OPERATING SYSTEMS
CCGC-5000

btn

Module - 05

Agenda

HUMBER
Faculty of Applied Sciences & Technology

Authentic information is available from the given resources in course outline and URL's mentioned from this slides, and this presentation is only supportive document to be read with the given resources and corrected accordingly if required..

- Listing Packages
- Installed & Available packages
- Creating local repositories
- Installing packages
- Removing packages
- Management of packages
- Installation from source

Must read

- Chapters 9,10 of RHEL8, 2nd Edition book
- RedHat documentation https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/

Package Management

- In Linux distributions, installation of softwares or applications are done through packages
- Debian distributions packages have extension .deb
- **apt** or **apt-get** command requires repository/repositories for installation in Debian distros to download and install packages and its dependencies.
- **dpkg** is package manager for Debian, do not use the repository, do not download and install dependent packages..
- RedHat based distributions packages have extension .rpm
- From RedHat 8, **dnf** is the package manager for rpm linux distros, it requires repositores and installs dependent packages.
- ***dnf*** *is an upgrade of* ***yum*** *the package manager in previous RedHat versions*.
- **rpm** is package manager for rpm linux distros, which does not look into repositories or download and install dependent packages.

Package Management - rpm

- To query all packages : **rpm –qa or rpmquery -a**

```
[user1@rhel ~]$ rpm -qa
qemu-kvm-block-gluster-2.12.0-63.module+el8+2833+c7d6d092.x86_64
boost-atomic-1.66.0-6.el8.x86_64
gnome-session-wayland-session-3.28.1-6.el8.x86_64
grub2-tools-2.02-66.el8.x86_64
```

```
[user1@rhel ~]$ rpmquery -a
qemu-kvm-block-gluster-2.12.0-63.module+el8+2833+c7d6d092.x86_64
boost-atomic-1.66.0-6.el8.x86_64
gnome-session-wayland-session-3.28.1-6.el8.x86_64
grub2-tools-2.02-66.el8.x86_64
```

- To query a package : **rpm –q** *package*
- To list all files in a package : **rpm –ql** *package*
- To list only documentation files in a package : **rpm –qd** *package*
- To identify package associated with the file : **rpm –qf** *file*
- To display information about package : **rpm –qi** *package*
- To install package : **rpm –ivh** *package*
  - *option **i** is to install the package, **v** for detailed information and **h** to show installation progress*
  - *Note in the screenshot below, unlike dnf, rpm command requires the full path of the package to install*
- Removing a package : **rpm –ev** *package*

Package Management - dnf

- A **dnf repository** is a storage where software packages are available and used to download and install when required with appropriate configuration which could be free or available on subscription.
- The configuration file is **/etc/dnf/dnf.conf**
- It is also called as repo and could be used to retrieve package and install, query, update installed packages.
- RHEL8 is distributed to two main repositories BaseOS and AppStream which is available in the ISO installation.
- These two repositories can be created locally configuring repo file with extension .repo in **/etc/yum.repos.d** directory
- The repo can be configured locally or for DVD, FTP, HTTP or NFS sources by setting the **baseurl** directive with two forward slash characters
- *gpgcheck GNU Privacy Guard* (GnuPG or GPG) public key signature for ensuring the credibility of its developer or publisher.

Local dnf (yum) repository

- **dnf** (yum) repository is required for package management.
- To create local repository, two package directories AppStream and BaseOS is required

1. Check where AppStream and BaseOS is copied locally *(if not copied then need to copy locally from iso file)*
   *(in Labs for Vcloud, it is already copied in **/AppStream** and **/BaseOS**)*
2. Create yum configuration file **local.repo** in **/etc/yum.repos.d** as below:

```
[AppStream]
name=AppStream repo
baseurl=file://enter fullpath of AppStream directory in the system
enabled=1
gpgcheck=0

[BaseOS]
name=BaseOS repo
baseurl=file://enter fullpath of BaseOS directory in the system
enabled=1
gpgcheck=0
```

3. Run **dnf -v clean all**
4. Check repository: **dnf -v repolist**

Creating file in linux:

- **vi**, **vim**, **nano**, **gedit** are text editors in linux
- to create a file & save with **vi** or **vim** *(**vim** is **vi** improved)*
  vim *filename*
- Press **ESC**, type **I** *(upper case i to insert text)*
- Now start typing, when completed
- Press **ESC**, type **:x** *(lower case x to save and exit)*
  (do not use upper case X which will encrypt the file, if X is shown just press ESC and then type lower case x)
- Other commands
  :q to quit without saving
  :wq! to write and quit.
- **ESC** key toggles between text and command mode
- Refer to man vi or vim for more options.

Package Management - dnf

- To list the installed packages : **dnf list installed**
- Listing packages available for installation from all configured dnf repos : **dnf list available**
- To list installed packages & available packages for installation from all configured dnf repos : **dnf list**
- To list packages available for update from all configured dnf repos : **dnf list updates**
- To list and find a specific package is installed or available from configured dnf repos : **dnf list** *package*
- To install package : **dnf –y install** *package* (-y option installs without confirmation)
- To remove package : **dnf –y remove** *package*

Package Management - dnf

- To list packages installed using wild card : **dnf list** *name**
- To list recently added packages : **dnf list recent**
- To install package : **dnf –y install *package*** (-y option installs without confirmation)
- To update an installed package : **dnf –y update** *package*
- To update all installed package : **dnf –y update**
- To view header info on package : **dnf info** *package*
- To search for packages of a specific file : **dnf provides** *file*
- To search package for a command : **dnf provides `which** *commandname*`
- To remove package : **dnf –y remove** *package*
- To review the transactions using dnf : **sudo dnf history**
- Alternatively **dnf** can be replaced by **yum**, example **yum install *package***
- Reveiw and familariase other options of dnf command and its purpose

Package Management - Groups

- To list a summary of installed and available groups :

  dnf groups summary

- To list all environment and other groups that are available for installation :

  dnf group list

- To display header information associated with a group :

  dnf group info *group_package*

- To install a group of packages : **dnf –y group install** *group_package*
- To update a group of packages : **dnf –y group update** *group_package*
- To remove group package : **dnf –y group remove** *group_package*
- *Yum cheat sheet :*
  https://access.redhat.com/sites/default/files/attachments/rh_dnf_cheatsheet_1214_jcs_print-1.pdf

Package Management - Modules

- Module Management - A module is a set of RPM packages that represent a component and are usually installed together.
- A typical module contains packages with an application, packages with the application-specific dependency libraries, packages with documentation for the application, and packages with helper utilities.
- Within modules, streams are organized by version and profiles are organized by purpose.
- To list all modules along with their stream, profile and summary information,

  dnf module list
- To list all streams for a specific module such as perl

  dnf module list perl
- To install a module perl : **dnf module install perl**
- To update a module: **dnf module update** *module_name*
- To display module info: **dnf module info --profile** *module_name*
- To remove a module : **dnf module remove** *module_name*:*stream*

Package management : Required reading of Chapter 9,10 of course book

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/installing_managing_and_removing_user-space_components/introduction-to-modules_using-appstream

Installing from source

- Installation is done from source files when they are not available in repositories
- Steps to install
  1. To check compression method
     file packagename
  2. Uncompress
     tar xvf source tarball tgz or tar.gz -C ~/source
     tar jxvg source tarball bz or tar.bz -C ~/source
  3. Check for dependencies
     - From the uncompressed directory, read README file, then run **./configure**, which checks for dependencies
  4. compile the source
     - After configure script is successful, run **make** command
     - Then run **make install**
- In case of failure, need to check for errors, fix it and run **make clean** before starting again
- To remove : **sudo make uninstall**

Refer Chapter-3, RHEL 2nd Edition book for **tar** usage