

systemd

- Création par : [👤lagrenouille](#)
- Objet : du tuto gestionnaire de système et de services pour Linux.
- Niveau requis :
[débutant](#), [avisé](#)
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[à-placer](#)

[lien vers doublon](#)

- Testé par :

Introduction



systemd est un gestionnaire de système et de services pour Linux. Il est le système d'init par défaut dans Debian depuis Jessie.

Systemd est compatible avec les scripts d'init SysV et LSB. Il peut fonctionner comme un remplaçant de sysvinit.

Systemd :systemd (voir systemd-boot) s'appelait avant "gummibot" Le Micrologiciel : Lorsque matériellement vous allumez votre ordinateur, c'est le premier programme à s'exécuter. En anglais on parle de firmware.

Astuce : On parle rarement de micro-logiciel, (et quelque soit son genre), on dira souvent BIOS par abus de langage

* BIOS, soit Basic Input Output System / Le système élémentaire des entrées et sortie

```
systemctl status systemd-journald
```

- systemd-journald.service - Journal Service
Loaded: loaded (/lib/systemd/system/systemd-journald.service; static)
Active: active (running) since Wed 2023-04-12 18:22:15 CEST; 6 days ago
TriggeredBy: ● systemd-journald-audit.socket
● systemd-journald-dev-log.socket

```

● systemd-journald.socket
  Docs: man:systemd-journald.service(8)
        man:journald.conf(5)
Main PID: 488 (systemd-journal)
Status: "Processing requests..."
Tasks: 1 (limit: 38366)
Memory: 146.1M
CPU: 2min 3.897s
CGroup: /system.slice/systemd-journald.service
        └─488 /lib/systemd/systemd-journald

```

```

avril 12 18:22:15 lebug systemd-journald[488]: Journal started
avril 12 18:22:15 lebug systemd-journald[488]: Runtime Journal
(/run/log/journal/ba457e39339a453aa2cfa330d4a3>
avril 12 18:22:15 lebug systemd-journald[488]: Time spent on flushing to
/var/log/journal/ba457e39339a453aa2c>
avril 12 18:22:15 lebug systemd-journald[488]: System Journal
(/var/log/journal/ba457e39339a453aa2cfa330d4a3a>
Warning: journal has been rotated since unit was started, output may be
incomplete.
lines 1-21/21 (END)

```

* systemd (pour « system daemon * systemd se manipule grâce à la commande "systemctl". * On utilisera le répertoire /etc/systemd/system pour stocker ses modifications et configurations personnelles, ce qui a le gros avantage que ces fichiers ne seront pas effacés en cas de mise à jour du système.

Les unités La configuration de Systemd se base sur des unités (units) qui ont un nom et un type. Ainsi, le fichier NetworkManager.service définira l'unité de type service qui s'occupe de la gestion réseau ; de même, systemd-shutdown.socket définira la socket pour l'arrêt. La commande suivante vous listera les unités disponibles sur votre système :

```
systemctl -t help
```

Pour lister toutes les unités présentes sur le système, on fera

```
systemctl list-units
```

La commande systemd-analyze blame, permet de lister le temps de démarrage des différents services. C'est assez pratique pour déterminer le ou les services qui mettent du temps à se lancer. si je veux voir l'état du service ssh le service systemd est il en cours d'exécution, je vérifie en tapant

```
systemctl status systemd-journald
```

```

● systemd-journald.service - Journal Service
  Loaded: loaded (/lib/systemd/system/systemd-journald.service; static;
vendor preset: enabled)
  Active: active (running) since Sun 2018-06-17 07:44:12 CEST; 1 day 16h
ago
  Docs: man:systemd-journald.service(8)
        man:journald.conf(5)

```

```
Main PID: 199 (systemd-journal)
Status: "Processing requests..."
Tasks: 1 (limit: 4915)
CGroup: /system.slice/systemd-journald.service
└─199 /lib/systemd/systemd-journald
```

* systemd (pour « system daemon »)

* systemd se manipule grâce à la commande "systemctl".

* On utilisera le répertoire /etc/systemd/system pour stocker ses modifications et configurations personnelles, ce qui a le gros avantage que ces fichiers ne seront pas effacés en cas de mise à jour du système. **Les unités**

La configuration de Systemd se base sur des unités (units) qui ont un nom et un type. Ainsi, le fichier NetworkManager.service

définira l'unité de type service qui s'occupe de la gestion réseau ; de même, systemd-shutdown.socket définira la socket pour l'arrêt.

La commande suivante vous listera les unités disponibles sur votre système :

Pour lister toutes les unités présentes sur le système, on fera

```
systemctl list-units
```

La commande systemd-analyze blame, permet de lister le temps de démarrage des différents services.

C'est assez pratique pour déterminer le ou les services qui mettent du temps à se lancer.

si je veux voir l'état du service ssh

```
systemctl status sshd
```

```
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Sun 2018-06-17 07:44:22 CEST; 1 day 16h
   ago
     Process: 928 ExecReload=/bin/kill -HUP $MAINPID (code=exited,
   status=0/SUCCESS)
     Process: 923 ExecReload=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
     Process: 648 ExecStartPre=/usr/sbin/sshd -t (code=exited,
   status=0/SUCCESS)
    Main PID: 688 (sshd)
      Tasks: 1 (limit: 4915)
     CGroup: /system.slice/ssh.service
             └─688 /usr/sbin/sshd -D
```

Pour configurer un service pour qu'il soit lancé automatiquement au démarrage du système, utiliser la commande :

```
systemctl enable nom_du_service.service
```

Désactiver un service au démarrage

```
systemctl disable nom_du_service.service
```

Pour connaître le statut d'un service, utiliser la commande :

exemple pour sshd

```
sudo systemctl is-active sshd.service
```

```
active.
```

Services activés :

```
systemctl list-unit-files | grep enabled
```

Services démarrés en ce moment :

```
systemctl | grep running
```

Visualiser le journal relatif à un service particulier

```
journalctl -u apache2
```

Visualiser le journal relatif à un processus particulier

```
journalctl _PID=1992
```

```
-- Journal begins at Sat 2021-10-23 07:58:27 CEST, ends at Wed 2023-04-26
18:05:01 CEST. --
oct. 24 21:41:44 arthur audit[1992]: AVC apparmor="DENIED" operation="open"
profile="/usr/b>
-- Boot 220b6a6f14f94c9ab6979af1ed339768 --
nov. 19 07:09:56 arthur at-spi-bus-launcher[1992]: dbus-daemon[1992]:
Activating service na>
nov. 19 07:09:56 arthur at-spi-bus-launcher[1992]: dbus-daemon[1992]:
Successfully activate>
-- Boot 728c228552864bd0ab04dbc272a37c24 --
```

Visualiser le journal relatif au dernier boot (-b) avec les messages explicatifs (-x)

```
journalctl -xb
```

Visualiser les erreurs dans le journal (-p err) en commençant par la plus récente (-r)

```
journalctl -r -p err
```

Whouahah! y'a du rouge, mais rien à boire 😊

commandes

Repérer les services inutiles Pour cela, pas de recette miracle.

les principales commandes

```
systemctl
```

Affiche les 5 derniers lignes.

```
journalctl -n 5
```

```
systemctl status
```

```
systemctl --failed
```

```
journalctl -r -p err
```

```
journalctl -q
```

```
journalctl -p emerg
```

```
journalctl -p alert
```

```
journalctl -p crit
```

```
journalctl -p warning
```

```
journalctl -p notice
```

```
journalctl -p info
```

```
journalctl -p debug
```

Afficher les logs du démarrage

```
journalctl -b
```

Afficher les logs à partir de 21:00

```
journalctl - -since 21:00
```

Afficher les logs depuis hier

```
journalctl - -since yesterday
```

#journalctl -since "8 hours ago"

Afficher les logs de temps à temps

```
journalctl sudo journalctl --since "2022-12-25 18:00" --until "2022-12-26 12:30"
```

Pour afficher l'enregistrement des journaux du noyau, exécutez

```
journalctl -k
```

Afficher en temps réelle

```
journalctl -f
```

ces commandes vous amèneront à d'autres commandes, suivant les erreurs rencontrées.

Autres commandes systemd

démarrer un service systemd.

```
systemctl start application.service
```

```
systemctl stop application .service
```

Pour redémarrer un service en cours d'exécution

```
systemctl restart application .service
```

```
systemctl reload application .service
```

Si vous ne savez pas si le service intègre la fonctionnalité qui lui permet de recharger sa configuration,

vous pouvez lancer la commande **reload-or-restart** Si disponible, vous rechargerez la configuration en

place. Sinon, le service redémarrera pour récupérer la nouvelle configuration :

```
systemctl reload-or-restart application .service
```

Les commandes ci-dessus vous seront utiles pour démarrer ou arrêter des services pendant la session en cours. Vous devez les activer pour demander à systemd de lancer automatiquement les services au démarrage.

```
systemctl enable application .service
```

Cela créera un lien symbolique à partir de la copie du fichier de service du système

Pour désactiver le démarrage automatique d'un service, vous pouvez saisir :

```
systemctl disable application.service
```

Cela supprimera le lien symbolique

Pour vérifier l'état d'un service sur votre système.

```
systemctl status application.service
```

```
systemctl status apache
```

```
apache2.service          apache-htcacheclean.service
lagrenouille@lebug:~$ systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-08-26 08:57:16 CEST; 1 weeks 3 days ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1073 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Process: 742043 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
  Main PID: 1328 (apache2)
    Tasks: 11 (limit: 38369)
   Memory: 16.3G
      CPU: 8min 30.341s
   CGroup: /system.slice/apache2.service
           └─ 1328 /usr/sbin/apache2 -k start
              └─ 742049 /usr/sbin/apache2 -k start
                 └─ 742053 /usr/sbin/apache2 -k start
                    └─ 742061 /usr/sbin/apache2 -k start
                       └─ 742078 /usr/sbin/apache2 -k start
                          └─ 742080 /usr/sbin/apache2 -k start
                             └─ 742081 /usr/sbin/apache2 -k start
                                └─ 743437 /usr/sbin/apache2 -k start
                                   └─ 744516 /usr/sbin/apache2 -k start
                                      └─ 745385 /usr/sbin/apache2 -k start
                                         └─ 745386 /usr/sbin/apache2 -k start
```

si vous souhaitez vérifier si une unité est actuellement active

```
systemctl is-active application.service
```

```
systemctl is-active mysql
```

```
active
```

les options de la commande

```
systemctl list-units
```

La sortie affiche les colonnes suivantes :

UNIT: le nom de l'unité systemd

LOAD si la configuration de l'unité a été analysée par systemd . La configuration des unités chargées est gardée en mémoire.

ACTIVE: un état résumé indiquant si l'unité est active. Il s'agit généralement d'une méthode assez simple d'établir si l'unité a bien démarré ou pas.

SUB: il s'agit d'un état de niveau inférieur qui donne des informations plus détaillées sur l'unité. Il • varie souvent en fonction du type, de l'état et du mode de fonctionnement réel de l'unité.

DESCRIPTION: une courte description textuelle de ce que l'unité est et fait.

Pour afficher le fichier de l'unité que systemd a chargé sur son système, vous pouvez utiliser la

commande cat qui a été ajoutée dans la version 209 de systemd). Par exemple, pour voir le fichier de l'unité de démon de planification atd , nous pourrions saisir :

```
systemctl cat 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 graceful-stop
ExecReload=/usr/sbin/apachectl graceful
KillMode=mixed
PrivateTmp=true
Restart=on-abort

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

Pour voir une arborescence des dépendances de l'unité, vous pouvez utiliser la commande list-dependencies :

```
systemctl list-dependencies sshd.service
```

Pour consulter les propriétés de niveau inférieur d'une unité, vous pouvez utiliser la commande show

```
systemctl show sshd.service
```

pour voir les conflits que l'unité sshd.service


```
systemctl show sshd.service -p Conflicts
```

MASK nous avons vu de quelle manière arrêter ou désactiver un service

mais systemd a également la possibilité de marquer une unité comme étant totalement impossible à démarrer, automatiquement ou manuellement, On dit alors que l'on « masque » l'unité, et il est possible de le faire avec la commande mask :

```
systemctl mask apache2.service
```

```
(c'est pas malin ça sur mon serveur :-))
```

Vous ne pouvez plus relancer le service à ce stade

Pour rendre ceci possible:

```
systemctl unmask apache2.service
```

il existe bien d'autres commandes.. mais je stoppe ici...

voyez le lien en pieds de page

Visualiser l'espace disque utilisé par les journaux

```
journalctl --disk-usage
```

pour finir

En 2014, la communauté Debian a organisé un vote controversé pour remplacer le système de démarrage (SysVinit) par systemd.

Cette décision n'a pas plu à une partie importante des utilisateurs, qui ont commencé à chercher des alternatives.

Devuan, un Debian sans Systemd

Artix Linux

Gentoo Linux

MX Linux

Où sont stockés les fichiers Systemd ?

Les conf situés dans /etc/systemd. Les fichiers unités sont stockés dans le répertoire /usr/lib/systemd et ses sous-répertoires, tandis que le répertoire /etc/systemd/ et ses sous-répertoires contiennent des liens symboliques vers les fichiers unités nécessaires à la configuration locale de cet hôte.

La copie du système des fichiers unitaires est généralement conservée dans le répertoire

/lib/systemd/system.

Lorsqu'un logiciel installe des fichiers d'unité sur le système, il s'agit de l'emplacement où ils sont placés par défaut.

Qu'est-ce qu'un fichier de service sous Linux ?

Un fichier SERVICE est un fichier d'unité de service inclus avec systemd, un système init (initialisation) utilisé par diverses

distributions Linux pour amorcer l'espace utilisateur et gérer les processus. ... systemd init system est une suite de programmes inclus dans diverses distributions Linux. Le système est utilisé pour gérer différents aspects d'un serveur.

Où se trouvent les démons sous Linux ?

Linux démarre souvent les démons au démarrage. Scripts shell stockés dans /etc/init.d sont utilisés pour démarrer et arrêter les démons.

Le répertoire /lib/systemd/system contient les fichiers unitaires fournis par le système ou fournis par les packages installés.

Le répertoire /etc/systemd/system stocke les fichiers d'unité fournis par l'utilisateur.

Les logiciels à l'échelle du système utilisent souvent des fichiers de configuration stockés dans /etc, tandis que les applications

utilisateur utilisent souvent un « dotfile » – un fichier ou un répertoire dans le répertoire personnel précédé d'un point,

qui sous Unix masque le fichier ou le répertoire de la liste occasionnelle. Certains fichiers de configuration exécutent un ensemble de commandes au démarrage.

La configuration du noyau Linux se trouve généralement dans les sources du noyau dans le fichier :

/usr/src/linux/. configuration. make menuconfig – démarre un outil de configuration orienté terminal (en utilisant ncurses)

... make xconfig – démarre un outil de configuration basé sur X.

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:systeme:admin:systemd>

Last update: **28/12/2023 23:10**

