

Managing Services with **systemd**

By

Prof. Subrahmanya Bhat

Srinivas University, Mangaluru.

systemd

- Is a system and service manager for Linux operating systems.
- It is designed to be backward compatible with SysV init scripts

Systemd..

- provides a number of features such as parallel startup of system services at boot time, on-demand activation of daemons, support for system state snapshots, or dependency-based service control logic

- .Systemd introduces the concept of systemd units.
- .These units are represented by unit configuration files located in one of the directories listed below
 - /usr/lib/systemd/system/
 - /run/systemd/system/
 - /etc/systemd/system/

systemd unit types.

- .Service unit
- .Target unit
- .Automount unit
- .Device unit
- .Mount unit
- .Path unit
- .Scope unit

systemd units..

- .Slice unit
- .Snapshot unit
- .Socket unit
- .Swap unit
- .Timer unit

systemd units..

.These **units** encapsulate information about system services, listening sockets, saved system state snapshots, and other objects that are relevant to the **init** system.

systemd provides following features

- .Socket-based activation
- .Bus-based activation
- .Device-based activation
- .Path-based activation
- .System state snapshots

Features..

- Mount and automount point management
- Aggressive parallelization
- Transactional unit activation logic
- Backwards compatibility with SysV init

Socket-based activation

- .Systemd uses socket units for socket-based activation
- .At boot time, systemd creates listening sockets for all system services that support this type of activation, and passes the sockets to these services as soon as they are started.

Bus-based activation

- .Systemd uses D-Bus service files for bus-based activation
- .System services that use D-Bus for inter-process communication can be started on-demand the first time a client application attempts to communicate with them.

Device-based activation

- .Systemd uses device units for device-based activation
- .System services that support device-based activation can be started on demand when a particular type of hardware is plugged in or becomes available

Path-based activation

- .Systemd uses path units for path-based activation
- .System services that support path-based activation can be started on-demand when a particular file or directory changes its state

System state snapshots

- .To store the current state of the system, systemd uses dynamically created snapshot units
- .Systemd can temporarily save the current state of all units or restore a previous state of the system from a dynamically created snapshot

Mount & automount point management

- .Systemd uses mount units for mount points and automount units for automount points
- .Systemd monitors and manages mount and automount points

Aggressive parallelization

- .Because of the use of socket-based activation, systemd can start system services in parallel as soon as all listening sockets are in place
- .In combination with system services that support on-demand activation, parallel activation significantly reduces the time required to boot the system

Transactional unit activation logic

- .Before activating or deactivating a unit, systemd calculates its dependencies, creates a temporary transaction, and verifies that this transaction is consistent.
- .If a transaction is inconsistent, systemd automatically attempts to correct it and remove non-essential jobs from it before reporting an error

Backwards compatibility with SysV init

- Systemd fully supports SysV init scripts as described in the Linux Standard Base Core Specification, which eases the upgrade path to systemd service units

Managing System Services

- .Service units end with the **.service** file extension and serve a similar purpose as init scripts.
- .To view, start, stop, restart, enable, or disable system services, use the ***systemctl*** command

systemctl

.systemctl stop name.service

.systemctl restart name.service

.systemctl try-restart name.service

.systemctl reload name.service

.systemctl status name.service

.systemctl list-units --type service --all

for example

.systemctl stop nfs-server.service

.systemctl stop nfs-server

.To list all currently loaded service units, type the following at a shell prompt

.systemctl list-units --type service

.You can also list all available service units to see if they are enabled or not

.systemctl list-unit-files --type service

Displaying Service Status

.To display detailed information about a service unit that corresponds to a system service

.systemctl status name.service

.Replace name with the name of the service unit you want to inspect

Example

.systemctl status gdm.service

Example o/p

- gdm.service - GNOME Display Manager
- Loaded: loaded
(/usr/lib/systemd/system/gdm.service; enabled)
- Active: active (running) since Thu 2013-10-17
17:31:23 CEST; 5min ago
- Main PID: 1029 (gdm)
- CGroup: /system.slice/gdm.service
 - └─1029 /usr/sbin/gdm
 - └─1037 /usr/libexec/gdm-simple-slave --
display-id

.command displays the name of the selected service unit followed by its short description like

- Loaded
- Active
- Main PID
- Status
- Process
- CGroup

Loaded

.Information whether the service unit has been loaded, the absolute path to the unit file, and a note whether the unit is enabled

Active

- Information whether the service unit is running followed by a time stamp.

MainPID

.The PID of the corresponding system service followed by its name.

Status

.Additional information about the corresponding system service.

Process

.Additional information about related processes.

CGroup

.Additional information about related Control Groups (cgroups).

Starting a Service

systemctl start name. Service

Starting Apache Service

.The service unit for the Apache HTTP Server is named `httpd.service`.

.To activate this service unit and start the `httpd` daemon in the current session

.systemctl start httpd.service

Stopping a Service

systemctl stop name.service

Ex. Stopping a Service

systemctl stop httpd.service

systemctl stop bluetooth.service

Restarting a Service

systemctl restart name. Service

.This command stops the selected service unit in the current session and immediately starts it again.

.Importantly, if the selected service unit is not running, this command starts it too

try-restart

.To tell systemd to restart a service unit only if the corresponding service is already running

.systemctl try-restart name. service

reload

.Certain system services also allow you to reload their configuration without interrupting their execution. To do so,

.systemctl reload name. service

.Note that system services that do not support this feature ignore this command altogether.

.For convenience, the systemctl command also supports the **reload-or-restart** and **reload-or-try-restart** commands that restart such services

Enabling a Service

.To configure a service unit that corresponds to a system service to be automatically started at boot time, type the following

.systemctl enable name.service

.Replace name with the name of the service unit you want to enable

Ex

.To configure the Apache HTTP Server to start automatically at boot time, run the following command as root

.systemctl enable httpd.service

Disabling a Service

systemctl disable name.service

.Replace name with the name of the service unit you want to disable

.To only verify that a particular service unit is running, run the following command:

.systemctl is-active name. service

.Similarly, to determine whether a particular service unit is enabled, type:

.systemctl is-enabled name. service

.Note that both **systemctl is-active** and **systemctl is-enabled** return an exit status of 0 if the specified service unit is running or enabled respectively

.To determine what services are ordered to start before the specified service, type the following at a shell prompt:

**.systemctl list-dependencies --before
gdm.service**

.To determine what services are ordered to start after the specified service, type the following at a shell prompt:

**.systemctl list-dependencies --after
gdm.service**

- .you can mask any service unit to prevent it from being started manually or by another service.
- .To do so, run the following command as root

.systemctl mask name.service

.To unmask a service unit, type as root

.Systemctl unmask name.service

Working with systemd Targets

.In Red Hat Enterprise Linux 7, the concept of runlevels has been replaced with systemd targets

Target units

- Systemd has only limited support for runlevels.
- It provides a number of target units that can be directly mapped to these runlevels

Target units

- .Systemd targets are represented by **target units**
- .Target units end with the **.target** file extension

Target units

- .Their only purpose is to group together other systemd units through a chain of dependencies.
- .For example, the **graphical.target** unit, which is used to start a graphical session, starts system services such as the GNOME Display Manager (**gdm.service**) or Accounts Service (**accounts-daemon.service**) and also activates the **multi-user.target** unit.

Target units

.Similarly, the **multi-user.target** unit starts other essential system services such as NetworkManager (**NetworkManager.service**) or D-Bus(**dbus.service**) and activates another target unit named **basic.target**.

Run Level 0

.runl evel0.target, poweroff.target

.Shut down and power off the system.

Run Level 1

.runlevel1.target, rescue.target

.Set up a rescue shell, single user mode

Run Level 2-4

.runlevel4.target, multi-user.target

.Set up a non-graphical multi-user system

Run Level 5

.runlevel5.target, graphical.target

.Set up a graphical multi-user system.

Run Level 6

.runlevel6.target, reboot.target

.Shut down and reboot the system.

Viewing the Current Target

.To list all currently loaded target units, type the following command at a shell prompt

.systemctl list-units --type target

Default Target

.To determine which target unit is used by default, run the following command

.systemctl get-default

Changing the Default Target

.To configure the system to use a different target unit by default, type the following at a shell prompt as root

.systemctl set-default name.target

Ex

systemctl set-default multi-user.target

Changing the Current Target

.To change to a different target unit in the current session, type the following at a shell prompt as root

.systemctl isolate name.target

.To turn off the graphical user interface and change to the multi-user.target unit in the current session

.systemctl isolate multi-user.target

Changing to Rescue Mode

- .Rescue mode provides a convenient single-user environment and allows you to repair your system
- .rescue mode is equivalent to single user mode and requires the root password

.In rescue mode, the system attempts to mount all local file systems and start some important system services, but it does not activate network interfaces or allow more users to be logged into the system at the same time

.To change the current target and enter rescue mode in the current session, type the following at a shell prompt as root

.systemctl rescue

.This command is similar to

.systemctl isolate rescue.target

.it also sends an informative message to all users that are currently logged into the system.

]# systemctl rescue

Broadcast message from root@ localhost on pts/0
(Fri 2013-10-25 18:23:15 CEST):

The system is going down to rescue mode NOW!

systemctl --no-wall rescue

Changing to Emergency Mode

.Emergency mode provides the most minimal environment possible and allows you to repair your system even in situations when the system is unable to enter rescue mode

.To change the current target and enter emergency mode, type the following at a shell prompt as root

.systemctl emergency

Systemctl isolate emergency.target

systemctl --no-wall emergency

Power Management Commands

Shutting Down

Suspending

Hibernating

.The **systemctl** utility provides commands for shutting down the system

Shutting Down the System

.To shut down the system and power off the machine, type the following at a shell prompt as root

.systemctl poweroff

.To shut down and halt the system without powering off the machine, run the following command as root

.systemctl halt

.Both of these commands causes **systemd** to send an informative message to all users that are currently logged into the system

systemctl --no-wall poweroff

systemctl --no-wall halt

Using shutdown Command

.To shut down the system and power off the machine at a certain time, use a command in the following format as root

.shutdown --poweroff hh:mm

Advantages of Shutdown cmd

- .As it supports a time argument, it is useful for scheduled maintenance
- .It allow more time for users to react to the warning that a system shutdown has been scheduled.
- .The option to cancel the shutdown can also be an advantage

.A pending ***shutdown*** can be cancelled by the root user as follows:

.shutdown -c

.To shut down and halt the system after a delay, without powering off the machine, use a command in the following format as root

.shutdown --halt +m

. +m is the delay time in minutes

Restarting the System

systemctl reboot

.To suspend the system, type the following at a shell prompt as root

.systemctl suspend

systemctl suspend

- This command saves the system state in RAM
- powers off most of the devices in the machine
- When you turn the machine back on, the system then restores its state from RAM without having to boot again

- Because the system state is saved in RAM and not on the hard disk, restoring the system from suspend mode is significantly faster than restoring it from hibernation
- a suspended system state is also vulnerable to power outages

Hibernating the System

.To hibernate the system, type the following at a shell prompt as root

.systemctl hibernate

systemctl hibernate

- .This command saves the system state on the hard disk drive and powers off the machine.
- .When you turn the machine back on, the system then restores its state from the saved data without having to boot again

.To hibernate and suspend the system, run the following command as root

.systemctl hybrid -sleep

Remote Machine Mgt

.The `systemctl` utility also allows you to interact with `systemd` running on a remote machine over the SSH protocol

.Provided that the **sshd** service on the remote machine is running, you can connect to this machine by running the **systemctl** command with the **--host** or **-H** command line option

systemctl --host user_name@host_name command

.Ex

***.systemctl -H root@server-0 1.example.com status
httpd.service***

Note

- When systemd starts a system service, it stores the ID of its main process in order to keep track of it
- The **systemctl** utility then uses this PID to query and manage the service.

Note

- With systemd, only running services are stopped on shutdown.
- earlier releases of the system used symbolic links located in the **/etc/rc0.d/** directory to stop all available system services regardless of their status.

Note

- System services are unable to read from the standard input stream.
- When systemd starts a service, it connects its standard input to **/dev/null** to prevent any interaction with the user

Note

- System services do not inherit any context (such as the HOME and PATH environment variables) from the invoking user and their session
- Each service runs in a clean execution context.

Note

- All operations on service units are subject to a default timeout of 5 minutes to prevent a malfunctioning service from freezing the system

Note

- Previous versions of Red Hat Enterprise Linux, which were distributed with SysV, used init scripts located in the **/etc/rc.d/init.d/** directory.
- These init scripts allowed the system administrator to control the state of services and daemons

Note

- In Red Hat Enterprise Linux 7, these init scripts have been replaced with service units
- Service units end with the **.service** file extension and serve a similar purpose as init scripts.
- To view, start, stop, restart, enable, or disable system services, use the **systemctl** command

systemd unit files

.systemctl commands work with unit files in the background

.A unit file contains configuration directives that describe the unit and define its behaviour

systemd unit files

.the /etc/systemd/system/ directory is reserved for unit files created or customized by the system administrator

systemd unit files

- Unit files typically consist of three sections
 - Unit
 - Unit Type
 - Install

Ex. postfix.service Unit File

[Unit]

- Description=Postfix Mail Transport Agent
- After=syslog.target network.target
- Conflicts=sendmail.service exim.service

[Service]

- Type=forking
- PIDFile=/var/spool/postfix/pid/master.pid
- EnvironmentFile=-/etc/sysconfig/network
- ExecStartPre=-/usr/libexec/postfix/aliasesdb

Unit

- Contains generic options that are not dependent on the type of the unit.
- These options provide unit description, specify the unit's behavior, and set dependencies to other units

Unit Type

.If a unit has type-specific directives, these are grouped under this section (named after the unit type, Ex. Service).

Install

.Contains information about unit installation used by **systemctl enable** and **disable** commands

Assignments

1 Mark Questions

What is systemd?

Name the systemd tool for Service Management.

Name the systemd tool for System Management.

How do you identify the unit files related to any service?

Assignments 1 Marks

How will you get default target unit in Linux?

Give the command to set your RedHat Linux in graphical mode by default

Which command used for changing target in the current session?

While Shutting down, which target is used by RedHat Linux?

While Rebooting, which target is used by RedHat Linux?

Subrahmanya Bhat, Srinivas University

SSH stands for_____

Assignments 7 Marks

What are the various system units? Give the significance of any three.

What is systemd? What are the features of systemd?

Give the command to get the status of any service. Also give different descriptions on its output

Give the procedure to check whether gdm service is enabled or not. If not start and then stop the service.

Give the difference between following

Assignments 7 Marks

Give the difference between following commands:

systemctl restart httpd.service and *systemctl try-restart httpd.service*.

What is the significance of *systemctl reload httpd.service* command? How it differs from *systemctl restart httpd.service*

How will you find default target unit for linux? How to change it to another target unit?

What is system targets in RedHat Linux? Name the different target units and its significance

Assignments 7 Marks

How to get in to *rescue mode* and *emergency mode* in case of RedHat Enterprise Linux? Give its significance.

What is the significance of *suspend* and *hibernate* in case of RedHat Enterprise server?

Give the different usage and significance of *shutdown* command in RedHat Linux

What is systemd? Give the procedure to manage the remote machine using systemd.

Subrahmanya Bhat, Srinivas University

Write a note on Unit Files and its structure in

Assignments 7 Marks

How to get in to *rescue mode* and *emergency mode* in case of RedHat Enterprise Linux? Give its significance.

What is the significance of *suspend* and *hibernate* in case of RedHat Enterprise server?

Give the different usage and significance of *shutdown* command in RedHat Linux

What is systemd? Give the procedure to manage the remote machine using systemd.

Subrahmanya Bhat, Srinivas University

Write a note on Unit Files and its structure in