



Red Hat System Administration I

Dr. Eng. Gamal Selim

Lecture 07

Agenda

- **Chapter 12: Install and Update Software Packages**

Goal	Download, install, update, and manage software packages from Red Hat and DNF package repositories.
Objectives	• Register a system to your Red Hat account and assign it entitlements for software updates and support services with Red Hat Subscription Management. • Explain how software is provided as RPM packages, and investigate the DNF and RPM installed system packages. • Find, install, and update software packages with the <code>dnf</code> command. • Enable and disable server use of Red Hat or third-party DNF repositories.
Sections	• Register Systems for Red Hat Support (and Quiz) • Explain and Investigate RPM Software Packages (and Guided Exercise) • Install and Update Software Packages with DNF (and Guided Exercise) • Enable DNF Software Repositories (and Guided Exercise)
Lab	Install and Update Software Packages

Register Systems for Red Hat Support

Objectives

Register a system to your Red Hat account and assign it entitlements for software updates and support services with Red Hat Subscription Management.

Red Hat Subscription Management

Red Hat Subscription Management provides tools to entitle machines to product subscriptions, for administrators to get updates to software packages and to track information about support contracts and subscriptions that the systems use. Standard tools such as the `dnf` command obtain software packages and updates through a content distribution network that the Red Hat Content Delivery Network provides.

You can perform the following main tasks with the Red Hat Subscription Management tools:

- *Register* a system to associate it with the Red Hat account with an active subscription. With the Subscription Manager, the system can register uniquely in the subscription service inventory. You can unregister the system when it is not in use.
- *Subscribe* a system to entitle it to updates for the selected Red Hat products. Subscriptions have specific levels of support, expiration dates, and default repositories. The tools help to either auto-attach or select a specific entitlement.
- *Enable repositories* to provide software packages. By default, each subscription enables multiple repositories; other repositories such as updates or source code are enabled or disabled. A repository is a central location for storing and maintaining software packages.
- *Review and track* available or consumed entitlements. In the Red Hat Customer Portal, you might view the subscription information locally on a specific system or for a Red Hat account.

Subscribe a System with the Command Line

Use the `subscription-manager` command to register a system without using a graphical environment. The `subscription-manager` command automatically attaches a system to the best-matched compatible subscriptions for the system.

Register a system by using the credentials of the Red Hat Customer Portal as the `root` user:

```
[root@host ~]# subscription-manager register --username <yourusername>
Registering to: subscription.rhsm.redhat.com:443/subscription
Password: yourpassword
The system has been registered with ID: 1457f7e9-f37e-4e93-960a-c94fe08e1b4f
The registered system name is: host.example.com
```

View available subscriptions for your Red Hat account:

```
[root@host ~]# subscription-manager list --available
```

```
-----  
    Available Subscriptions  
-----
```

```
...output omitted...
```

Explain and Investigate RPM Software Packages

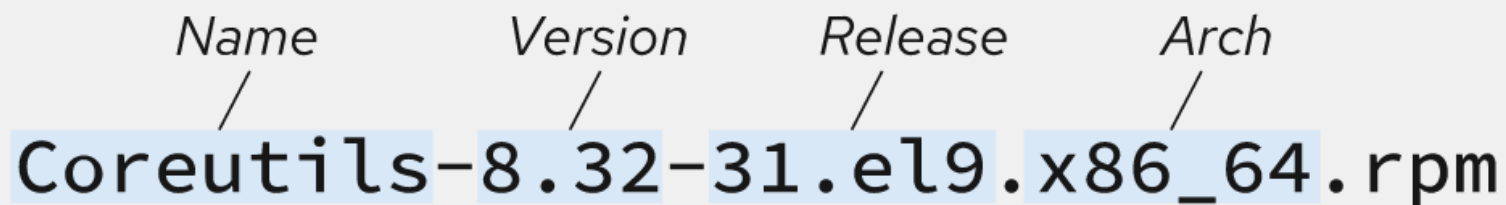
Objectives

Explain how software is provided as RPM packages, and investigate the DNF and RPM installed system packages.

Software Packages and RPM

The RPM Package Manager, which Red Hat originally developed, provides a standard way to package software for distribution. Managing software in the form of RPM packages is simpler than working with software that is extracted to a file system from an archive. With RPM packages, administrators can track which files the software package installs, which files the software package removes if you uninstall it, and it verifies that supporting packages are present when you install it. The local RPM database on your system stores the information about installed packages. Red Hat provides all software for Red Hat Enterprise Linux as an RPM package.

RPM package file names consist of four elements (plus the `.rpm` suffix): `name-version-release.architecture`:



The diagram shows the file name `Coreutils-8.32-31.el9.x86_64.rpm` with labels above each part: *Name* above `Coreutils`, *Version* above `8.32`, *Release* above `31.el9`, and *Arch* above `x86_64`. Each label is connected to its corresponding part of the file name by a diagonal slash (/).

Figure 12.3: RPM file name elements

- NAME is one or more words to describe the contents (`coreutils`).
- VERSION is the version number of the original software (`8.32`).
- RELEASE is the release number of the package based on that version, and is set by the packager, who might not be the original software developer (`31.el9`).
- ARCH is the processor architecture that the package is compiled to run on. The `x86_64` value indicates that this package is built for the 64-bit version of the x86 instruction set (as opposed to `aarch64` for 64-bit ARM, and so on).

RPM packages are often downloaded from repositories. A repository is a central location for storing and maintaining RPM software packages.

You require only the package name to install RPM packages from repositories.

- If multiple versions exist, then the RPM Package Manager installs the package with the later version number.
- If multiple releases of a single version exist, then the RPM Package Manager installs the package with the later release number.

Each RPM package is an archive with the following components:

- The files that the package installs in your system.
- Information about the package (metadata), such as the name, version, release, and architecture; a summary and description of the package; whether it requires other packages to be installed; licensing; a package change log; and other details.
- Scripts that might run when you install, update, or remove the package. These scripts might also run when you install, update, or remove other packages.

Examine RPM Packages

The `rpm` utility is a low-level tool that can retrieve information about the contents of package files and installed packages. By default, the tool gets information from a local database of installed packages. Use the `rpm` command `-p` option to get information about a downloaded but uninstalled package file. Use this option to inspect the package contents before installing.

Retrieve general information about installed packages:

- `rpm -qa` : List all installed packages.
- `rpm -qf FILENAME` : Determine which package provides *FILENAME*.

```
[user@host ~]$ rpm -qf /etc/yum.repos.d  
redhat-release-9.1-1.0.el9.x86_64
```

Get information about specific packages:

- `rpm -q` : List the currently installed package version.

```
[user@host ~]$ rpm -q dnf  
dnf-4.10.0-4.el9.noarch
```

- `rpm -qi` : Get detailed package information.
- `rpm -ql` : List the files that the package installs.

```
[user@host ~]$ rpm -ql dnf  
/usr/bin/dnf  
/usr/lib/systemd/system/dnf-makecache.service  
/usr/lib/systemd/system/dnf-makecache.timer  
/usr/share/bash-completion  
/usr/share/bash-completion/completions  
/usr/share/bash-completion/completions/dnf  
...output omitted...
```

- **rpm -qc** : List only the configuration files that the package installs.

```
[user@host ~]$ rpm -qc openssh-clients
/etc/ssh/ssh_config
/etc/ssh/ssh_config.d/50-redhat.conf
```

- **rpm -qd** : List only the documentation files that the package installs.

```
[user@host ~]$ rpm -qd openssh-clients
/usr/share/man/man1/scp.1.gz
/usr/share/man/man1/sftp.1.gz
/usr/share/man/man1/ssh-add.1.gz
/usr/share/man/man1/ssh-agent.1.gz
...output omitted...
```

- `rpm -q --scripts`: List the shell scripts that run before or after you install or remove the package.

```
[user@host ~]$ rpm -q --scripts openssh-server
preinstall scriptlet (using /bin/sh):
getent group sshd >/dev/null || groupadd -g 74 -r sshd || :
getent passwd sshd >/dev/null || \
    useradd -c "Privilege-separated SSH" -u 74 -g sshd \
    -s /sbin/nologin -r -d /usr/share/empty.sshd sshd 2> /dev/null || :
postinstall scriptlet (using /bin/sh):

if [ $1 -eq 1 ] && [ -x "/usr/lib/systemd/systemd-update-helper" ]; then
    # Initial installation
    /usr/lib/systemd/systemd-update-helper install-system-units sshd.service sshd.socket || :
fi
...output omitted...
```

- `rpm -q --changelog`: List the change log information for the package.

```
[user@host ~]$ rpm -q --changelog audit
* Tue Feb 22 2022 Sergio Correia <scorreia@redhat.com> - 3.0.7-101
- Adjust sample-rules dir permissions
  Resolves: rhbz#2054432 - /usr/share/audit/sample-rules is no longer readable by non-root users

* Tue Jan 25 2022 Sergio Correia <scorreia@redhat.com> - 3.0.7-100
- New upstream release, 3.0.7
  Resolves: rhbz#2019929 - capability=unknown-capability(39) in audit messages
...output omitted...
```

Query local package files:

- `rpm -qlp`: List the files that the local package installs.

```
[user@host ~]$ ls -l podman-4.0.0-6.el9.x86_64.rpm
-rw-r--r--. 1 student student 13755101 Mar 22 11:35 podman-4.0.0-6.el9.x86_64.rpm2637-15.el9.x86_64.rpm
```

```
[user@host ~]$ rpm -qlp podman-4.0.0-6.el9.x86_64.rpm
/etc/cni/net.d
/etc/cni/net.d/87-podman-bridge.conflist
/usr/bin/podman
...output omitted...
```

Install RPM Packages

Use the `rpm` command to install an RPM package that you downloaded to your local directory.

```
[root@host ~]# rpm -ivh podman-4.0.0-6.el9.x86_64.rpm
Verifying...                               ##### [100%]
Preparing...                               ##### [100%]
Updating / installing...
   podman-2:4.0.0-6                         ##### [100%]
```

Warning

Be careful when installing packages from third parties, not just because of the software that the packages might install, but because the RPM package might include arbitrary scripts that run as the `root` user as part of the installation process.

Extracting RPM packages

Use the `rpm2cpio` command to extract files from an RPM package file without installing the package.

The `rpm2cpio` command converts an RPM package to a `cpio` archive. After the RPM package is converted to a `cpio` archive, the `cpio` command can extract a list of files.

Use the `cpio` command with the `-i` option to extract files from standard input. Use the `-d` option to create subdirectories as needed, starting in the current working directory. Use the `-v` option for verbose output.

```
[user@host tmp-extract]$ rpm2cpio httpd-2.4.51-7.el9_0.x86_64.rpm | cpio -idv
./etc/httpd/conf
./etc/httpd/conf.d/autoindex.conf
./etc/httpd/conf.d/userdir.conf
./etc/httpd/conf.d/welcome.conf
./etc/httpd/conf.modules.d
./etc/httpd/conf.modules.d/00-base.conf
./etc/httpd/conf.modules.d/00-dav.conf
./etc/httpd/conf.modules.d/00-mpm.conf
./etc/httpd/conf.modules.d/00-optional.conf
./etc/httpd/conf.modules.d/00-proxy.conf
./etc/httpd/conf.modules.d/00-systemd.conf
./etc/httpd/conf.modules.d/01-cgi.conf
./etc/httpd/conf.modules.d/README
./etc/httpd/conf/httpd.conf
...output omitted...
```

Extract individual files by specifying the path of the file:

```
[user@host ~]$ rpm2cpio httpd-2.4.51-7.el9_0.x86_64.rpm | cpio -id "*/etc/httpd/conf/httpd.conf"
9774 blocks
[user@host ~]$ ls etc/httpd/conf/
httpd.conf
```

Manage Software Packages with DNF

DNF (Dandified YUM) replaced YUM as the package manager in Red Hat Enterprise Linux 9. DNF commands are functionally the same as YUM commands. For compatibility, YUM commands still exist as symbolic links to DNF:

```
[user@host ~]$ ls -l /bin/ | grep yum | awk '{print $9 " " $10 " " $11}'  
yum -> dnf-3  
yum-builddep -> /usr/libexec/dnf-utils  
yum-config-manager -> /usr/libexec/dnf-utils  
yum-debug-dump -> /usr/libexec/dnf-utils  
yum-debug-restore -> /usr/libexec/dnf-utils  
yumdownloader -> /usr/libexec/dnf-utils  
yum-groups-manager -> /usr/libexec/dnf-utils
```

In this course, you work with the `dnf` command. Some documentation might still refer to the `yum` command, but the files are the same linked command.

The low-level `rpm` command can be used to install packages, but it is not designed to work with package repositories or to resolve dependencies from multiple sources automatically.

DNF improves RPM-based software installation and updates. With the `dnf` command, you can install, update, remove, and get information about software packages and their dependencies. You can get a history of transactions and work with multiple Red Hat and third-party software repositories.

Find Software with DNF

The `dnf help` command displays usage information. The `dnf list` command displays installed and available packages.

```
[user@host ~]$ dnf list 'http*'
Available Packages
http-parser.i686                2.9.4-6.el9    rhel-9.0-for-x86_64-appstream-rpms
http-parser.x86_64              2.9.4-6.el9    rhel-9.0-for-x86_64-appstream-rpms
httpcomponents-client.noarch    4.5.13-2.el9   rhel-9.0-for-x86_64-appstream-rpms
httpcomponents-core.noarch      4.4.13-6.el9   rhel-9.0-for-x86_64-appstream-rpms
httpd.x86_64                    2.4.51-5.el9   rhel-9.0-for-x86_64-appstream-rpms
httpd-devel.x86_64              2.4.51-5.el9   rhel-9.0-for-x86_64-appstream-rpms
httpd-filesystem.noarch        2.4.51-5.el9   rhel-9.0-for-x86_64-appstream-rpms
httpd-manual.noarch             2.4.51-5.el9   rhel-9.0-for-x86_64-appstream-rpms
httpd-tools.x86_64              2.4.51-5.el9   rhel-9.0-for-x86_64-appstream-rpms
```

The `dnf search KEYWORD` command lists packages by keywords that are in the name and summary fields only. To search for packages with "web server" in their name, summary, and description fields, use `search all`:

```
[user@host ~]$ dnf search all 'web server'
===== Summary & Description Matched: web server =====
nginx.x86_64 : A high performance web server and reverse proxy server
pcp-pmda-weblog.x86_64 : Performance Co-Pilot (PCP) metrics from web server logs
===== Summary Matched: web server =====
libcurl.x86_64 : A library for getting files from web servers
libcurl.i686 : A library for getting files from web servers
===== Description Matched: web server =====
freeradius.x86_64 : High-performance and highly configurable free RADIUS server
git-instaweb.noarch : Repository browser in gitweb
http-parser.i686 : HTTP request/response parser for C
http-parser.x86_64 : HTTP request/response parser for C
httpd.x86_64 : Apache HTTP Server
mod_auth_openidc.x86_64 : OpenID Connect auth module for Apache HTTP Server
mod_jk.x86_64 : Tomcat mod_jk connector for Apache
mod_security.x86_64 : Security module for the Apache HTTP Server
varnish.i686 : High-performance HTTP accelerator
varnish.x86_64 : High-performance HTTP accelerator
...output omitted...
```

The `dnf info PACKAGENAME` command returns detailed information about a package, including the needed disk space for installation. For example, the following command retrieves information about the `httpd` package:

```
[user@host ~]$ dnf info httpd
Available Packages
Name           : httpd
Version        : 2.4.51
Release        : 5.el9
Architecture   : x86_64
Size           : 1.5 M
Source         : httpd-2.4.51-5.el9.src.rpm
Repository     : rhel-9.0-for-x86_64-appstream-rpms
Summary        : Apache HTTP Server
URL            : https://httpd.apache.org/
License        : ASL 2.0
Description    : The Apache HTTP Server is a powerful, efficient, and extensible
                  : web server.
```

The `dnf provides PATHNAME` command displays packages that match the specified path name (the path names often include wildcard characters). For example, the following command finds packages that provide the `/var/www/html` directory:

```
[user@host ~]$ dnf provides /var/www/html
httpd-filesystem-2.4.51-5.el9.noarch : The basic directory layout for the Apache HTTP Server
Repo          : rhel-9.0-for-x86_64-appstream-rpms
Matched from:
Filename      : /var/www/html
```

Install and Remove Software with DNF

The `dnf install PACKAGENAME` command obtains and installs a software package, including any dependencies.

```
[root@host ~]# dnf install httpd
Dependencies resolved.
=====
Package            Arch    Version      Repository                                Size
=====
Installing:
httpd              x86_64  2.4.51-5.el9  rhel-9.0-for-x86_64-appstream-rpms      1.5 M
Installing dependencies:
apr                x86_64  1.7.0-11.el9  rhel-9.0-for-x86_64-appstream-rpms      127 k
apr-util           x86_64  1.6.1-20.el9  rhel-9.0-for-x86_64-appstream-rpms       98 k
apr-util-bdb       x86_64  1.6.1-20.el9  rhel-9.0-for-x86_64-appstream-rpms       15 k
httpd-filesystem   noarch  2.4.51-5.el9  rhel-9.0-for-x86_64-appstream-rpms       17 k
httpd-tools        x86_64  2.4.51-5.el9  rhel-9.0-for-x86_64-appstream-rpms       88 k
redhat-logos-httpd
                  noarch  90.4-1.el9    rhel-9.0-for-x86_64-appstream-rpms       18 k
Installing weak dependencies:
apr-util-openssl   x86_64  1.6.1-20.el9  rhel-9.0-for-x86_64-appstream-rpms       17 k
mod_http2          x86_64  1.15.19-2.el9 rhel-9.0-for-x86_64-appstream-rpms      153 k
mod_lua            x86_64  2.4.51-5.el9  rhel-9.0-for-x86_64-appstream-rpms       63 k

Transaction Summary
=====
Install 10 Packages

Total download size: 2.1 M
```

The `dnf update PACKAGENAME` command obtains and installs a later version of the specified package, including any dependencies. Generally, the process tries to preserve configuration files in place, but in some cases, those files might be renamed if the packager considers that the earlier name will not work after the update. If no `PACKAGENAME` is specified, then it installs all relevant updates.

```
[root@host ~]# dnf update
```

Because a new kernel can be tested only by booting to that kernel, the package specifically supports the installation of multiple versions at once. If the new kernel fails to boot, then the earlier kernel is still available. Running the `dnf update kernel` command installs the new kernel. The configuration files hold a list of packages to always install even if the administrator requests an update.

Use the `dnf list kernel` command to list all installed and available kernels. To view the currently running kernel, use the `uname` command. The `uname` command `-r` option shows only the kernel version and release, and the `uname` command `-a` option shows the kernel release and additional information.

Use the `dnf list kernel` command to list all installed and available kernels. To view the currently running kernel, use the `uname` command. The `uname` command `-r` option shows only the kernel version and release, and the `uname` command `-a` option shows the kernel release and additional information.

```
[user@host ~]$ dnf list kernel
Installed Packages
kernel.x86_64                5.14.0-70.el9                @System
[user@host ~]$ uname -r
5.14.0-70.el9.x86_64
[user@host ~]$ uname -a
Linux workstation.lab.example.com 5.14.0-70.el9.x86_64 #1 SMP PREEMPT Thu Feb 24 19:11:22 EST 2022 x86_64 x86_64 x86_64 GNU/Linux
```

The `dnf remove PACKAGENAME` command removes an installed software package, including any supported packages.

```
[root@host ~]# dnf remove httpd
```

Warning

The `dnf remove` command removes the listed packages *and any package that requires the packages to be removed* (and packages which require those packages, and so on). This command can lead to unexpected removal of packages, so review the list of packages to be removed.

Install and Remove Groups of Software with DNF

The `dnf` command also has the concept of *groups*, which are collections of related software that are installed together.

In Red Hat Enterprise Linux 9, the `dnf` command can install two kinds of package groups. Regular groups are collections of packages. Environment groups are collections of regular groups. The packages or groups that these collections provide might be listed as `mandatory` (they must be installed if the group is installed), `default` (normally installed if the group is installed), or `optional` (not installed when the group is installed, unless specifically requested).

Similar to the `dnf list` command, the `dnf group list` command shows the names of installed and available groups.

```
[user@host ~]$ dnf group list
Available Environment Groups:
  Server with GUI
  Server
  Minimal Install
...output omitted...
Available Groups:
  Legacy UNIX Compatibility
  Console Internet Tools
  Container Management
...output omitted...
```

The `dnf group info` command displays information about a group. It includes a list of mandatory, default, and optional package names.

```
[user@host ~]$ dnf group info "RPM Development Tools"
Group: RPM Development Tools
Description: Tools used for building RPMs, such as rpmbuild.
Mandatory Packages:
  redhat-rpm-config
  rpm-build
Default Packages:
  rpmdevtools
Optional Packages:
  rpmlint
```

The `dnf group install` command installs a group that installs its mandatory and default packages and their dependent packages.

```
[root@host ~]# dnf group install "RPM Development Tools"
```

```
...output omitted...
```

```
Installing Groups:
```

```
  RPM Development Tools
```

```
Transaction Summary
```

```
=====
```

```
Install 19 Packages
```

```
Total download size: 4.7 M
```

```
Installed size: 15 M
```

```
Is this ok [y/N]: y
```

```
...output omitted...
```

View Transaction History

All installation and removal transactions are logged in the `/var/log/dnf.rpm.log` file.

```
[user@host ~]$ tail -5 /var/log/dnf.rpm.log
2022-03-23T16:46:43-0400 SUBDEBUG Installed: python-srpm-macros-3.9-52.el9.noarch
2022-03-23T16:46:43-0400 SUBDEBUG Installed: redhat-rpm-config-194-1.el9.noarch
2022-03-23T16:46:44-0400 SUBDEBUG Installed: elfutils-0.186-1.el9.x86_64
2022-03-23T16:46:44-0400 SUBDEBUG Installed: rpm-build-4.16.1.3-11.el9.x86_64
2022-03-23T16:46:44-0400 SUBDEBUG Installed: rpmdevtools-9.5-1.el9.noarch
```

The `dnf history` command displays a summary of installation and removal transactions.

```
[root@host ~]# dnf history
```

ID	Command line	Date and time	Action(s)	Altered
7	group install RPM Develop	2022-03-23 16:46	Install	20
6	install httpd	2022-03-23 16:21	Install	10 EE
5	history undo 4	2022-03-23 15:04	Removed	20
4	group install RPM Develop	2022-03-23 15:03	Install	20
3		2022-03-04 03:36	Install	5
2		2022-03-04 03:33	Install	767 EE
1	-y install patch ansible-	2022-03-04 03:31	Install	80

The `dnf history undo` command reverses a transaction.

```
[root@host ~]# dnf history undo 6
...output omitted...
Removing:
apr-util-openssl x86_64 1.6.1-20.el9 @rhel-9.0-for-x86_64-appstream-rpms 24 k
httpd             x86_64 2.4.51-5.el9 @rhel-9.0-for-x86_64-appstream-rpms 4.7 M
...output omitted...
```

Summary of DNF Commands

Packages can be located, installed, updated, and removed by name or by package groups.

Task:	Command:
List installed and available packages by name.	<code>dnf list [NAME-PATTERN]</code>
List installed and available groups.	<code>dnf group list</code>
Search for a package by keyword.	<code>dnf search KEYWORD</code>
Show details of a package.	<code>dnf info PACKAGENAME</code>
Install a package.	<code>dnf install PACKAGENAME</code>
Install a package group.	<code>dnf group install GROUPNAME</code>
Update all packages.	<code>dnf update</code>
Remove a package.	<code>dnf remove PACKAGENAME</code>
Display transaction history.	<code>dnf history</code>

Introduction to BaseOS and Application Stream

Red Hat Enterprise Linux 9 distributes the content through two main software repositories: *BaseOS* and *Application Stream* (AppStream).

The BaseOS repository provides the core operating system content for Red Hat Enterprise Linux as RPM packages. BaseOS components have the same lifecycle as content in previous Red Hat Enterprise Linux releases. The Application Stream repository provides content with varying lifecycles as both modules and traditional packages.

Application Stream contains necessary parts of the system, as well as a wide range of applications that were previously available as part of Red Hat Software Collections and other products and programs. Each Application Stream has a lifecycle that is either the same as Red Hat Enterprise Linux 9 or shorter.

Both BaseOS and AppStream are necessary parts of a Red Hat Enterprise Linux 9 system.

Module Streams

Each module has one or more *module streams*, which hold different versions of the content. Each of the streams receives updates independently. Think of the module stream as a virtual repository in the Application Stream physical repository.

For each module, you can enable only one of its streams, and this stream provides its packages.

Module Profiles

Each module can have one or more profiles. A profile is a list of packages that you can install together for a particular use case, such as for a server, client, development, minimal installation, or other.

Installing a module profile installs a particular set of packages from the module stream. You can subsequently install or uninstall packages normally. If you do not specify a profile, then the module installs its default profile.

Manage Modules with DNF

Red Hat Enterprise Linux 9 supports modular features of Application Stream. To handle the modular content, you can use the `dnf module` command. Otherwise, the `dnf` command works with similar modules to regular packages.

See the following list for some important commands to manage modules:

- `dnf module list`: List the available modules with the module name, stream, profiles, and a summary.
- `dnf module list module-name`: List the module streams for a specific module and retrieve their status.
- `dnf module info module-name`: Display details of a module, including the available profiles and a list of the packages that the module installs. Running the `dnf module info` command without specifying a module stream lists the packages that are installed from the default profile and stream. Use the `module-name:stream` format to view a specific module stream. Add the `--profile` option to display information about packages that each of the module's profiles installed.
- `dnf module provides package`: Display which module provides a specific package.

Enable DNF Software Repositories

Objectives

Enable and disable server use of Red Hat or third-party DNF repositories.

Enable Red Hat Software Repositories

Systems often have access to many Red Hat repositories. The `dnf repolist all` command lists all available repositories and their statuses:

```
[user@host ~]$ dnf repolist all
```

repo id	repo name	status
rhel-9.0-for-x86_64-appstream-rpms	RHEL 9.0 AppStream	enabled
rhel-9.0-for-x86_64-baseos-rpms	RHEL 9.0 BaseOS	enabled

The `dnf config-manager` command can enable and disable repositories. For example, the following command enables the `rhel-9-server-debug-rpms` repository:

```
[user@host ~]$ dnf config-manager --enable rhel-9-server-debug-rpms
```

Add DNF Repositories

The `dnf config-manager` command can also add repositories to the machine. The following command creates a `.repo` file by using an existing repository's URL.

```
[user@host ~]$ dnf config-manager \
--add-repo="https://dl.fedoraproject.org/pub/epel/9/Everything/x86_64/"
Adding repo from: https://dl.fedoraproject.org/pub/epel/9/Everything/x86_64/
```

The corresponding `.repo` file is visible in the `/etc/yum.repos.d/` directory:

```
[user@host ~]$ cd /etc/yum.repos.d
[user@host yum.repos.d]$ cat \
dl.fedoraproject.org_pub_epel_9_Everything_x86_64_.repo
[dl.fedoraproject.org_pub_epel_9_Everything_x86_64_]
name=created by dnf config-manager from https://dl.fedoraproject.org/pub/epel/9/Everything/x86_64/
baseurl=https://dl.fedoraproject.org/pub/epel/9/Everything/x86_64/
enabled=1
```

RPM Configuration Packages for Local Repositories

Some repositories provide a configuration file and a GPG public key as part of an RPM package to simplify their installation. You can import the GPG public key by using the `rpm --import` command. The `dnf install` command can download and install these RPM packages.

For example, the following command imports the `RPM-GPG-KEY-EPEL-9` (EPEL) GPG public key and installs the RHEL9 Extra Packages for Enterprise Linux (EPEL) repository RPM:

```
[user@host ~]$ rpm --import \  
https://dl.fedoraproject.org/pub/epel/RPM-GPG-KEY-EPEL-9  
[user@host ~]$ dnf install \  
https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

The `.repo` files often list multiple repository references in a single file. Each repository reference begins with a single-word name in square brackets.

```
[user@host ~]$ cat /etc/yum.repos.d/epel.repo
[epel]
name=Extra Packages for Enterprise Linux $releasever - $basearch
#baseurl=https://download.example/pub/epel/$releasever/Everything/$basearch/
metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-$releasever&arch=$basearch&infra=$infra&content=$contentdir
enabled=1
gpgcheck=1
countme=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-$releasever
...output omitted...
[epel-source]
name=Extra Packages for Enterprise Linux $releasever - $basearch - Source
#baseurl=https://download.example/pub/epel/$releasever/Everything/source/tree/
metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-source-$releasever&arch=$basearch&infra=$infra&content=$contentdir
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-$releasever
gpgcheck=1
```

Summary

- Red Hat Subscription Management provides tools to entitle machines to product subscriptions, get updates to software packages, and track information about support contracts and subscriptions that the systems use.
- Software is provided as RPM packages, to install, upgrade, and uninstall software on the system.
- The `rpm` command can query a local database to provide information about the contents of installed packages and to install downloaded package files.
- The `dnf` utility is a powerful command-line tool to install, update, remove, and query software packages.
- Red Hat Enterprise Linux uses Application Streams to provide a single repository to host multiple versions of an application's packages and its dependencies.

End of Chapter 12

*Thank
you*

