

6 - Réseau, Web et accès distant

Contrôler la configuration réseau, tester DNS et ports, télécharger et administrer à distance.

- [6.1 - Adresses IP et tests de connectivité](#)
- [6.2 - DNS, routes et ports réseau](#)
- [6.3 - Requêtes et téléchargements avec curl et wget](#)
- [6.4 - SSH, SCP et rsync](#)

6.1 - Adresses IP et tests de connectivité

Le diagnostic réseau doit progresser du local vers le distant : interface, adresse, route, passerelle, DNS, port puis application.

Afficher les interfaces et les adresses

```
ip -brief link
ip -brief address
ip address show
ip link show enp1s0
```

Activer ou désactiver une interface temporairement :

```
sudo ip link set enp1s0 up
sudo ip link set enp1s0 down
```

Sur une machine distante, désactiver l'interface utilisée par SSH coupe immédiatement la connexion.

Afficher les routes

```
ip route
ip -6 route
ip route get 1.1.1.1
```

`ip route get` indique la route, l'interface et l'adresse source choisies par le noyau.

Tester la pile locale et une destination

```
ping -c 4 127.0.0.1
ping -c 4 192.168.1.1
ping -c 4 1.1.1.1
ping -c 4 example.com
```

Interprétation progressive :

1. `127.0.0.1` teste la pile IP locale.
2. La passerelle teste le lien au réseau local.
3. Une IP Internet teste le routage sans dépendre du DNS.
4. Un nom de domaine teste aussi la résolution DNS.

Certains équipements bloquent ICMP. Un échec de `ping` ne prouve donc pas à lui seul que le service est indisponible.

Voir le chemin suivi

```
tracpath example.com
traceroute example.com
```

`tracpath` est souvent utilisable sans privilèges. Les étoiles peuvent correspondre à un routeur qui ne répond pas aux sondes, sans que le trafic final soit nécessairement bloqué.

NetworkManager

Si la machine utilise NetworkManager :

```
nmcli device status
nmcli connection show
nmcli device show enp1s0
```

Statistiques et erreurs d'interface

```
ip -s link show enp1s0  
sudo ethtool enp1s0
```

Rechercher pertes, erreurs, négociation de vitesse et état du lien. `ethtool` peut nécessiter une installation séparée.

Les commandes `ip` modifient souvent l'état en mémoire seulement. Pour une configuration persistante, utiliser l'outil de la distribution : NetworkManager, systemd-networkd, Netplan ou les fichiers réseau prévus.

6.2 - DNS, routes et ports réseau

Vérifier la résolution DNS

```
getent hosts example.com
resolvectl query example.com
dig example.com
dig +short example.com
dig @1.1.1.1 example.com
```

`getent` suit la configuration complète du système, y compris `/etc/hosts`, DNS et éventuellement un annuaire. `dig` interroge directement le DNS et fournit davantage de détails.

Afficher la configuration DNS gérée par systemd-resolved :

```
resolvectl status
```

Examiner aussi :

```
cat /etc/resolv.conf
grep '^hosts:' /etc/nsswitch.conf
```

Ne pas modifier `/etc/resolv.conf` avant de vérifier s'il est généré automatiquement.

Afficher la table de routage et les voisins

```
ip route
ip neigh
```

`ip neigh` remplace les usages courants de l'ancienne commande `arp`.

Voir les ports en écoute

```
ss -tuln
sudo ss -tulpn
ss -tan state established
```

Options : `t` TCP, `u` UDP, `l` écoute, `n` valeurs numériques, `p` processus.

Exemples ciblés :

```
sudo ss -ltnp 'sport = :22'
sudo lsof -iTCP:8080 -sTCP:LISTEN
```

Tester un port distant

```
nc -vz serveur.example 22
nc -vz 192.168.1.10 443
```

Pour HTTPS, tester aussi le protocole réel :

```
curl -I --connect-timeout 5 https://serveur.example
openssl s_client -connect serveur.example:443 -servername serveur.example </dev/null
```

Un port ouvert ne garantit pas que l'application fonctionne correctement. À l'inverse, un délai peut venir du pare-feu, du routage ou d'un filtrage intermédiaire.

Ordre de diagnostic conseillé

Interface UP → adresse IP → route par défaut → passerelle → DNS → port → application

Commandes correspondantes :

```
ip -brief address
ip route
```

```
ping -c 2 adresse-passerelle  
getent hosts serveur.example  
nc -vz serveur.example 443  
curl -I https://serveur.example
```

Cette progression permet de séparer un problème réseau d'un problème applicatif au lieu de modifier plusieurs couches en même temps.

6.3 - Requêtes et téléchargements avec curl et wget

`curl` envoie des requêtes et affiche ou enregistre la réponse. `wget` est particulièrement pratique pour télécharger des fichiers de manière non interactive.

Utiliser curl

Afficher une page :

```
curl https://example.com
```

Afficher uniquement les en-têtes :

```
curl -I https://example.com
```

Suivre les redirections et enregistrer :

```
curl -L -o fichier.zip https://example.com/fichier.zip  
curl -LO https://example.com/fichier.zip
```

Échouer si le serveur renvoie une erreur HTTP :

```
curl --fail --show-error --location -o fichier.zip URL
```

Afficher les détails de connexion :

```
curl -v https://example.com  
curl -w '\nHTTP=%{http_code} Temps=%{time_total}s\n' -o /dev/null -sS https://example.com
```

Tester une API JSON :

```
curl --fail --silent --show-error \  
-H 'Accept: application/json' \  
https://api.example.com/status
```

Ne placez pas un jeton secret directement dans une capture d'écran, l'historique du shell ou une documentation partagée.

Utiliser wget

```
wget https://example.com/fichier.iso  
wget -O image.iso https://example.com/fichier.iso  
wget -c https://example.com/gros-fichier.iso
```

`-c` tente de reprendre un téléchargement interrompu si le serveur le permet.

Contrôler un téléchargement

```
file fichier.zip  
sha256sum fichier.zip
```

Comparer la somme SHA-256 avec une valeur obtenue depuis une source officielle indépendante :

```
echo 'SOMME_ATTENDUE fichier.zip' | sha256sum -c -
```

Sécurité TLS

Éviter les options `curl -k` ou `wget --no-check-certificate`, qui désactivent la validation du certificat. Sur un réseau interne, installer l'autorité de certification correcte constitue la solution durable.

Ne pas exécuter directement un téléchargement

Éviter :

```
curl URL | sudo sh
```

Procédure plus sûre : télécharger, vérifier la source et la somme, lire le script, puis l'exécuter avec les droits strictement nécessaires.

6.4 - SSH, SCP et rsync

SSH permet d'administrer une machine distante de manière chiffrée. SCP copie des fichiers via SSH et rsync synchronise efficacement des arborescences.

Se connecter en SSH

```
ssh utilisateur@serveur.example  
ssh -p 2222 utilisateur@serveur.example  
ssh utilisateur@192.168.1.20
```

Exécuter une commande distante :

```
ssh utilisateur@serveur.example 'hostname && uptime'
```

Créer une clé moderne

```
ssh-keygen -t ed25519 -a 64 -C "alex-poste"  
ssh-copy-id utilisateur@serveur.example
```

Tester ensuite la connexion avant de désactiver une authentification par mot de passe.

Permissions recommandées :

```
chmod 700 ~/.ssh  
chmod 600 ~/.ssh/id_ed25519  
chmod 644 ~/.ssh/id_ed25519.pub  
chmod 600 ~/.ssh/authorized_keys
```

Ne jamais partager la clé privée `id_ed25519`.

Copier avec scp

```
scp rapport.pdf utilisateur@serveur:/srv/documents/  
scp utilisateur@serveur:/var/log/app.log .  
scp -r dossier/ utilisateur@serveur:/srv/sauvegarde/
```

Synchroniser avec rsync

Simulation locale :

```
rsync -avhn source/ destination/
```

Exécution :

```
rsync -avh --progress source/ destination/  
rsync -avh --progress source/ utilisateur@serveur:/srv/destination/
```

La barre finale change le sens :

```
rsync -avh source/ destination/  # contenu de source  
rsync -avh source destination/  # dossier source lui-même
```

Synchronisation miroir avec suppression :

```
rsync -avhn --delete source/ destination/  # toujours simuler  
rsync -avh --delete source/ destination/
```

`--delete` supprime dans la destination ce qui n'existe plus dans la source. Vérifier les chemins et la simulation avant l'exécution.

Diagnostic SSH

```
ssh -v utilisateur@serveur.example  
ssh -G serveur.example | less
```

Vérifier côté serveur :

```
sudo systemctl status ssh  
sudo ss -ltnp 'sport = :22'  
sudo journalctl -u ssh -n 100
```

Sur Fedora ou Arch, le service peut s'appeler `sshd`. Une empreinte d'hôte modifiée doit être vérifiée auprès de l'administrateur avant de supprimer l'ancienne entrée de `known_hosts`.