# M3105 - Administration des Serveurs Linux: Systemd, Paquets

Guillaume Urvoy-Keller

September 24, 2020

## Systemd (System Daemon)

# "Rappel" sur les services

- Appelés aussi démon, d'où leur nom, souvent avec un "d" : sshd, snmpd
- Un service est démarré via un script
  - /etc/init.d/apache2 start – OBSOLETE
  - service apache2 start – OBSOLETE
  - systemctl start apache2
- Les services écrivent leurs messages dans le log général /var/log/syslog → **c'est là qu'il faut aller voir pour debugger**

```
root@buster:/home/vagrant# grep apache2 /var/log/syslog
Sep 21 18:58:00 buster systemd[1]: apache2.service: Control
     process exited, code=exited, status=1/FAILURE
Sep 21 18:58:00 buster systemd[1]: apache2.service: Failed
    with result 'exit-code'.
Sep 21 19:35:37 buster systemd[1]: apache2.service:
    Succeeded.
Sep 21 19:35:43 buster systemd[1]: apache2.service: Unit
    cannot be reloaded because it is inactive.
```

# Systemd: la gestion des services

- Adopté par toutes les distributions Linux depuis 2015: RHEL 7, CentOS sans oublier Debian et Ubuntu.
- Remplace SysVInit, basé sur des scripts dans /etc/init.d/...
- Fait beaucoup plus :
  - la gestion des ressources,
  - l'arrêt et le démarrage des services,
  - la détection des périphériques,
  - la journalisation,
  - la virtualisation par conteneur
- Projet initié par Lennart Poettering, un ingénieur de Red Hat.

# Systemd

- Le premier processus démarré (PID=1)
- ..qui active ensuite les autres
- Les scripts dans SysVInit pouvaient être lents si blocage car exécution séquentielle :
  Ex: attente de l'activation de la carte réseau
- Dans Systemd introduit la notion d'unité
  - Peut être vue comme un container bas niveau dans lequel s'exécute ... un service
  - ou autre chose, par exemple une socket, le montage d'un système de fichier
- Unités peuvent être parallélisées $\rightarrow$ rapidité du boot!!!
- Gestion des blocages
- Journalisation
- Gestion VM et containers via libvirt et cgroups

# Systemd: les fichiers clefs

On liste ce qui a été installé dans le paquet systemd.

Ce qui va nous intéresser : systemctl (= system control) et journalctl

```
root@buster:/home/vagrant# dpkg -L systemd
/bin
/bin/journalctl
/bin/loginctl
/bin/networkctl
/bin/systemctl
/bin/systemd-ask-password
/bin/systemd-escape
/bin/systemd-inhibit
/bin/systemd-machine-id-setup
/bin/systemd-notify
```

# Systemd: les fichiers clefs

Les fichiers des unités dans /usr/lib/systemd/system.

```
root@buster:/home/vagrant# ls /usr/lib/systemd/system/
[...]
network-pre.target  systemd-halt.service
[...]
killprocs.service       sudo.service
[...]
ssh.service     syslog.socket
```

Extensions correspondent au type de l'unité : .service, .socket, .mount, .path....

# Un exemple d'unité : le script pour Apache2

Le langage est déclaratif (décrit ce qui doit être fait) et non impératif (comment le faire) → plus compact (peu de code)

```
root@buster:/home/vagrant# more  /usr/lib/systemd/system/apache2.service
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
Documentation=https://httpd.apache.org/docs/2.4/

[Service]
Type=forking
Environment=APACHE_STARTED_BY_SYSTEMD=true
ExecStart=/usr/sbin/apachectl start
ExecStop=/usr/sbin/apachectl stop
ExecReload=/usr/sbin/apachectl graceful
PrivateTmp=true
Restart=on-abort

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

# Un exemple d'unité : le script pour Apache2

- Unit : Description et règles de dépendance vis à vis des autres unités
  Ex: démarrer le serveur après que le réseau soit opérationnel
- Service : commande effectivement lancée au démarrage via un

```
root@buster:/home/vagrant# systemctl start apache2
root@buster:/home/vagrant# systemctl stop apache2
root@buster:/home/vagrant# systemctl reload apache2
```

- WantedBy indique que le service est nécessaire à la cible multi-user.target = niveau d'exécution sans interface graphique.

# Un exemple d'unité : le script pour Apache2

```
root@buster:/home/vagrant# systemctl status apache2
 apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
      enabled)
   Active: active (running) since Mon 2020-09-21 19:08:01 GMT; 6min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 2817 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 2821 (apache2)
    Tasks: 55 (limit: 545)
   Memory: 5.1M
   CGroup: /system.slice/apache2.service
           I--2821 /usr/sbin/apache2 -k start
           I--2822 /usr/sbin/apache2 -k start
           I-- 2823 /usr/sbin/apache2 -k start

Sep 21 19:08:01 buster systemd[1]: Starting The Apache HTTP Server...
Sep 21 19:08:01 buster systemd[1]: Started The Apache HTTP Server.
```

Des infos sur l'état du serveur, mais aussi : numéro du processus, fichier de log qui le concerne, et si il démarre au boot (=enabled)

# Activation /désactivation au boot

```
root@buster:/home/vagrant# systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/
     systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service -> /lib/
     systemd/system/apache2.service.
root@buster:/home/vagrant# systemctl disable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/
     systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable apache2
Removed /etc/systemd/system/multi-user.target.wants/apache2.service.
```

# Afficher tous les services

```
root@buster:/home/vagrant# systemctl -t service
UNIT                                LOAD   ACTIVE SUB     DESCRIPTION
apache2.service                     loaded active running The Apache HTTP Server
console-setup.service               loaded active exited  Set console font and keymap
cron.service                        loaded active running Regular background program
     processing daemon
getty@tty1.service                  loaded active running Getty on tty1
ifup@eth0.service                   loaded active exited  ifup for eth0
ifupdown-pre.service                loaded active exited  Helper to synchronize boot
     up for ifupdown
keyboard-setup.service              loaded active exited  Set the console keyboard
     layout
kmod-static-nodes.service           loaded active exited  Create list of required
     static device nodes for the current kernel
networking.service                  loaded active exited  Raise network interfaces
rsyslog.service                     loaded active running System Logging Service
ssh.service                         loaded active running OpenBSD Secure Shell server
[...]
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.
```

# Journalisation

Systemd démarre un service de journalisation qui capture tous les messages par unité, mais aussi le log du boot et le log global ( syslog)

```
root@buster:/home/vagrant# journalctl -u apache2
-- Logs begin at Mon 2020-09-21 15:06:35 GMT, end at Mon 2020-09-21 19:21:48 GMT. --
Sep 21 18:58:00 buster systemd[1]: Starting The Apache HTTP Server...
Sep 21 18:58:00 buster apachectl[2534]: (98)Address already in use: AH00072:
     make_sock: could not bind to address [::]:80
Sep 21 18:58:00 buster apachectl[2534]: (98)Address already in use: AH00072:
     make_sock: could not bind to address 0.0.0.0:80
Sep 21 18:58:00 buster apachectl[2534]: no listening sockets available, shutting down
Sep 21 18:58:00 buster apachectl[2534]: AH00015: Unable to open logs
Sep 21 18:58:00 buster apachectl[2534]: Action 'start' failed.
Sep 21 18:58:00 buster apachectl[2534]: The Apache error log may have more
     information.
Sep 21 18:58:00 buster systemd[1]: apache2.service: Control process exited, code=
     exited, status=1/FAILURE
Sep 21 18:58:00 buster systemd[1]: apache2.service: Failed with result 'exit-code'.
Sep 21 18:58:00 buster systemd[1]: Failed to start The Apache HTTP Server.
Sep 21 19:08:01 buster systemd[1]: Starting The Apache HTTP Server...
Sep 21 19:08:01 buster systemd[1]: Started The Apache HTTP Server.
```

# Journalisation - le boot

```
root@buster:/home/vagrant# journalctl -b
-- Logs begin at Mon 2020-09-21 15:06:35 GMT, end at Mon 2020-09-21 19:35:56 GMT. --
Sep 21 15:06:35 buster kernel: Linux version 4.19.0-5-amd64 (debian-kernel@lists.
     debian.org) (gcc version 8.3.0 (Debian 8.3.0-7)) #1 SMP Debian 4.19.37-5
     (2019-06-19)
Sep 21 15:06:35 buster kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.19.0-5-amd64
     root=UUID=b9ffc3d1-86b2-4a2c-a8be-f2b2f4aa4cb5 ro net.ifnames=0 quiet
Sep 21 15:06:35 buster kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating
      point registers'
Sep 21 15:06:35 buster kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE
     registers'
Sep 21 15:06:35 buster kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX
     registers'
Sep 21 15:06:35 buster kernel: x86/fpu: xstate_offset[2]:  576, xstate_sizes[2]:  256
Sep 21 15:06:35 buster kernel: x86/fpu: Enabled xstate features 0x7, context size is
     832 bytes, using 'standard' format.
Sep 21 15:06:35 buster kernel: BIOS-provided physical RAM map:
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff]
      usable
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff]
      reserved
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff]
      reserved
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x0000000000100000-0x00000001ffefffff]
      usable
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x00000001fff0000-0x00000001ffffffff]
      ACPI data
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff]
      reserved
```

# Journalisation - les données de /var/log/syslog

```
root@buster:/home/vagrant# journalctl -f
-- Logs begin at Mon 2020-09-21 15:06:35 GMT. --
Sep 21 19:21:17 buster systemd[1]: Reloading.
Sep 21 19:21:48 buster systemd[1]: Reloading.
Sep 21 19:21:48 buster systemd[1]: Reloading.
Sep 21 19:21:48 buster systemd[1]: Reloading.
Sep 21 19:35:37 buster systemd[1]: Stopping The Apache HTTP
    Server...
Sep 21 19:35:37 buster systemd[1]: apache2.service: Succeeded.
Sep 21 19:35:37 buster systemd[1]: Stopped The Apache HTTP
    Server.
Sep 21 19:35:43 buster systemd[1]: apache2.service: Unit cannot
    be reloaded because it is inactive.
Sep 21 19:35:56 buster systemd[1]: Starting The Apache HTTP
    Server...
Sep 21 19:35:56 buster systemd[1]: Started The Apache HTTP
    Server.
```

# Gestion allumage/extinction/redémarrage

```
$ systemctl reboot
$ systemctl poweroff
$ systemctl suspend
```

## La gestion des paquets avec apt

# Apt

- Outil de gestion standard des paquets au dessus de dpkg pour Debian/Ubuntu
- Gère les dépendances entre paquets $\rightarrow$ évite les installations manuelles
- Permet de nommer les paquets d'après leur nom et non leur numéro de version.
  Ex: libc6 et non libc6_1.9.6-2.deb

# Apt

## Pourquoi faut-il toujours faire un update?

Pour pointer vers la bonne version dans le dépôt.
Ex: apt croit que libc6 c'est libc6_1.9.5-1.deb et non libc6_1.9.6-2.deb
$\rightarrow$ engendrera une erreur à l'installation

Nota: apt upgrade met à jour tous les paquets existants (mais ne met pas à jour la distribution Debian, par exemple Debian 9 à Debian 10).

# Apt: la gestion des sources

Dans le fichier /etc/apt/sources.list (ou dans le répertoire /etc/apt/sources.list.d)

```
root@buster:/home/vagrant# more /etc/apt/sources.list

# deb cdrom:[Debian GNU/Linux 10.0.0 _Buster_ - Official amd64 NETINST
     20190706-10:23]/ buster main
#deb cdrom:[Debian GNU/Linux 10.0.0 _Buster_ - Official amd64 NETINST
     20190706-10:23]/ buster main

# Base
deb http://deb.debian.org/debian buster main
deb-src http://deb.debian.org/debian buster main

# Security
deb http://security.debian.org/debian-security buster/updates main
deb-src http://security.debian.org/debian-security buster/updates main
```

La source peut-être un cdrom, un serveur en http ou https

# Apt: les sources

Format :

```
deb url distribution component1 component2 component3 [..] componentX
deb-src url distribution component1 component2 component3 [..] componentX
```

- Les composants sont typiquement : main, contrib, non free
- deb: les binaires
- deb-src: les sources
- les distributions peuvent avoir un type associé : oldstable, stable, testing, unstable. Défaut : stable.
- Les dépôts qui sont au minimum présents sont Base et Security.

# Apt: la gestion des sources

- Main les paquets conformes à Debian Free Software Guidelines.
- Non-free paquets non conformes à cette politique, mais qui peuvent être distribués librement.
- Contrib paquets open source mais qui ont besoin d'élements non-free

# Apt: Ajout de source externe

1. Il faut assez souvent ajouter une clef pour authenfier le dépot

```
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

2. Puis ajouter les noms des dépôts dans le fichier sources.list :

```
$ sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/debian \
   $(lsb_release -cs) \
   stable"
```

3. Puis faire un update :

```
root@buster:/home/vagrant# apt update
Hit:1 http://security.debian.org/debian-security buster/updates InRelease
Hit:2 http://deb.debian.org/debian buster InRelease
Get:3 https://download.docker.com/linux/debian buster InRelease [44.4 kB]
Get:4 https://download.docker.com/linux/debian buster/stable Packages [13.3 kB]
Fetched 57.8 kB in 0s (155 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
89 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

4. Puis apt-install docker-ce