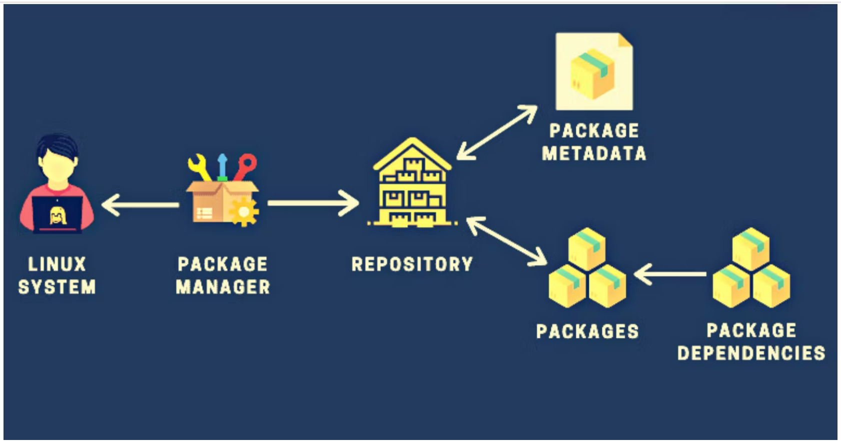


Day 7 : Understanding package manager and systemctl

Kunal Maurya · Mar 1, 2023 · 5 min read

TABLE OF CONTENTS

Linux is a popular operating system that is known for its flexibility and customization. One of the key components of Linux that makes it so powerful is its package management system.In this blog post, we'll dive deeper into how package managers work in Linux and the benefits they provide.

What is a package manager?

A package manager is a software tool that helps you install, update, and remove software packages on your Linux system. It provides an easy way to manage software on your system, by handling all the dependencies and configurations for you.

How do Package Managers Work?

Package managers in Linux work by downloading software packages from a central repository. A repository is a collection of software packages that are maintained by a specific organization or community. Each package in the repository is associated with metadata that includes information about the package, such as its name, version, and dependencies.

When you use a package manager to install software, it checks the repository for the package and downloads it along with any dependencies that the software requires. The package manager then installs the software and sets up any necessary configurations.

Different Package Managers in Linux

There are several package managers available in Linux, each with its own strengths and weaknesses. Some of the most popular package managers include:

1. **APT**: Advanced Package Tool (APT) is a package manager used by Debian, Ubuntu, and other Debian-based distributions.
2. **YUM**: Yellowdog Updater, Modified (YUM) is a package manager used by Red Hat, CentOS, and other Red Hat-based distributions.
3. **Pacman**: Pacman is a package manager used by Arch Linux and its derivatives.
4. **Zypper**: Zypper is a package manager used by SUSE Linux and its derivatives.

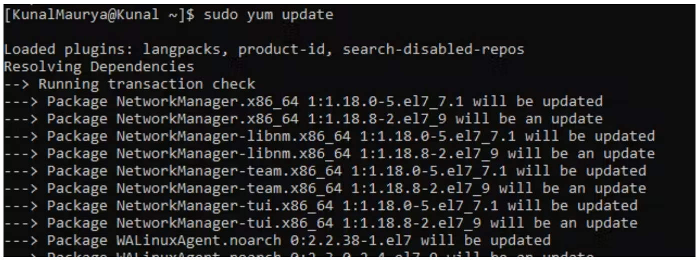
YUM Commands

Red Hat 7.7 uses the YUM (Yellowdog Updater Modified) package manager.

- Update the package list:

```
sudo yum update
```

This command updates the package list on your system. It checks the Red Hat repositories for updated packages and installs them.



```
[KunalMaurya@Kunal ~]$ sudo yum update

Loaded plugins: langpacks, product-id, search-disabled-repos
Resolving Dependencies
--> Running transaction check
---> Package NetworkManager.x86_64 1:1.18.0-5.el7_7.1 will be updated
---> Package NetworkManager.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package NetworkManager-libnm.x86_64 1:1.18.0-5.el7_7.1 will be updated
---> Package NetworkManager-libnm.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package NetworkManager-team.x86_64 1:1.18.0-5.el7_7.1 will be updated
---> Package NetworkManager-team.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package NetworkManager-tui.x86_64 1:1.18.0-5.el7_7.1 will be updated
---> Package NetworkManager-tui.x86_64 1:1.18.8-2.el7_9 will be an update
---> Package WALinuxAgent.noarch 0:2.2.38-1.el7 will be updated
---> Package WALinuxAgent.noarch 0:2.2.0.2.4.el7_9 will be an update
```

- Install a package:

```
sudo yum install PACKAGE-NAME
```

This command installs the specified package and any dependencies that it requires.

- Remove a package:

```
sudo yum remove PACKAGE-NAME
```

- Check for package updates:

```
sudo yum check-update
```

This command checks for available package updates without installing them.

What is systemd?

`systemd` is a system and service manager that provides a central way to manage system services and processes. It replaces the traditional init system used by most Linux distributions.

`systemd` provides several benefits, including:

- Faster boot times
- Improved process management
- Improved service management
- Better logging and monitoring capabilities
- Enhanced security features

`systemd` is designed to be backwards compatible with the traditional init system, so most of the commands and tools you're familiar with (such as `service`, `chkconfig`, and `init`) can still be used.

What is systemctl?

`systemctl` is a command-line tool used to control the systemd system and service manager. It allows you to manage and control various aspects of the system's services, including starting, stopping, enabling, disabling, and reloading services.

Some common commands that you can use with `systemctl` include:

- `systemctl start <service>`: Starts a service
- `systemctl stop <service>`: Stops a service
- `systemctl restart <service>`: Restarts a service
- `systemctl enable <service>`: Enables a service to start automatically at boot
- `systemctl disable <service>`: Disables a service from starting automatically at boot
- `systemctl status <service>`: Displays the status of a service

What is service command?

The `service` command in Linux is used to manage system services. It allows you to start, stop, restart, enable, disable, and check the status of services that are managed by the system's init system.

The basic syntax for using the `service` command is:

```
service <service-name> <action>
```

where `<service-name>` is the name of the service you want to manage and `<action>` is the action you want to perform on the service.

Here are some common actions that you can perform with the `service` command:

- `start`: Starts a service
- `stop`: Stops a service
- `restart`: Restarts a service
- `reload`: Reloads the configuration of a service
- `status`: Displays the status of a service
- `enable`: Enables a service to start automatically at boot
- `disable`: Disables a service from starting automatically at boot

Note that the `service` command is being phased out in favor of `systemctl` and `systemd` on newer Linux distributions. However, it is still supported on many systems and can be a useful tool for managing services.

Install Docker using YUM package managers

1. Install Docker using the YUM package manager:

    ```
    sudo yum install docker
    ```

2. Start the Docker service:

    ```
    sudo systemctl start docker
    ```

3. Verify that Docker is running:

    ```
    sudo systemctl status docker
    ```

Install Jenkins using YUM package managers

Prerequisites for installing Jenkins:

- Java needs to be installed and configured on the server on which you want to configure Jenkins.

    ```
     sudo yum install java-11-openjdk-devel
    ```

- Install the `wget` tool in your operating system to fetch the Jenkins repository:

```
sudo yum install wget
```

You can enable the Jenkins repository by following these steps:

- Add the Jenkins repository to the YUM package manager:

```
sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/red
```

- Import the Jenkins GPG key:

  ```
  sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.
  ```

- Install Jenkins using the YUM package manager:

  ```
  sudo yum install jenkins
  ```

- **Start the Jenkins service:**

  ```
  sudo systemctl start jenkins
  ```

- **Verify that Jenkins is running:**

  ```
  sudo systemctl status jenkins
  ```

- **Stop the Jenkins service:**

  ```
  sudo systemctl stop jenkins
  sudo systemctl status jenkins
  ```

Thank you for reading! Hope you find this article helpful.

~Kunal