yum

RPM通过rpm database来实现管理:位于 /var/lib/rpm

这个过程中的依赖关系比较难以管理

yum最大作用：自动解决依赖关系：

x-->y

x: 依赖哪些包;

y: 安装情况；

yum如何解决依赖关系

依赖一个核心的东西：repository: 仓库，yum源 一个rpm包的目录，一个xml文件

RHEL5：中所有的软件包是分开存放的，四个目录中有不同的东西

Server

如果把这个目录当做源，这里一定有一个：repodata/

Cluster

ClusterStorage

VT

RHEL6:所以的rpm包都在

Package中存放

Server中没有了rpm包只不过只是定义了数据包的元信息

rpm： 每一个rpm包依赖于其它的哪些capacity，是rpm包的制作者定义的；

name-version-release.arch.rpm：

获取数据的方法：

本地：[file:///](NULL) /var/yumrepo/Server

ftp服务器：ftp://192.168.0.254/pub/Server/

ftp://USERNAME:PASSWORD@192.168.0.254/pub/Server/

**yum仓库包含：**

1)rpm包

**2)repodata/:** 只有包含此目录为其子目录的那个路径才可以用作yum源：(都是源信息)

repomd.xml

primary.xml.gz：当前Yum仓库内每一个rpm包的名字及其彼此间的依赖关系；此外还包含每一个rpm能够安装生成的文件；

filelists.xml.gz: 所有rpm包包含的安装生成的文件的文件列表；时间戳 校验值，yum源是否改变

other.xml.gz: 额外信息，主要包含的是rpm包的changelog；

rpm包包组的定义：人为的不是必须的

yum配置文件：

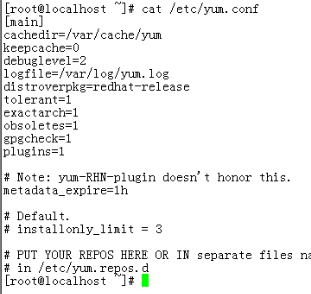
/etc/yum.conf 主配置文件，这里还能指定包含其他文件

/etc/yum.repos.d/\*.repo

/etc/yum/pluginconf.d/ 插件 有时候有

配置文件：分为两段

全局配置：对所有的yum仓库都生效



分段配置：配置一个个的yum仓库

[repo]

para=value

[repo2]

para=value

[repo3]

样例：

[repo]

name=Primary Yum repository //详细说明名字,=号两段不能有空格

baseurl=ftp://192.168.0.254/pub/Server

enabled={0|1}

gpgcheck={0|1}

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

yum的命令：

repolist 显示所有可用yum源信息

list {installed|available|updates|all}

clean {all|packages|metadata} 清除缓存

/var/cache/yum: yum包的缓存

元数据

rpm包

grouplist 显示包组的相关信息

groupinfo "GROUP\_NAME"

groupinstall "GROUP\_NAME"

groupremove "GROUP\_NAME"

info PKG\_NAME

search KEYWORD

whatprovides FILE 查找某文件时是由哪一个文件生成

provides FILE

install 不需要指定版本号

remove

update

reinstall

check-update

localinstall

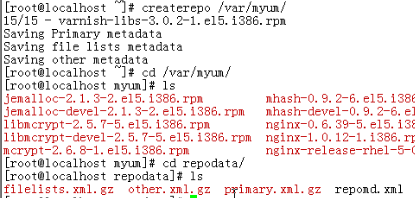
yum命令的选项：

-y: 自动回答为yes

--nogpgcheck

createrepo

-g FILE: 指定包组定义文件



mirrorlist Specifies a URL to a file containing a list of baseurls. This can be used instead of or with the baseurl option.

Substitution variables, described below, can be used with this option. As a special hack is the mirrorlist URL contains the

word "metalink" then the value of mirrorlist is copied to metalink (if metalink is not set).

enabled Either '1' or '0'. This tells yum whether or not use this repository.

gpgcheck Either '1' or '0'. This tells yum whether or not it should perform a GPG signature check on the packages gotten from

this repository.

repo\_gpgcheck Either '1' or '0'. This tells yum whether or not it should perform a GPG signature check on the repodata from

this repository.

gpgkey A URL pointing to the ASCII-armored GPG key file for the repository. This option is used if yum needs a public key to

verify a package and the required key hasn't been imported into the RPM database. If this option is set, yum will automatically

import the key from the specified URL. You will be prompted before the key is installed unless the assumeyes option is set.

Multiple URLs may be specified here in the same manner as the baseurl option (above). If a GPG key is required to install a

package from a repository, all keys specified for that repository will be installed.

VARIABLES

There are a number of variables you can use to ease maintenance of yum's configuration files. They are available in the values of sev-

eral options including name, baseurl and commands.

$releasever This will be replaced with the value of the version of the package listed in distroverpkg. This defaults to the

version of 'redhat-release' package.

$arch This will be replaced with your architecture as listed by os.uname()[4] in Python.

$basearch This will be replaced with your base architecture in yum. For example, if your $arch is i686 your $basearch will be

i386.

$YUM0-$YUM9 These will be replaced with the value of the shell environment variable of the same name. If the shell environment