Machina Sanjeevi
sanjeevim@yahoo.com
Document: Systemctl, Systemd

Controlling Services and Daemons:-

System startup and server processes are managed by the systemd System and Service Manager. This program provides a method for activating system resources, server daemons, and other processes, both at boot time and on a running system.

Daemons are processes that wait or run in the background performing various tasks. Generally, daemons start automatically at boot time and continue to run until shutdown or until they are manually stopped. By convention, the names of many daemon programs end in the letter "d".

To listen for connections, a daemon uses a socket. This is primary communication channel with local or remote clients. Sockets may be created by daemons or may be separated from the daemon and be created by another process, such as systemd. The socket is passed to the daemon when a connection is established by the client.

As service often refers to one or more daemons, but starting or stopping a service may instead make a one-time change to the state of the system, which does not involve leaving a daemon process running afterwards(called one-shot)

In Redhat Enterprise Linux 7, process ID 1 is systemd, the new init system. A few of the new features provided by systemd include:

1)Parallelization capabilities, which increase the boot speed of a system.

2)On-demand starting of daemons without requiring a separate service.

3)Automatic service dependency management, which can prevent log timeout, such as by not starting a network service when the network is not available.

4)A method of tracking related processes together by using Linux control groups.

Systemctl and Systemd units:-

The systemctl command is used to manage different types of systemd objects, called units. A list of available unit types can be displayed with ***systemctl -t help***

```
# systemctl -t help
Available unit types:
service
socket
target
device
mount
automount
snapshot
timer
swap
path
slice
scope
#
```

Some common unit types are listed below:

Machina Sanjeevi
sanjeevim@yahoo.com
Document: Systemctl, Systemd

- Service units have a .service extension and represent system services. This type of unit is used to start frequently accessed daemons, such as as webserver.
- Socket units have a .socket extension and represent inter-process communication (IPC) sockets. control of the socket will be passed to a daemon or newly started service when a client connection is made. Socket units are used to delay the start of a service at boot time and to start less frequently used services on demond. These are similar in principle to services which use a xinetd super server to start ondemand.
- Path units have a .path extension and are used to delay the activation of a service unitil a specific file system change occurs. This is commonly used for services which use spool directories, such as a printing system.

Service States:-

The status of a service can be viewed with systemctl status name.type command. If the unit type is not provided, systemctl will show the status of service until, of non exits.

Example:- systemctl status sshd.service

Several keywords indicating the state of the service can be found in the status output:

*Loaded:*Unit configuration file has been processed

Active(running): Running with one or more continuing processes.

Active(Exited): Successfully completed a one-time configuration

*Active(Waiting):*Running for but waiting for an event

Inactive: Not running

*enabled:*Will be started at boot time.

Disabled: Will not be started at boot time

static : Can not be enabled, but may be started by an enabled unit automatically.

Identifying the status of systemd units - Examples:-

- systemctl list-units --type service
- systemctl list-units --type=socket --all
- systemctl status chronyd
- systemctl is-enabled sshd
- systemctl is-active sshd
- systemctl status sshd
- systemctl list-unit-files --type=service
- systemctl is-enabled sshd
- systemctl is-active sshd
- systemctl status --type swap
- systemctl status --type mount
- systemctl status dvd71.mount

Machina Sanjeevi
sanjeevim@yahoo.com
Document: Systemctl, Systemd

- systemctl --type socket
- systemctl --type mount
- systemctl --type service
- systemctl --type device
- systemctl -l status sshd.service
- systemctl status sshd.service

Controlling system services:-

Starting and stopping system daemons on a running system:-

1. View the status of a service

    systemctl status sshd.service

2. Verify that the process is running

    ps -up PID

3. Stop the service and verify the status

    systemctl stop sshd.service

    systemctl status sshd.service

4.Start the srvice and view the status. The process ID has changed.

    systemctl start sshd.service

    systemctl status sshd.service

5.Stop, then start the servcice in single command

    systemctl restart sshd.service

    systemctl status sshd.service

6.Issue instructions for a service to read and reload its configureation file without a complete stop and start. The rpocess ID will not change.

    systemctl reload sshd.service

    systemctl status sshd.service

Unit Dependencies: -

Services may be started as dependencies of other services. If a socket unit is enabled and the service unit with the same name is not, the service will automatically be started when a request is made on the network socket. Service may also be triggered by path units when a filesystem condition is met. For example, a file placed into the print spool directory will cause the cups print service to be started if it is not running.

Machina Sanjeevi
sanjeevim@yahoo.com

#systemctl stop cups.service

Warning: Stopping cups, but it can still be activated by:

cups.path

cups.socket

To completely stop printing services on a system, stop all three units. Disabling the service will disable the dependencies.

The systemctl list-dependencies UNIT command can be used to print out a tree of what other units must be started if the specified unit is started. Depending on the exact dependency, the other unit may need to be running before or after the specified unit starts. The --reverse option to this command will show what units need to have the specified unit started in order to run.

Masking Services:-

At times, a system may have conflicting services installed. For example, there are multiple methods to manage networks (network and network manager) and firewalls (iptables and firewalld). to prevent an administrator from accidentally starting a service, that service may be masked. Masking will create a link in the configuration directories so that if the service is started, nothing will happen.

#systemctl mask network

#systemctl unmask network

Note:- A disabled service will not be started automatically at boot or by other unit files, but can be started manually. A masked service cannot be started manually or automatically.

Enabling System daemon to start or stop at boot:-

Starting a service on a running system does not guarantee that the service will be started when the system reboots. Similarly, stopping a service on a running system will not keep it from starting again when the system reboots. Services are started at boot time when links are created in the appropriate systemd configuration directories. These links are created and removed with systemctl commands.

In the below example,

1.View the status of a service

systemctl status sshd.service

2.Disable the service and verify the status. Note that disabling a service does not stop the service.

systemctl disable sshd.service

systemctl status sshd.serivce

Machina Sanjeevi
sanjeevim@yahoo.com

3.Enable the service and verify the status

systemctl enable sshd.service

systemctl is-enabled sshd.service

Systemd targets:-

Systemd target is a set of systemd units that should be started to reach a desired state. Below are the important Targets:

1.multi-user.target

2.graphical.target

3.rescue.target

4.emergency.target

Setting a default target:-

Normally the default target will be a symbolic link (/etc/systemd/system/default) that linked to either graphical.target or multi-user.target. The systemctl tool provides two commands to manage the link. ie get-default and set-default.

ex:- systemctl get-default

The above command will print the default target that is currently set

systemctl set-default graphical.target

or

systemctl set-default multi-user.target

The above commands will set the default target.

To move to particular target, use the below command:

systemctl isolate TargetName

ex:-

systemctl isolate graphical.target

or

systemctl isolate multi-user.target

Summary of systemctl commands:-

1.View detailed information about a unit state

systemctl status UNIT

Machina Sanjeevi
sanjeevim@yahoo.com
Document: Systemctl, Systemd
2.Stop a service ona running system

systemctl stop UNIT

3.Start a service on a running system

systemctl start UNIT

4.Restart a service on a running system

systemctl restart UNIT

5.Reload configuration files of a running service

systemctl reload UNIT

6.Completely disable a service from being started, both manually and at boot

systemctl mask UNIT

7.Make a masked service available

systemctl unmask UNIT

8.Configure a servicxe to start at boot time

systemctl enable UNIT

9.Disable a service from starting at boot time

systemctl dsiable UNIT

10.List units which are required and wanted by the specified unit

systemctl list-dependencies UNIT

11.To query if a particular service is active?

systemctl is-active ServiceName

12.To list active state of all loaded units

systemctl list-units --type service --all

13.To list only failed services

systemctl --failed --type=service

Exercises for Controlling Services and Daemons:-

Start the psacct Service and configure the psacct service so that it starts at system boot

systemctl start psacct

systemctl status psacct

systemctl enable psacct

Machina Sanjeevi
sanjeevim@yahoo.com
Document: Systemctl, Systemd
systemctl status psacct

Stop rsyslog service and configure so that it doesnt start at system boot

systemctl stop rsyslog

systemctl status rsyslog

systemctl disable rsyslog

systemctl status rsyslog

Questions: -

1. How will you change run levels using systemd (changing targets)

2. How will you change default run level (or default targe)

3.compare rhel 6 run levels to rhel targets

0	runlevel0.target, poweroff.target	Shut down and power off
1	runlevel1.target, rescue.target	Set up a rescue shell
2,3,4	runlevel[234].target, multi- user.target	Set up a nongraphical multi-user shell
5	runlevel5.target, graphical.target	Set up a graphical multi-user shell
6	runlevel6.target, reboot.target	Shut down and reboot the system

4.How will you set a default target.

5.How will you verify a defualt target is set to which which target

6. How will you display the status of service?

7. How will you enable/disable/mask a service. What are the differences between masking and disable.