

## Chapitre 3

# Configuration du système

### 1. Configuration du réseau

La configuration du réseau sur les systèmes Linux est une compétence fondamentale pour les administrateurs de systèmes, les ingénieurs réseau et toute personne intéressée par la gestion des technologies de l'information. Cette opération consiste à ajuster un ensemble de paramètres et de services pour permettre à un ordinateur sous Linux de communiquer sur un réseau informatique. À la base de cette configuration se trouvent divers fichiers de configuration, outils en ligne de commande et interfaces graphiques, permettant de contrôler le comportement réseau de la machine.

Linux offre une flexibilité remarquable en matière de configuration réseau grâce à son architecture ouverte et à la variété d'outils disponibles. Parmi ces outils, on retrouve `ifconfig` et `ip`, qui servent à configurer les interfaces réseau, ainsi que `netplan` ou `NetworkManager`, plus récents, qui offrent une approche plus automatisée et conviviale pour gérer les réseaux. Chaque distribution Linux peut proposer ses propres outils ou préférences pour la gestion du réseau, ce qui souligne l'importance de consulter la documentation spécifique à chaque distribution.

La configuration réseau sur Linux peut également impliquer la mise en place de services réseau plus avancés, tels que le routage IP, le pare-feu avec `iptables` ou `nftables`, et la sécurité des communications avec des outils comme OpenSSL pour la mise en place de VPN ou de connexions sécurisées. La compréhension de ces services et de leur configuration est cruciale pour assurer la sécurité et l'efficacité du réseau. Nous étudierons en détail ces concepts dans un chapitre ultérieur.

Il est nécessaire de maîtriser la configuration du réseau sur Linux pour une gestion robuste et sécurisée des infrastructures informatiques. Que ce soit pour des serveurs, des postes de travail ou des systèmes embarqués, Linux offre les outils et la flexibilité nécessaires pour répondre à presque tous les besoins en matière de réseau. Avec l'évolution constante de la technologie réseau, rester informé des dernières pratiques et outils disponibles sous Linux est indispensable pour tout professionnel du secteur.

La gestion et le diagnostic des réseaux informatiques sous Linux impliquent une suite d'outils puissants conçus pour aider les administrateurs à configurer, surveiller et résoudre les problèmes de réseau. Chacun de ces outils joue un rôle crucial dans l'arsenal des administrateurs réseau, leur permettant de gérer efficacement les interfaces réseau, de surveiller les connexions et le flux de données, et de diagnostiquer divers problèmes de réseau.

`ifconfig` et `ip` sont essentiels pour la configuration des interfaces réseau. `ifconfig`, bien que considéré comme obsolète dans les distributions modernes, était traditionnellement utilisé pour configurer les interfaces réseau, définir les adresses IP et gérer les états des interfaces. Son successeur, `ip`, offre des fonctionnalités plus robustes et détaillées, permettant une gestion précise des adresses IP, des routes, et des politiques de routage, ainsi que la surveillance détaillée des interfaces réseau. Ces outils offrent une interface de commande puissante pour ajuster finement le réseau selon les exigences des administrateurs.

D'autre part, `netstat` et `ss` fournissent une vue détaillée des différentes statistiques de réseau, telles que les connexions actives, les tables de routage, les écoutes de port, et plus encore. Tandis que `netstat` a longtemps été la référence pour obtenir des informations réseau, `ss` est apparu comme un outil plus rapide et plus capable, offrant une meilleure vision des sockets et des détails de connexion. Pour le diagnostic et le dépannage, `ping` et `traceroute` sont inestimables, permettant aux utilisateurs de tester la connectivité réseau et de tracer les itinéraires que les paquets prennent à travers le réseau, fournissant ainsi des indices vitaux pour résoudre les problèmes de réseau.

Il est aussi nécessaire de comprendre et de maîtriser la configuration des interfaces réseau via la ligne de commande car cruciale dans un contexte où les réseaux sont de plus en plus complexes et où les exigences en matière de sécurité et de performance sont élevées. Les administrateurs peuvent ainsi assurer une gestion efficace du réseau, avec la capacité d'automatiser des tâches, de scripter des configurations et d'effectuer des ajustements précis, garantissant que le réseau répond aux besoins de l'organisation tout en restant agile et sécurisé face aux évolutions technologiques constantes.

Nous allons approfondir quelques-uns de ces outils.

### 1.1 ifconfig

L'outil `ifconfig` (interface configuration) est traditionnellement utilisé pour configurer, gérer et afficher les paramètres des interfaces réseau. Bien qu'il soit désormais considéré comme dépassé par `ip`, il reste disponible sur de nombreux systèmes pour la compatibilité. Voici les commandes principales :

- **Afficher toutes les interfaces réseau** : simplement en exécutant `ifconfig` sans argument, vous pouvez afficher toutes les interfaces réseau actives, y compris leurs adresses IP, l'état de l'interface, le masque de sous-réseau, etc.

```
ifconfig
```

- **Configurer une adresse IP** : permet de configurer manuellement une adresse IP et un masque de sous-réseau sur une interface réseau spécifique. Par exemple :

```
ifconfig eth0 192.168.1.100 netmask 255.255.255.0
```

- **Activer/désactiver une interface** : active ou désactive l'interface réseau spécifiée, ce qui peut être utile pour la gestion de la connectivité ou la résolution de problèmes réseau.
  - Activer l'interface `eth0` : `ifconfig eth0 up`
  - Désactiver l'interface `eth0` : `ifconfig eth0 down`
- **Configurer une adresse MAC** : change l'adresse MAC (*Media Access Control*) de l'interface réseau, ce qui peut être requis pour des raisons de sécurité ou de tests. Par exemple : `ifconfig eth0 hw ether 02:01:02:03:04:05` modifie l'adresse MAC de l'interface `eth0` en `02:01:02:03:04:05`.
- **Afficher une interface spécifique** : montre les détails tels que l'adresse IP, le masque de sous-réseau, et d'autres informations de la configuration réseau pour une interface spécifique.  
`ifconfig eth0` affiche les détails de l'interface `eth0`, y compris son adresse IP, son masque de sous-réseau, son statut (*active* ou *inactive*).

```
root@debian-networking:~# ifconfig
Dev: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.60  netmask 255.255.255.0  broadcast 192.168.1.255
    inet6 fe80::6b7e:cd1b:a907:38d0  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:01:af:ad  txqueuelen 1000  (Ethernet)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 7464  bytes 2428287 (2.3 MiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.18  netmask 255.255.255.0  broadcast 192.168.1.255
    ether 08:00:27:48:c6:c8  txqueuelen 1000  (Ethernet)
    RX packets 441973  bytes 402288423 (383.6 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 68247  bytes 7430328 (7.0 MiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.44  netmask 255.255.255.0  broadcast 192.168.1.255
    ether 08:00:27:0f:ef:10  txqueuelen 1000  (Ethernet)
    RX packets 489928  bytes 414977765 (395.7 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 55970  bytes 5108867 (4.8 MiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.48  netmask 255.255.255.0  broadcast 192.168.1.255
    inet6 2a01:e0a:15e:b5b0:e8d7:5b40:6d84:586f  prefixlen 64  scopeid 0x0<global>
    inet6 2a01:e0a:15e:b5b0:1b28:8a2d:a43a:f12b  prefixlen 64  scopeid 0x0<global>
```

```
root@debian-networking:~# ifconfig enp0s3 192.168.1.19 netmask 255.255.255.0
root@debian-networking:~# ifconfig enp0s3 down
root@debian-networking:~# ifconfig enp0s3 up
root@debian-networking:~# ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.19 netmask 255.255.255.0 broadcast 192.168.1.255
    ether 08:00:27:48:c6:c8 txqueuelen 1000 (Ethernet)
    RX packets 442526 bytes 402364174 (383.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 68311 bytes 7436112 (7.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@debian-networking:~#
```

## 1.2 ip

`ip` est un outil plus récent et plus puissant que `ifconfig`, offrant une gamme étendue de fonctionnalités pour la gestion des interfaces réseau, des routes, des adresses IP et plus encore.

- **Afficher les interfaces** : cette commande est utilisée pour afficher les détails de toutes les interfaces réseau sur le système, y compris les adresses IP, les états des interfaces et d'autres informations réseau pertinentes.

```
ip addr show
```

- **Ajouter ou supprimer des adresses IP** : permet de modifier la configuration IP d'une interface, en ajoutant ou en supprimant des adresses IP.

- Ajouter : attribue une nouvelle adresse IP à une interface réseau spécifique.

```
ip addr add 192.168.1.10/24 dev eth0
```

- Supprimer : enlève une adresse IP d'une interface réseau.

```
ip addr del 192.168.1.10/24 dev eth0
```

- **Gérer l'état des interfaces** : active ou désactive les interfaces réseau, ce qui est essentiel pour la gestion de la connectivité et le contrôle de l'interface.

- Activer l'interface `eth0` : `ip link set eth0 up`

- Désactiver l'interface `eth0` : `ip link set eth0 down`

- **Modifier l'adresse MAC** : change l'adresse MAC de l'interface réseau, ce qui peut être nécessaire pour des raisons de sécurité, de confidentialité ou pour des tests.

```
ip link set dev eth0 address aa:bb:cc:dd:ee:ff
```

modifie l'adresse MAC de l'interface `eth0` en `aa:bb:cc:dd:ee:ff`.

- **Afficher les routes** : affiche la table de routage IP du système, montrant comment les paquets sont dirigés d'une interface à une autre ou vers des réseaux spécifiques.

`ip route show` montre la table de routage actuelle, y compris les routes par défaut, les routes spécifiques à des réseaux et les métriques associées.

- **Ajouter ou supprimer des routes** : modifie les itinéraires que les paquets utilisent pour atteindre leur destination, un aspect crucial de la configuration du réseau.

- Ajouter : crée une nouvelle route dans la table de routage.

`ip route add 192.168.2.0/24 via 192.168.1.1` ajoute une route pour atteindre le réseau 192.168.2.0/24 via la passerelle 192.168.1.1.

- Supprimer : retire une route existante de la table de routage.

`ip route del 192.168.2.0/24` supprime la route vers le réseau 192.168.2.0/24.

```
root@debian-networking:~# ip add show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:48:c6:c8 brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.19/24 brd 192.168.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:0f:ef:10 brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.44/24 brd 192.168.1.255 scope global dynamic enp0s8
       valid_lft 28648sec preferred_lft 28648sec
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:01:af:ad brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.48/24 brd 192.168.1.255 scope global enp0s9
       valid_lft forever preferred_lft forever
   inet6 2a01:e0a:15e:b5b0:b7f3:325f:7a5a:2f74/64 scope global temporary deprecated dynamic
       valid_lft 86386sec preferred_lft 0sec
   inet6 2a01:e0a:15e:b5b0:4d4f:a34f:8d2f:4296/64 scope global temporary deprecated dynamic
       valid_lft 86386sec preferred_lft 0sec
   inet6 2a01:e0a:15e:b5b0:e8d7:5b40:6d84:586f/64 scope global temporary deprecated dynamic
       valid_lft 86386sec preferred_lft 0sec
   inet6 2a01:e0a:15e:b5b0:1b28:8a2d:a43a:f12b/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86386sec preferred_lft 86386sec
   inet6 fe80::946a:fd84:3f6a:c08e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
5: enp0s10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
```

```

root@debian-networking:~# ip addr add 192.168.119/24 dev enp0s3
root@debian-networking:~# ip link set enp0s3 up
root@debian-networking:~# ip route show
default via 192.168.1.254 dev enp0s8
default via 192.168.1.254 dev Dev proto static metric 400
10.8.0.0/24 dev tun0 proto kernel scope link src 10.8.0.1
192.168.1.0/24 dev enp0s8 proto kernel scope link src 192.168.1.44
192.168.1.0/24 dev enp0s10 proto kernel scope link src 192.168.1.28
192.168.1.0/24 dev enp0s9 proto kernel scope link src 192.168.1.48
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.19
192.168.1.0/24 dev Dev proto kernel scope link src 192.168.1.60 metric 400
192.168.119.0/24 dev enp0s3 proto kernel scope link src 192.168.119.0
root@debian-networking:~# ip route add 192.168.2.0/24 via 192.168.1.254
root@debian-networking:~# ip route show
default via 192.168.1.254 dev enp0s8
default via 192.168.1.254 dev Dev proto static metric 400
10.8.0.0/24 dev tun0 proto kernel scope link src 10.8.0.1
192.168.1.0/24 dev enp0s8 proto kernel scope link src 192.168.1.44
192.168.1.0/24 dev enp0s10 proto kernel scope link src 192.168.1.28
192.168.1.0/24 dev enp0s9 proto kernel scope link src 192.168.1.48
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.19
192.168.1.0/24 dev Dev proto kernel scope link src 192.168.1.60 metric 400
192.168.2.0/24 via 192.168.1.254 dev enp0s8
192.168.119.0/24 dev enp0s3 proto kernel scope link src 192.168.119.0
root@debian-networking:~#

```

## 1.3 netstat

netstat (*Network Statistics*) est un outil de ligne de commande qui affiche les connexions réseau, les tables de routage, les statistiques d'interface, les connexions masquées, et les membres multicast. Bien qu'il soit moins utilisé de nos jours, remplacé en grande partie par ss, ses principales commandes incluent :

- netstat -a : affiche toutes les connexions et les ports d'écoute.
- netstat -t : montre les connexions TCP.
- netstat -u : liste les connexions UDP.
- netstat -l : affiche uniquement les sockets en écoute.
- netstat -r : affiche la table de routage.
- netstat -s : présente les statistiques par protocole.
- netstat -p : montre le programme associé à chaque socket.

**Remarque**

*Il est possible de regrouper les arguments, en faisant par exemple `netstat -atu` pour afficher toutes les connexions TCP et UDP.*

```
root@debian-networking:~# netstat -atu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:domain         0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:ssh            0.0.0.0:*               LISTEN
tcp6     0      0 [::]:domain            [::]:*                  LISTEN
tcp6     0      0 [::]:ssh                [::]:*                  LISTEN
tcp6     0      0 localhost:ipp           [::]:*                  LISTEN
udp      0      0 0.0.0.0:631            0.0.0.0:*               LISTEN
udp      0      0 0.0.0.0:38801          0.0.0.0:*               LISTEN
udp      0      0 0.0.0.0:domain         0.0.0.0:*               LISTEN
udp      0      0 0.0.0.0:bootpc         0.0.0.0:*               LISTEN
udp      0      0 0.0.0.0:openvpn        0.0.0.0:*               LISTEN
udp      0      0 0.0.0.0:mdns            0.0.0.0:*               LISTEN
udp6     0      0 [::]:46480             [::]:*                  LISTEN
udp6     0      0 [::]:domain            [::]:*                  LISTEN
udp6     0      0 [::]:mdns               [::]:*                  LISTEN
root@debian-networking:~#
```

Les champs affichés dans la sortie de la commande sont expliqués comme suit :

- **Proto** : indique le protocole de la connexion, qui peut être TCP, UDP, TCP6 (TCP sur IPv6), ou UDP6 (UDP sur IPv6).
- **Recv-Q** : représente la queue de réception, qui indique le nombre d'octets en attente de réception par une application locale.
- **Send-Q** : représente la queue d'envoi, qui montre le nombre d'octets en attente d'être envoyés par une application locale.
- **Local Address** : affiche l'adresse et le port sur lequel l'ordinateur local écoute ou est connecté. Il peut s'agir d'une adresse IP spécifique à l'une des interfaces de la machine ou de « 0.0.0.0 » (ou « :: » pour IPv6), ce qui signifie que la machine écoute sur toutes les adresses IP disponibles.



- `Foreign Address` : indique l'adresse IP et le port de la machine distante à laquelle la connexion locale est établie. Un astérisque (\*) dans cette colonne signifie que le système écoute les connexions entrantes de n'importe quelle adresse distante.
- `State` : montre l'état de la connexion. Par exemple, `LISTEN` signifie que l'application est en attente d'une connexion entrante, `ESTABLISHED` signifie qu'il y a une connexion active.

Les ports sont indiqués après les adresses IP, séparés par des deux-points. Par exemple, « `localhost:ipp` » indique que le service d'impression Internet (IPP) écoute les connexions entrantes sur l'adresse de loopback (`localhost`), qui est généralement `127.0.0.1` pour IPv4 ou `::1` pour IPv6.

Voici quelques détails spécifiques sur certaines lignes de la sortie :

- Les lignes avec `domain` se réfèrent généralement au port 53, utilisé par le service DNS.
- `ssh` fait référence au service Secure Shell, qui par défaut écoute sur le port 22.
- `mdns` sur le port 5353 est utilisé par Multicast DNS, une partie du protocole Zeroconf.
- `ipp` sur le port 631 concerne le protocole Internet Printing Protocol, utilisé pour les services d'impression réseau.

### 1.4 ss

`ss` (*Socket Statistics*) est un outil moderne qui remplace `netstat`, offrant plus d'informations et étant plus rapide. Il est utilisé pour afficher des informations détaillées sur les différents sockets. Ses commandes principales incluent :

- `ss -t` : affiche les sockets TCP.
- `ss -u` : liste les sockets UDP.
- `ss -l` : montre les sockets en écoute.
- `ss -a` : affiche toutes les sockets.
- `ss -p` : affiche le processus utilisant le socket.
- `ss -s` : montre les statistiques de socket.