**Boot, reboot, and shut down a system normally.**

**Prerequisites**

Switch on a physical server or a virtual machine (in this case, type: # **virsh start vm**).  
Connect to the console.

**Basic Management**

To reboot the system, choose one command among these:

# **reboot**

# **systemctl reboot**

# **shutdown -r now**

# **init 6**

# **telinit 6**

To shutdown the system, choose one command among these:

# **halt**

# **systemctl halt**

# **shutdown -h now**

# **init 0**

# **telinit 0**

To switch off the system, choose one command among these:

# **poweroff**

# **systemctl poweroff**

**Advanced Management**

To suspend the system, type:

# **systemctl suspend**

To put the system into hibernation, type:

# **systemctl hibernate**

To put the system into hibernation and suspend it, type:

# **systemctl hybrid-sleep**

# use systemctl to start, restart and shutdown rhel 7 and centos 7

use systemctl to start, restart and shutdown rhel 7 and centos 7

In this article we are going to learn how to restart, shutdown and change **run Levels = Targets** in RHEL 7. This article is useful for the guys who are preparing for the RHCSA and RHCE examination.

## Using operating systems

* Start, restart and shutdown a system normally
* Start systems manually in different destinations
* Interrupting the boot process in order to gain access to a system
* Identify the processes that make heavy use of the CPU and memory, adjust the priority of processes renice and kill processes
* Locate and interpret system log files and newspapers
* Access the console of a virtual machine
* Start and stop virtual machines
* Transfer files between different systems securely

These objectives, though trivial for an experienced Linux administrator, presented a new challenge, since in this version of the system has been incorporated Redcap systemd, adding many more features and a little complexity.

We’ll break briefly the first 3 points in this post  
Start, restart and shutdown a system normally

This statement may tempt us to think ‘shutdown’, ‘reboot’ and ‘init’ are the commands needed to handle the issue. While it is true, it is not less, now the whole process startup, shutdown and restart is managed by systemd, therefore, we need to incorporate into our repertoire systemd command.

[root@TechTutorial ~]# systemctl reboot

[root@TechTutorial ~]# systemctl halt

[root@TechTutorial ~]# systemctl poweroff

Anyway we can continue you to use the old classic commands also

## How to restart server

[root@TechTutorial ~]# reboot

[root@TechTutorial ~]# shutdown -r now [

[root@TechTutorial ~]# init 6

## Turning off server

[root@TechTutorial ~]# halt

[root@TechTutorial ~]# shutdown -h now

[root@TechTutorial ~]# init 0

## Off ACPI ( ACPI stands for Advanced configuration and power interface )

[root@TechTutorial ~]# poweroff

## use systemctl to suspend and hibernate

On the other hand, we can manage with systemd suspend / hibernate from the same using systemctl commands

[root@TechTutorial ~]# systemctl suspend

[root@TechTutorial ~]# systemctl hibernate

[root@TechTutorial ~]# systemctl hybrid-sleep

Start systems manually in different destinations

Also, with systemctl we can manage what was formerly known as ‘run levels’, now systemd introduces the concept of ‘targets’ greatly improving dependencies to move from a ‘target’ to another.  
Known to all, the run levels are

**init 0: halt**  
**init 1: single: Single user or maintenance**  
**init 2: no network**   
**init 3: multi user without GUI**  
**init 5: multi user with GUI**  
**init 6: reboot**

And we could go from one to another with the unit command

[root@TechTutorial ~]# init 1

And to know current run level we can use the following command

[root@TechTutorial ~]# runlevel N 5

[root@TechTutorial ~]# who -r

run-level 5  2016-04-26 14:06

Systemd is compatible with all of these commands, but the use of systemctl command is recommended, the use of which are briefly described below

Single user with local mounted file system

[root@TechTutorial ~]# systemctl rescue

Single user without local mounted file systems (only / )

[root@TechTutorial ~]# systemctl emergency

Multi user without GUI

[root@TechTutorial ~]# systemctl isolate multi-user.target

Multi User with GUI

[root@TechTutorial ~]# systemctl isolate graphical.target

We can also set default target, without editing any files. Set default target using below command

[root@TechTutorial ~]# systemctl set-default multi-user.target

To know default run level you have to below command

[root@TechTutorial ~]# systemctl get-default

## Conclusion

Run levels are obsolete now they called targets in RHEL 7 and up coming versions. As of now in RHEL 7 version old commands also works but make a good practice to use systemctl always.