

Operating Running Systems

- **Boot, reboot, and shut down a system normally**
- **Boot systems into different targets manually**
- Interrupt the boot process to gain access to a system
- **Identify CPU/memory intensive processes and kill processes**
- **Adjust process scheduling**
- **Manage tuning profiles**
- **Locate and interpret system log files and journals**
- **Preserve system journals**
- Start, stop, and check the status of network services
- Securely transfer files between systems

Reboot, Shutdown and Suspending System

Command	Action/Description
<code>systemctl reboot</code> or <code>shutdown -r now</code>	To reboot System
<code>systemctl halt</code> or <code>shutdown --halt</code>	To halt the System
<code>systemctl poweroff</code> or <code>shutdown --poweroff</code>	To power off System
<code>systemctl suspend</code>	To suspend System (no-power mode)
<code>systemctl hibernate</code>	To hibernate System (low-power mode)
<code>shutdown --halt HH:MM</code>	To schedule System halt
<code>shutdown --halt +10</code>	To Schedule System halt after 10 minutes
<code>shutdown --halt +10 "System will go down in 10 minutes"</code>	To schedule System halt with message to users
<code>shutdown -c</code>	To cancel the scheduled shutdown command

- Run CPU intensive process on System with nice value of -5 to give more CPU attention than default.
 - Change the niceness value to 5 so that CPU pays less attention to this process

Niceness value :

nice value can be between -20 to 19. Lesser the NICE value, more CPU resources will be used. Higher the nice value, less CPU attention will be given. Never run process with nice value of -20 ,CPU will give highest priority and no other jobs will be able to run.

Command	Action/Description
<code>nice -n -5 dd if=/dev/zero of=/dev/null</code>	To start process with nice value -5
<code>renice -n 5 -p PID</code>	To change nice value
<code>top</code>	To verify changes

- Run below command in background with nice value of 10.
sleep 3600

Command	Action/Description
nice -n 10 sleep 3600 &	To start a process with pre-defined nice value
ps aux grep sleep	To get PID
top -p PID	To display resource usage using top

➤ Kill all running dd processes to stop forcefully...

Command	Action/Description
top	To check running dd processes and PID's
kill -9 PID	To kill process forcefully
man kill	Manual page for kill command
kill --help	To display help for kill command

- Modify kernel parameter `vm.swappiness` to set the value to 50.
 - Changes done should persist after reboot.

Definition :

it represents the percentage of the free memory before activating swap. The lower the value, the less swapping is used, and the more memory pages are kept in physical memory.

Changing the value directly influences the performance of the Linux system. These values are defined:

0: swap is disabled

1: minimum amount of swapping without disabling it entirely

10: recommended value to improve performance when sufficient memory exists in a system

100: aggressive swapping

Command	Action/Description
<code>sysctl -a grep swappiness</code>	To display existing value of parameter
<code>sysctl -w vm.swappiness=50</code>	To change value of parameter in run time
<code>vim /etc/sysctl.conf</code> <code>vm.swappiness=50</code> <code>:wq</code>	To make changes persistently
<code>systemctl reboot</code>	To reboot System
<code>sysctl -a grep swappiness</code>	To verify results

- **Modify kernel parameter to enable ipv4 packet forwarding.**
 - Changes done should persist after reboot.

Definition :

IP forwarding allows an operating system (here on Linux) to forward packets as a router does or more generally to route them through other networks.

Command	Action/Description
<code>sysctl -a grep 'net.ipv4.ip_forward'</code>	To display existing value
<code>sysctl -w net.ipv4.ip_forward=1</code>	To enable ipv4 packet forwarding in run time
<code>vim /etc/sysctl.d/ipv4_forward.conf</code> <code>net.ipv4.ip_forward=1</code> <code>:wq</code>	To enable ipv4 packet forwarding persistently
<code>systemctl reboot</code>	Reboot the System
<code>sysctl -a grep 'net.ipv4.ip_forward'</code>	To verify results

Introducing tuned and tuned profiles

Tuned is service which monitors the system and optimizes the performance of system for different use cases.

There are pre-defined tuned profiles which are present on path **/usr/lib/tuned**.

Tuned profiles are designed keeping in mind three parameters linked closely to performance of system.

- High throughput
- Low latency
- Saving power

Predefined profiles are divided into two categories :

- Power-saving profiles
- Performance-boosting profiles

You can customise a tuned profile based on standard profile or can create a completely new profile. Such profiles are always created under directory **/etc/tuned**. In case you want to create a new profile by adding different settings to pre-defined profile, copy the profile from **/usr/lib/tuned** to **/etc/tuned** and add/modify different settings.

If two profiles have same name under **/usr/lib/tuned** and **/etc/tuned** , profile under /etc/tuned takes precedence.

tuned.conf : TuneD profile definition

Tuned profile is defined in **tuned.conf** file on directory path **/usr/lib/tuned/<profile>/tuned.conf** or on path **/etc/tuned/<profile>/tuned.conf**.

Profile on path **/etc/tuned** takes precedence.

We will walk through manual page of **tuned.conf** for details.

Introducing important TuneD performance variables

- **CPUFreq Governor (governor):** Sets CPU frequency depending on the profile
governor can be set to :
 - performance
 - powersave
 - conservative
 - ondemand....
- **Energy performance Bias (energy_perf_bias) :** Defines priority of one over another
energy_perf_bias can be set to :
 - performance
 - powersave
 - normal
- **ALMP(almp) :** Aggressive Link Power Management protocol to manage HDD/SDD power consumption.
almp can be set to :
 - min_power
 - medium_power
- **AC97 Audio Driver timeout (timeout) & Radeon Graphics Card power save options (radeon_powersave)**

Tuned profiles distributed with RHEL-9

- balanced
- powersave
- throughput-performance
- latency-performance
- network-latency
- network-throughput
- virtual-guest
- virtual-host

.....

Important topics for RHCSA exam

- Listing and setting tuned profile
- Customizing tuned profile
- Merging tuned profiles

Recommended manual Pages

- man tuned
- man tuned-adm
- man tuned.conf
- man tuned-profiles

- Set recommended tuned profile on **System** and verify same after changes are done.

Command	Action/Description
dnf install tuned	Installing tuned service
systemctl enable --now tuned	Starting and configuring tuned to start at boot
tuned-adm list	Listing available tuned profiles
tuned-adm recommend	Listing tuned recommend profile
tuned-adm active	Listing active profile on System
tuned-adm profile recommended_profile	Activating recommended tuned profile
tuned-adm active	Verifying active profile after changes
man tuned-adm	To check manual page for tuned-adm

- Create a custom tuned profile with name `virtual-guest-custom` based on base profile `virtual-guest` and add below `sysctl` setting.

```
vm.swappiness=40
```

- Make sure user-provided system `sysctl` settings are not overridden by tuned `sysctl` settings.

Command	Action/Description
<code>dnf install tuned</code>	Installing tuned service
<code>systemctl enable --now tuned</code>	Starting/enabling tuned service
<code>tuned-adm active</code>	Listing active tuned profile
<code>mkdir /etc/tuned/virtual-guest-custom</code>	Creating directory under /etc/tuned to create new profile with name virtual-guest-custom
<code>vim /etc/tuned/virtual-guest-custom/tuned.conf</code> <pre>[main] summary = Customized Virtual Guest profile include = virtual-guest [sysctl] type = sysctl vm.swappiness = 40 :wq</pre>	Creating tuned configuration file for profile virtual-guest-custom
<code>tuned-adm profile virtual-guest-custom</code>	Activating customized tuned profile virtual-guest-custom
<code>systemctl restart tuned</code>	Restarting tuned service to make changes effective

- Activate additional tuned profile **powersave** to merge with existing profile.
 - **powersave** profile must have high priority under conflicting conditions.

Command	Action/Description
dnf install tuned	Installing tuned service
systemctl enable --now tuned	Starting/enabling tuned service
tuned-adm list	Listing available tuned profile
tuned-adm active	Listing active tuned profile
tuned-adm profile virtual-guest-custom powersave	Merging existing active profile with additional profile powersave
tuned-adm list	Verifying active profiles

Locating and interpreting system logs and journals

rsyslog :

rsyslog have been in use since long to manage logging on Linux systems and by default it stores the logs under [/var/log](#) directory in different files.

[/var/log/messages](#) - Stores global system messages (kernel,systemd,boot messages etc.)

[/var/log/secure](#) - Stores messages related to authentication, authorization (sshd, selinux ,user sessions, failed logins etc.)

[/var/log/cron](#) - Stores cron related messages

[/var/log/mail](#) - Stores mail server logs

How does rsyslog function?

- **rsyslog** uses input modules to collect log messages from different sources e.g. logs from applications running on system.
- Collected log messages are passed through ruleset and defined action is done when some specified condition is met.
- Output modules are used to implement actions e.g. **omfile** (built-in output module) stores the logs in local filesystem.

rsyslog rules:

rsyslog rules are formed using specific format (**facility.severity**).

rsyslog facility or **subsystem** describes source of log messages.

For example, subsystem(facility) **kern** means kernel log messages and there is subsystem information field in each log message.

rsyslog log severity can be **debug(7), info(6),notice(5),warn(4),err(3),crit(2),alert(1) and emerg(0)**.

Example rule:

#log all cron log messages(with starting severity level info) to specified file)

```
cron.info          /var/log/cron
```

[systemd-journald](#):

Since RHEL 7, **systemd** is the system and service manager and **systemd** introduced its own logging system (**system-journald service**) .

This service does not store the messages in separate files like **rsyslog** does but stores the system events together in binary format and these events cannot be read by using commands like rsyslog files.

We need to use [journalctl](#) command to query system journals.

Both logging services co-exist on Red Hat Enterprise Linux.

➤ Configure persistent storage to preserve system journals.

Command	Action/Description
<pre>vim /etc/systemd/journald.conf storage=persistent ----- :wq</pre>	Set storage to persistent
<pre>systemctl restart systemd-journald</pre>	Restart the service for changes to take into effect
<pre>journalctl --flush</pre>	Flush all journal data from /run to /var
<pre>ls -l /var/log/journal/</pre>	Listing contents in directory