

Автоматизация

Thursday, August 27, 2020 4:38 PM

Systemd менажера сервисите в CentOS както и в много други дистрибуции.

Отговаря също за много други обекти като:

- устройства;
- системни таймери;
- targets -> това са еквивалентите на Runlevels

"Нивото на изпълнение е състояние на init и цялата система, която определя какви системни услуги работят. Нивата на изпълнение се идентифицират с числа. Някои с нива на изпълнение, за да определят кои подсистеми работят, напр. Дали X работи, дали мрежата работи и т.н."

Systemd objects се наричат още units за всеки unit има unitfile за конфигурация.

Това което трябва да знаем по тази тема са само:

Service units и Service unit files.

Командата отговорна за тях е systemctl.

За да видим нашите service unit files напишете в конзолата:

`systemctl list-unit-files --at service`

Изохода за на тази команда ще изглежда ето така:

```
nick@pop-os:~$ sudo systemctl list-unit-files --at service
UNIT FILE                                STATE      VENDOR PRESET
accounts-daemon.service                 enabled    enabled
acpid.service                           disabled   enabled
alsa-restore.service                    static     enabled
alsa-state.service                       static     enabled
alsa-utils.service                       masked     enabled
apparmor.service                        enabled    enabled
apport-autoreport.service                static     enabled
apport-forward@.service                  static     enabled
apport.service                           generated enabled
apt-daily-upgrade.service                 static     enabled
apt-daily.service                        static     enabled
autovt@.service                          enabled    enabled
avahi-daemon.service                     enabled    enabled
blk-availability.service                  enabled    enabled
bluetooth.service                       enabled    enabled
bolt.service                             static     enabled
brltty-udev.service                      static     enabled
brltty.service                           disabled   enabled
clean-mount-point@.service                static     enabled
colord.service                           static     enabled
configure-printer@.service                static     enabled
console-getty.service                     disabled   disabled
console-setup.service                     enabled    enabled
container-getty@.service                  static     enabled
cron.service                             enabled    enabled
cryptdisks-early.service                  masked     enabled
cryptdisks.service                       masked     enabled
cups-browsed.service                     enabled    enabled
cups.service                             enabled    enabled
dbus-fi.w1.wpa_supplicant1.service        enabled    enabled
dbus-org.bluez.service                   enabled    enabled
dbus-org.freedesktop.Avahi.service         enabled    enabled
dbus-org.freedesktop.hostname1.service     static     enabled
dbus-org.freedesktop.locale1.service       static     enabled
dbus-org.freedesktop.login1.service         static     enabled
dbus-org.freedesktop.ModemManager1.service enabled    enabled
dbus-org.freedesktop.nm-dispatcher.service enabled    enabled
dbus-org.freedesktop.resolve1.service       enabled    enabled
dbus-org.freedesktop.thermald.service       enabled    enabled
dbus-org.freedesktop.timedate1.service      static     enabled
dbus-org.freedesktop.timesync1.service      enabled    enabled
dbus.service                             static     enabled
debug-shell.service                       disabled   disabled
display-manager.service                   static     enabled
```

Обикновено `systemctl list-unit-files -t service` ще ви покаже всички сервиси които са конфигурирани да се включват сам

Добавяйки -а всъщност сортира всичките --> включени и изключени.

systemctl list-unit-files имайте в предвид, че тази команда само показва статуса на сервизите.

Enabled	Сервиза ще се включи автоматично по време на старт
Disabled	Сервиза няма да стартира автоматично
Static	Сервиза не е включен

Някой сервизи не са имани в предвид да стартират автоматично.

sudo systemctl list-units -at service

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
accounts-daemon.service	loaded	active	running	Accounts Service
acpid.service	loaded	active	running	ACPI event daemon
alsa-restore.service	loaded	active	exited	Save/Restore Sound Card State
alsa-state.service	loaded	inactive	dead	Manage Sound Card State (restore and store)
apparmor.service	loaded	active	exited	Load AppArmor profiles
apport-autoreport.service	loaded	inactive	dead	Process error reports when automatic reporting is enabled
apport.service	loaded	active	exited	LSB: automatic crash report generation
apt-daily-upgrade.service	loaded	inactive	dead	Daily apt upgrade and clean activities
apt-daily.service	loaded	inactive	dead	Daily apt download activities
auditd.service	not-found	inactive	dead	auditd.service
avahi-daemon.service	loaded	active	running	Avahi mDNS/DNS-SD Stack
blk-availability.service	loaded	active	exited	Availability of block devices
bluetooth.service	loaded	active	running	Bluetooth service
bolt.service	loaded	active	running	Thunderbolt system service
colord.service	loaded	active	running	Manage, Install and Generate Color Profiles
connman.service	not-found	inactive	dead	connman.service
console-screen.service	not-found	inactive	dead	console-screen.service
console-setup.service	loaded	active	exited	Set console font and keymap
cron.service	loaded	active	running	Regular background program processing daemon
cups-browsed.service	loaded	active	running	Make remote CUPS printers available locally
cups.service	loaded	active	running	CUPS Scheduler
dbus.service	loaded	active	running	D-Bus System Message Bus
dm-event.service	loaded	inactive	dead	Device-mapper event daemon
dmesg.service	loaded	inactive	dead	Save initial kernel messages after boot
e2scrub_all.service	loaded	inactive	dead	Online ext4 Metadata Check for All filesystems
e2scrub_reap.service	loaded	inactive	dead	Remove Stale Online ext4 Metadata Check Snapshots
emergency.service	loaded	inactive	dead	Emergency Shell
fcoe.service	not-found	inactive	dead	fcoe.service
finalrd.service	loaded	active	exited	Create final runtime dir for shutdown pivot root
fstrim.service	loaded	inactive	dead	Discard unused blocks on filesystems from /etc/fstab
fwupd-refresh.service	loaded	inactive	dead	Refresh fwupd metadata and update motd
gdm.service	loaded	active	running	GNOME Display Manager
geoclue.service	loaded	active	running	Location Lookup Service
getty-static.service	loaded	inactive	dead	getty on tty2-tty6 if dbus and logind are not available
getty@tty1.service	loaded	inactive	dead	Getty on tty1
gpu-manager.service	loaded	inactive	dead	Detect the available GPUs and deal with any system changes
ifupdown-pre.service	loaded	active	exited	Helper to synchronize boot up for ifupdown
irqbalance.service	loaded	active	running	irqbalance daemon
iscsi-shutdown.service	not-found	inactive	dead	iscsi-shutdown.service
iscsi.service	not-found	inactive	dead	iscsi.service
iscsid.service	not-found	inactive	dead	iscsid.service
kbd.service	not-found	inactive	dead	kbd.service
keyboard-setup.service	loaded	active	exited	Set the console keyboard layout
kmmod-static-nodes.service	loaded	active	exited	Create list of static device nodes for the current kernel
logrotate.service	loaded	inactive	dead	Rotate log files
lvm2-activation.service	not-found	inactive	dead	lvm2-activation.service
lvm2-lvmpolld.service	loaded	inactive	dead	LVM2 poll daemon
lvm2-monitor.service	loaded	active	exited	Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling

Първата колона е със името на сервиза, втората колона показва дали юнит файла е зареден, третата колона показва о сервиза,

четвъртата колона показва по детайлна информация за статута на сервиза - дали работи или не и последната показва

Можем да ги сортираме както си искаме например:

sudo systemctl list-units -t service --state running

Можем да cat-нем сервизите за да видим конфигурациите им:

systemctl cat rsyslog.service

```
nick@pop-os:~$ systemctl cat rsyslog.service
# /lib/systemd/system/rsyslog.service
[Unit]
Description=System Logging Service
Requires=syslog.socket
Documentation=man:rsyslogd(8)
Documentation=https://www.rsyslog.com/doc/

[Service]
Type=notify
ExecStart=/usr/sbin/rsyslogd -n -iNONE
```

```
ExecStart=/usr/sbin/rsyslogd -n -iNONE
StandardOutput=null
Restart=on-failure

# Increase the default a bit in order to allow many simultaneous
# files to be monitored, we might need a lot of fds.
LimitNOFILE=16384

[Install]
WantedBy=multi-user.target
Alias=syslog.service
```

това, което виждаме тук, е файлът на сервисната единица, който определя зависимостите на услугата, каква команда се изпълнява и какво трябва да направи услугата, ако не успее

Можем да взем статуса на единична услуга:

```
systemctl status rsyslog
```

Системните сервиси са процеси стартирани от операционната система и са менежирани с командата systemctl и с нег

```
systemctl list-unit-files --at service
```

Нека спрем atd cron.service ще ви трябват високи привилегии.

```
sudo systemctl stop cron
```

Нека проверим статута му:

```
sudo systemctl status cron
```

Нека го стартираме:

```
systemctl start cron
```

Верифицирайте, че пак работи с:

```
sudo systemctl status cron
```

За да рестартираме ползваме systemctl restart.

Ако искаме просто да разберем дали някой сервис просто работи може да ползваме is-active

```
systemctl is-active cron
```

Ако искаме да ограничим сервис от това да бъде активен или сам да се включва можем да го маскираме.

```
systemctl mask cron
```

```
systemctl unmask cron
```

Можем да ги enable и disable

```
sudo systemctl disable cron
```

```
sudo systemctl enable cron
```

Проверката е:

```
systemctl is-enabled cron
```

systemd ни позволява да дефинираме таймерни единици:

systemd timers те са много по силна и алтернативна версия на cron jobs

Имаме два типа timer-units:

Real-time timers;

- Активират се на календарни събития.
- Подобни на cron jobs.
- Стартират базирано на датата и часа.

Можем да сложим таймер на 31 декември с цел да ни каже Happy New Year или може да сложим

таймер с цел да направи цял backup на системата.

Monotonic timers;

- Активират се на времеви интервал спрямо начална точка .
- Например ако таймер трябва да се активира 5 минути след буут процеса.
- Или 30 секунди след като се логнем.

Ще използваме монотон таймер.

Защо е по добре да ползваме таймери вместо cron джобове е защото всеки джоб ще си има свой сървис файл също ,

- Могат да работят в тяхната си среда.
- Могат да се добавят в sgroups. Могат да имат зависимости спрямо други systemd юнити.
- Например мрежи, примерно може да имаме джоб който ще тръгва в момента в който имаме връзка с интернет.
- Също джобовете ще бъдат логнати в systemd journal.

Cron jobs

<https://crontab.guru/>

vim /etc/crontab

- Лесни за създаването и почти винаги са еднолинии.
- Имат съпорт за имейли базирано на резултата от джоба.

Тази функционалност може да бъде заменена с таймери от systemd.

Systemd timers имат .timer extension и сервис със .service екстеншън.

Можем да имаме backup.timer и backup.service файл където таймера ще активира сървиса.

След това сервиса може да пусне скрипт със всякакво име като например backup.sh.

systemctl list-timers

Изхода показва кога за последно таймера е ръннал за последно и кога пак.

Казва ни името на таймера и сервиса.

Примерен сервис файл:

[Unit]

Description=System Backup

[Service]

Type=simple

ExecStart=/root/bin/sysbackup.sh

[Install]

WantedBy=multi-user.target

Примерен timer файл:

[Unit]

Description=System backup every day at 2 AM

[Timer]

OnCalendar=*-*-* 02:00:00 Всеки ден в 02:00:00 AM

Unit=backup.service

[Install]

WantedBy=multi-user.target

Един бърз пример:

```
nick@pop-os:~$ cat /etc/systemd/system/apt-update.service
[Unit]
Description=Updates OS automatically
Wants=apt-update.timer
```

```
[Service]
Type=Simple
ExecStart=/root/bin/apt-update.sh

[Install]
WantedBy=multi-user.target
```

```
nick@pop-os:/etc/systemd/system$ cat apt-update.timer
[Unit]
Description=System automatically updates

[Timer]
OnCalendar=*-** 04:04:00
Unit=apt-update.service

[Install]
WantedBy=multi-user.target
```

Monotonic timers

Те са специфични до стартираща точка

OnActiveSec	Това определя таймер спрямо момента, в който таймерът е активиран.
OnBootSec	Това дефинира таймер по отношение на момента на зареждане на машината.
OnStartupSec	Това дефинира таймер спрямо времето, когато мениджърът на услуги за първи път стартира. За системният мениджър на услуги обикновено започва много рано при стартиране. Това е предим всеки потребител, тъй като мениджърът на потребителски услуги обикновено започва само при
OnUnitActiveSec	Това определя таймер по отношение на времето, когато таймерът, който трябва да се активира,
OnUnitInactiveSec	Това определя таймер по отношение на времето, когато таймерът, който трябва да се активира,

На мен ми е нагласено след като буутне да минат 2 мин и тогава да се стартира.

<https://opensource.com/article/20/7/systemd-timers>

Файловете crontab са мястото, където се съхраняват списъците със задачи и други инструкции към демона cron.

```
* * * * command to execute
| | | |
| | |   day of week (0 - 6) (0 to 6 are Sunday to Saturday, or use names; 7 is Sunday, the same as 0)
| |     month (1 - 12)
|       day of month (1 - 31)
|       hour (0 - 23)
|       min (0 - 59)
```

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.
```

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
```

```
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
```

Entry	Description	Equivalent to	Example
@yearly (or @annually)	Run once a year at midnight on January 1	0 0 1 1 *	@yearly.php /home/example_username/mail.php
@monthly	Run once a month at midnight on the first day of the month	0 0 1 * *	@monthly.php /home/example_username/mail.php
@weekly	Run once a week at midnight on Sunday morning	0 0 * * 0	@weekly.php /home/example_username/mail.php
@daily (or @midnight)	Run once a day at midnight	0 0 * * *	@daily.php /home/example_username/mail.php
@hourly	Run once an hour at the beginning of the hour	0 * * * *	@hourly.php /home/example_username/mail.php
@reboot	Run at startup (of the cron daemon)	@reboot	@reboot.php /home/example_username/mail.php

Един елементарен cron е:

```
@daily root apt update -y && apt upgrade -y
crontab -e
```

Променяте файла crontab и може да впишете важите автоматични дейности.

<https://crontab.guru/>

```
crontab -e
```

```
5 0 * 8 * root apt update -y && apt upgrade -y
```

За да проверите дали cron job работи отворете:

```
cat /var/log/syslog | grep cron
```

Ако не работи отворете и махнете # пред кронджоба ви.

```
/etc/rsyslog.d/50-default.conf
```