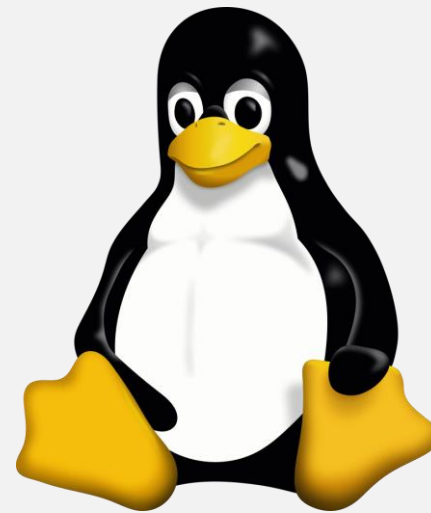


Intégrer des
services réseau
multi-
plateformes

300

Systeme de
fichiers





Applications

Applications - Gestionnaire de paquets

- But : installer, mettre à jour des applications
- Principe : Les applications sont disponibles dans des **dépôts logiciel** ou **software repositories**
- Le Play store de google, l'App store de Apple et le Microsoft store sont des dépôts avec un fonctionnement similaire (le principe a été directement repris de Linux).
- Sous Linux (contrairement au Play store ou à l'App store), les **dépôts** sont **configurables**.



dpkg – Gestionnaire bas niveau

- Outil de base pour gérer les paquets .deb.
- **Ne gère pas les dépendances.**
- Souvent utilisé par APT ou pour installer un .deb téléchargé.
- Précis mais moins convivial.

- Exemple d'utilisation :

```
# Installer un paquet local
sudo dpkg -i paquet.deb

# Supprimer un paquet (sans supprimer config)
sudo dpkg -r nom_du_paquet

# Supprimer avec configuration
sudo dpkg -P nom_du_paquet

# liste les paquets
sudo dpkg -l
```

APT – Gestionnaire haut niveau

- Outil plus convivial pour gérer les paquets .deb.
- Télécharge depuis les dépôts, installe et **gère les dépendances**.
- Recommandé pour la plupart des installations.

- Commandes fréquentes avec APT :

*Note: **apt-get** est l'ancêtre d'**apt**.*

*→ Vous trouverez encore souvent des exemples avec **apt-get** sur le web. **apt-get** est toujours supporté.*

```
# Met à jour la liste des paquets disponibles  
sudo apt update
```

```
# Met à jour les paquets installés  
sudo apt upgrade
```

```
# installe un logiciel  
sudo apt install nom_du_paquet
```

```
# supprime un logiciel  
sudo apt remove nom_du_paquet
```

Snap – Paquets universels



- Développé par Canonical (Ubuntu).
- Fonctionne sur plusieurs distributions Linux.
- Inclut toutes les dépendances (paquet autonome).
- S'exécute dans un bac à sable (sandbox).

- Exemple de commande :

```
# Rechercher une application
snap find nom_application

# Installer une application
sudo snap install nom_application

# Mettre à jour toutes les applications Snap
sudo snap refresh
```



Mises à jour

Mises à jour du système et des applications

Question : Comment automatiser les mises à jour ?

- Via APT

```
#met à jour à la dernière version
$sudo apt update && apt install <package-name>
#met à jour à la version souhaitée
$sudo apt update && apt install <package-name>=<version-number>
#met à jour toutes les applications gérées par APT
$sudo apt update && apt upgrade
```

- Via SNAP

```
#met à jours toutes les applications gérées par snap
$sudo snap refresh
#met à jours une application gérée par snap
$sudo snap refresh <nom-du-snap>
```

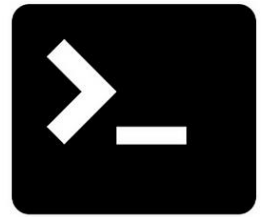

Mise à jour de sécurité

L'objectif principal d'**unattended-upgrades** est de garantir que les systèmes restent à jour en termes de correctifs de sécurité sans intervention manuelle, ce qui est crucial pour maintenir la sécurité du système.

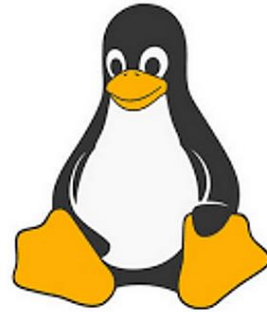
 **A utiliser sur les serveurs de production pour éviter les effets de bord.**

- Configuration : Configurez les paramètres dans le fichier **/etc/apt/apt.conf.d/50unattended-upgrades**. Vous pouvez y spécifier quels types de mises à jour (comme les mises à jour de sécurité) doivent être installés automatiquement.
- Planification : Ajustez la fréquence des mises à jour automatiques dans le fichier **/etc/apt/apt.conf.d/20auto-upgrades**.





systemd



Les services

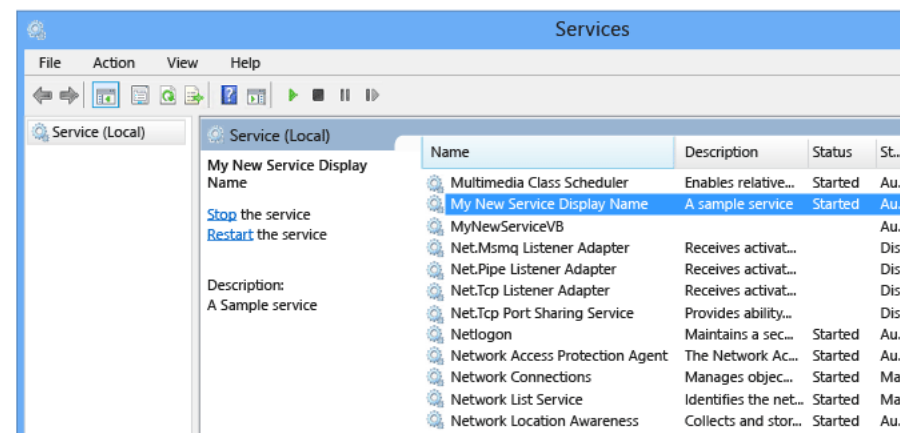
Gestion des services avec systemd

- **systemd** est le système d'initialisation et de gestion des services par défaut sous Ubuntu.
- Il permet de démarrer, arrêter, activer ou désactiver des services.
- Les commandes se font principalement avec **systemctl**.

Sur Ubuntu:

```
team@itsfoss: ~  
$ systemctl list-units --type=service --state=running  
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION  
accounts-daemon.service            loaded active running Accounts Service  
acpi-fakekey.service               loaded active running ACPI fakekey daemon  
acpid.service                       loaded active running ACPI event daemon  
avahi-daemon.service                loaded active running Avahi mDNS/DNS-SD Stack  
chrony.service                     loaded active running chrony, an NTP client/server  
cron.service                       loaded active running Regular background program processing daemon  
cups-browsed.service                loaded active running Make remote CUPS printers available locally  
cups.service                       loaded active running CUPS Scheduler  
dbus.service                       loaded active running D-Bus System Message Bus  
haveged.service                    loaded active running Entropy Daemon based on the HAVEGE algorithm  
ModemManager.service               loaded active running Modem Manager  
NetworkManager.service             loaded active running Network Manager  
nfs-blkmap.service                  loaded active running pNFS block layout mapping daemon  
nfs-idmapd.service                  loaded active running NFSv4 ID-name mapping service  
nfs-mountd.service                  loaded active running NFS Mount Daemon  
nfsdclld.service                    loaded active running NFSv4 Client Tracking Daemon  
nmbd.service                        loaded active running Samba NMB Daemon  
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
```

Equivalent Windows:



Commandes systemctl – Services

- Commandes principales pour gérer les services :

```
# Vérifier l'état d'un service
systemctl status ssh.service

# Démarrer / Arrêter / Redémarrer un service
sudo systemctl start apache2.service
sudo systemctl stop apache2.service
sudo systemctl restart apache2.service

# Activer / Désactiver au démarrage
sudo systemctl enable apache2.service
sudo systemctl disable apache2.service

# Lister tous les services
systemctl list-unit-files --type=service
```

Note: le « .service » n'est pas toujours obligatoire

Planification de tâches

JANVIER

08 Lundi
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22

- Paint Nail
- organisation de la semaine
- Inscription Salon
- Repas avec Charlotte
- Sport

09 Mardi
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22

- élancements
- Formation Budget
- Reunion
- Vacances
- Expedition
- Point Vente

10 Mercredi
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22

- Repos avec les enfants
- Panier de légumes + Batchcooking

11 Jeudi
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22

- élancements
- RDV Imprimeur
- Sport

12 Vendredi
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22

- Repas d'équipe pour la nouvelle année
- Balade en Forêt

13 Samedi
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22

- Bilan de la semaine + orga de la semaine pro
- Soirée Ciné

14 Dimanche
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22

- BATCHCOOKING
- Dhal de lentilles
- Soupe de Potimarron
- Légumes rôtis

URGENT

- Inscription Salon
- Idées de repas pour la semaine prochaine
- Parcours Balade en Forêt
- Commander repas de vendredi chez Petite Nature

Janvier
L M M J V S D
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

Automatiser les mises à jour - Cron

- 2 fichiers pour stocker les informations :
 - Tâches de l'utilisateur -> /var/spool/cron/crontabs/
 - Tâches globales (système) -> /etc/crontab
- Pour éditer les tâches : crontab -e
- Pour afficher les tâches : crontab -l
- Syntaxe : mm hh d m dow [user] command
 - mm = minute
 - hh = heure
 - d = jour du mois
 - m = mois (1 à 12)
 - dow = jour de la semaine (1 à 7)
 - command = commande à exécuter
- Exemples (à faire sur serveur de test, pas de production):

```
0 20 * * * sudo apt update && sudo apt upgrade -y
```


Automatiser les mises à jour - Cron

```
# syntax of cron
# _____ sec (0 - 59)
# |_____ min (0 - 59)
# ||_____ hour (0 - 23)
# |||_____ day (1 - 31)
# |||_____ month (1 - 12)
# |||_____ weekday (0 - 6)
#
#
#
#
#
#
#
# * * * * * user-name command to execute
# 0 0 0 * * 6 root /scripts/have_fun
```

Special characters :

- * any value
- , value list separator
- range of values
- / step values

Pour vous aider à définir la période d'exécution

<https://cron.help/>

Cron Helper Crontab syntax for us humans

Every 5 minutes

`*/5 * * * *`

Minutes	Hours	Day of Month	Month	Day of Week
:00, :05, :10, :15, :20, :25, :30, :35, :40, :45, :50, and :55	all	all	all	all

- * Expands to all values for the field
- , List separator
- Range separator
- / Specifies step for ranges
- @hourly Run at the start of each hour
- @daily Run every day at midnight UTC
- @weekly Run at every Sunday at midnight UTC
- @monthly Run on the 1st of each month at midnight UTC
- @yearly Run on Jan 1st at midnight UTC
- @annually Same as @yearly