一个. repo文件下定义了很多个repo源。

name=Some name for this server  
#name，是对repository的描述，支持像$releasever $basearch这样的变量; name=Fedora Core $releasever - $basearch - Released Updates

baseurl=url://path/to/repository/  
#baseurl是服务器设置中最重要的部分，只有设置正确，才能从上面获取软件。它的格式是：  
baseurl=url://server1/path/to/repository/  
url://server2/path/to/repository/  
url://server3/path/to/repository/  
#其中url支持的协议有 http:// ftp:// file://, https://。baseurl后可以跟多个url，你可以自己改为速度比较快的镜像站，但baseurl只能有一个，也就是说不能像如下格式：  
baseurl=url://server1/path/to/repository/  
baseurl=url://server2/path/to/repository/  
baseurl=url://server3/path/to/repository/  
其中url指向的目录必须是这个repository header目录的上一级，它也支持$releasever $basearch这样的变量。  
  
#mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=fedora-$releasever&arch=$basearch  
#上面的这一行是指定一个镜像服务器的地址列表，通常是开启的，本例中加了注释符号禁用了，我们可以试试，将$releasever和$basearch替换成自己对应的版本和架构，例如10和i386，在浏览器中打开，我们就能看到一长串镜可用的镜像服务器地址列表。  
  
url之后可以加上多个选项，如gpgcheck、exclude、failovermethod等，比如：  
gpgcheck=1  
exclude=gaim  
#其中gpgcheck，exclude的含义和[main]部分相同，但只对此服务器起作用，  
failovermethod=priority  
#failovermethode 有两个选项roundrobin和priority，意思分别是有多个url可供选择时，yum选择的次序，roundrobin是随机选择，如果连接失 败则使用下一个，依次循环，priority则根据url的次序从第一个开始。如果不指明，默认是roundrobin。  
  
enabled=[1 or 0]  
#当某个软件仓库被配置成 enabled=0 时，yum 在安装或升级软件包时不会将该仓库做为软件包提供源。使用这个选项，可以启用或禁用软件仓库。  
#通过 yum 的 --enablerepo=[repo\_name] 和 --disablerepo=[repo\_name] 选项，或者通过 PackageKit 的"添加/删除软件"工具，也能够方便地启用和禁用指定的软件仓库  
  
几个变量：  
$releasever，发行版的版本，从[main]部分的distroverpkg获取，如果没有，则根据redhat-release包进行判断。  
$arch，cpu体系，如i686,athlon等  
$basearch，cpu的基本体系组，如i686和athlon同属i386，alpha和alphaev6同属alpha。

4. Yum GPG key  
Reposity的GPG key，前面说过，yum可以使用gpg对包进行校验，确保下载包的完整性，所以我们先要到各个repository站点找到gpg key，一般都会放在首页的醒目位置，一些名字诸如 RPM-GPG-KEY.txt之类的纯文本文件，把它们下载，然后用rpm --import xxx.txt命令将它们导入，最好把发行版自带GPG-KEY也导入，rpm --import /usr/share/doc/redhat-release-\*/RPM-GPG-KEY 官方软件升级用的上。

5. EPEL

如果既想获得 RHEL 的高质量、高性能、高可靠性，又需要方便易用(关键是免费)的软件包更新功能，那么 Fedora Project 推出的 EPEL(Extra Packages for Enterprise Linux)正好适合你。EPEL(http://fedoraproject.org/wiki/EPEL) 是由 Fedora 社区打造，为 RHEL 及衍生发行版如 CentOS、Scientific Linux 等提供高质量软件包的项目。  
  
特点：  
1，不用去换原来yum源，安装后会产生新repo  
2，epel会有很多源地址，如果一个下不到，会去另外一个下  
   http://mirror.suhu.com/fedora-epel/6/x86\_64/epel-release-6-8.noarch.rpm

安装epel：  
32位系统选择：  
rpm -ivh http://mirror.suhu.com/fedora-epel/6/i386/epel-release-6-8.noarch.rpm  
  
64位系统选择：  
rpm -ivh http://mirror.suhu.com/fedora-epel/6/x86\_64/epel-release-6-8.noarch.rpm  
  
导入key：  
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6  
  
如果用比较新的软件，用epel-test.repo这个文件就行了  
  
6. Yum源更新的步骤

----1. 首先备份/etc/yum.repos.d/CentOS-Base.repo

mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.backup

-----2. 下载对应版本repo文件, 放入/etc/yum.repos.d/(操作前请做好相应备份)

Centos7:

wget -P /etc/yum.repos.d <http://mirrors.163.com/.help/CentOS7-Base-163.repo>

Centos6:

wget -P /etc/yum.repos.d <http://mirrors.163.com/.help/CentOS6-Base-163.repo>

Centos5:

wget -P /etc/yum.repos.d <http://mirrors.163.com/.help/CentOS5-Base-163.repo>

-----3. 运行以下命令生成缓存

yum clean all  
yum makecache

7. yum repolist all

这个命令列出所有的repo源状态，不管是enabled还是disabled的。注意这里列出的是repo源的名称，不是.repo文件的名称，一个. repo文件中定义了许多的repo源。每个repo源才是独立的个体，repo文件名称并不重要。

8. yum总共有两种缓存

第一种是cachedir=/var/cache/yum  
#cachedir：yum缓存的目录，yum在此存储下载下来的rpm包和数据库，一般是/var/cache/yum。

第二种yum makecache,

他的作用是服务器的包信息下载到本地电脑缓存起来，以后安装的时候yum不需要询问，直接按照缓存下来的去请求下载安装。如果我们清除cache以后不手动执行这个命令，那么在做yum list installed, yum repolist all等yum命令的时候，yum也会自动自己做一次。

9. yum-config-manager 用于管理YUM, 可以通过CLI来启用某个repo 加入某个repo等。而不需要到YUM配置文件中进行配置。

10.搜索显示配置的YUM中指定包可安装的版本号：

yum --showduplicates list [package名字]

安装执行版本号的包

yum install [package名字]- [version info]

11.

yum clean all + yum list installed 首先系统清除yum缓存，然后yum list installed因为系统不存在yum缓存所以会先生成缓存，生成系统缓存的时候会读取repo信息。如果设置的repo中有失效repo则会报出信息。可以用于清理yum repo。

12.