

# SYSTEMD

Systemd, qui est le gestionnaire de services et de démarrage par défaut dans la plupart des distributions modernes.

## Gestion des services avec systemd :

- **Démarrer/Arrêter un service :**
  - Commandes `systemctl start`, `systemctl stop`, `systemctl restart`.
- **Activer/Désactiver un service au démarrage :**
  - Commandes `systemctl enable`, `systemctl disable`.
- **Vérification du statut d'un service :**
  - Utilisation de `systemctl status` pour vérifier l'état d'un service.
- **Affichage des services actifs :**
  - Commandes `systemctl list-units --type=service`.

## Structure des fichiers de configuration des services dans Linux avec systemd

### Fichiers d'unité de service (.service)

Les fichiers de service dans systemd se trouvent généralement dans deux répertoires principaux :

- **/etc/systemd/system/** : Répertoire pour les fichiers de service définis par l'utilisateur ou l'administrateur. Ils ont priorité sur les fichiers situés dans `/usr/lib/systemd/system`.
- **/usr/lib/systemd/system/** : Répertoire utilisé pour les fichiers de service fournis par les paquets installés via le gestionnaire de paquets.

Ces fichiers définissent comment un service doit être lancé, arrêté, et supervisé par systemd.

## Structure des fichiers `.service`

Un fichier de service est un fichier texte qui contient plusieurs sections principales. Chaque section a un rôle précis. Voici un exemple de fichier de service minimaliste, suivi de l'explication des sections.

### Exemple d'un fichier `.service`

```
[Unit]
Description=Mon Service Personnalisé
After=network.target

[Service]
ExecStart=/usr/bin/mon-script.sh
ExecStop=/usr/bin/mon-script-stop.sh
Restart=on-failure
User=monutilisateur
WorkingDirectory=/home/monutilisateur/

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

## 1. Section [Unit]

Cette section fournit une description générale du service, ainsi que des dépendances avec d'autres services ou des conditions spécifiques. Elle définit quand et comment le service doit être démarré, en tenant compte des autres composants du système.

- **Description=** : C'est un texte descriptif du service. Il apparaîtra lorsque tu vérifieras le statut du service avec `systemctl status`.
  - Ex : `Description=Mon Service Personnalisé`
- **After=** : Indique que le service doit être démarré après un autre service ou une autre cible. Par exemple, `After=network.target` signifie que ce service doit être démarré après que la couche réseau soit prête.
  - Ex : `After=network.target`
- **Requires=** et **Wants=** (optionnel) : Ces directives spécifient des dépendances obligatoires ou optionnelles. Si un service listé dans `Requires` échoue, le service principal échouera également.

## 2. Section [Service]

La section [Service] est celle qui contient la configuration du fonctionnement du service, spécifiquement comment il doit être démarré, arrêté, redémarré, et sous quel utilisateur il s'exécute.

- **ExecStart=** : Cette directive spécifie le chemin vers la commande ou le script à exécuter pour démarrer le service.
  - Ex : ExecStart=/usr/bin/mon-script.sh
- **ExecStop=** (optionnel) : Commande à exécuter pour arrêter le service proprement. C'est utile pour les services qui doivent exécuter une commande spécifique lors de l'arrêt.
  - Ex : ExecStop=/usr/bin/mon-script-stop.sh
- **Restart=** (optionnel) : Définit la politique de redémarrage du service. Par exemple, Restart=on-failure redémarre le service uniquement s'il échoue.
  - Ex : Restart=on-failure
- **User=** (optionnel) : Exécuter le service en tant qu'utilisateur spécifique. Cela peut être important pour des raisons de sécurité.
  - Ex : User=monutilisateur
- **WorkingDirectory=** (optionnel) : Spécifie le répertoire dans lequel le service sera exécuté.
  - Ex : WorkingDirectory=/home/monutilisateur/

D'autres options peuvent être utilisées :

- **Type=** : Spécifie le type du service. Les valeurs possibles incluent simple, forking, oneshot, etc. Le type simple est le plus courant, et signifie que le processus démarre immédiatement et que systemd ne doit pas attendre de retour du processus pour marquer le service comme actif.
  - Ex : Type=simple
- **TimeoutStartSec=** et **TimeoutStopSec=** : Temps d'attente avant de marquer le démarrage ou l'arrêt d'un service comme échoué.
  - Ex : TimeoutStartSec=30

### 3. Section [Install]

Cette section détermine comment et quand le service sera activé. Elle contrôle principalement le comportement lors du démarrage et du redémarrage du système.

- **WantedBy=** : Cette directive détermine sous quelle cible le service sera activé. Par exemple, multi-user.target signifie que le service sera activé dans un environnement multi-utilisateur (niveau d'exécution 3 ou 5, ce qui correspond à un environnement sans ou avec interface graphique).
  - Ex : WantedBy=multi-user.target

Tu peux aussi avoir la directive RequiredBy=, qui précise que le service est requis par d'autres services ou cibles.