

Services

A partir de Redhat/CentOS 7 et Ubuntu 16.04 la gestion des services se fait par systemd

systemd

target boot

```
systemctl get-default      Interrogation de la target
systemctl set-default <mode> Positionne la target
```

La commande gère un lien default.target dans le répertoire [/etc/systemd/system](#) qui pointe vers [/lib/systemd/system/<target>](#).

Equivalence target/runlevel

Runlevel	Target
0	poweroff.target
1	rescue.target
2,3,4	multi-user.target
5	graphical.target
6	reboot.target

Interrogation runlevel

```
who -r      # retourne le runlevel et l'heure du dernier démarrage.
```

Déclaration des services

Emplacements

- [/etc/systemd/system](#) : services personnalisés ou modifiés localement (priorité la plus élevée)
- [/run/systemd/system](#) : services générés pendant l'exécution
- [/usr/lib/systemd/system](#) ou [/lib/systemd/system](#): services installés par les paquets du système

Création d'un service

Créer un fichier .service dans le répertoire [/etc/systemd/system](#)

[Unit]**Description**=Description **du** service

dépendances

Wants=svc1.service svc2.service # ces services seront démarrés en même temps que celui-ci qui démarrera que les précédents démarrent avec succès ou pas**Requires**=titi.service # si le service spécifié échoue ou est arrêté, celui-ci le sera également

ordre de démarrage

Before=svc1.service # le service svc1 démarrera après que celui-ci soit démarré**After**=svc2.service # ce service démarrera après le service svc2**StartLimitIntervalSec**=0 # contrôle la limitation des redémarrages**[Service]****Type**=simple # simple | forking | oneshot | notify | dbus | idle**User**=user # utilisateur avec lequel exécuter le service**Group**=group # groupe de l'utilisateur à utiliser pour le service**StandardOutput**=null # destination de la sortie console : syslog | null**StandardError**=null # destination des messages d'erreur : syslog | null**SyslogIdentifier**=appli # Identifiant dans les logs**Environment**=VAR1=VAL1 # définition de variables d'environnement**Environment**=VAR2=VAL2**EnvironmentFile**=/.../fichier # définition de variables d'environnement dans un fichier externe (attention accessibilité pas dans répertoire user)**ExecStartPre**=/bin/mkdir -p /var/run/monapplication #
exécution de commandes avant le démarrage du service**ExecStartPre**=-/bin/chown monuser:monuser /var/run/monapplication # il est possible d'ajouter un - devant la commande pour ignorer les erreurs**ExecStartPre**=/opt/monapplication/scripts/check_dependencies.sh #
exécution script avant le démarrage du service**TimeoutStartSec**=300 # temps d'attente maximum pour ExecStartPre**WorkingDirectory**=/home/pi # répertoire de travail**ExecStart**=commande # commande pour lancer le service**Restart**=always # politique de redémarrage du service si plantage**RestartSec**=20 # délai avant redémarrage**ExecStopPost**=/opt/monapplication/scripts/cleanup.sh # exécution script après l'arrêt du service

```
[Install]
WantedBy=multi-user.target # niveau pour le lancement du service
```

<https://www.freedesktop.org/software/systemd/man/latest/systemd.unit.html>

Exemple

```
[Unit]
Description=Service toto

[Service]
Type=simple
User=utilisateur
Group=groupe
ExecStart=/usr/bin/java -jar /chemin/toto.jar
EnvironmentFile=/etc/toto.conf
Restart=always
RestartSec=20
SyslogIdentifier=TOTO

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

Management des services

Afficher l'emplacement du fichier de service

```
systemctl show -p FragmentPath <service>
```

Afficher le contenu du fichier de service

```
systemctl cat <service>
```

Prise en compte modifications

```
sudo systemctl daemon-reload # recharge le gestionnaire de configuration de systemd
```

Démarrage/arrêt des services

```
sudo systemctl start <service> # démarrage du service
sudo systemctl stop <service> # arrêt du service
sudo systemctl restart <service> # redémarrage du service
```

```
systemctl help <service>           # affiche les pages d'aide associé au
service (si existent)
systemctl status <service>         # interrogation de l'état du service
```

Activation/désactivation des services au démarrage

```
sudo systemctl enable <service>    # activation du service au démarrage
sudo systemctl disable <service>   # désactivation du service au démarrage
systemctl list-unit-files *.service # liste l'état des services au
démarrage
```

Interrogations

```
systemctl status                   # état du système
systemctl list-units                # liste les services en cours d'exécution
systemctl                           # idem
systemctl --failed                 # liste les services en échec
systemctl list-unit-files           # liste tous les services installés
systemctl is-enabled <service>     # vérifie si le service est activé
```

Analyse démarrage

Pour voir les services qui prennent le plus de temps au démarrage du système utiliser la commande :

```
systemd-analyze blame
```

Debug

Fichier de log du service

Si le paramètre SyslogIdentifier est utilisé :

```
sudo journalctl -u appli           # pour voir les derniers messages
sudo journalctl -u appli -f        # pour suivre en temps réel
sudo journalctl -xe                # journaux du système
```

Analyse démarrage du système

```
systemd-analyze blame # liste les services lancé au démarrage, triés
selon leur temps d'exécution.
```

[Haut de page](#)

Ancien management des services

Démarrage/arrêt des services

```

service <nom> start      démarrage du service
service <nom> stop       arrêt du service
service <nom> restart    redémarre le service
service <nom> reload     recharge les fichiers de configuration du service
service <nom> status     interroge l'état du service

```

Activation/désactivation des services au démarrage

Redhat

```

chkconfig <service> <option>      défini l'état du service dans
/etc/rcx.d. Avec option=on|off|reset|resetpriorities
chkconfig --list <service>        liste la configuration d'un service
donné en fonction du niveau d'exécution
chkconfig --list                   liste de l'ensemble des services
chkconfig --add <service>         ajoute un service
chkconfig --del <service>         supprime le service
chkconfig --level x <service> on|off active ou désactive le service pour
les niveaux d'exécution indiqués.

```

Ubuntu

```

sudo update-rc.d <service> enable   activation du service
sudo update-rc.d <service> disable  désactivation du service

```

Ancien niveaux d'exécution system V

Niveau	Effet
0	Halt : arrête le système d'exploitation, éteint la machine
1	mode mono utilisateur pour la maintenance, mode console
2	mode multi utilisateur, sans réseau, mode console
3	mode multi utilisateur, avec réseau, mode console
4	idem que 3 laissé à la convenance de l'administrateur
5	mode multi utilisateur, avec réseau, avec interface graphique
6	reboot, redémarrage de la machine
S,s	mode single user, le mode le plus bas en cas de soucis

Le niveau d'exécution est positionné dans le fichier `/etc/inittab` :

```
id:5:initdefault:
```

Pour chaque niveau d'exécution il existe un répertoire `/etc/rcn.d` qui contient des liens symboliques vers les services, à lancer ou arrêter, présents dans `/etc/init.d`

Les liens sont de la forme `Sxx<service>` pour start ou `Kxx<service>` pour kill. `xx` défini l'ordre d'exécution (00= premier, 99=dernier), les kill sont exécutés en premier, puis tous les start.

[Haut de page](#)

From:

<https://wiki.iot-acs.fr/> - **Wiki**

Permanent link:

<https://wiki.iot-acs.fr/doku.php?id=all:bibles:linux:services>

Last update: **2026/02/26 09:22**

