Linux 云计算集群架构师

学神 IT 教育：从零基础到实战，从入门到精通！

联系方式：

学神 IT 教育官方网站: http://www.xuegod.cn

Linux 云计算架构师进阶学习群 QQ 群: 1072932914

学习顾问：小语老师　　学习顾问：边边老师　　学神微信公众号

微信扫码添加学习顾问微信，同时扫码关注学神公众号了解最新行业动态，获取更多学习资料及答疑就业服务！

第八章　Centos8 软件包的管理与安装

本节所讲内容：

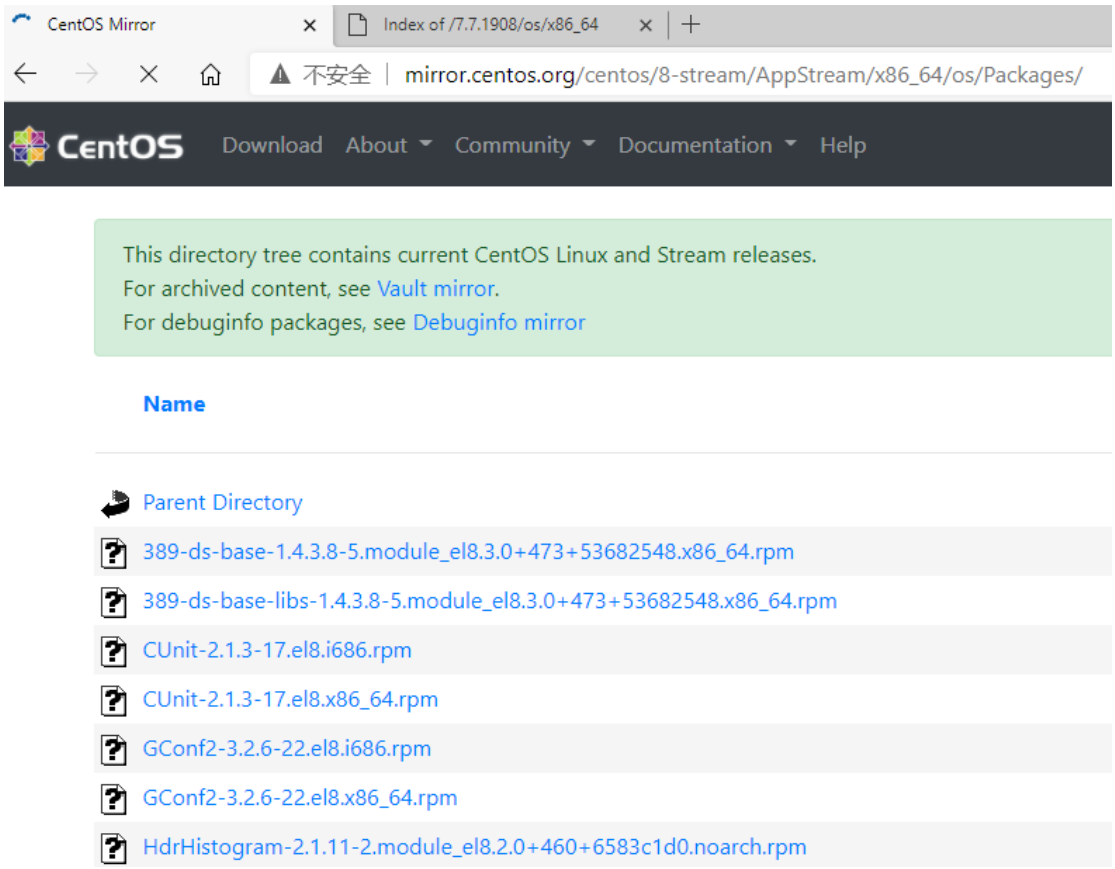
8.1　使用 rpm 命令-安装-查看-卸载-rpm 软件包
8.2　yum 管理软件包
8.3　CentOS8 中使用 DNF 管理软件包
8.4　实战 tar 源码包管理-源码包安装方法

8.1　软件包的管理

软件包的类型
rpm 二进制包------》已经使用 GCC 编译后的（二进制已经可以被操作系统直接执行了）
tar 源码包-----》需要编译（源码包就是你能看懂的，基于字符的，还需要进行编译）
RPM 概述：RPM 是 RPM RedHat Package Manager（RPM 软件包管理器）的缩写，这一文件格式名称虽然打上了 RedHat 的标志，但是其原始设计理念是开放式的，现在包括 OpenLinux、SUSE 以及 Turbo Linux 等 Linux 的分发版本都有采用，可以算是公认的行业标准了。
kaili　apt install name.deb

8.1.1　rpm 软件包的管理

rpm 包的获取方式：
1、Centos 系统镜像光盘
2、网站 rpmfind.net
3、比如安装 mysql、nginx 软件，我们可以去它的官方网站下载:
http://nginx.org/en/download.html
4、centos yum 源上，也有 rpm 可以手动下载
https://www.centos.org/download/



rpm 包格式的说明

例 1：

[root@xuegod63 ~]# ls /mnt/BaseOS/Packages/zsh-5.5.1-6.el8_1.2.x86_64.rpm

/mnt/BaseOS/Packages/zsh-5.5.1-6.el8_1.2.x86_64.rpm

zsh -5. 5. 1- 6. el8. x86_64.rpm

软件名 主版本号 次版本号 修订 release 发布 操作系统版本 软件包是 64 位包

release（rpm 自身的发布版本号，表示这个 rpm 软件包是第几次编译生成的，与程序源码的发行号无关）

#修订指是的第几次修改 bug。 发布指的是：第几次发布。 发布时，可能只是对软件安装的默认参数做了修改，而没有其它改动，就做了一次编译。

el8 redhat8.x/centos8.x

x86_64：表示软件包是 64 位

.rpm：.rpm 和.src.rpm，是 rpm 包类型后缀，rpm 是编译好的二进制包，.src.rpm 是源码包

devel：表示这个 rpm 包是软件的开发包

noarch：说明这样的软件包可以在任何平台安装和运行，不需要特定的硬件平台

例 2:

[root@xuegod63 ~]# ls /mnt/BaseOS/Packages/atlas-3.10.3-7.el8.i686.rpm

/mnt/BaseOS/Packages/atlas-3.10.3-7.el8.i686.rpm

注： .i686 代表，此包是 32 位操作系统包。 64 位操作系统是可以安装 32 位操作系统的包。 32

位操作系统，安装不了 64 位的包。 而且在 centso7 开始，就没有 32 位操作系统。

例 3：

[root@xuegod63 ~]# ls /mnt/AppStream/Packages/zsh-html-5.5.1-6.el8_1.2.noarch.rpm

注：结尾有 noarch，代表此包在 32 位和 64 位操作系统上都可以运行。 这类型包，里面通常是文本文件，如： shell 脚本，html，txt 等。

root@xuegod63 ~]# uname -r #查看内核版本。

4.18.0-193.el8.x86_64 #我的内核版本是 4.18.0-193.el8，有 x86_64 就是 64 位操作系统

8.1.2 安装 rpm 软件

RPM 工具使用分为安装、查询、验证、更新、删除等操作

命令格式：rpm [参数] 软件包

参数：

-i 是 install 的意思， 安装软件包

-v 显示附加信息，提供更多详细信息

-V 校验，对已经安装的软件进行校验

-h --hash 安装时输出####标记

```
准备中...                          ################################# [100%]
```

互动：rpm 使用时，什么情况下使用软件包全名，什么时候使用软件包名?

全名：在安装和更新升级时候使用

包名：对已经安装过的软件包进行操作时，比如查找已经安装的某个包，卸载包等 ，使用包名。它默认是去目录/var/lib/rpm 下面进行搜索。 当一个 rpm 包安装到系统上之后,安装信息通常会保存在本地的 /var/lib/rpm/目录下。

例 1：从本地安装

[root@xuegod63 ~]# mount /dev/sr0 /mnt #挂载，确保光盘镜像已经在虚拟机开机加载

[root@xuegod63 ~]# rpm -ivh /mnt/BaseOS/Packages/lrzsz-0.12.20-43.el8.x86_64.rpm

#本地安装 lrzsz 包，安装后可以使用 rz 和 sz 命令。

8.1.3 rpm 查询功能

用法：rpm -q（query） 常与下面参数组合使用

-a（all） 查询所有已安装的软件包

-f（file）系统文件名（查询系统文件所属哪个软件包），反向查询

-i 显示已经安装的 rpm 软件包信息，后面直接跟包名

-l（list） 查询软件包中文件安装的位置

-p 查询未安装软件包的相关信息，后面要跟软件的命名

-R 查询软件包的依赖性

例：

[root@xuegod63 mnt]# rpm -q lrzsz --->查询指定的包是否安装

[root@xuegod63 mnt]# rpm -qa --->查询所有已安装包

例：查看 passwd 文件中包括 bash 的行。

[root@xuegod63 ~]# grep bash /etc/passwd #grep 后面加关键字，可以查找文件中的内容。

root:x:0:0:root:/root:/bin/bash

mk:x:1000:1000:mk:/home/mk:/bin/bash

[root@xuegod63 mnt]# rpm -qa | grep lrzsz --->查询所有已安装包中带 vim 关键字的包
[root@xuegod63 ~]# which find #查看 find 命令的路径
/usr/bin/find
[root@xuegod63 ~]# rpm -qf /usr/bin/find #查询文件或命令属于哪个安装包
查询已经安装的 rpm 包的详细信息或作用 rpm -qi rpm 包名
[root@xuegod63 ~]# rpm -qf `which find` #反引号中可以执行 shell 命令
[root@xuegod63 ~]# rpm -qi lrzsz

[root@xuegod83 ~]# rpm -qf `which vim`
[root@xuegod83 ~]# rpm -qi vim-enhanced

针对没有安装的 RPM 包，要加参数： -p
[root@xuegod63 ~]# rpm -qpi \
/mnt/AppStream/Packages/php-mysqlnd-7.2.11-2.module_el8.1.0+209+03b9a8ff.x86_64.rpm
Summary : A module for PHP applications that use MySQL databases
#php 使用 mysql 数据库的一个模块

[root@xuegod63 mnt]# rpm -qpl \
/mnt/AppStream/Packages/nginx-1.14.1-9.module_el8.0.0+184+e34fea82.x86_64.rpm
#查看 rpm 安装后，将生成哪些文件

8.1.4 查看软件包内容是否被修改
rpm -V 包名
rpm -Vf 文件路径
例：
[root@xuegod63 ~]# which find
/usr/bin/find
[root@xuegod63 ~]# rpm -qf /usr/bin/find
findutils-4.5.11-5.el7.x86_64
注：上面两条命令，等价于以下面这条命令：
[root@xuegod63 ~]# rpm -qf `which find` # 这是反引号。 一行命令中，如果有反引号，那么先执行反引号中的命令，把反引号中的命令的输出，作为前面命令输入。

[root@xuegod63 ~]# rpm -Vf /usr/bin/find # 参数-Vf 后面加文件的路径，查看每个命令或文件，有没有被修改。
[root@xuegod63 ~]# echo aaa >> /usr/bin/find
[root@xuegod63 ~]# rpm -Vf /usr/bin/find
S.5....T. /usr/bin/find
[root@xuegod63 ~]# rpm -V findutils #检查包
S.5....T. /usr/bin/find
注：如果出现的是点，表示测试这一项，没有被修改

出现下面的字符代表某测试的失败:

5 — MD5 校验和是否改变，你也看成文件内容是否改变

S — 文件长度，大小是否改变

L — 符号链接，文件路径是否改变

T — 文件修改日期是否改变

D — 设备

U — 用户，文件的属主

G — 用户组

M — 模式（包含许可和文件类型）

? — 不可读文件

再后面的 c 文件名,它表示的是文件类型

c 配置文件

d 普通文件

g 不该出现的文件，意思就是这个文件不该被这个包所包含

l 授权文件（license file）

r 描述文件

[root@xuegod63 ~]# rpm -V lrzsz # -V 后面加软件包的名字，查看这个包安装的所有文件，没有被修改。

互动： 查看系统中所有的 rpm 包及安装的文件有没有被黑客修改

root@xuegod63 ~]# rpm -Va > rpm_check.txt

注： 检验时参考了 /var/lib/rpm 目录下的 rpm 数据库信息

```
[root@xuegod63 ~]# ls /var/lib/rpm
Basenames      Enhancename       Packages       Suggestname
Conflictname   Filetriggername   Providename    Supplementname
__db.001       Group             Recommendname  Transfiletriggername
__db.002       Installtid        Requirename    Triggername
__db.003       Name              Sha1header
Dirnames       Obsoletename      Sigmd5
[root@xuegod63 ~]#
```

8.1.5 rpm 包卸载和升级

用法：rpm -e（erase） 包名

[root@xuegod63 ~]# rpm -q lrzsz 错误

lrzsz-0.12.20-43.el8.x86_64

[root@xuegod63 ~]# rpm -e lrzsz

[root@xuegod63 ~]# rpm -q lrzsz #已经找不到 lrzsz 包了，说明卸载成功了

参数： --nodeps 忽略依赖，建议在卸载时不要用 rpm 去卸载有依赖关系的包，应该用 yum

[root@xuegod63 ~]# rpm -e --nodeps lrzsz

升级:

[root@xuegod63 ~]# rpm -Uvh \

/mnt/BaseOS/Packages/lrzsz-0.12.20-43.el8.x86_64.rpm #centos8 下安装

[root@xuegod63 ~]# rpm -Uvh /mnt/Packages/lrzsz-0.12.20-36.el7.x86_64.rpm #centos7 下升级或安装 lrzsz 包

#因为升级时会有一些依赖包要解决。 所以一般我们使用 yum update 包来升级。

8.1.6 解决 rpm 依赖关系：

[root@xuegod63 ~]# rpm -ivh /mnt/AppStream/Packages/httpd-tools-2.4.37-21.module_el8.2.0+382+15b0afa8.x86_64.rpm

报错：

```
[root@xuegod63 ~]# rpm -ivh /mnt/AppStream/Packages/httpd-tools-2.4.37-21.module_el8.2.0+
382+15b0afa8.x86_64.rpm
错误：依赖检测失败：
        libapr-1.so.0()(64bit) 被 httpd-tools-2.4.37-21.module_el8.2.0+382+15b0afa8.x86_6
4 需要
        libaprutil-1.so.0()(64bit) 被 httpd-tools-2.4.37-21.module_el8.2.0+382+15b0afa8.x
86_64 需要
```

我们需要把依赖包安装上，才可以。

[root@xuegod63 ~]# rpm -ivh /mnt/AppStream/Packages/apr-1.6.3-9.el8.x86_64.rpm

[root@xuegod63 ~]# rpm -ivh /mnt/AppStream/Packages/apr-util-1.6.1-6.el8.x86_64.rpm

[root@xuegod63 ~]# rpm -ivh /mnt/AppStream/Packages/httpd-tools-2.4.37-21.module_el8.2.0+382+15b0afa8.x86_64.rpm

现在就可以安装成功了。

8.2 YUM 的使用

yum（全称为 Yellow dog Updater Modified）是一个前端软件包管理器。基于 RPM 包管理，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件，无须繁琐地一次次下载、安装。

例如我们需要安装一个软件 A，而软件 A 依赖 B，而 B 可能还继续依赖其他软件，通过 yum 我们只需要安装 A，其他依赖包会自动安装。

yum 提供了查找、安装、删除某一个、一组甚至全部软件包的命令，而且命令简洁而又好记

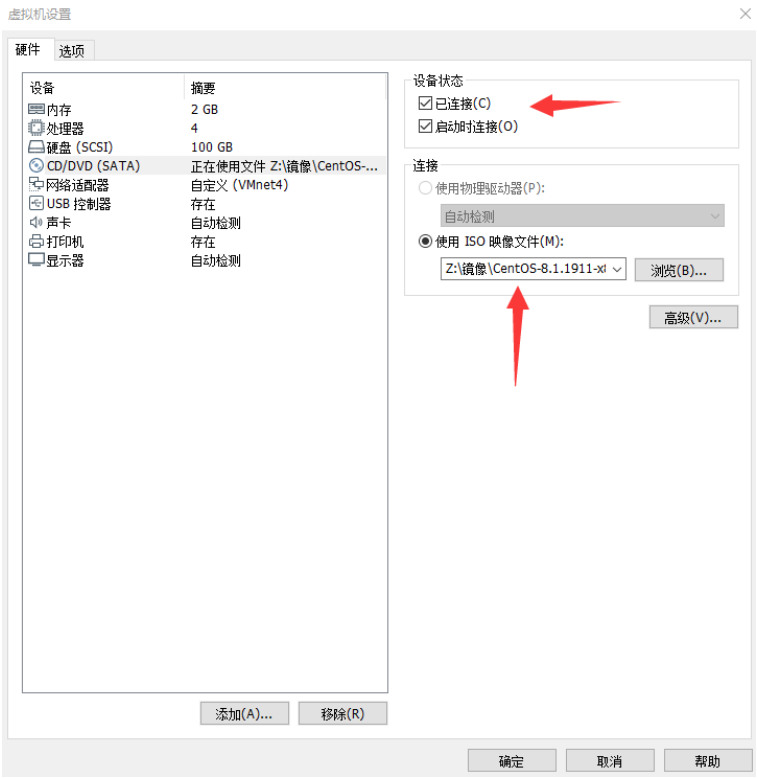
YUM：解决依赖关系问题，自动下载软件包，它是基于 C/S 架构

C=client S=server、ftp\http\file

8.2.1 配置 yum 源

1、挂载镜像：

先确定虚拟机光驱中有加载系统镜像



[root@xuegod63 Packages]# mount /dev/cdrom /mnt/

[root@xuegod63 Packages]# ls /mnt/

2、本地配置 yum 源文件：

centos8 本地 YUM 源配置：

在 centos8 当中，本地光盘中的安装包被分别放在了两个路径下。假设挂载路径为/mnt，那么两个 Package 路径分别为：/mnt/AppStream/Packages 和/mnt/BaseOS/Packages。

因为 IOS 镜像内的设置，所以导致了在编辑本地 yum 源的时候需要分别写两个路径在配置文件中。同时网络 yum 源也被分别写到了两个配置文件内。所以就需要把两个网络 yum 源配置文件改名。

```
[root@bogon ~]# ls /etc/yum.repos.d/
CentOS-AppStream.repo    CentOS-Debuginfo.repo  CentOS-PowerTools.repo
CentOS-Base.repo         CentOS-Extras.repo     CentOS-Sources.repo
CentOS-centosplus.repo   CentOS-fasttrack.repo  CentOS-Vault.repo
CentOS-CR.repo           CentOS-Media.repo
[root@bogon ~]#
```

#centos8 与之前版本不同的是增加了一个 CentOS-AppStream.repo 文件，CentOS-AppStream.repo 和 CentOS-Base.repo 文件都需要移除目录或者改名。

BaseOS 类似于原来的软件仓库，主要提供了系统的基础组件，它支持与之前版本兼容

AppStream:它提供的是系统以外的应用程序，如 httpd，nginx，php，mariadb-server 等

(1)首先挂载光盘：

[root@localhost ~]# mount /dev/sr0 /mnt/

mount: /mnt: WARNING: device write-protected, mounted read-only.

(2)创建一个本地 yum 配置文件 centos8.repo：

[root@bogon ~]# vim /etc/yum.repos.d/centos8.repo　　#写入以下内容

[c8-media-BaseOS]

name=CentOS-BaseOS-$releasever - Media
baseurl=file:///mnt/BaseOS
gpgcheck=1
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial

[c8-media-AppStream]
name=CentOS-AppStream-$releasever - Media
baseurl=file:///mnt/AppStream
gpgcheck=1
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial

#添加修改完两个区域后保存退出。
注：如果 gpgcheck=1 ，需要导入 rpm 公钥。方便后期校对 rpm 包。一般情况，写为 0
工作中就写成 1 并导入公钥。这样安全。

(3)移动网络 yum 源配置文件位置：
[root@xuegod83 ~]# mkdir /opt/yum
[root@ xuegod83 ~]# mv /etc/yum.repos.d/C* /opt/yum
#将网络 yum 配置文件移动到任意位置，使其不被 yum 所识别。
[root@ xuegod83 ~]# yum makecache
注：在 centos8 版本当中可以继续使用 yum 命令进行安装包管理。
[root@xuegod63 ~]# ll /usr/bin/yum
lrwxrwxrwx. 1 root root 5 4 月 25 03:57 /usr/bin/yum -> dnf-3
[root@xuegod83 ~]# ll /usr/bin/dnf
lrwxrwxrwx. 1 root root 5 8 月 5 2020 /usr/bin/dnf -> dnf-3
注：但是我们执行的 yum 命令是一个软连接，它被链接到了 dnf-3 命令上。
使用 DNF 来管理软件包，提升了包括用户体验，内存占用，依赖分析，运行速度等多方面内容。

3.网络 yum 源
Centos 使用阿里网络源：
https://developer.aliyun.com/mirror/centos

[root@xuegod63 ~]# wget -O /etc/yum.repos.d/Centos-8.repo http://mirrors.aliyun.com/repo/Centos-8.repo

例：安装 centos epel 扩展 yum 源。
注：epel 源是对 centos 系统中自带的 base 源的扩展。(因为不是所有软件包都放在 base 源里)
[root@xuegod63 ~]# yum -y install epel-release
[root@xuegod63 ~]# ls /etc/yum.repos.d/epel.repo #这就是安装的 epel 源
阿里 epel 源安装
https://developer.aliyun.com/mirror/epel

8.2.2 yum 使用

yum 常用操作：

[root@xuegod63 ~]# yum -y install httpd #安装软件包， -y 直接安装

[root@xuegod63 ~]# yum update #升级软件包，改变软件设置和系统设置,系统版本内核都升级。这里选择：n ，先不升级，因为升级需要的下载的包太多

```
事务概要
==========================
安装    4 软件包
升级   48 软件包

总下载： 262 M
确定吗？ [y/N]： n
操作中止。
[root@xuegod63 ~]#
```

[root@xuegod63 ~]# yum upgrade

#升级软件包，不改变软件设置和系统设置，系统版本升级，内核不改变。工作中推荐使用这种升级方式。

[root@xuegod63 ~]# yum info httpd #查询 rpm 包作用

[root@xuegod63 ~]# yum provides /usr/bin/find #查看命令是哪个软件包安装的

```
[root@xuegod63 ~]# yum provides  /usr/bin/find
Repository extras is listed more than once in the c
Repository centosplus is listed more than once in t
Repository PowerTools is listed more than once in t
Repository AppStream is listed more than once in th
Repository c8-media-BaseOS is listed more than once
Repository c8-media-AppStream is listed more than o
上次元数据过期检查： 0:05:07 前，执行于 2020年07月01
findutils-1:4.6.0-20.el8.x86_64 : The GNU versions
仓库          : @System
匹配来源：
文件名        : /usr/bin/find
```

注： 发现 find 命令是 findutils 包中的文件

[root@xuegod63 ~]# yum -y remove 包名 #卸载包

例：[root@xuegod63 ~]# yum -y remove httpd-tools

[root@xuegod63 ~]# yum search keyword #在软件包的包名和详细描述信息中搜索包括指定字符串的软件包

例：[root@xuegod63 ~]#　yum search httpd　　#查找包括 httpd 字符的软件包
[root@xuegod63 ~]# yum search lrzsz

yum 报错，注意的几个小问题：
1、确定光盘是否链接，光盘是否挂载
2、配置文件中格式是否正确，字母，符号有没有少写,挂载点和配置文件中设置的是否一致
3、网络源需要联网，操作和 RPM 类似，只是会自动安装依赖项。

8.2.3　yum 安装开发工具软件包组

[root@xuegod63 ~]#　yum grouplist　#查看有哪些软件包组
语法：yum groupinstall GROUPNAME
yum grouplist　#显示中文，如果想变成英文，则执行以一下命令
[root@bogon Packages]# echo $LANG
zh_CN.UTF-8
[root@bogon Packages]#LANG=en_US.UTF-8
yum grouplist
[root@localhost Packages]# yum grouplist

测试：
当你最小化安装系统后，在源码编译安装软件包时，觉得很需要安装很多依赖包，很痛苦，可以先安装好这个 Development tools 开发工具软件包组。
[root@xuegod63 ~]# yum groupinstall 'Development tools'　#安装开发工具软件包组。

8.3　CentOS8 中使用 DNF 管理软件包-了解

DNF：Dandified YUM，是基于 RPM 的 Linux 发行版的软件包管理器。它用于在 Fedora / RHEL / CentOS 操作系统中安装，更新和删除软件包。 它是 Fedora 22，CentOS8 和 RHEL8 的默认软件包管理器。 DNF 是 YUM 的下一代版本，并打算在基于 RPM 的系统中替代 YUM。 DNF 功能强大且具有健壮的特征。DNF 使维护软件包组变得容易，并且能够自动解决依赖性问题。

Dandified　[ˈdændɪfaɪd]　打扮时髦; 打扮得华丽的;
注：目前 DNF 命令和 YUM 命令相互兼容，软件包仓库依旧使用 YUM 仓库。
已安装软件包
[root@xuegod63 ~]# dnf list installed

查找软件包
[root@xuegod63 ~]# dnf search httpd

```
[root@xuegod63 ~]# dnf search httpd
Repository AppStream is listed more than once in the configuration
Repository extras is listed more than once in the configuration
Repository PowerTools is listed more than once in the configuration
Repository centosplus is listed more than once in the configuration
上次元数据过期检查：0:05:34 前，执行于 2020年07月02日 星期四 17时21分38秒。
========================== 名称 精准匹配：httpd ==========================
httpd.x86_64 : Apache HTTP Server
httpd.x86_64 : Apache HTTP Server
========================== 名称 和 概况 匹配：httpd ==========================
centos-logos-httpd.noarch : CentOS-related icons and pictures used by httpd
centos-logos-httpd.noarch : CentOS-related icons and pictures used by httpd
keycloak-httpd-client-install.noarch : Tools to configure Apache HTTPD as
                                     : Keycloak client
```

安装软件包

[root@xuegod63 ~]# dnf install httpd -y

卸载软件包

[root@xuegod63 ~]# dnf remove httpd -y

下载软件包

[root@xuegod63 ~]# dnf download httpd

查看软件包信息

[root@xuegod63 ~]# dnf info httpd

检查系统中可更新软件包

[root@xuegod63 ~]# dnf check-update

更新所有软件包

[root@xuegod63 ~]# dnf update

或者

[root@xuegod63 ~]# dnf upgrade

更新指定软件包

[root@xuegod63 ~]# dnf update httpd

列出软件包组

[root@xuegod63 ~]# dnf grouplist

安装软件包组

[root@xuegod63 ~]# dnf groupinstall '开发工具'

更新软件包组

[root@xuegod63 ~]# dnf groupupdate '开发工具'

清空所有缓存

在使用 DNF 的过程中，会因为各种原因在系统中残留各种过时的文件和未完成的编译工程。我们

可以使用该命令来删除这些没用的垃圾文件。并且软件仓库中的软件包依赖也会被清空，再次安装软件时则重新下载软件包依赖信息。

[root@xuegod63 ~]# dnf clean all

[root@xuegod63 ~]# yum clean all

重新创建新的软件包依赖关系

[root@xuegod63 ~]# dnf makecache

或直接

[root@xuegod63 ~]# yum makecache

我喜欢使用 ：

[root@xuegod63 ~]# yum clean all

[root@xuegod63 ~]# yum list　#当清空后， 列出软件列表时，会自动创建新的软件包依赖关系

8.4　实战 tar 源码包管理-源码包安装方法

8.4.1　源码安装 nginx

1. 编译环境如 gcc 和 gcc-c++编译器，make
2. 准备软件 ： nginx-1.18.0.tar.gz

部署 Nginx

安装 nginx 源码编译，需要的依赖包：

[root@xuegod63 ~]# yum -y install gcc gcc-c++ make zlib-devel pcre pcre-devel openssl-devel

软件包说明：

gcc　　c 语言编译器。

gcc-c++　　c++ 语言编译器。

make 用于 configure 和 make 编译的工具。

zlib :nginx 提供 gzip 压缩 模块，需要 zlib 库支持。

pcre 包作用是让 nginx 支持正则表达式，地址重写 rewrite

openssl-devel :让 nginx 提供 ssl 功能。

开始安装：

源码编译 3 把斧：./configure ， make ，make install

[root@xuegod63 ~]# tar xvf nginx-1.18.0.tar.gz

[root@xuegod63 ~]# cd nginx-1.18.0

[root@xuegod63 nginx-1.18.0]# ./configure --prefix=/usr/local/nginx

[root@xuegod63 ~]# make -j 4

[root@xuegod63 ~]# make install

3. 详解源码安装 3 把斧 配置(configure)、编译(make)、安装(make install)

./configure

a. 指定安装路径，例如 --prefix=/usr/local/nginx

b. 启用或禁用某项功能，例如 --enable-ssl,--disable-filter

c. 和其它软件关联，例如--with-pcre --with-http_ssl_module

d. 检查安装环境，例如是否有编译器 gcc，是否满足软件的依赖需求

最终生成：Makefile

make -j 4 #把源代码文件编译成可执行的二进制文件，按 Makefile 文件编译，可以使用-j 4 指定 4 核心 CPU 编译，提升速度

make install #按 Makefile 定义的文件路径安装

make clean //清除上次的 make 命令所产生的 object 和 Makefile 文件。使用场景：当需要重新配置执行 configure 时，需要先执行 make clean。如下：

[root@xuegod63 nginx-1.18.0]# make clean (仅仅清除之前的可执行文件及配置文件)

rm -rf Makefile objs

[root@xuegod63 nginx-1.18.0]# ./configure --prefix=/usr/local/nginx #重新进行配置

8.4.2 删除源码包：

安装完，删除源码包：

有时删除不干净，所以建议大家安装时，在 configure 步骤添加一个： --prefix 参数。这样删除或备份时，直接对删除--prefix 指定的安装目录操作就可以了。

[root@xuegod63 ~]# rm -rf /usr/local/nginx/

8.4.3 实战 2：源码编译出错的 5 种完美解决方法

[root@xuegod63 extundelete-0.2.4]# tar xvf extundelete-0.2.4.tar.bz2

[root@xuegod63 ~]# cd extundelete-0.2.4

#安装则解决依赖。

[root@xuegod63 extundelete-0.2.4]# ./configure #检查系统安装环境

Configuring extundelete 0.2.4

configure: error: Can't find ext2fs library

源码编译出错后，常见解决方法：

共 5 种方法

方法 1：

[root@xuegod63 ~]# mount /dev/sr0 /mnt

[root@xuegod63 Packages]# cd /mnt/BaseOS/Packages

[root@xuegod63 Packages]# rpm -ivh ext2fs

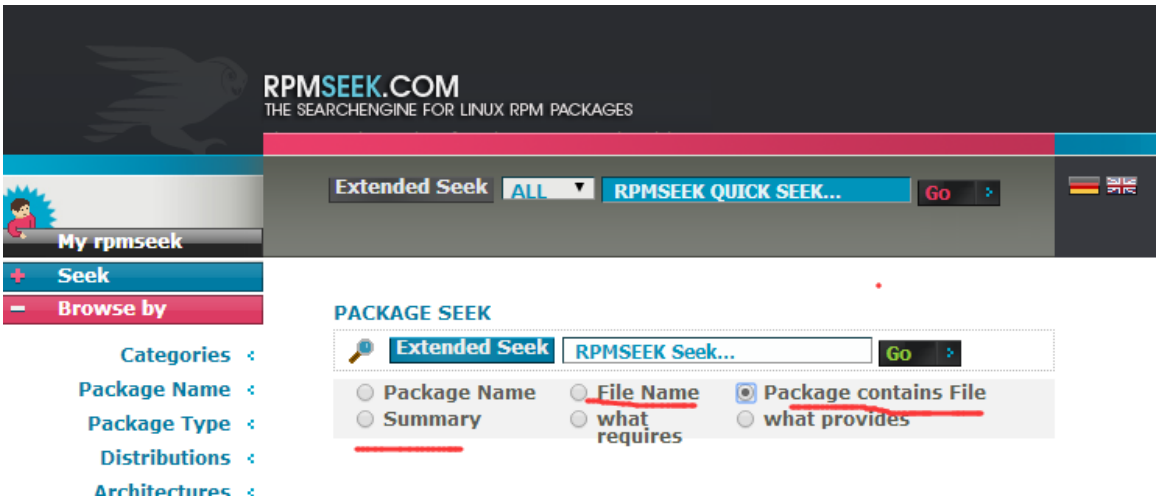
#切到本地软件包所在路径，按两下 tab 键。 一般情况，ext2fs 就是要安装的软件包的名字开头。如果存在 自动补全

方法 2：[root@xuegod63 Packages]# ls *ext2fs* #查找完整关键字

方法 3：[root@xuegod63 Packages]# ls *2fs* #查找部分关键字

方法 4： 终极大招

http://www.rpmseek.com/index.html



方法 5： 使用 yum 去搜索，推荐使用这个方法

[root@xuegod63 Packages]# yum search 2fs　　#centos7 中比较好用

安装库：

[root@xuegod63 ~]# cd /mnt/BaseOS/Packages/

[root@xuegod63 Packages]# rpm -ivh e2fsprogs-libs-1.45.6-1.el8.x86_64.rpm

Verifying...　　################################# [100%]

Preparing...　　################################# [100%]

package e2fsprogs-libs-1.45.6-1.el8.x86_64 is already installed

互动：这里显示库已经安装，但是 configure 时又说找不到。怎么办?

解决：安装了库，却显示找不到。 这种情况： 需要安装库的开发文件

[root@xuegod63 Packages]# rpm -ivh e2fsprogs-devel-1.45.6-1.el8.x86_64.rpm

devel = development (开发)

错误：依赖检测失败：

libcom_err-devel(x86-64) = 1.45.6-1.el8 被 e2fsprogs-devel-1.45.6-1.el8.x86_64 需要

[root@xuegod83 Packages]# ls libcom_err-devel*

[root@xuegod83 Packages]# rpm -ivh libcom_err-devel-1.45.6-1.el8.x86_64.rpm

或

[root@xuegod83 extundelete-0.2.4]# yum -y install e2fsprogs-devel.x86_64

扩展：技巧

查看 rpm 包安装后生成的文件：

[root@xuegod63 Packages]#rpm -qpl e2fsprogs-devel-1.41.12-11.el6.x86_64.rpm | more

```
[root@xuegod63 Packages]# rpm -qpl e2fsprogs-devel-1.45.4-3.el8.x86_64.rpm
/usr/include/e2p
/usr/include/e2p/e2p.h
/usr/include/ext2fs
/usr/include/ext2fs/bitops.h
/usr/include/ext2fs/ext2_err.h
/usr/include/ext2fs/ext2_ext_attr.h
```

注：可以看到很多.h 结尾的文件，这些文件叫头文件。有了这些头文件， ./configure 通过.h 头文件，才能找到对应的库文件。所以库文件和 devel 开发包都要安装。

[root@xuegod63 extundelete-0.2.4]# ./configure

Configuring extundelete 0.2.4

Writing generated files to disk

已经解决了，ext2fs 库文件缺失的问题，这里不进行编译和安装步骤，因为 extundelete 只能在 centos6 下安装。

总结,软件安装方法特点：

rpm + yum：方便，软件版本低。稳定性好、管理方便。性能稍差。

源码编译安装：麻烦，软件版本新，可以定制。稳定性稍差、管理稍差。性能好。

源码编译安装：主要是安装 LAMP 或 LNMP 架构时，我们会用

总结：

8.1 使用 rpm 命令-安装-查看-卸载-rpm 软件包

8.2 yum 管理软件包

8.3 CentOS8 中使用 DNF 管理软件包

8.4 实战 tar 源码包管理-源码包安装方法