Service & Daemons:

Service:

A service is an application / set of application that run in the Background , waiting to be used or carrying out essential tasks.

Daemons:

Daemons are the process that run in background or wait in background to perform various tasks. (non-interactive)

- It start at background and stop till it shutdown or until they are manually stopped
- Name of many Daemons end with "d"

{System start-up and server process are managed by SYSTEMD & SERVICE MANAGER}

Features of SYSTEMD

- Parallelization capacity:
  - It can start multiple services at a time
- On demand starting of daemons with out requiring separate service
- Automatic service dependency Management:
- A methods to track related process together

It has some units to manage diff service and or objects

- Service unit (.service) : system services /
- Socket (.socket) : helps for IPC (Inter Process Communication) if a client connect to socket systemd start a daemon and connect to it . it used to start require daemons on-demand
- Path (.path) it used to delay the activation of service until a file system changes occur Eg: printing system

#systemctl is the command to manage units

```
#systemctl -t help
#systemctl list-unit --type=service
```

unit name : Load (to memory) : Active (started or not) : SUB (State) : Description

for check the service installed but not enabled

```
#systemctl list-unit-files --type=service
```

for check status of a particular service

```
#systemctl status sshd.service
```

- loaded : whether it loaded in memory or not
- Active :
- PID :
- Status:

Service States in the Output of systemctl

- loaded : unit processed
- active(running) : Running with 1 or < cont process
- active(exited) : successfully completed one-time conf
- active(waiting) : Running but wait for event
- inactive : not runing
- nabled : it start at boot
- disable : can't start at boot
- static

verifying status

```
# systemctl is-active sshd.service
               is-enabled
               is-failed
```

for check dependency

```
#systemctl list-dependency sshd.service
```

STARTING AND STOPPING SERVICES

```
[root@host ~]# systemctl start sshd.service
[root@host ~]# systemctl stop sshd.service
[root@host ~]# systemctl restart sshd.service
[root@host ~]# systemctl reload-or-restart sshd.service
```

MASKING & UNMASKING

So when we installing any package it install all dependency files,more than one service which are conflicting to each other.

Eg. for mail server

- POSTFIX
- SENDMAIL

For overcome that we do mask / unmask

When we mask a service it create a link in conf. dir to the /dev/null

Which prevent to start the service.

```
[root@host ~]# systemctl mask sendmail.service
[root@host ~]# systemctl list-unit-files --type=service
[root@host ~]# systemctl start sendmail.service
[root@host ~]# systemctl unmask sendmail
```

ENABLING SERVICES TO START OR STOP AT BOOT

When we enable a service it create a link on Systemd Conf directories to start the service at boot

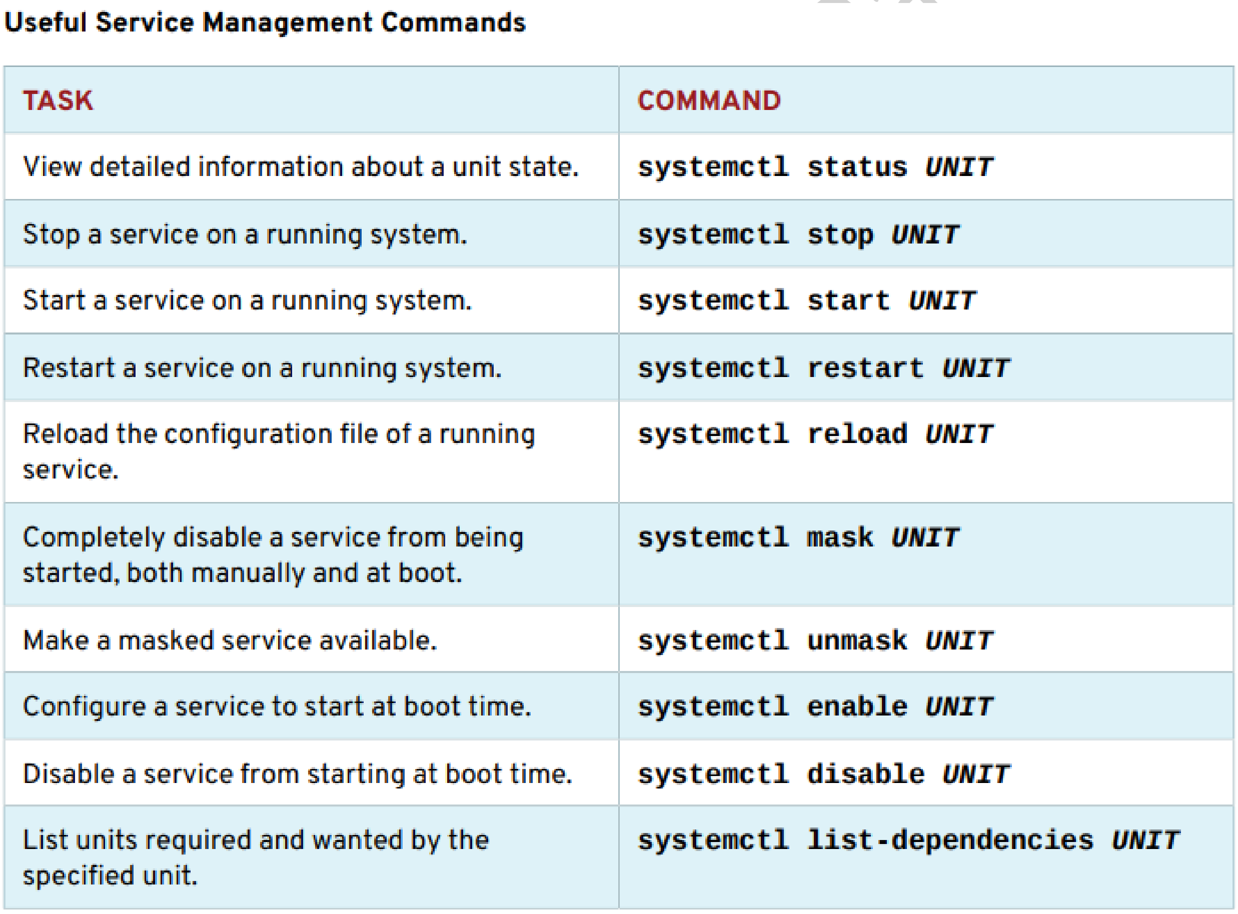
When we disable it, delete the link

(/etc/systemd/system/multi-user.target.wants/---),

```
[root@root ~]# systemctl enable sshd.service
```

It create a symbolic link on ***/etc/systemd/system/TARGETNAME.target.wants*** directory

```
[root@host ~]# systemctl disable sshd.service
```



Useful Service Management Commands

TASK	COMMAND
View detailed information about a unit state.	`systemctl status UNIT`
Stop a service on a running system.	`systemctl stop UNIT`
Start a service on a running system.	`systemctl start UNIT`
Restart a service on a running system.	`systemctl restart UNIT`
Reload the configuration file of a running service.	`systemctl reload UNIT`
Completely disable a service from being started, both manually and at boot.	`systemctl mask UNIT`
Make a masked service available.	`systemctl unmask UNIT`
Configure a service to start at boot time.	`systemctl enable UNIT`
Disable a service from starting at boot time.	`systemctl disable UNIT`
List units required and wanted by the specified unit.	`systemctl list-dependencies UNIT`