

# RHEL: Services basic management - systemd

Article Number: 137 | Rating: Unrated | Last Updated: Sat, Jun 2, 2018 9:04 AM

## RHEL: Services basic management - systemd

```
# Tested on CentOS 7

# systemd is the new Fedora init system adopted by Red Hat from RHEL
7 on.

# It is backwards compatible with SysV init scripts, enhances the
administrative process
# and provides new features such as parallel startup of system
services at boot time or
# dependency-based service control, for instance.

# systemd introduces the concept of systemd units. These units are
represented by unit
# configuration files and encapsulate information about system
services and other objects
# that are relevant to the init system.
#
# systemd unit locations:
#
# /usr/lib/systemd/system      Systemd units distributed with installed
RPM packages.
#
# /run/systemd/system         Systemd units created at run time. This
directory takes
#
                                precedence over the directory with
installed service units.
#
```

```

# /etc/systemd/system      Systemd units created and managed by the
system administrator.
#                          This directory takes precedence over the
directory with runtime
#                          units.

# "systemctl" command shows the state of all services. It queries the
state of services,
# both systemd native and SysV/LSB services. It shows for each
service, whether it managed
# to start up or failed (time-out, non-zero exit code, abnormal
termination):

systemctl
  UNIT                                LOAD    ACTIVE SUB
DESCRIPTION
[...]
  auditd.service                    loaded active running
Security Auditing Service
  chronyd.service                   loaded active running   NTP
client/server
  crond.service                     loaded active running
Command Scheduler
  dbus.service                     loaded active running   D-Bus
System Message Bus
  getty@tty1.service                loaded active running   Getty
on tty1
* kdump.service                   loaded failed failed   Crash
recovery kernel arming
  kmod-static-nodes.service         loaded active exited
Create list of required static device nodes for the current kernel
  lvm2-monitor.service              loaded active exited
Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress
polling
  network.service                  loaded active exited   LSB:
Bring up/down networking
[...]

```

local-fs-pre.target	loaded active active	Local
File Systems (Pre)		
local-fs.target	loaded active active	Local
File Systems		
multi-user.target	loaded active active	Multi-
User System		
network-online.target	loaded active active	
Network is Online		
network.target	loaded active active	
Network		
paths.target	loaded active active	Paths
remote-fs.target	loaded active active	
Remote File Systems		
slices.target	loaded active active	
Slices		
sockets.target	loaded active active	
Sockets		
swap.target	loaded active active	Swap
sysinit.target	loaded active active	
System Initialization		
timers.target	loaded active active	
Timers		
systemd-tmpfiles-clean.timer	loaded active waiting	Daily
Cleanup of Temporary Directories		

LOAD = Reflects whether the unit definition was properly loaded.

ACTIVE = The high-level unit activation state, i.e. generalization of SUB.

SUB = The low-level unit activation state, values depend on unit type.

109 loaded units listed. Pass --all to see loaded but inactive units, too.

To show all installed unit files use 'systemctl list-unit-files'.

# To have a quick view of system status, run "**systemctl status**":

## systemctl status

```
* myserver
  State: running
    Jobs: 0 queued
  Failed: 0 units
    Since: Tue 2016-02-02 09:13:06 CET; 1h 11min ago
  CGroup: /
          |-1 /usr/lib/systemd/systemd --switched-root --system
--deserialize 21
          |-user.slice
          |  |-user-0.slice
          |    |-session-3.scope
          |      |-2162 /usr/sbin/anacron -s
          |        |-session-2.scope
          |          |-2114 sshd: root@pts/1
          |            |-2116 -bash
          |              |-2212 man systemd-cgls
          |                |-2221 less -s
          |                  |-session-1.scope
          |                    |-2068 sshd: root@pts/0
          |                      |-2070 -bash
          |                        |-2323 systemctl status
          |                          |-2324 less
          |-system.slice
          |  |-mysqld.service
          |    |-1014 /usr/sbin/mysqld --daemonize --pid-
file=/var/run/mysqld/mysqld.pid
          |  |-tuned.service
          |    |-939 /usr/bin/python -Es /usr/sbin/tuned -l -P
          [...]
          |-system-getty.slice
          |  |-getty@tty1.service
          |    |-764 /sbin/agetty --noclear tty1 linux
```

# To have a little bit more information about a service, use

```
"systemctl status <service>".
```

```
# systemd tracks and remembers whether the service started up  
successfully or not  
# both during start-up and runtime.
```

```
# Example of a service that failed to stay up, when it ran as PID  
2148, and indicates that  
# the process failed with exit status of 1:
```

```
systemctl status kdump
```

```
* kdump.service - Crash recovery kernel arming  
   Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled;  
vendor preset: enabled)  
   Active: failed (Result: exit-code) since Mon 2016-01-25 14:11:55  
CET; 1h 50min ago  
   Main PID: 2148 (code=exited, status=1/FAILURE)
```

```
Jan 25 14:11:21 myserver systemd[1]: Starting Crash recovery kernel  
arming...
```

```
Jan 25 14:11:55 myserver kdumpctl[2148]: No memory reserved for crash  
kernel.
```

```
Jan 25 14:11:55 myserver kdumpctl[2148]: Starting kdump: [FAILED]
```

```
Jan 25 14:11:55 myserver systemd[1]: kdump.service: main process  
exited, code=exited, status=1/FAILURE
```

```
Jan 25 14:11:55 myserver systemd[1]: Failed to start Crash recovery  
kernel arming.
```

```
Jan 25 14:11:55 myserver systemd[1]: Unit kdump.service entered  
failed state.
```

```
Jan 25 14:11:55 myserver systemd[1]: kdump.service failed.
```

```
# Managing of system services with "systemctl"
```

```
# -----  
-----
```

```
# Start a service
```

```
systemctl start <service>
```

```
# Stop a service
```

```
systemctl stop <service>
```

```
# Restart a service
```

```
systemctl restart <service>
```

```
# Restart a service only if it is running
```

```
systemctl try-restart <service>
```

```
# Reload configuration
```

```
systemctl reload <service>
```

```
# Check if a service is running
```

```
systemctl status <service>
```

```
systemctl is-active <service>
```

```
# Display the status of all services
```

```
systemctl list-units --type service
```

```
systemctl list-units --all
```

```
# Enable a service
```

```
systemctl enable <service>
```

```
# Disable a service
```

```
systemctl disable <service>
```

```
# Check if a service is enabled
```

```
systemctl status <service>
```

```
systemctl is-enabled <service>
```

```
# List all services and check if they are enabled
```

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

```
# List services that are ordered to start before the specified unit
```

```
systemctl list-dependencies --after [<service>]
```

```

# List services that are ordered to start after the specified unit
systemctl list-dependencies --before [<service>]

# SYSTEMD TARGETS
# -----
-----

# In RHEL 7, the concept of runlevels has been replaced with systemd
"targets".

# Systemd targets are represented by target units. Target units end
with the .target file
# extension and their only purpose is to group together other systemd
units through a chain
# of dependencies.

# RHEL 7 has a number of predefined targets similar to the standard
set of runlevels from
# the previous releases.

# systemd Targets

runlevel0.target, poweroff.target      shut down and power off the
system
runlevel1.target, rescue.target        set up a rescue shell
runlevel2.target, multi-user.target    set up a non-graphical multi-
user system
runlevel3.target, multi-user.target    set up a non-graphical multi-
user system
runlevel4.target, multi-user.target    set up a non-graphical multi-
user system
runlevel5.target, graphical.target     set up a graphical multi-
user system

```

```
runlevel6.target, reboot.target          shut down and reboot the
system

# List currently loaded target units:
systemctl list-units --type target

# Change the current target:
systemctl isolate <name.target>

# Verify / modify the default target:
systemctl get-default
systemctl set-default <name.target>

# The last command will replace the
/etc/systemd/system/default.target file with a
# symbolic link to /usr/lib/systemd/system/<name.target>:

systemctl set-default graphical.target
    Removed symlink /etc/systemd/system/default.target.
    Created symlink from /etc/systemd/system/default.target to
/usr/lib/systemd/system/graphical.target.

# Switching to default/rescue/emergency mode

systemctl default          # Enter default mode. Equivalent to systemctl
isolate default.target

systemctl rescue           # or systemctl isolate rescue.target
systemctl --no-wall rescue # prevent sending informative
message to users
```



```
# In emergency mode, the system only mounts the root file system only
for reading and
# starts a few essential services. In RHEL 7, emergency mode requires
the root password.
```

```
systemctl emergency                # or systemctl isolate
emergency.target
systemctl --no-wall emergency      # prevent sending informative
message to users
```

```
# Halting, powering off and rebooting system
```

```
# In RHEL 7, systemctl replaces power management commands; these
commands are available in
# the system for compatibility reasons but it is recommended to use
systemctl when
# possible:
```

```
ll /usr/sbin/halt /usr/sbin/poweroff /usr/sbin/shutdown
/usr/sbin/reboot
    lrwxrwxrwx. 1 root root 16 Jan 15 18:33 /usr/sbin/halt ->
../bin/systemctl
    lrwxrwxrwx. 1 root root 16 Jan 15 18:33 /usr/sbin/poweroff ->
../bin/systemctl
    lrwxrwxrwx. 1 root root 16 Jan 15 18:33 /usr/sbin/reboot ->
../bin/systemctl
    lrwxrwxrwx. 1 root root 16 Jan 15 18:33 /usr/sbin/shutdown ->
../bin/systemctl
```

```
# Halt the system
systemctl halt
```

```
# Power off the system
systemctl poweroff
```

```
# Restart the system
```

```
systemctl reboot
```

Posted - Sat, Jun 2, 2018 9:04 AM. This article has been viewed 18897 times.

Online URL: <http://kb.ictbanking.net/article.php?id=137>