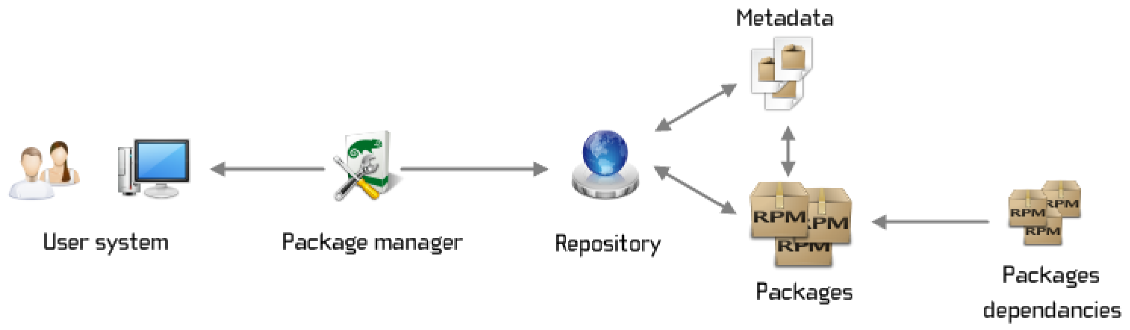
Day 7 Task: Understanding package manager and systemctl

What is a package manager in Linux?

A package manager is a group of software tools and process of handling the many and varied dependencies and artifacts for your servers, applications, and developers. It automates the installation process, upgrading process, configuration process, and removing process of the computer programs for an operating system of the computer in an efficient manner. These are the archives, binaries, libraries, tools, scripts, modules, snippets, metadata, assets and even datasets that power your processes, products, and solutions. A package manager works with packages, data within archive files, and software distributions.



What is a package?

A package is usually referred to an application but it could be a GUI application, command line tool or a software library. A package is essentially an archive file containing the binary executable, configuration file and sometimes information about the dependencies.

Common types of Linux packages include .deb, .rpm, and .tgz. Since Linux packages do not usually contain the dependencies necessary to install them, many Linux distributions use package managers that automatically read dependencies files and download the packages needed before proceeding with the installation. Some examples of package managers are APT, YUM, and Pacman.

Different kinds of package managers

There are many different kinds of package managers available but the matters is to install a new software onto a system we should able to know which OS is used.

Package Managers differ based on packaging system but same packaging system may have more than one package manager.

- apt-get package manager
- dnf package manage
- yum package manager

Tasks

1. **You have to install docker in your system from your terminal using package managers**

   Ubuntu:-

   Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

   Set up the repository

   1. Update the apt package index and install packages to a allow apt to use a repository over HTTPS:

      ```
      sudo apt-get update
      sudo apt-get install \
          ca-certificates \
          curl \
          gnupg \
          lsb-release
      ```

   2. Add Docker's official GPG key:

   ```
   sudo mkdir -p /etc/apt/keyrings
   curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
   /etc/apt/keyrings/docker.gpg
   ```

   3. Use the following command to set up the repository:

      ```
      echo \
        "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
      https://download.docker.com/linux/ubuntu \
        $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
      ```

      After setting up docker repository, try to install docker engine.

1. Update the apt package index

   ```
   sudo apt-get update
   ```

2. Install the docker engine,container and docker-compose

   ```
   Sudo apt-get install docker-ce docker-ce-cli containerd,io docker-compose-plugin
   ```

3. Verfiy docker installation by running hello-world image

   ```
   Sudo docker run hello-world
   ```

2. **You have to install jenkins in your system from your terminal using package managers**

   It is recommended to install Jenkins using the project-maintained repository, rather than from the default Ubuntu repository.

   Set up repository

   1. Start by importing the GPG key. The GPG key verifies package integrity but there is no output. Run:

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee
/usr/share/keyrings/jenkins-keyring.asc &gt; /dev/null
```

2. Add the Jenkins software repository to the source list and provide the authentication key:

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]
https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list
&gt; /dev/null
```

After setting up Jenkins repository, try to install Jenkins engine.

1. Update the system repository one more time. Updating refreshes the cache and makes the system aware of the new Jenkins repository.

```
sudo apt update
```

2. Install Jenkins by running:

```
sudo apt install jenkins -y
```

3. To check if Jenkins is installed and running, run the following command:

```
sudo systemctl status Jenkins
```

systemctl and systemd

systemctl is used to examine and control the state of “systemd” system and service manager. systemd is system and service manager for Unix like operating systems(most of the distributions, not all).

The *systemctl* command interacts with the SystemD service manager to manage the services. Contrary to *service* command, it manages the services by interacting with the SystemD process instead of running the init script.