Day 7: Understanding package manager and systemctl

This is #90DaysofDevops challenge under the guidance of Shubham Londhe sir.

Day 7 TASK

check this for task:

https://github.com/LondheShubham153/90DaysOfDevOps/blob/master/2023/day07/tasks.md

What is a package manager in Linux?

In simpler words, a package manager is a tool that allows users to install, remove, upgrade, configure and manage software packages on an operating system. The package manager can be a graphical application like a software center or a command line tool like apt-get, YUM or pacman.

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 (required by other software programs). A package is essentially an archive file containing the binary executable, configuration file and sometimes information about the dependencies.

Different kinds of package managers

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

For example, RPM has Yum and DNF package managers. For DEB, you have apt-get, aptitude command line based package managers.

Task 1 : You have to install docker and jenkins in your system from your terminal using package managers

Commands used to install Docker and Manage Docker :

1. Remove any Docker files that are running in the system, using the following command:

```
$ sudo apt-get remove docker docker-engine docker.io
```

2. Check if the system is up-to-date using the following command:

```
$ sudo apt-get update
```

3. Install Docker using the following command:

```
$ sudo apt install docker.io
```

```
ubuntu@ip-172-31-44-140:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
```

4. Install all the dependency packages using the following command:

```
$ sudo snap install docker
```

```
ubuntu@ip-172-31-44-140:~$ sudo snap install docker
docker 20.10.17 from Canonical✓ installed
```

5. Before testing Docker, check the version installed using the following command:

```
$ docker --version
```

```
ubuntu@ip-172-31-44-140:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
```

6 . Pull an image from the Docker hub using the following command:

```
$ sudo docker run hello-world
```

Here, *hello-world* is the docker image present on the Docker hub.

7. Check if the docker image has been pulled and is present in your system using the following command:

```
$ sudo docker images
```

```
root@ip-172-31-44-140:/home/ubuntu# docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    feb5d9fea6a5   16 months ago   13.3kB
```

8. To display all the containers pulled, use the following command:

```
$ sudo docker ps -a
```

```
root@ip-172-31-44-140:/home/ubuntu# sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED              STATUS                          PORTS     NAMES
a2171a7545da   hello-world   "/hello"   27 seconds ago       Exited (0) 26 seconds ago                 awesome_meitner
585174eb63ec   hello-world   "/hello"   47 seconds ago       Exited (0) 46 seconds ago                 elegant_wilbur
0101816d12c7   hello-world   "/hello"   About a minute ago   Exited (0) About a minute ago             hungry_herschel
```

9. To check for containers in a running state, use the following command:

```
$ sudo docker ps
```

```
root@ip-172-31-44-140:/home/ubuntu# docker run -d  --name raj1 hello-world:latest
c9b739720b8383d0b6548c8646b54ea3ed5c68a31352e87eba67c55035e5cf6b
root@ip-172-31-44-140:/home/ubuntu# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-44-140:/home/ubuntu# docker ps -a
CONTAINER ID   IMAGE                COMMAND    CREATED          STATUS                      PORTS     NAMES
c9b739720b83   hello-world:latest   "/hello"   10 seconds ago   Exited (0) 9 seconds ago              raj1
6af93db2bc87   hello-world:latest   "/hello"   2 minutes ago    Exited (0) 2 minutes ago              raj
a2171a7545da   hello-world          "/hello"   6 minutes ago    Exited (0) 6 minutes ago              awesome_meitner
585174eb63ec   hello-world          "/hello"   6 minutes ago    Exited (0) 6 minutes ago              elegant_wilbur
0101816d12c7   hello-world          "/hello"   7 minutes ago    Exited (0) 7 minutes ago              hungry_herschel
```

You've just successfully installed Docker on Ubuntu!

Install Jenkins:-

Check the below website for step by step installation of Jenkins on your Ubuntu server.

https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-22-04

What is systemctl and systemd ?

`systemd` is a Linux init system and system manager that is widely used in modern Linux distributions as the default init system. It provides a way to manage and control the various services that run on a Linux system, as well as other system-level functionality.

`systemctl` is the command line tool used to control and manage the `systemd` system and service manager. It provides various commands to start, stop, restart, enable, and disable services, as well as other functionalities such as inspecting the status of services, displaying log messages, and managing system-level settings and configurations.

Here are a few examples of common tasks that can be performed using `systemctl`:

- Start a service: `systemctl start <service-name>`
- Stop a service: `systemctl stop <service-name>`
- Restart a service: `systemctl restart <service-name>`
- Enable a service to start automatically at boot: `systemctl enable <service-name>`

- Disable a service from starting automatically at boot: `systemctl disable <service-name>`
- Check the status of a service: `systemctl status <service-name>`

Overall, `systemd` and `systemctl` provide a centralized and standardized way to manage services and other system-level components on Linux, making it easier to configure and maintain a Linux system.

1. **check the status of docker service in your system (make sure you completed above tasks, else docker won't be installed)**

```
Last login: Sun Feb 12 14:58:02 2023 from 3.16.146.5
ubuntu@ip-172-31-44-140:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu4
```

2. stop the service jenkins and post before and after screenshots

Before :

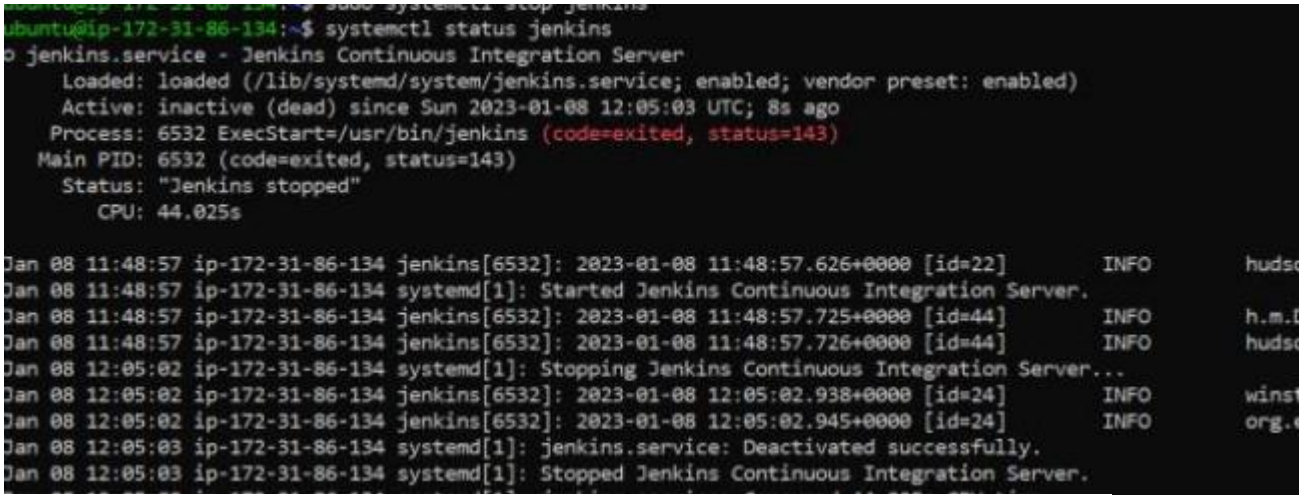
```
root@ubuntu:~# systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Mon 2019-03-18 09:09:33 UTC; 2min 3s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 1152)
   CGroup: /system.slice/jenkins.service

Mar 18 09:09:29 ubuntu systemd[1]: Starting LSB: Start Jenkins at boot time...
Mar 18 09:09:29 ubuntu jenkins[8659]: Correct java version found
Mar 18 09:09:29 ubuntu jenkins[8659]:  * Starting Jenkins Automation Server jenk
Mar 18 09:09:32 ubuntu su[8705]: Successful su for jenkins by root
Mar 18 09:09:32 ubuntu su[8705]: + ??? root:jenkins
Mar 18 09:09:32 ubuntu su[8705]: pam_unix(su:session): session opened for user j
Mar 18 09:09:32 ubuntu su[8705]: pam_unix(su:session): session closed for user j
Mar 18 09:09:33 ubuntu jenkins[8659]:    ...done.
Mar 18 09:09:33 ubuntu systemd[1]: Started LSB: Start Jenkins at boot time.
lines 1-16/16 (END)
```

After:

systemctl stop jenkins

systemctl status jenkins



3. Read about the commands systemctl vs service

`systemctl` and `service` are both tools used to manage and control services on a Linux system. However, they have some differences:

1. `systemctl` is the newer tool and is used on systems that use the Systemd init system, which is now widely adopted as the default init system for many popular Linux distributions, including Fedora, Red Hat Enterprise Linux, and Ubuntu.
2. `service` is the older tool and is used on systems that use the System V init system, which was the previous standard init system used in many popular Linux distributions.
3. `systemctl` provides more advanced features compared to `service`, such as the ability to manage units, which are the basic building blocks of Systemd. This allows you to manage not just services, but also other system components, such as sockets, devices, and mount points, with a unified interface.
4. `service` is limited to managing services only, and its syntax and options are not as advanced as those of `systemctl`.

systemctl commands :

- `systemctl start <service-name>`
- `systemctl stop <service-name>`
- `systemctl restart <service-name>`
- `systemctl enable <service-name>`
- `systemctl disable <service-name>`
- `systemctl status <service-name>`

service commands :

- `service <service-name> start`
- `service <service-name> status`
- `service <service-name> stop`
- `service <service-name> restart`

Please, feel free to drop any questions in the comments below. I would be happy to answer them.

If this post was helpful, please do follow and click the clap

_Thank you for reading

_Rajani