# Administration Système Linux

---



## 1. Prise en main de Linux

### 1.1 Dossiers et configuration

- `/etc` : contient les fichiers de configuration du système (réseau, services, utilisateurs, etc.).
- `/var` : contient les données variables, notamment les fichiers de log dans `/var/log`.

### 1.2 Gestion des utilisateurs

- `adduser nom` : ajoute un nouvel utilisateur avec création de répertoire personnel.
- `groups nom` : affiche les groupes auxquels appartient un utilisateur.

### 1.3 Droits et permissions

- Les droits sont notés en **lecture (r=4), écriture (w=2), exécution (x=1)**.
- Exemple : `chmod 755 fichier` signifie :
  - Propriétaire : lecture, écriture, exécution.
  - Groupe : lecture, exécution.
  - Autres : lecture, exécution.
- `chown -R utilisateur:groupe /chemin` change le propriétaire et le groupe.

### 1.4 Services système

- `systemctl start ssh` : démarre le service SSH.
- `systemctl enable ssh` : active SSH au démarrage du système.
- `journalctl -xe` : consulte les logs système détaillés.

---

# 2. Réseau sous Linux

## 2.1 Commandes de base

- `ip a` : affiche les interfaces réseau et leurs IP.
- `ping google.com` : vérifie la connectivité en passant par le DNS.
- `ping 8.8.8.8` : teste la connectivité sans DNS.

## 2.2 Résolution DNS

- `nslookup nom` / `dig nom` / `host nom` : résolvent les noms en IP.

## 2.3 Diagnostic réseau

- `traceroute nom` : affiche le chemin réseau vers une IP.
- `ss -tuln` : liste les ports en écoute (t: TCP, u: UDP, l: listen, n: IP numériques).

---

# 3. Pare-feu : iptables & ufw

## 3.1 Concepts

- **INPUT** : trafic entrant vers le système.
- **OUTPUT** : trafic sortant.
- **FORWARD** : trafic traversant la machine (en mode routeur).

## 3.2 iptables

- `iptables -P INPUT DROP` : bloque tout le trafic entrant par défaut.
- `iptables -A INPUT -p tcp --dport 22 -j ACCEPT` : autorise SSH.
- `-m conntrack --ctstate ESTABLISHED,RELATED` : autorise les connexions existantes.
- `iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE` : fait du NAT (utile pour les passerelles).

## 3.3 ufw (Uncomplicated Firewall)

- `ufw allow 443/tcp` : autorise HTTPS.

- `ufw enable` / `ufw status` : active ou consulte l'état du pare-feu.

---

# 4. SSH, DNS, DHCP et transferts

## 4.1 SSH (Secure Shell)

- `ssh utilisateur@ip -p 2222` : connexion SSH sur port personnalisé.
- Config dans `/etc/ssh/sshd_config`.

## 4.2 DNS

- Sert à résoudre un nom de domaine en IP.
- Utilise souvent 8.8.8.8 comme serveur DNS (Google).

## 4.3 DHCP

- Attribue dynamiquement une IP aux machines.
- Le client envoie une requête, le serveur lui propose une IP et la configuration réseau.

## 4.4 SCP

- `scp fichier utilisateur@ip:/chemin` : transfert de fichier via SSH.
- `scp -r dossier utilisateur@ip:/chemin` : transfert de dossier récursif.

---

# 5. Serveur Web Apache + PHP

## 5.1 Installation

```
apt update
apt install apache2 php libapache2-mod-php
```

## 5.2 Configuration

- Les fichiers du site sont dans `/var/www/html/`.
- Modifier ou créer un VirtualHost dans `/etc/apache2/sites-available/mon-site.conf`.
- Activer avec `a2ensite mon-site` puis `systemctl reload apache2`.

## 5.3 Permissions

```
chown -R www-data:www-data /var/www/html/
chmod -R 755 /var/www/html/
```

### Tester PHP

- Créer un fichier `info.php` avec :

```
<?php phpinfo(); ?>
```

- Puis y accéder via le navigateur.

### VirtualHosts

- Permet d'héberger plusieurs sites web sur la même IP (ex: `site1.local`, `site2.local`).
- Nécessite de modifier `/etc/hosts` sur le client :

```
192.168.1.100 site1.local site2.local
```

---

## 6. Serveur FTP avec vsftpd

### 6.1 Installation

```
apt install vsftpd
```

### 6.2 Configuration

- Fichier : `/etc/vsftpd.conf`

Options utiles :

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
chroot_local_user=YES
```

- Activer TLS :
  `ssl_enable=YES`

```
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

### 6.3 Accès

- Port par défaut : **21**
- Utilisable avec **FileZilla**.

- Vérifier que le pare-feu autorise le port 21 :

```
ufw allow 21/tcp
```

---

# 7. Partage de fichiers (Samba et NFS)

## 7.1 Samba

- Partage accessible depuis Windows.
- Installer :

```
apt install samba
```

- Configurer `/etc/samba/smb.conf` :

```
[partage]
   path = /srv/partage
   read only = no
   browsable = yes
```

- Redémarrer :

```
systemctl restart smbd
```

## 7.2 NFS (entre machines Linux)

- Installer :

```
apt install nfs-kernel-server
```

- Déclarer le partage dans `/etc/exports` :

```
/srv/partage 192.168.1.0/24(rw,sync,no_subtree_check)
```

- Activer :

```
exportfs -a
systemctl restart nfs-kernel-server
```

- Sur le client :

```
apt install nfs-common
mount 192.168.1.100:/srv/partage /mnt
```

# 8. Sauvegardes sous Linux

---

## 8.1 – Pourquoi sauvegarder ?

Sauvegarder consiste à copier des données importantes afin de pouvoir les restaurer en cas de perte, de panne, d'erreur humaine ou de cyberattaque. C'est une **étape essentielle** dans l'administration système.

**Objectifs :**

- Préserver l'intégrité et la disponibilité des données.
- Permettre un retour arrière (rollback) en cas de problème.
- Garantir la continuité d'activité (PRA/PCA).

**Bonnes pratiques :**

- Sauvegarder **régulièrement** (quotidiennement ou hebdomadairement).
- Stocker les sauvegardes sur **un support externe ou distant**.
- Conserver plusieurs versions (rétention).
- **Tester la restauration** de temps en temps.

---

## 8.2 – Sauvegarde locale avec `cp` ou `rsync`

**Avec `cp` (simple mais peu efficace pour les grosses structures) :**

```
cp -r /etc /home/user/sauvegardes/etc_backup
```

**Avec `rsync` (plus rapide, synchronise efficacement) :**

```
rsync -av --delete /etc /home/user/sauvegardes/
```

- `-a` : archive (conserve les permissions, dates, etc.).
- `-v` : verbose (affiche les détails).
- `--delete` : supprime les fichiers disparus de la source.

---

## 8.3 – Sauvegarde compressée avec `tar`

```
tar -czvf sauvegarde-home.tar.gz /home/user
```

- `-c` : création.
- `-z` : compression gzip.
- `-v` : verbose.
- `-f` : fichier de destination.

Pour extraire :

```
tar -xzvf sauvegarde-home.tar.gz
```

---

### 8.4 – Sauvegarde automatique avec `cron`

Éditer la crontab :

```
crontab -e
```

Exemple pour lancer un script de sauvegarde tous les jours à 2h du matin :

```
0 2 * * * /home/user/scripts/backup.sh
```

Contenu possible du script `backup.sh` :

```
#!/bin/bash
DATE=$(date +%F)
tar -czf /sauvegardes/home-$DATE.tar.gz /home/user
```

Ne pas oublier de rendre le script exécutable :

```
chmod +x /home/user/scripts/backup.sh
```

---

### 8.5 – Sauvegarde distante avec `scp` ou `rsync`

**Avec `scp` :**

```
scp sauvegarde.tar.gz user@192.168.1.100:/home/user/backups/
```

**Avec `rsync` (plus efficace) :**

```
rsync -avz /home/user/sauvegardes/
user@192.168.1.100:/home/user/backup/
```

> Astuce : utiliser une **clé SSH** pour éviter d'avoir à entrer le mot de passe.

---

### 8.6 – Sauvegarde incrémentielle avec `rsync`

Rsync ne recopie que les fichiers **modifiés** :

```
rsync -av --delete /var/www/ /media/usb/backup-www/
```

Pour une stratégie plus poussée, on peut combiner `rsync` avec `--link-dest` ou utiliser **`rsnapshot`** (outil basé sur rsync).

---

### 8.7 – Outils professionnels de sauvegarde

- **Bacula** / **Bareos** : systèmes de sauvegarde en réseau.
- **Timeshift** : snapshots pour postes Linux (desktop).
- **Rsync + cron** : combinaison simple et puissante pour serveurs.
- **Déduplication avec BorgBackup** ou **Restic**.

---

### 8.8 – Restauration

Restaurer une archive :

```
tar -xzvf sauvegarde-home.tar.gz -C /
```

Restaurer un dossier avec `rsync` :

```
rsync -av /sauvegardes/etc/ /etc/
```

---

### 8.9 – Cas d’usage

- **/etc/** : contient les fichiers de configuration.
- **/home/** : contient les données utilisateur.
- **/var/www/** : sites web hébergés par Apache/Nginx.
- **/var/lib/mysql/** : bases de données (à ne pas copier "à chaud").

---

# 9. Supervision avec Zabbix

---

## 9.1 – Qu’est-ce que Zabbix ?

**Zabbix** est une plateforme de supervision **open-source** qui permet de :

- Suivre la santé des serveurs, postes clients, services et équipements réseau.
- Collecter des métriques (charge CPU, RAM, espace disque, état des services, etc.).
- Générer des alertes automatiquement en cas de problème.
- Visualiser les données avec des **graphiques, tableaux de bord et cartes réseau**.
- Envoyer des notifications par e-mail, SMS, webhook...

---

## 9.2 – Architecture de Zabbix

Zabbix est composé de plusieurs éléments :

- **Serveur Zabbix** : le cœur du système.
- **Base de données** : stocke les données collectées.
- **Frontend Web** : interface graphique pour gérer la supervision.
- **Agents Zabbix** : installés sur les machines à surveiller.
- **Proxies Zabbix** (facultatifs) : pour surveiller à distance.

---

## 9.3 – Installation de Zabbix sur Ubuntu Server 22.04

### Étape 1 – Ajouter le dépôt officiel

```
wget
https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release
/zabbix-release_6.0-4+ubuntu22.04_all.deb
sudo dpkg -i zabbix-release_6.0-4+ubuntu22.04_all.deb
sudo apt update
```

### Étape 2 – Installer les composants

```
sudo apt install zabbix-server-mysql zabbix-frontend-php
zabbix-apache-conf zabbix-sql-scripts zabbix-agent mariadb-server
```

### Étape 3 – Créer la base de données Zabbix

```
sudo mysql -u root -p
```

Dans MariaDB :

```
CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'motdepasse';
GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

### Étape 4 – Importer le schéma de base

```
zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -u zabbix -p zabbix
```

### Étape 5 – Configurer le serveur Zabbix

Éditer le fichier :

```
sudo nano /etc/zabbix/zabbix_server.conf
```

Changer la ligne suivante :

```
DBPassword=motdepasse
```

### Étape 6 – Démarrer les services

```
sudo systemctl restart zabbix-server zabbix-agent apache2
sudo systemctl enable zabbix-server zabbix-agent apache2
```

---

## 9.4 – Accès à l'interface web

Accéder via un navigateur à l'adresse :
`http://IP_DU_SERVEUR/zabbix`

Suivre l'installation graphique :

- DB Host : `localhost`
- DB Name : `zabbix`
- User : `zabbix`
- Password : `motdepasse`
- Timezone : `Europe/Paris`

Identifiants par défaut :

- Nom : `Admin`

- Mot de passe : `zabbix`

---

## 9.5 – Ajouter un hôte (machine à surveiller)

### Sur le client à surveiller :

```
sudo apt install zabbix-agent
```

Configurer :

```
sudo nano /etc/zabbix/zabbix_agentd.conf
```

Modifier :

```
Server=IP_DU_SERVEUR_ZABBIX
```

Redémarrer le service :

```
sudo systemctl restart zabbix-agent
sudo systemctl enable zabbix-agent
```

### Depuis l'interface web :

- Aller dans : *Configuration → Hosts → Create Host*
- Donner un nom, une IP
- Ajouter un template (ex : **Linux by Zabbix agent**)

---

## 9.6 – Fonctions principales

- **Triggers** : alertes automatiques (ex : CPU > 90%)
- **Templates** : modèles de surveillance prêts à l'emploi
- **Cartes réseau** : visualisation des hôtes
- **Graphiques** : CPU, RAM, disque, etc.
- **Alertes** : mails, SMS, webhook...

---

## 9.7 – Sécurisation

- Changer le mot de passe `Admin`
- Restreindre l'accès à l'interface Web
- Utiliser **TLS** entre serveur et agents (optionnel)

---

### 9.8 –Dépannage et test

Commandes utiles :

```
systemctl status zabbix-server
tail -f /var/log/zabbix/zabbix_server.log
zabbix_get -s IP_CLIENT -k system.hostname
```