

Systemd de sys Init V a kernel



Présentation pour le groupe xstra

Le 27 janvier 2015

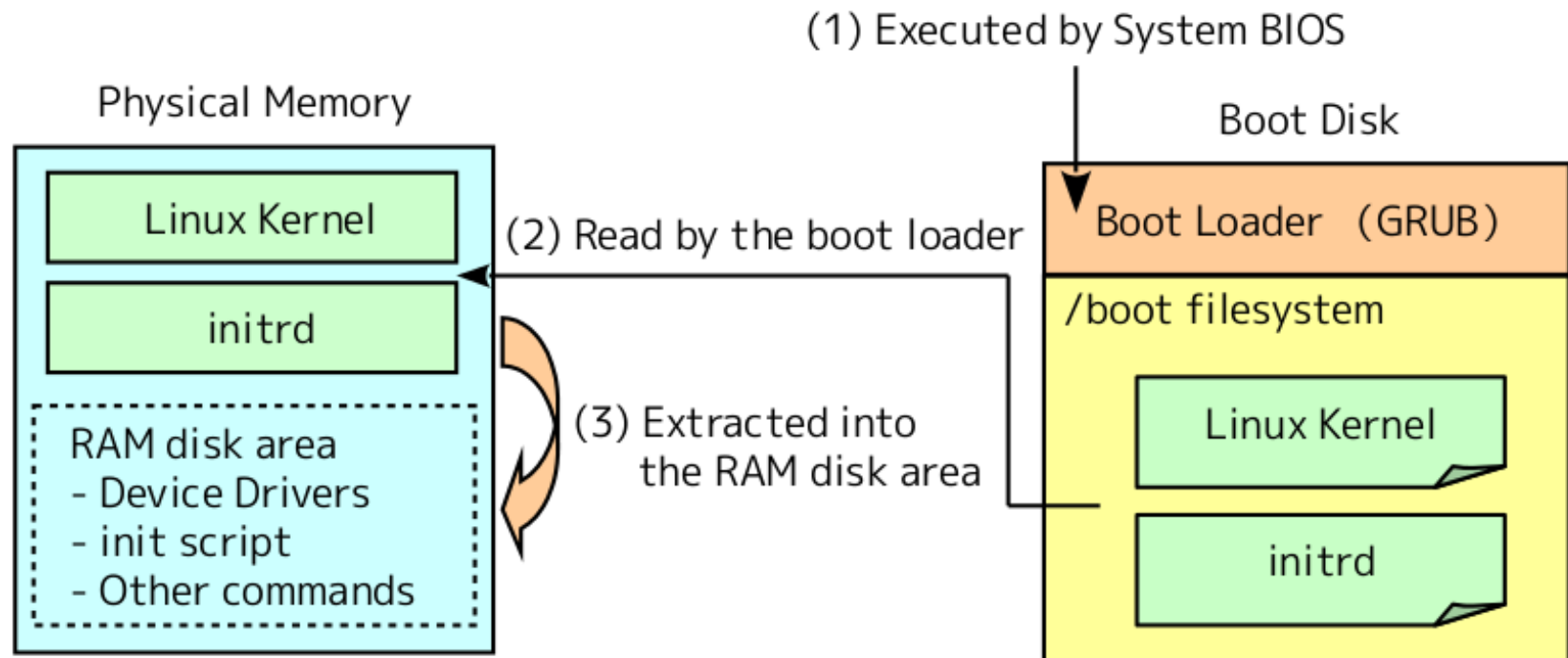
Plan de la présentation

- System init V
- Les Services
- Systemctl
- Journalctl
- extra

A blue sky with many seagulls flying over the ocean. The seagulls are in various stages of flight, some with wings spread wide, others in a more compact shape. The ocean is visible at the bottom of the frame, and there are some faint clouds in the sky.

SysVinit & Upstart

Linux boot process

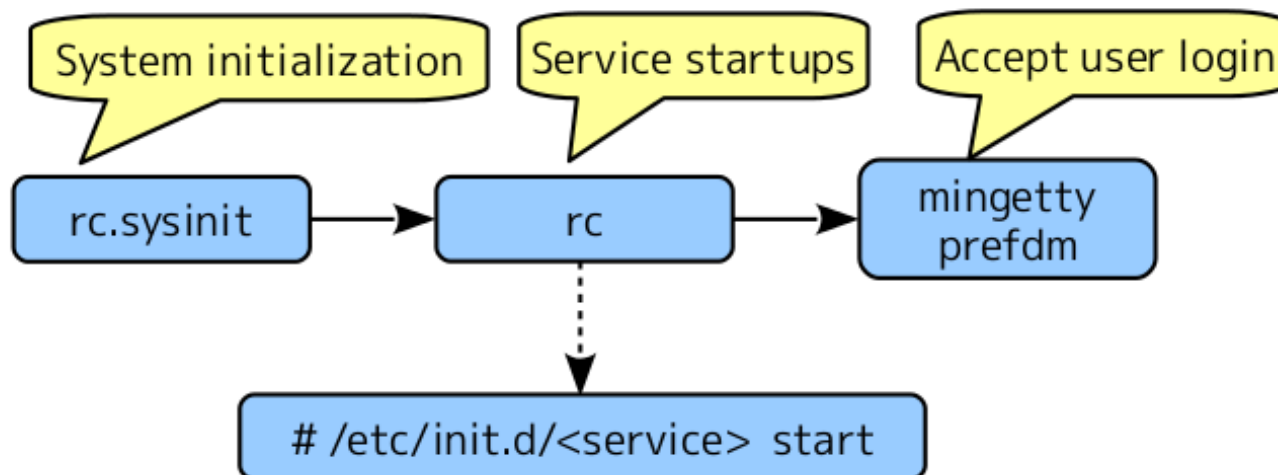


Processus de boot pour Linux

- ***/sbin/init*** est traditionnellement le coeur du système d'init pour SysinitV ou Upstart
- Pour systemd ***/usr/bin/systemd*** est utilisé

Processus de boot pour Linux

- Execute the system initialization script “/etc/rc.d/rc.sysinit” which does fsck and mount of filesystems, and other various initialization works.
- Execute the service startup script “/etc/rc.d/rc” which starts services according to the current runlevel.
 - The actual startup tasks are done by the separate service scripts under /etc/init.d/.
- Start mingetty processes which accepts the console user login. And start the GUI login screen for runlevel 5.



Example of /etc/inittab from RHEL5

```
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
...snip...

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
```

• Quels Problèmes avec `sysvinit`?

- synchronous
- everything started at boot time
- can't control double-forked child processes

Objectifs de systemd

- Réduire le temp de boot
- Gérer dynamiquement les changements de configuration
- Standardiser le processus de shutdown
- Contrôler la séquence d'exécution des processus
- Isoler les processus grâce aux cgroups

Architecture

systemd Utilities

systemctl journalctl notify analyze cglc cgtop loginctl nspawn

systemd Daemons

systemd
journald networkd
logind user session

systemd Targets

bootmode basic multi-user graphical user-session
dbus telephony display service
shutdown reboot dlog logind user-session tizen service

systemd Core

manager unit login namespace log
systemd service timer mount target multiseat inhibit cgroup dbus
snapshot path socket swap session pam

systemd Libraries

dbus-1 libpam libcap libcryptsetup tcpwrapper libaudit libnotify

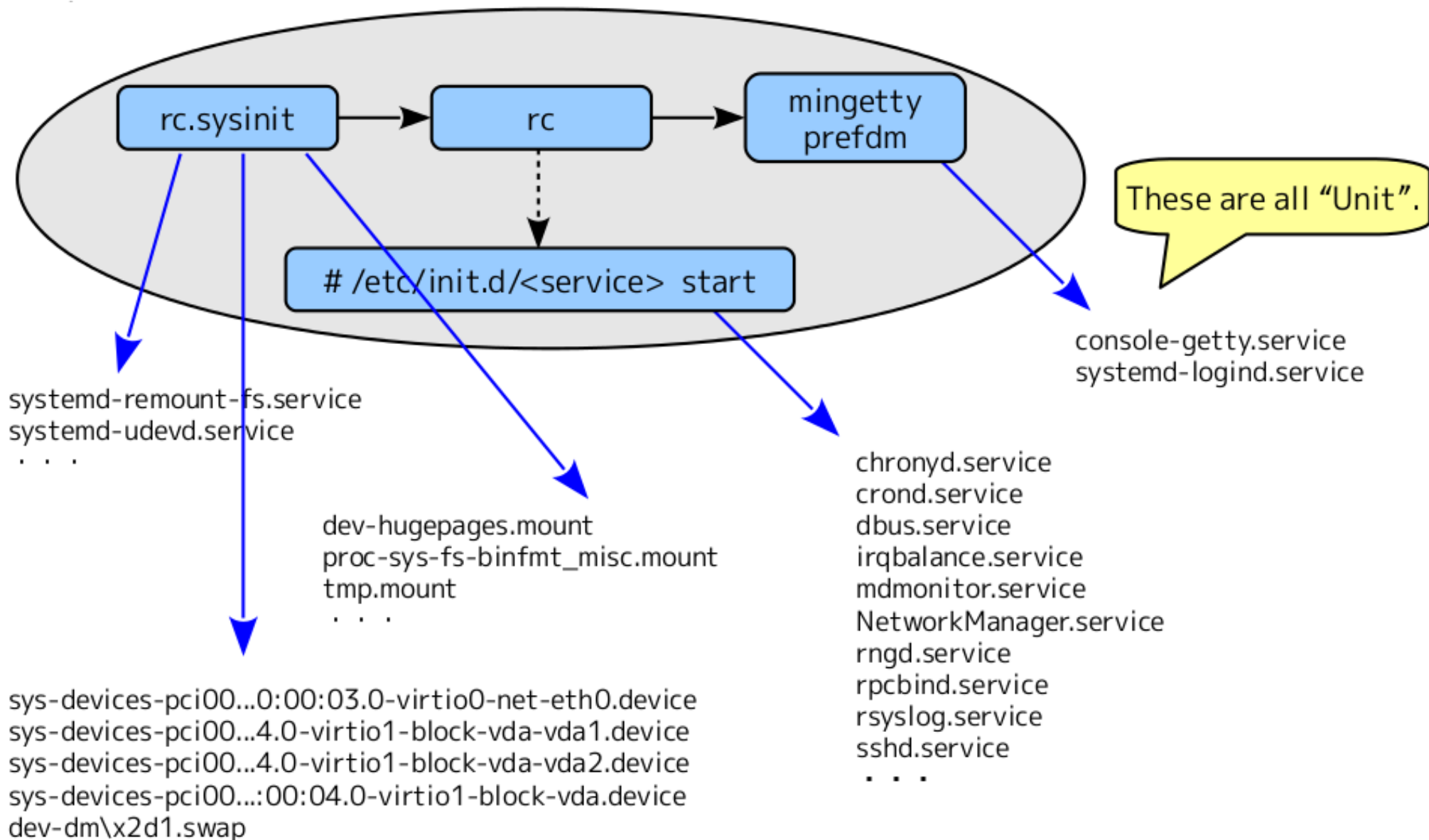
Linux Kernel

cgroups autofs kdbus



Boot process with systemd

Boot process with systemd



Type d'unité

- .service : Service type
 - When activated, an associated daemon is started.
- .target : Target type
 - Do nothing. This is used to group other units when defining unit dependencies, or to provide a timing synchronization point when defining order relationships.
- .mount : Mount point type
 - When activated, an associated filesystem is mounted.
- .swap : Swap area type
 - When activated, an associated swap area is enabled.
- .device : Device type
 - When udev recognizes a new device, the associated unit is defined and activated automatically.

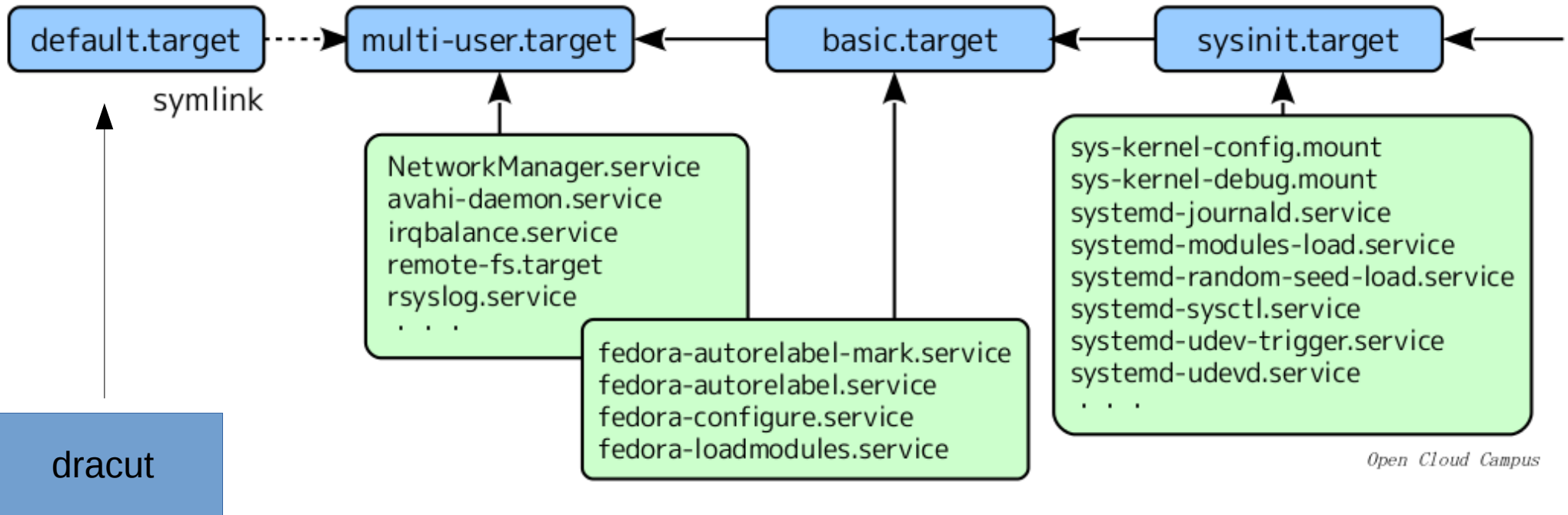
configuration

- */etc/systemd/system/* : Admin customized files
- */usr/lib/systemd/system/* : System default files
- Rajouter un service est aussi simple que de rajouter un lien symbolique dans “<unit name>.wants”

Par exemple:

/etc/systemd/system/graphical.target.wants/

Boot sequence



Adoption

- systemd adoption
- Ubuntu: upstart 2006 (native init 2009)
- Fedora: upstart 2008, systemd 2011
- RHEL: RHEL 7 (just released) use systemd
- SUSE: upstart option 2010, systemd 2011
- SLES: SLES 12 (Q3) will use systemd
- Debian: systemd 2014, Ubuntu will follow

Units

- Example d'unit
- `/usr/lib/systemd/system/sshd.service`
- `<example>`



`/usr/lib/systemd/system/httpd.service`

```
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
...
```

`/usr/lib/systemd/system/firewalld.service`

```
[Unit]
Description=firewalld - dynamic firewall daemon
Before=network.target
Before=libvirtd.service
Before=NetworkManager.service
...
```

httpd is assured to start after firewall has been configured.

Units

- Connaitre les dépendances:

```
# systemctl list-dependencies <unit name>
```

- En amont

```
# systemctl list-dependencies <unit name>  
--before
```

- En aval

```
# systemctl list-dependencies <unit name> --after
```

Units

- some systemd unit types:
 - service**: traditional daemon (ssh, http, kdm...)
 - socket**: listener socket for service activation
 - target**: like a runlevel, but not exclusive
 - device**: for device-based activation
 -
- “man systemd” pour la liste complète

Units

- Conversion d'un script de base sysinit V par simple ajout de commentaires.

Exemple de `/etc/init.d/network`

```
### BEGIN INIT INFO
# Provides: $network
# Should-Start: iptables ip6tables NetworkManager-wait-online NetworkManager
# Short-Description: Bring up/down networking
# Description: Bring up/down networking
### END INIT INFO
```

cgroup

- Topologie des cgroup
systemd-cgls
- Afficher les cgroup avec ps
ps xawf -eo pid,user,cgroup,args
- Activité des cgroup
systemd-cgtop

Controller les units

- Chkconfig équivalent
systemctl enable/disable <unit name>
- Service start/stop équivalent
systemctl start/stop/restart <unit name>
- Service status équivalent
systemctl status <unit name>
- Forcefull kill
#systemctl kill -s9 <unit name>
- Aide en ligne (man)
- # systemctl help <unit name>

Init level

- Init 3 == `systemctl isolate multi-user.target`
- Kernel option == `systemd.unit=<target-name>`
- Shutdown == `systemctl poweroff`
- Reboot == `systemctl reboot`
- Analyse du boot == `Systemd-analyse blame`

Service modèle

- Exemple d'un service
`/usr/lib/systemd/system/abrttd.service`

```
[Unit]
Description=Daemon to detect crashing apps
After=syslog.target

[Service]
ExecStart=/usr/sbin/abrttd
Type=forking

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

Units security

By network isolation

- [Service]

ExecStart=...

PrivateNetwork=yes

Units security

By fs isolation (external)

- [Service]

ExecStart=...

PrivateTmp=yes

Units security

By fs isolation (full)

- [Service]
- ExecStart=...
- DeviceAllow=/dev/null rw

Units security

By resources

- [Service]

ExecStart=...

LimitNPROC=1

LimitFSIZE=0

Units security

Full security

- [Service]

ExecStart=...

CapabilityBoundingSet=CAP_CHOWN CAP_KILL

InaccessibleDirectories=/home

ReadOnlyDirectories=/var

Manager les ressources

- [Service]

CPUShares=1500

MemoryLimit=1G

BlockIOWeight=/dev/disk/by-id/ata..... 750

Socket activation

- **`/etc/systemd/system/mycontainer.service`**

```
[Unit]
Description=My little container

[Service]
ExecStart=/usr/bin/systemd-nspawn -jbD /srv/mycontainer 3
KillMode=process
```

- **`/etc/systemd/system/mycontainer.socket`**

```
[Unit]
Description=The SSH socket of my little container

[Socket]
ListenStream=23
```

Journald

- Remplace le basic syslog
- Format binaire du journal (par défaut)
- Fonction de recherche avancé
- **-u <unit name>** : Show log messages related to this unit.
- **--since="YYYY-MM-DD hh:mm:ss"** : Show log messages since this date/time.
- **--until="YYYY-MM-DD hh:mm:ss"** : Show log messages until this date/time.
- **-b [-n]** : Show log messages since the last (or iteration -n) system boot.
- **-f** : Works as "tail -f"
- **--no-pager**: Don't use the pager(less)
- **-a** : Don't cut long messages.

journalctl

journalctl -b -p err → *erreur du dernier boot*

journalctl /usr/sbin/vpnc → *filtrer par binaires*

journalctl /dev/sdc → *filtrer par device*

journalctl -o verbose -n → *connaitre les champs*

journalctl _UID=70 → *filtrer par utilisateur*

journalctl _SE<tab> → *completion des options*

journalctl <tab> → *découverte des commandes*

journalctl -u sshd.service -b --no-pager -a

journal

- Fichier de configuration
`/etc/systemd/journald.conf`
- Le flux peut être rediriger vers syslog ou `/var/log`
`ForwardToSyslog / ForwardToWall`
- Pour plus de fonctions voir les tableaux en annexe

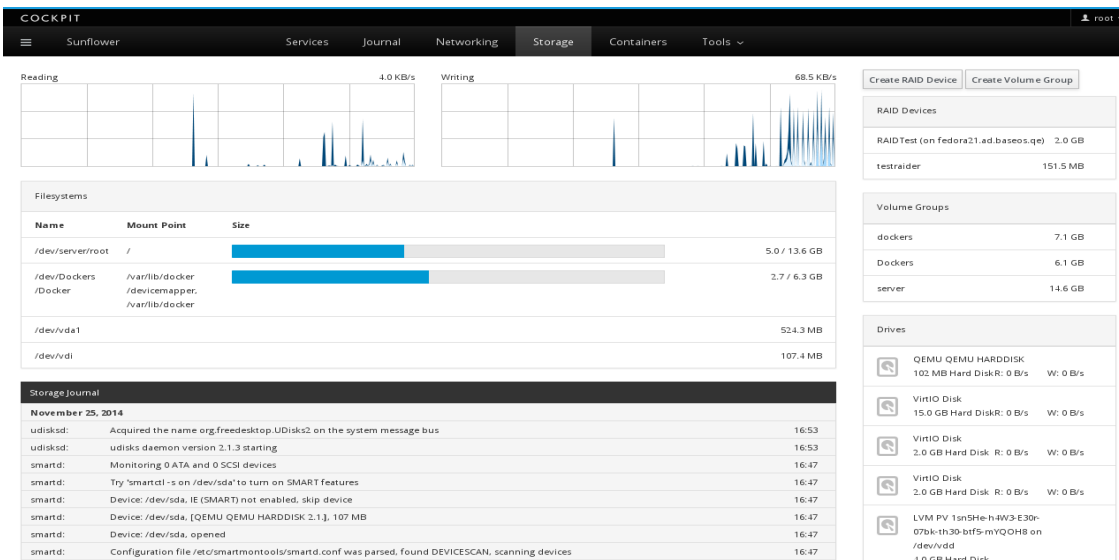
Extra

- GUI

package systemd-ui fournit une interface graphique a systemd

→ **systemadm**

- GUI distant avec **cockpit**



Extra

- Systemd dispose d'un grand nombre de binaires réalisant les tâches de base d'un OS, un exemple:
- **Systemd-nspawn** (conteneur avec/sans boot)

```
yum -y --releasever=21 --nogpg --installroot=/srv/mycontainer --disablerepo='*'
--enablerepo=fedora install systemd passwd yum fedora-release vim-minimal
# start container
systemd-nspawn -D /srv/mycontainer -b # poweroff or ctrl+] x 3 to quit
# on the host
machinectl
machinectl status mycontainer
machinectl poweroff mycontainer
```

- **Toutes les commandes systemes acceptent “-M nommachine”**
systemctl -M mycontainer

Références

- Site de documentation

Site du projet

Redhat 7 doc

Archwiki

blog de lennart Poettering

- Slides sur systemd ayant servis à la présentation (anglais)
- Slide1
- slide2

Références (tableaux)

- Cheatsheet [1,2,3,4](#)
- [Site de présentation des commandes usuelles](#)

Systemd vs SysVinit

Systemd Commands: <http://linoxide.com/linux-command/linux-systemd-commands/>

Service Related Commands

Comments	SysVinit	Systemd
Start a service	service dummy start	systemctl start dummy.service
Stop a service	service dummy stop	systemctl stop dummy.service
Restart a service	service dummy restart	systemctl restart dummy.service
Reload a service	service dummy reload	systemctl reload dummy.service
Service status	service dummy status	systemctl status dummy.service
Restart a service if already running	service dummy condrestart	systemctl condrestart dummy.service
Enable service at startup	chkconfig dummy on	systemctl enable dummy.service
Disable service at startup	chkconfig dummy off	systemctl disable dummy.service
Check if a service is enabled at startup	chkconfig dummy	systemctl is-enabled dummy.service
Create a new service file or modify configuration	chkconfig dummy --add	systemctl daemon-reload

Note : New version of systemd support "systemctl start dummy" format.

Runlevels

Comments	SysVinit	Systemd
System halt	0	runlevel0.target, poweroff.target
Single user mode	1, s, single	runlevel1.target, rescue.target
Multi user	2	runlevel2.target, multi-user.target
Multi user with Network	3	runlevel3.target, multi-user.target
Experimental	4	runlevel4.target, multi-user.target
Multi user, with network, graphical mode	5	runlevel5.target, graphical.target
Reboot	6	runlevel6.target, reboot.target
Emergency Shell	emergency	emergency.target
Change to multi user runlevel/target	telinit 3	systemctl isolate multi-user.target (OR systemctl isolate runlevel3.target)
Set multi-user target on next boot	sed s/^id:.*initdefault:/id:3:initdefault:/	ln -sf /lib/systemd/system/multi-user.target /etc/systemd/system/default.target
Check current runlevel	runlevel	systemctl get-default
Change default runlevel	sed s/^id:.*initdefault:/id:3:initdefault:/	systemctl set-default multi-user.target

Miscellaneous Commands

Comments	SysVinit	Systemd
System halt	halt	systemctl halt
Power off the system	poweroff	systemctl poweroff
Restart the system	reboot	systemctl reboot
Suspend the system	pm-suspend	systemctl suspend
Hibernate	pm-hibernate	systemctl hibernate
Follow the system log file	tail -f /var/log/messages or tail -f /var/log/syslog	journalctl -f

Systemd New Commands

Comments	Systemd
Execute a systemd command on remote host	systemctl dummy.service start -H user@host
Check boot time	systemd-analyze or systemd-analyze time
Kill all processes related to a service	systemctl kill dummy
Get logs for events for today	journalctl --since=today
Hostname and other host related information	hostnamectl
Date and time of system with timezone and other information	timedatectl

Divers

- Bash

```
# alias jxn=' journalctl -xn | vim -c "setl  
filetype=messages" -'
```

- Vim

```
cd ~/.vim  
git clone https://fedorapeople.org/cgit/wwoods/public\_git/vim-scripts.git/
```