Linux Package management

As a server administrator you will need to install different software on your server on different occasion .Most of the Linux operating system(Ubuntu Server/Centos server/Open SUSE server) has two different ways of installing software. First are the software packages that contain the programs that are ready to install and that integrate with the server easily. The server keeps the list of installed packages in the database that makes maintaining very easy. The second option to install software in via tarball. Which basically just an archive of the software. Archive can be anything (can be any record of the data) but it can b also used to deliver software. The first method is proffered most of the time Because server can keep track of the software that are installed via packages .Software installed via tarball are not tracked. There is a second difference between packages and tarballs that some software need other packages for working properly (this is called dependency).both tarball and packages have program installed that check if the dependencies are met but only the software packages interact with the package manager. And in that way it can install the missing dependencies which other installation system cant do. So now a days software packages are preferred. Software packages mostly made in two different formats .On Red Hat and openSUSE and similar distribution rpm packages is used .And debian based operating system like ubuntu server deb package is used.But this packages can be converted. And the other advantage is software can be install by compiling the source code too.

High level and Low level Pckage management Tools

in order to interact with the software packages there are two types of available tools. low level package management also known as local package management system. and the high level tools are known as online package management tools.

Distribution	Low-Level Tools	High Level Tools
Debian based distribution	dpkg	apt/aptitude
Centos/Red Hat	rpm	yum
Open SUSE	rpm	zypper

If you already download or create your own .deb package you can manage it with **dpkg** command.

Installing package with dpkg:

For installing packages with dpkg . command is

=>dpkg -i <package_name>

List of current package:

To list all the current packages that are currently installed in Ubuntu server the command is
=>dpkg -L

it will show the name,version,architecture and a small description

```
vagrant@ubuntu-bionic:~$ sudo dpkg -l | more
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|| Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||| Name                          Version                                      Architecture Description
+++-==============================-============================================-============-=====================================
==================================
ii  accountsservice               0.6.45-1ubuntu1                              amd64        query and manipulate user account informati
on
ii  acl                           2.2.52-3build1                               amd64        Access control list utilities
ii  acpid                         1:2.0.28-1ubuntu1                            amd64        Advanced Configuration and Power Interface
event daemon
ii  adduser                       3.116ubuntu1                                 all          add and remove users and groups
ii  apache2                       2.4.29-1ubuntu4.6                            amd64        Apache HTTP Server
ii  apache2-bin                   2.4.29-1ubuntu4.6                            amd64        Apache HTTP Server (modules and other binar
y files)
ii  apache2-data                  2.4.29-1ubuntu4.6                            all          Apache HTTP Server (common files)
ii  apache2-utils                 2.4.29-1ubuntu4.6                            amd64        Apache HTTP Server (utility programs for we
b servers)
ii  apparmor                      2.12-4ubuntu5.1                              amd64        user-space parser utility for AppArmor
ii  apport                        2.20.9-0ubuntu7.6                            all          automatically generate crash reports for de
bugging
ii  apport-symptoms               0.20                                         all          symptom scripts for apport
ii  apt                           1.6.11                                       amd64        commandline package manager
ii  apt-utils                     1.6.11                                       amd64        package management related utility programs
ii  at                            3.1.20-3.1ubuntu2                            amd64        Delayed job execution and batch processing
ii  base-files                    10.1ubuntu2.4                                amd64        Debian base system miscellaneous files
ii  base-passwd                   3.5.44                                       amd64        Debian base system master password and grou
p files
```

Check packages installation status:

if you need to know any packages installed or not then following command can show if the package installed or not

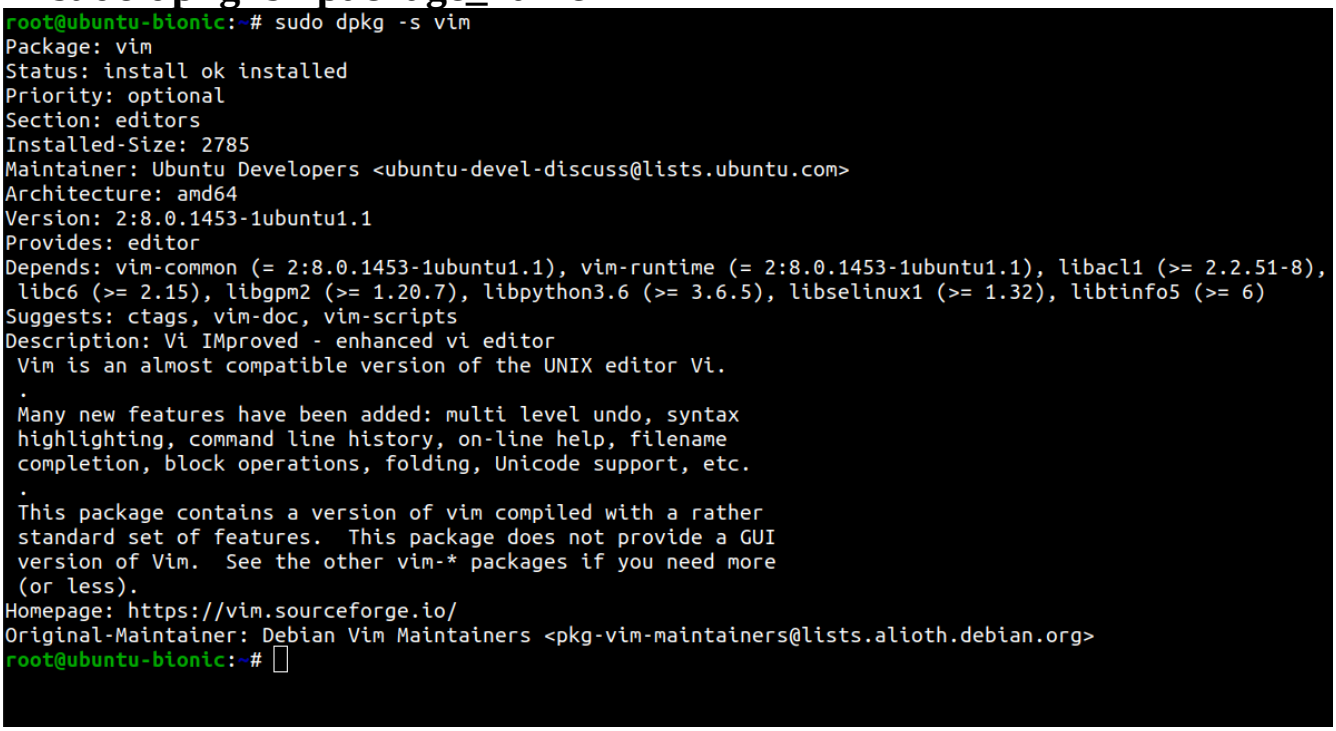
=>dpkg –get-selections <package_name>

```
root@ubuntu-bionic:~# dpkg --get-selections git
git                                             install
root@ubuntu-bionic:~# dpkg --get-selections postgresql
postgresql                                      install
root@ubuntu-bionic:~# dpkg --get-selections java
dpkg: no packages found matching java
root@ubuntu-bionic:~#
```

Check Details information about packages:

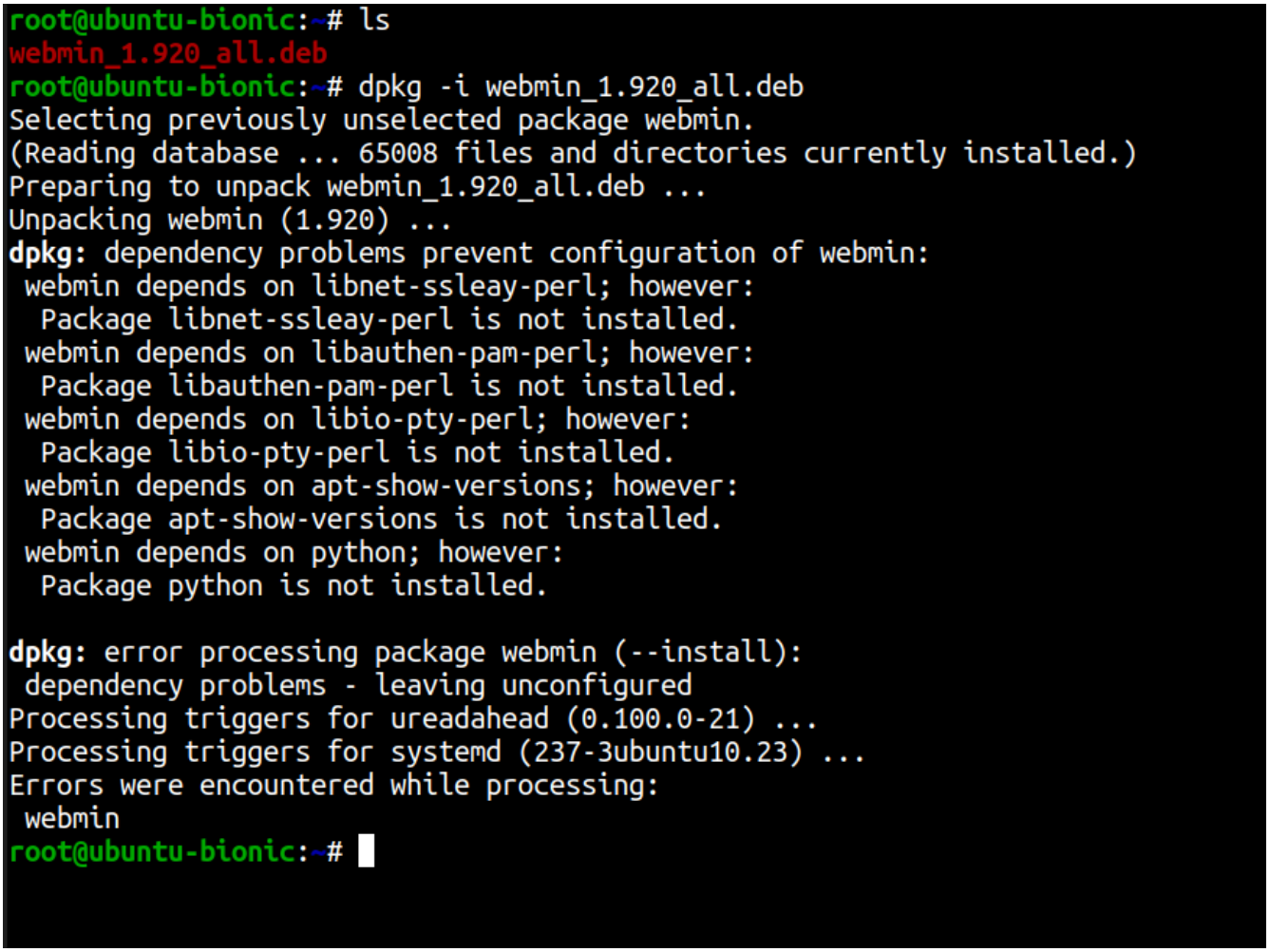
To check details about a installed packages use this command

=>sudo dpkg -s <package_name>



Disadvantage of dpkg:

suppose we want to install a downloaded packages **webmin.deb.** We will show some dependency problem like this and it install the program without the dependency and the program wont run you have to install dependency manually the other dependencies that's a big complexity .If you remove the program it still create the problem if you try to install other program.



[To fix this problem we can use the online package management system
=>sudo apt-get install -f
it will search the dependencies and install them

```
root@ubuntu-bionic:~# apt-get install -f
Reading package lists... Done
Building dependency tree
Reading state information... Done
Correcting dependencies... Done
The following packages were automatically installed and are no longer required:
  dbconfig-common dbconfig-mysql default-mysql-client fontconfig-config fonts-dejavu-core libfontconfig1
  libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libjs-sphinxdoc libjs-underscore libtiff5 libwebp6 libxpm4
  libzip4 php-bz2 php-curl php-gd php-mbstring php-php-gettext php-phpseclib php-tcpdf php-zip php7.3-bz2
  php7.3-curl php7.3-gd php7.3-mbstring php7.3-zip
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  apt-show-versions libapt-pkg-perl libauthen-pam-perl libio-pty-perl libnet-ssleay-perl libpython-stdlib
  libpython2.7-minimal libpython2.7-stdlib perl-openssl-defaults python python-minimal python2.7
  python2.7-minimal
Suggested packages:
  python-doc python-tk python2.7-doc binutils binfmt-support
The following NEW packages will be installed:
  apt-show-versions libapt-pkg-perl libauthen-pam-perl libio-pty-perl libnet-ssleay-perl libpython-stdlib
  libpython2.7-minimal libpython2.7-stdlib perl-openssl-defaults python python-minimal python2.7
  python2.7-minimal
0 upgraded, 13 newly installed, 0 to remove and 15 not upgraded.
1 not fully installed or removed.
Need to get 4401 kB of archives.
After this operation, 18.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

]

Remove packages:

to remove packages from the system this command is used

=>dpkg -r <package_name>

Completely remove package and configuration file:

to completely remove package and the related configuration file this command is used

=>dpkg -P <package_name>

```
root@ubuntu-bionic:~# dpkg -r webmin
(Reading database ... 94784 files and directories currently installed.)
Removing webmin (1.920) ...
Running uninstall scripts ..
Can't locate WebminCore.pm in @INC (you may need to install the WebminCore module) (@INC contains: /etc/perl
 /usr/local/lib/x86_64-linux-gnu/perl/5.26.1 /usr/local/share/perl/5.26.1 /usr/lib/x86_64-linux-gnu/perl5/5.
26 /usr/share/perl5 /usr/lib/x86_64-linux-gnu/perl/5.26 /usr/share/perl/5.26 /usr/local/lib/site_perl /usr/l
ib/x86_64-linux-gnu/perl-base) at /usr/share/webmin/run-uninstalls.pl line 6.
BEGIN failed--compilation aborted at /usr/share/webmin/run-uninstalls.pl line 6.
root@ubuntu-bionic:~# dpkg -p webmin
dpkg-query: package 'webmin' is not available
Use dpkg --info (= dpkg-deb --info) to examine archive files,
and dpkg --contents (= dpkg-deb --contents) to list their contents.
root@ubuntu-bionic:~# dpkg -P webmin
(Reading database ... 66012 files and directories currently installed.)
Purging configuration files for webmin (1.920) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.23) ...
root@ubuntu-bionic:~#
```

If you find a file and want to know which package it belongs to use this command

=>dpkg -S <file_path>

```
root@ubuntu-bionic:~# dpkg -S /bin/cp
coreutils: /bin/cp
root@ubuntu-bionic:~# dpkg -S /bin/cat
coreutils: /bin/cat
root@ubuntu-bionic:~# dpkg -S /bin/ping
iputils-ping: /bin/ping
root@ubuntu-bionic:~#
```

Reconfigure packages:

if you face any problem in your package configuration. You can reconfigure the package with this command

=>dpkg-reconfigure <package_name>

But to do this you need to know the exact name of the package. It will automatically rewind the installation process and give you chance to reconfigure.

```
root@ubuntu-bionic:~# dpkg-reconfigure webmin
Webmin install complete. You can now login to https://ubuntu-bionic:10000/
as root with your root password, or as any user who can use sudo
to run commands as root.
root@ubuntu-bionic:~#
```

Installing packages with apt:

The **apt** utility is a powerful and free package management command line program, that is used to work with Ubuntu's APT (Advanced Packaging Tool) library to perform installation of new software packages, removing existing software packages, upgrading of existing software packages and even used to upgrading the entire operating system

On ubuntu server or any debian based OS there is a list repository url which is populated during the installation in **'/etc/apt/sources.list'** but you can add repository.

Update repository:

Before installing any package you need to update the software repository.
Command
=>sudo apt update
[pic]
[you need to be root to perform the action]

Upgrade existing Software:

To upgrade every package in the latest version use this command

=>sudo apt upgrade

[pic]

Update OS distribution :

to upgrade the distribution for example upgrading ubuntu 16.0 to ubuntu latest version this command is used

=>sudo apt dist-upgrade

Install Packages:

for installing packages this command is used

=>sudo apt install <package_name>

[pic]

for example

to install vim editor we use this command

=>sudo apt install vim

[pic]

Remove Packages:

for removing packages this command is used

=>sudo apt remove <package_name>

[pic]

for example

to remove vim editor we use this command

=>sudo apt remove vim

[pic]

[this command will remove the packages but not the dependencies.To remove this command is used

=>sudo apt autoremove

[pic]

]

apt-cache command

The apt-cache command line tool is used for searching apt software package cache. In simple words, this tool is used to search software packages, collects information of packages and also used to search for what available packages are ready for installation on Ubuntu based systems.

Apt-cache search command:

=>sudo apt-cache search <package_name>
[pic]

This command show all the program will show all the program that depends on the packages. suppose you install gmail packages this command

=>sudo apt-cache search gmail
[pic]
will show all the packages that are depends on this packages like ‘thunderbird’

Package Details:

You can also see the details of any packages with apt just like the **dpkg -s.** command

=>sudo apt-cache show vim
[pic]

Find Unmet Dependencies:

This command will find all the unmet dependencies of the system

=>sudo apt-cache unmet
[pic]

Find Specific Dependency of Packages:

=>sudo apt-cache depends <package_name>
[pic]
This command will give all the dependencies of the Packages.

Find Reverse Dependencies:

=>sudo apt-cache rdepends <package_name>
[pic]
This command will find the reverse dependencies of the programn.That means it will show all the packages that depends on that packages.

For example:
=>sudo apt-cache rdepends git
[pic]
this command will show all the other program that depends on the git program.

Aptitude package management tool:

There is a new package management tools called aptitude. to use that first you have to install it with this command

=>sudo apt install aptitude
[pic]

Install package via aptitude:

installing command with aptitude is

=>sudo aptitude install <package_name>
[pic]

example:
=>sudo aptitude install emacs
[pic]

Search package via aptitude

For searching any packages this command is used

=>sudo apttitude search <package_name>
[pic]
The main advantage of the aptitude is when you run the aptitude program without any flag

=>aptitude
[pic]
this will open a menu based installer inside the terminal. That means you will get almost a gui based installer inside a terminal.

Graphical Package management System:

If you want to use a graphical Package management system you can use synaptic package management software. its very easy to install,remove,and upgrade packages with synaptic package management.

[pic]