Version : **2022.01**

Dernière mise-à-jour : 2022/05/04 05:49

LDF606 - Gestion du Réseau

Contenu du Module

Présentation

Debian 11 utilise **Network Manager** pour gérer le réseau. Network Manager est composé de deux éléments :

- un service qui gère les connexions réseaux et rapporte leurs états,
- des front-ends qui passent par un API de configuration du service.

Important : Notez qu'avec cette version de NetworkManager, IPv6 est activée par défaut.

Le service NetworkManager doit toujours être lancé :

```
root@debian11:~# systemctl status NetworkManager.service
● NetworkManager.service - Network Manager
     Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor pres>
     Active: active (running) since Sun 2022-05-01 18:00:05 CEST; 20h ago
       Docs: man:NetworkManager(8)
   Main PID: 499 (NetworkManager)
      Tasks: 3 (limit: 4632)
     Memory: 13.3M
        CPU: 1.811s
     CGroup: /system.slice/NetworkManager.service
             └─499 /usr/sbin/NetworkManager --no-daemon

May 01 18:00:05 debian11 NetworkManager[499]: <info>  [1651420805.4957] device (ens18>
May 01 18:00:05 debian11 NetworkManager[499]: <info>  [1651420805.5102] device (ens18>
May 01 18:00:05 debian11 NetworkManager[499]: <info>  [1651420805.5136] device (ens18>
May 01 18:00:05 debian11 NetworkManager[499]: <info>  [1651420805.5141] device (ens18>
May 01 18:00:05 debian11 NetworkManager[499]: <info>  [1651420805.5147] manager: Netw>
May 01 18:00:05 debian11 NetworkManager[499]: <info>  [1651420805.5157] manager: Netw>
May 01 18:00:05 debian11 NetworkManager[499]: <info>  [1651420805.5159] policy: set '>
May 01 18:00:05 debian11 NetworkManager[499]: <info>  [1651420805.6555] device (ens18>
May 01 18:00:05 debian11 NetworkManager[499]: <info>  [1651420805.6577] manager: Netw>
May 01 18:00:05 debian11 NetworkManager[499]: <info>  [1651420805.6604] manager: star>
lines 1-21/21 (END)
[q]
```

La Commande nmcli

La commande **nmcli** (Network Manager Command Line Interface) est utilisée pour configurer NetworkManager.

Les options et les sous-commandes peuvent être consultées en utilisant les commandes suivantes :

```
root@debian11:~# nmcli help
Usage: nmcli [OPTIONS] OBJECT { COMMAND | help }

OPTIONS
  -a, --ask                                ask for missing parameters
  -c, --colors auto|yes|no                 whether to use colors in output
  -e, --escape yes|no                      escape columns separators in values
  -f, --fields <field,...>|all|common      specify fields to output
  -g, --get-values <field,...>|all|common  shortcut for -m tabular -t -f
  -h, --help                               print this help
  -m, --mode tabular|multiline             output mode
  -o, --overview                           overview mode
  -p, --pretty                             pretty output
  -s, --show-secrets                       allow displaying passwords
  -t, --terse                              terse output
  -v, --version                            show program version
  -w, --wait <seconds>                     set timeout waiting for finishing operations

OBJECT
  g[eneral]       NetworkManager's general status and operations
  n[etworking]    overall networking control
  r[adio]         NetworkManager radio switches
  c[onnection]    NetworkManager's connections
  d[evice]        devices managed by NetworkManager
  a[gent]         NetworkManager secret agent or polkit agent
  m[onitor]       monitor NetworkManager changes
```

LAB #1 - Configuration du Réseau

1.1 - Connections et Profils

NetworkManager inclus la notion de **connections** ou **profils** permettant des configurations différentes en fonction de la localisation. Pour voir les connections actuelles, utilisez la commande **nmcli c** avec la sous-commande **show** :

```
root@debian11:~# nmcli c show
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  77c569e6-3176-4c10-8008-40d7634d2504  ethernet  ens18
```

Créez donc un profil IP fixe rattaché au périphérique **ens18** :

```
root@debian11:~# nmcli connection add con-name ip_fixe ifname ens18 type ethernet ip4 10.0.2.41/24 gw4 10.0.2.1
Connection 'ip_fixe' (c52994fc-0918-4108-81d2-d86dade62c7a) successfully added.
```

Constatez sa présence :

```
root@debian11:~# nmcli c show
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  77c569e6-3176-4c10-8008-40d7634d2504  ethernet  ens18
ip_fixe             c52994fc-0918-4108-81d2-d86dade62c7a  ethernet  --
```

Notez que la sortie n'indique pas que le profil **ip_fixe** soit associé au periphérique **ens18** car le profil **ip_fixe** n'est pas activé :

```
root@debian11:~# nmcli d show
GENERAL.DEVICE:                         ens18
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         F6:35:D1:39:09:72
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     Wired connection 1
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnect>
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         10.0.2.40/24
IP4.GATEWAY:                            10.0.2.1
IP4.ROUTE[1]:                           dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 10.0.2.1, mt = 100
```

```
IP4.DNS[1]:                             8.8.8.8
IP4.DNS[2]:                             8.8.4.4
IP6.ADDRESS[1]:                         fe80::f435:d1ff:fe39:972/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]:                           dst = ff00::/8, nh = ::, mt = 256, table=255

GENERAL.DEVICE:                         lo
GENERAL.TYPE:                           loopback
GENERAL.HWADDR:                         00:00:00:00:00:00
GENERAL.MTU:                            65536
lines 1-23
[q]
```

Pour activer le profil ip_fixe, utilisez la commande suivante :

```
[root@centos8 ~]# nmcli connection up ip_fixe
```

Notez que votre terminal est bloqué à cause du changement de l'adresse IP.

A faire - Revenez à votre Gateway et re-connectez-vous à la VM en tant que trainee en utilisant l'adresse IP 10.0.2.41.

Le profil ip_fixe est maintenant activé tandis que le profil enp0s3 a été désactivé :

```
root@debian11:~# nmcli c show
NAME                UUID                                  TYPE      DEVICE
ip_fixe             c52994fc-0918-4108-81d2-d86dade62c7a  ethernet  ens18
Wired connection 1  77c569e6-3176-4c10-8008-40d7634d2504  ethernet  --
root@debian11:~# nmcli d show
GENERAL.DEVICE:                         ens18
```

```
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         F6:35:D1:39:09:72
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     ip_fixe
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveC>
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         10.0.2.41/24
IP4.GATEWAY:                            10.0.2.1
IP4.ROUTE[1]:                           dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 1>
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 10.0.2.1, mt = 100
IP6.ADDRESS[1]:                         fe80::7958:e23f:31e:62cd/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]:                           dst = ff00::/8, nh = ::, mt = 256, tabl>

GENERAL.DEVICE:                         lo
GENERAL.TYPE:                           loopback
lines 1-19
[q]
```

Pour consulter les paramètres du profil **Wired connection 1**, utilisez la commande suivante :

```
root@debian11:~# nmcli -p connection show "Wired connection 1"
===============================================================================
                Connection profile details (Wired connection 1)
===============================================================================
connection.id:                          Wired connection 1
connection.uuid:                        77c569e6-3176-4c10-8008-40d7634d2504
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              --
connection.autoconnect:                 yes
connection.autoconnect-priority:        0
```

```
connection.autoconnect-retries:         -1 (default)
connection.multi-connect:               0 (default)
connection.auth-retries:                -1
connection.timestamp:                   1651494383
connection.read-only:                   no
connection.permissions:                 --
connection.zone:                        --
connection.master:                      --
connection.slave-type:                  --
connection.autoconnect-slaves:          -1 (default)
connection.secondaries:                 --
connection.gateway-ping-timeout:        0
connection.metered:                     unknown
connection.lldp:                        default
connection.mdns:                        -1 (default)
connection.llmnr:                       -1 (default)
connection.wait-device-timeout:         -1
---------------------------------------------------------------------------
802-3-ethernet.port:                    --
802-3-ethernet.speed:                   0
802-3-ethernet.duplex:                  --
802-3-ethernet.auto-negotiate:          no
802-3-ethernet.mac-address:             --
802-3-ethernet.cloned-mac-address:      --
802-3-ethernet.generate-mac-address-mask:--
802-3-ethernet.mac-address-blacklist:   --
802-3-ethernet.mtu:                     auto
802-3-ethernet.s390-subchannels:        --
802-3-ethernet.s390-nettype:            --
802-3-ethernet.s390-options:            --
802-3-ethernet.wake-on-lan:             default
802-3-ethernet.wake-on-lan-password:    --
---------------------------------------------------------------------------
ipv4.method:                            manual
```

```
ipv4.dns:                               8.8.8.8,8.8.4.4
ipv4.dns-search:                        --
ipv4.dns-options:                       --
ipv4.dns-priority:                      0
ipv4.addresses:                         10.0.2.40/24
ipv4.gateway:                           10.0.2.1
ipv4.routes:                            --
ipv4.route-metric:                      -1
ipv4.route-table:                       0 (unspec)
ipv4.routing-rules:                     --
ipv4.ignore-auto-routes:                no
ipv4.ignore-auto-dns:                   no
lines 1-56
[q]
```

De même, pour consulter les paramètres du profil **ip_fixe**, utilisez la commande suivante :

```
root@debian11:~# nmcli -p connection show ip_fixe
===============================================================================
                    Connection profile details (ip_fixe)
===============================================================================
connection.id:                          ip_fixe
connection.uuid:                        c52994fc-0918-4108-81d2-d86dade62c7a
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              ens18
connection.autoconnect:                 yes
connection.autoconnect-priority:        0
connection.autoconnect-retries:         -1 (default)
connection.multi-connect:               0 (default)
connection.auth-retries:                -1
connection.timestamp:                   1651496105
connection.read-only:                   no
connection.permissions:                 --
```

```
connection.zone:                        --
connection.master:                      --
connection.slave-type:                  --
connection.autoconnect-slaves:          -1 (default)
connection.secondaries:                 --
connection.gateway-ping-timeout:        0
connection.metered:                     unknown
connection.lldp:                        default
connection.mdns:                        -1 (default)
connection.llmnr:                       -1 (default)
connection.wait-device-timeout:         -1
---------------------------------------------------------------------------
802-3-ethernet.port:                    --
802-3-ethernet.speed:                   0
802-3-ethernet.duplex:                  --
802-3-ethernet.auto-negotiate:          no
802-3-ethernet.mac-address:             --
802-3-ethernet.cloned-mac-address:      --
802-3-ethernet.generate-mac-address-mask:--
802-3-ethernet.mac-address-blacklist:   --
802-3-ethernet.mtu:                     auto
802-3-ethernet.s390-subchannels:        --
802-3-ethernet.s390-nettype:            --
802-3-ethernet.s390-options:            --
802-3-ethernet.wake-on-lan:             default
802-3-ethernet.wake-on-lan-password:    --
---------------------------------------------------------------------------
ipv4.method:                            manual
ipv4.dns:                               --
ipv4.dns-search:                        --
ipv4.dns-options:                       --
ipv4.dns-priority:                      0
ipv4.addresses:                         10.0.2.41/24
ipv4.gateway:                           10.0.2.1
```

```
ipv4.routes:                            --
ipv4.route-metric:                      -1
ipv4.route-table:                       0 (unspec)
ipv4.routing-rules:                     --
ipv4.ignore-auto-routes:                no
ipv4.ignore-auto-dns:                   no
lines 1-56
[q]
```

Pour consulter la liste profils associés à un périphérique, utilisez la commande suivante :

```
root@debian11:~# nmcli -f CONNECTIONS device show ens18
CONNECTIONS.AVAILABLE-CONNECTION-PATHS: /org/freedesktop/NetworkManager/Settings/1,/o>
CONNECTIONS.AVAILABLE-CONNECTIONS[1]:   77c569e6-3176-4c10-8008-40d7634d2504 | Wired >
CONNECTIONS.AVAILABLE-CONNECTIONS[2]:   c52994fc-0918-4108-81d2-d86dade62c7a | ip_fixe
lines 1-3/3 (END)
[q]
```

Les fichiers de configuration pour le periphérique **ens18** se trouvent dans le répertoire **/etc/NetworkManager/system-connections/** :

```
root@debian11:~# ls -l /etc/NetworkManager/system-connections
total 8
-rw------- 1 root root 284 May  2 14:23  ip_fixe.nmconnection
-rw------- 1 root root 249 Apr 25 07:01 'Wired connection 1'
```

1.2 - Résolution des Noms

L'étude du fichier **/etc/NetworkManager/system-connections/ip_fixe.nmconnection** démontre l'abscence de directives concernant les DNS :

```
root@debian11:~# cat /etc/NetworkManager/system-connections/ip_fixe.nmconnection
[connection]
id=ip_fixe
```

```
uuid=c52994fc-0918-4108-81d2-d86dade62c7a
type=ethernet
interface-name=ens18
permissions=

[ethernet]
mac-address-blacklist=

[ipv4]
address1=10.0.2.41/24,10.0.2.1
dns-search=
method=manual

[ipv6]
addr-gen-mode=stable-privacy
dns-search=
method=auto

[proxy]
```

La résolution des noms est donc inactive :

```
root@debian11:~# ping www.free.fr
ping: www.free.fr: Temporary failure in name resolution
```

Modifiez donc la configuration du profil **ip_fixe** :

```
root@debian11:~# nmcli connection mod ip_fixe ipv4.dns 8.8.8.8
```

L'étude du fichier **/etc/NetworkManager/system-connections/ip_fixe.nmconnection** démontre que la directive concernant le serveur DNS a été ajoutée :

```
root@debian11:~# cat /etc/NetworkManager/system-connections/ip_fixe.nmconnection
[connection]
```

```
id=ip_fixe
uuid=c52994fc-0918-4108-81d2-d86dade62c7a
type=ethernet
interface-name=ens18
permissions=
timestamp=1651499105

[ethernet]
mac-address-blacklist=

[ipv4]
address1=10.0.2.41/24,10.0.2.1
dns=8.8.8.8;
dns-search=
method=manual

[ipv6]
addr-gen-mode=stable-privacy
dns-search=
method=auto

[proxy]
```

Afin que la modification du serveur DNS soit prise en compte, re-démarrez le service NetworkManager :

```
root@debian11:~# systemctl restart NetworkManager.service
```

Vérifiez que le fichier **/etc/resolv.conf** ait été modifié par NetworkManager :

```
root@debian11:~# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 8.8.8.8
```

Dernièrement vérifiez la resolution des noms :

```
root@debian11:~# ping www.free.fr
PING www.free.fr (212.27.48.10) 56(84) bytes of data.
64 bytes from www.free.fr (212.27.48.10): icmp_seq=1 ttl=47 time=10.8 ms
64 bytes from www.free.fr (212.27.48.10): icmp_seq=2 ttl=47 time=11.1 ms
^C
--- www.free.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 10.804/10.931/11.058/0.127 ms
```

Important : Notez qu'il existe un front-end graphique en mode texte, **nmtui**, pour configurer NetworkManager.

1.3 - Ajouter une Deuxième Adresse IP à un Profil

Pour ajouter une deuxième adresse IP à un profil sous Debian 11, il convient d'utiliser la commande suivante :

```
root@debian11:~# nmcli connection mod ip_fixe +ipv4.addresses 192.168.1.2/24
```

Rechargez la configuration du profil :

```
root@debian11:~# nmcli con up ip_fixe
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/2)
```

Saisissez ensuite la commande suivante :

```
root@debian11:~# nmcli connection show ip_fixe
connection.id:                          ip_fixe
connection.uuid:                        c52994fc-0918-4108-81d2-d86dade62c7a
connection.stable-id:                   --
```

```
connection.type:                        802-3-ethernet
connection.interface-name:              ens18
connection.autoconnect:                 yes
connection.autoconnect-priority:        0
connection.autoconnect-retries:         -1 (default)
connection.multi-connect:               0 (default)
connection.auth-retries:                -1
connection.timestamp:                   1651499367
connection.read-only:                   no
connection.permissions:                 --
connection.zone:                        --
connection.master:                      --
connection.slave-type:                  --
connection.autoconnect-slaves:          -1 (default)
connection.secondaries:                 --
connection.gateway-ping-timeout:        0
connection.metered:                     unknown
connection.lldp:                        default
connection.mdns:                        -1 (default)
connection.llmnr:                       -1 (default)
connection.wait-device-timeout:         -1
802-3-ethernet.port:                    --
802-3-ethernet.speed:                   0
802-3-ethernet.duplex:                  --
802-3-ethernet.auto-negotiate:          no
802-3-ethernet.mac-address:             --
802-3-ethernet.cloned-mac-address:      --
802-3-ethernet.generate-mac-address-mask:--
802-3-ethernet.mac-address-blacklist:   --
802-3-ethernet.mtu:                     auto
802-3-ethernet.s390-subchannels:        --
802-3-ethernet.s390-nettype:            --
802-3-ethernet.s390-options:            --
802-3-ethernet.wake-on-lan:             default
```

```
802-3-ethernet.wake-on-lan-password:    --
ipv4.method:                            manual
ipv4.dns:                               8.8.8.8
ipv4.dns-search:                        --
ipv4.dns-options:                       --
ipv4.dns-priority:                      0
ipv4.addresses:                         10.0.2.41/24, 192.168.1.2/24
ipv4.gateway:                           10.0.2.1
ipv4.routes:                            --
ipv4.route-metric:                      -1
ipv4.route-table:                       0 (unspec)
ipv4.routing-rules:                     --
ipv4.ignore-auto-routes:                no
ipv4.ignore-auto-dns:                   no
ipv4.dhcp-client-id:                    --
ipv4.dhcp-iaid:                         --
ipv4.dhcp-timeout:                      0 (default)
ipv4.dhcp-send-hostname:                yes
ipv4.dhcp-hostname:                     --
lines 1-56
[Space Bar]
IP4.ADDRESS[1]:                         10.0.2.41/24
IP4.ADDRESS[2]:                         192.168.1.2/24
IP4.GATEWAY:                            10.0.2.1
IP4.ROUTE[1]:                           dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 192.168.1.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[3]:                           dst = 0.0.0.0/0, nh = 10.0.2.1, mt = 100
IP4.DNS[1]:                             8.8.8.8
lines 57-112
[q]
```

Important : Notez l'ajout de l'adresse secondaire à la ligne **ipv4.addresses:** ainsi que l'ajout de la ligne **IP4.ADDRESS[2]:**.

Consultez maintenant le contenu du fichier **/etc/NetworkManager/system-connections/ip_fixe.nmconnection** :

```
root@debian11:~# cat /etc/NetworkManager/system-connections/ip_fixe.nmconnection
[connection]
id=ip_fixe
uuid=c52994fc-0918-4108-81d2-d86dade62c7a
type=ethernet
interface-name=ens18
permissions=
timestamp=1651499263

[ethernet]
mac-address-blacklist=

[ipv4]
address1=10.0.2.41/24,10.0.2.1
address2=192.168.1.2/24
dns=8.8.8.8;
dns-search=
method=manual

[ipv6]
addr-gen-mode=stable-privacy
dns-search=
method=auto

[proxy]
```

Important : Notez l'ajout de la ligne **address2=192.168.1.2/24**.

1.4 - La Commande hostname

La procédure de la modification du hostname est simplifiée et sa prise en compte est immédiate :

```
root@debian11:~# hostname
debian11

root@debian11:~# nmcli general hostname debian11.ittraining.loc

root@debian11:~# cat /etc/hostname
debian11.ittraining.loc

root@debian11:~# hostname
debian11.ittraining.loc
```

1.5 - La Commande ip

Sous Debian 11 la commande **ip** est préférée par rapport à la commande ifconfig :

```
root@debian11:~# ip address
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
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether f6:35:d1:39:09:72 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 10.0.2.41/24 brd 10.0.2.255 scope global noprefixroute ens18
       valid_lft forever preferred_lft forever
    inet 192.168.1.2/24 brd 192.168.1.255 scope global noprefixroute ens18
```

```
       valid_lft forever preferred_lft forever
    inet6 fe80::7958:e23f:31e:62cd/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

En cas de besoin, pour extraire les adresses IP de cette sortie, utilisez les commandes suivantes :

```
root@debian11:~# ip addr show ens18 | grep "inet" | grep -v "inet6" | awk '{ print $2; }' | sed 's/\/.*$//'
10.0.2.41
192.168.1.2
```

Les options de cette commande sont :

```
root@debian11:~# ip --help
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [ -force ] -batch filename
where  OBJECT := { link | address | addrlabel | route | rule | neigh | ntable |
                   tunnel | tuntap | maddress | mroute | mrule | monitor | xfrm |
                   netns | l2tp | fou | macsec | tcp_metrics | token | netconf | ila |
                   vrf | sr | nexthop | mptcp }
       OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
                    -h[uman-readable] | -iec | -j[son] | -p[retty] |
                    -f[amily] { inet | inet6 | mpls | bridge | link } |
                    -4 | -6 | -I | -D | -M | -B | -0 |
                    -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
                    -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
                    -rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
                    -c[olor]}
```

1.6 - Activer/Désactiver une Interface Manuellement

Deux commandes existent pour désactiver et activer manuellement une interface réseau :

```
# nmcli device disconnect enp0s3
# nmcli device connect enp0s3
```

Important : Veuillez ne **PAS** exécuter ces deux commandes.

1.7 - Routage Statique

La commande ip

Sous Debian 11, pour supprimer la route vers le réseau 192.168.1.0 il convient d'utiliser la commande ip et non pas la commande route :

```
root@debian11:~# ip route
default via 10.0.2.1 dev ens18 proto static metric 100
10.0.2.0/24 dev ens18 proto kernel scope link src 10.0.2.41 metric 100
192.168.1.0/24 dev ens18 proto kernel scope link src 192.168.1.2 metric 100

root@debian11:~# ip route del 192.168.1.0/24 via 0.0.0.0

root@debian11:~# ip route
default via 10.0.2.1 dev ens18 proto static metric 100
10.0.2.0/24 dev ens18 proto kernel scope link src 10.0.2.41 metric 100
```

Pour ajouter la route vers le réseau 192.168.1.0 :

```
root@debian11:~# ip route add 192.168.1.0/24 via 10.0.2.1

root@debian11:~# ip route
default via 10.0.2.1 dev ens18 proto static metric 100
10.0.2.0/24 dev ens18 proto kernel scope link src 10.0.2.41 metric 100
```

```
192.168.1.0/24 via 10.0.2.1 dev ens18
```

Important - La commande utilisée pour ajouter une passerelle par défaut prend la forme suivante **ip route add default via *adresse ip***.

Activer le routage sur le serveur

Pour activer le routage sur le serveur, il convient d'activer la retransmission des paquets:

```
root@debian11:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@debian11:~# cat /proc/sys/net/ipv4/ip_forward
1
```

LAB #2 - Diagnostique du Réseau

2.1 - ping

Pour tester l'accessibilité d'une machine, vous devez utiliser la commande **ping** :

```
root@debian11:~# ping -c4 10.0.2.1
PING 10.0.2.1 (10.0.2.1) 56(84) bytes of data.
64 bytes from 10.0.2.1: icmp_seq=1 ttl=64 time=0.184 ms
64 bytes from 10.0.2.1: icmp_seq=2 ttl=64 time=0.167 ms
64 bytes from 10.0.2.1: icmp_seq=3 ttl=64 time=0.168 ms
^C
--- 10.0.2.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2036ms
```

```
rtt min/avg/max/mdev = 0.167/0.173/0.184/0.007 ms
```

Les options de cette commande sont :

```
root@debian11:~# ping --help
ping: invalid option -- '-'

Usage
  ping [options] <destination>

Options:
  <destination>      dns name or ip address
  -a                 use audible ping
  -A                 use adaptive ping
  -B                 sticky source address
  -c <count>         stop after <count> replies
  -D                 print timestamps
  -d                 use SO_DEBUG socket option
  -f                 flood ping
  -h                 print help and exit
  -I <interface>     either interface name or address
  -i <interval>      seconds between sending each packet
  -L                 suppress loopback of multicast packets
  -l <preload>       send <preload> number of packages while waiting replies
  -m <mark>          tag the packets going out
  -M <pmtud opt>     define mtu discovery, can be one of <do|dont|want>
  -n                 no dns name resolution
  -O                 report outstanding replies
  -p <pattern>       contents of padding byte
  -q                 quiet output
  -Q <tclass>        use quality of service <tclass> bits
  -s <size>          use <size> as number of data bytes to be sent
  -S <size>          use <size> as SO_SNDBUF socket option value
  -t <ttl>           define time to live
```

```
  -U                print user-to-user latency
  -v                verbose output
  -V                print version and exit
  -w <deadline>     reply wait <deadline> in seconds
  -W <timeout>      time to wait for response

IPv4 options:
  -4                use IPv4
  -b                allow pinging broadcast
  -R                record route
  -T <timestamp>    define timestamp, can be one of <tsonly|tsandaddr|tsprespec>

IPv6 options:
  -6                use IPv6
  -F <flowlabel>    define flow label, default is random
  -N <nodeinfo opt> use icmp6 node info query, try <help> as argument

For more details see ping(8).
```

2.2 - netstat -i

Pour visualiser les statistiques réseaux, vous disposez de la commande **netstat** :

```
root@debian11:~# netstat -i
-bash: netstat: command not found

root@debian11:~# apt -y install net-tools

root@debian11:~# netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
ens18     1500     7861      0      0 0          7299      0      0      0 BMRU
```

```
lo       65536       82      0      0 0            82      0      0      0 LRU
```

Les options de cette commande sont :

```
root@debian11:~# netstat --help
usage: netstat [-vWeenNcCF] [<Af>] -r         netstat {-V|--version|-h|--help}
       netstat [-vWnNcaeol] [<Socket> ...]
       netstat { [-vWeenNac] -i | [-cnNe] -M | -s [-6tuw] }

        -r, --route              display routing table
        -i, --interfaces         display interface table
        -g, --groups             display multicast group memberships
        -s, --statistics         display networking statistics (like SNMP)
        -M, --masquerade         display masqueraded connections

        -v, --verbose            be verbose
        -W, --wide               don't truncate IP addresses
        -n, --numeric            don't resolve names
        --numeric-hosts          don't resolve host names
        --numeric-ports          don't resolve port names
        --numeric-users          don't resolve user names
        -N, --symbolic           resolve hardware names
        -e, --extend             display other/more information
        -p, --programs           display PID/Program name for sockets
        -o, --timers             display timers
        -c, --continuous         continuous listing

        -l, --listening          display listening server sockets
        -a, --all                display all sockets (default: connected)
        -F, --fib                display Forwarding Information Base (default)
        -C, --cache              display routing cache instead of FIB
        -Z, --context            display SELinux security context for sockets

  <Socket>={-t|--tcp} {-u|--udp} {-U|--udplite} {-S|--sctp} {-w|--raw}
```

```
          {-x|--unix} --ax25 --ipx --netrom
  <AF>=Use '-6|-4' or '-A <af>' or '--<af>'; default: inet
  List of possible address families (which support routing):
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
```

2.3 - traceroute

La commande ping est à la base de la commande **traceroute**. Cette commande sert à découvrir la route empruntée pour accéder à un site donné :

```
root@debian11:~# traceroute www.free.fr
traceroute to www.free.fr (212.27.48.10), 30 hops max, 60 byte packets
 1  10.0.2.1 (10.0.2.1)  0.476 ms  0.440 ms  0.419 ms
 2  51.68.180.253 (51.68.180.253)  2.730 ms  2.786 ms  2.927 ms
 3  10.161.41.253 (10.161.41.253)  0.315 ms 10.161.41.252 (10.161.41.252)  0.293 ms  0.274 ms
 4  10.17.242.66 (10.17.242.66)  0.467 ms 10.17.242.62 (10.17.242.62)  0.620 ms 10.17.242.66 (10.17.242.66)
0.795 ms
 5  10.73.40.38 (10.73.40.38)  0.189 ms  2.403 ms 10.73.40.42 (10.73.40.42)  2.339 ms
 6  10.73.249.68 (10.73.249.68)  4.894 ms * *
 7  fra-fr5-sbb1-nc5.de.eu (91.121.215.116)  1.909 ms  1.491 ms fra-fr5-sbb2-nc5.de.eu (94.23.122.246)  1.475 ms
 8  be101.sbg-g1-nc5.fr.eu (94.23.122.136)  4.182 ms be101.sbg-g2-nc5.fr.eu (91.121.215.196)  4.504 ms  4.918 ms
 9  be103.par-gsw-sbb1-nc5.fr.eu (91.121.215.219)  10.471 ms be103.par-th2-sbb1-nc5.fr.eu (94.23.122.139)  10.448
ms  10.198 ms
10  10.200.2.65 (10.200.2.65)  10.174 ms 10.200.2.71 (10.200.2.71)  10.211 ms 10.200.2.65 (10.200.2.65)  10.111
ms
11  * * *
12  194.149.166.61 (194.149.166.61)  10.289 ms *  10.111 ms
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
```

```
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Les options de cette commande sont :

```
root@debian11:~# traceroute --help
Usage:
  traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [ -m max_ttl ] [ -N squeries ] [ -
p port ] [ -t tos ] [ -l flow_label ] [ -w MAX,HERE,NEAR ] [ -q nqueries ] [ -s src_addr ] [ -z sendwait ] [ --
fwmark=num ] host [ packetlen ]
Options:
  -4                          Use IPv4
  -6                          Use IPv6
  -d  --debug                 Enable socket level debugging
  -F  --dont-fragment         Do not fragment packets
  -f first_ttl  --first=first_ttl
                              Start from the first_ttl hop (instead from 1)
  -g gate,...  --gateway=gate,...
                              Route packets through the specified gateway
                              (maximum 8 for IPv4 and 127 for IPv6)
  -I  --icmp                  Use ICMP ECHO for tracerouting
  -T  --tcp                   Use TCP SYN for tracerouting (default port is 80)
  -i device  --interface=device
```

```
                              Specify a network interface to operate with
-m max_ttl  --max-hops=max_ttl
                              Set the max number of hops (max TTL to be
                              reached). Default is 30
-N squeries  --sim-queries=squeries
                              Set the number of probes to be tried
                              simultaneously (default is 16)
-n                            Do not resolve IP addresses to their domain names
-p port  --port=port          Set the destination port to use. It is either
                              initial udp port value for "default" method
                              (incremented by each probe, default is 33434), or
                              initial seq for "icmp" (incremented as well,
                              default from 1), or some constant destination
                              port for other methods (with default of 80 for
                              "tcp", 53 for "udp", etc.)
-t tos  --tos=tos             Set the TOS (IPv4 type of service) or TC (IPv6
                              traffic class) value for outgoing packets
-l flow_label  --flowlabel=flow_label
                              Use specified flow_label for IPv6 packets
-w MAX,HERE,NEAR  --wait=MAX,HERE,NEAR
                              Wait for a probe no more than HERE (default 3)
                              times longer than a response from the same hop,
                              or no more than NEAR (default 10) times than some
                              next hop, or MAX (default 5.0) seconds (float
                              point values allowed too)
-q nqueries  --queries=nqueries
                              Set the number of probes per each hop. Default is
                              3
-r                            Bypass the normal routing and send directly to a
                              host on an attached network
-s src_addr  --source=src_addr
                              Use source src_addr for outgoing packets
-z sendwait  --sendwait=sendwait
                              Minimal time interval between probes (default 0).
```

```
                              If the value is more than 10, then it specifies a
                              number in milliseconds, else it is a number of
                              seconds (float point values allowed too)
  -e  --extensions            Show ICMP extensions (if present), including MPLS
  -A  --as-path-lookups       Perform AS path lookups in routing registries and
                              print results directly after the corresponding
                              addresses
  -M name  --module=name      Use specified module (either builtin or external)
                              for traceroute operations. Most methods have
                              their shortcuts (`-I' means `-M icmp' etc.)
  -O OPTS,...  --options=OPTS,...
                              Use module-specific option OPTS for the
                              traceroute module. Several OPTS allowed,
                              separated by comma. If OPTS is "help", print info
                              about available options
  --sport=num                 Use source port num for outgoing packets. Implies
                              `-N 1'
  --fwmark=num                Set firewall mark for outgoing packets
  -U  --udp                   Use UDP to particular port for tracerouting
                              (instead of increasing the port per each probe),
                              default port is 53
  -UL                         Use UDPLITE for tracerouting (default dest port
                              is 53)
  -D  --dccp                  Use DCCP Request for tracerouting (default port
                              is 33434)
  -P prot  --protocol=prot    Use raw packet of protocol prot for tracerouting
  --mtu                       Discover MTU along the path being traced. Implies
                              `-F -N 1'
  --back                      Guess the number of hops in the backward path and
                              print if it differs
  -V  --version               Print version info and exit
  --help                      Read this help and exit

Arguments:
```

```
+     host          The host to traceroute to
      packetlen     The full packet length (default is the length of an IP
                    header plus 40). Can be ignored or increased to a minimal
                    allowed value
```

LAB #3 - Connexions à Distance

3.1 - Telnet

La commande **telnet** est utilisée pour établir une connexion à distance avec un serveur telnet :

```
  # telnet numero_ip
```

Important - Le service telnet revient à une redirection des canaux standards d'entrée et de sortie. Notez que la connexion n'est **pas** sécurisée. Pour fermer la connexion, il faut saisir la commande **exit**. La commande telnet n'offre pas de services de transfert de fichiers. Pour cela, il convient d'utiliser la command **ftp**.

Les options de cette commande sont :

```
root@debian11:~# which telnet
/usr/bin/telnet
root@debian11:~# telnet --help
telnet: invalid option -- '-'
Usage: telnet [-4] [-6] [-8] [-E] [-L] [-a] [-d] [-e char] [-l user]
        [-n tracefile] [ -b addr ] [-r] [host-name [port]]
```

3.2 - wget

La commande **wget** est utilisée pour récupérer un fichier via http, https ou ftp :

```
root@debian11:~# wget https://www.dropbox.com/s/wk79lkfr6f12u9j/wget_file.txt
--2022-05-03 10:07:50--  https://www.dropbox.com/s/wk79lkfr6f12u9j/wget_file.txt
Resolving www.dropbox.com (www.dropbox.com)... 162.125.67.18, 2620:100:6023:18::a27d:4312
Connecting to www.dropbox.com (www.dropbox.com)|162.125.67.18|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: /s/raw/wk79lkfr6f12u9j/wget_file.txt [following]
--2022-05-03 10:07:51--  https://www.dropbox.com/s/raw/wk79lkfr6f12u9j/wget_file.txt
Reusing existing connection to www.dropbox.com:443.
HTTP request sent, awaiting response... 302 Found
Location:
https://uc64dcd84ee4be2a0c2f111bd2ab.dl.dropboxusercontent.com/cd/0/inline/BkjB7WuIg8oRJsAnsjaq0YeQrof8LaM6svUdaz
V8cmiN5MWJqHVYZlMfbR33nIokof9ZHmK551ixoOU8CgteUybofFdo_vD62VcF-
WNi4ACD8o6JsjdD7kiiuge-8mfbD1unwz4UtpyHIuVPQ9sUrh0QNNTVtsj3kcjDJAgEDg-uKQ/file# [following]
--2022-05-03 10:07:51--
https://uc64dcd84ee4be2a0c2f111bd2ab.dl.dropboxusercontent.com/cd/0/inline/BkjB7WuIg8oRJsAnsjaq0YeQrof8LaM6svUdaz
V8cmiN5MWJqHVYZlMfbR33nIokof9ZHmK551ixoOU8CgteUybofFdo_vD62VcF-
WNi4ACD8o6JsjdD7kiiuge-8mfbD1unwz4UtpyHIuVPQ9sUrh0QNNTVtsj3kcjDJAgEDg-uKQ/file
Resolving uc64dcd84ee4be2a0c2f111bd2ab.dl.dropboxusercontent.com
(uc64dcd84ee4be2a0c2f111bd2ab.dl.dropboxusercontent.com)... 162.125.67.15, 2620:100:6023:15::a27d:430f
Connecting to uc64dcd84ee4be2a0c2f111bd2ab.dl.dropboxusercontent.com
(uc64dcd84ee4be2a0c2f111bd2ab.dl.dropboxusercontent.com)|162.125.67.15|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46 [text/plain]
Saving to: ‘wget_file.txt’

wget_file.txt         100%[======================>]      46  --.-KB/s    in 0s

2022-05-03 10:07:51 (26.8 MB/s) - ‘wget_file.txt’ saved [46/46]
```

```
root@debian11:~# cat wget_file.txt
This is a file retrieved by the wget command.
```

Les options de cette commande sont :

```
root@debian11:~# wget --help | more
GNU Wget 1.21, a non-interactive network retriever.
Usage: wget [OPTION]... [URL]...

Mandatory arguments to long options are mandatory for short options too.

Startup:
  -V,  --version                   display the version of Wget and exit
  -h,  --help                      print this help
  -b,  --background                go to background after startup
  -e,  --execute=COMMAND           execute a `.wgetrc'-style command

Logging and input file:
  -o,  --output-file=FILE          log messages to FILE
  -a,  --append-output=FILE        append messages to FILE
  -d,  --debug                     print lots of debugging information
  -q,  --quiet                     quiet (no output)
  -v,  --verbose                   be verbose (this is the default)
  -nv, --no-verbose                turn off verboseness, without being quiet
       --report-speed=TYPE         output bandwidth as TYPE.  TYPE can be bits
  -i,  --input-file=FILE           download URLs found in local or external FILE
  -F,  --force-html                treat input file as HTML
  -B,  --base=URL                  resolves HTML input-file links (-i -F)
                                     relative to URL
       --config=FILE               specify config file to use
       --no-config                 do not read any config file
       --rejected-log=FILE         log reasons for URL rejection to FILE

Download:
```

```
  -t,  --tries=NUMBER              set number of retries to NUMBER (0 unlimits)
       --retry-connrefused         retry even if connection is refused
       --retry-on-http-error=ERRORS    comma-separated list of HTTP errors to retry
  -O,  --output-document=FILE      write documents to FILE
  -nc, --no-clobber                skip downloads that would download to
                                     existing files (overwriting them)
       --no-netrc                  don't try to obtain credentials from .netrc
  -c,  --continue                  resume getting a partially-downloaded file
       --start-pos=OFFSET          start downloading from zero-based position OFFSET
       --progress=TYPE             select progress gauge type
       --show-progress             display the progress bar in any verbosity mode
  -N,  --timestamping              don't re-retrieve files unless newer than
                                     local
       --no-if-modified-since      don't use conditional if-modified-since get
                                     requests in timestamping mode
       --no-use-server-timestamps  don't set the local file's timestamp by
                                     the one on the server
  -S,  --server-response           print server response
       --spider                    don't download anything
  -T,  --timeout=SECONDS           set all timeout values to SECONDS
       --dns-timeout=SECS          set the DNS lookup timeout to SECS
       --connect-timeout=SECS      set the connect timeout to SECS
       --read-timeout=SECS         set the read timeout to SECS
  -w,  --wait=SECONDS              wait SECONDS between retrievals
                                     (applies if more then 1 URL is to be retrieved)
--More--
[q]
```

3.3 - ftp

Important - Si la commande **ftp** n'est pas installée sous Debian 11, installez-le à l'aide de la commande **apt install ftp** en tant que root.

La commande **ftp** est utilisée pour le transfert de fichiers. Une fois connecté, il convient d'utiliser la commande **help** pour afficher la liste des commandes disponibles :

```
ftp> help
Commands may be abbreviated.  Commands are:

!               dir             mdelete         qc              site
$               disconnect      mdir            sendport        size
account         exit            mget            put             status
append          form            mkdir           pwd             struct
ascii           get             mls             quit            system
bell            glob            mode            quote           sunique
binary          hash            modtime         recv            tenex
bye             help            mput            reget           tick
case            idle            newer           rstatus         trace
cd              image           nmap            rhelp           type
cdup            ipany           nlist           rename          user
chmod           ipv4            ntrans          reset           umask
close           ipv6            open            restart         verbose
cr              lcd             prompt          rmdir           ?
delete          ls              passive         runique
debug           macdef          proxy           send
ftp>
```

Le caractère **!** permet d'exécuter une commande sur la machine cliente

```
ftp> !pwd
/root
```

Pour transférer un fichier vers le serveur, il convient d'utiliser la commande **put** :

```
ftp> put nom_fichier_local nom_fichier_distant
```

Vous pouvez également transférer plusieurs fichiers à la fois grâce à la commande **mput**. Dans ce cas précis, il convient de saisir la commande

suivante:

```
ftp> mput nom*.*
```

Pour transférer un fichier du serveur, il convient d'utiliser la commande **get** :

```
ftp> get nom_fichier
```

Vous pouvez également transférer plusieurs fichiers à la fois grâce à la commande **mget** (voir la commande **mput** ci-dessus).

Pour supprimer un fichier sur le serveur, il convient d'utiliser la commande **del** :

```
ftp> del nom_fichier
```

Pour fermer la session, il convient d'utiliser la commande **quit** :

```
ftp> quit
root@debian11:~#
```

3.4 - SSH

Présentation

La commande **ssh** est le successeur et la remplaçante de la commande **rlogin**. Il permet d'établir des connexions sécurisées avec une machine distante. SSH comporte cinq acteurs :

- Le **serveur SSH**
  - le démon sshd, qui s'occupe des authentifications et autorisations des clients,
- Le **client SSH**
  - ssh ou scp, qui assure la connexion et le dialogue avec le serveur,
- La **session** qui représente la connexion courante et qui commence juste après l'authentification réussie,
- Les **clefs**

    - **Couple de clef utilisateur asymétriques** et persistantes qui assurent l'identité d'un utilisateur et qui sont stockés sur disque dur,
    - **Clef hôte asymétrique et persistante** garantissant l'identité du serveur er qui est conservé sur disque dur
    - **Clef serveur asymétrique et temporaire** utilisée par le protocole SSH1 qui sert au chiffrement de la clé de session,
    - **Clef de session symétrique qui est générée aléatoirement** et qui permet le chiiffrement de la communication entre le client et le serveur. Elle est détruite en fin de session. SSH-1 utilise une seule clef tandis que SSH-2 utilise une clef par direction de la communication,
- La **base de données des hôtes connus** qui stocke les clés des connexions précédentes.

SSH fonctionne de la manière suivante pour la la mise en place d'un canal sécurisé:

- Le client contacte le serveur sur son port 22,
- Les client et le serveur échangent leur version de SSH. En cas de non-compatibilité de versions, l'un des deux met fin au processus,
- Le serveur SSH s'identifie auprès du client en lui fournissant :
    - Sa clé hôte,
    - Sa clé serveur,
    - Une séquence aléatoire de huit octets à inclure dans les futures réponses du client,
    - Une liste de méthodes de chiffrage, compression et authentification,
- Le client et le serveur produisent un identifiant identique, un haché MD5 long de 128 bits contenant la clé hôte, la clé serveur et la séquence aléatoire,
- Le client génère sa clé de session symétrique et la chiffre deux fois de suite, une fois avec la clé hôte du serveur et la deuxième fois avec la clé serveur. Le client envoie cette clé au serveur accompagnée de la séquence aléatoire et un choix d'algorithmes supportés,
- Le serveur déchiffre la clé de session,
- Le client et le serveur mettent en place le canal sécurisé.

SSH-1

SSH-1 utilise une paire de clefs de type RSA1. Il assure l'intégrité des données par une **Contrôle de Redondance Cyclique** (CRC) et est un bloc dit **monolithique**.

Afin de s'identifier, le client essaie chacune des six méthodes suivantes :

- **Kerberos**,
- **Rhosts**,
- **RhostsRSA**,
- Par **clef asymétrique**,

- **TIS**,
- Par **mot de passe**.

SSH-2

SSH-2 utilise **DSA** ou **RSA**. Il assure l'intégrité des données par l'algorithme **HMAC**. SSH-2 est organisé en trois **couches** :

- **SSH-TRANS** – Transport Layer Protocol,
- **SSH-AUTH** – Authentification Protocol,
- **SSH-CONN** – Connection Protocol.

SSH-2 diffère de SSH-1 essentiellement dans la phase authentification.

Trois méthodes d'authentification :

- Par **clef asymétrique**,
  - Identique à SSH-1 sauf avec l'algorithme DSA,
- **RhostsRSA**,
- Par **mot de passe**.

Les options de cette commande sont :

```
root@debian11:~# ssh --help
unknown option -- -
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
```

Authentification par mot de passe

L'utilisateur fournit un mot de passe au client ssh. Le client ssh le transmet de façon sécurisée au serveur ssh puis le serveur vérifie le mot de passe et l'accepte ou non.

Avantage:

- Aucune configuration de clef asymétrique n'est nécessaire.

Inconvénients:

- L'utilisateur doit fournir à chaque connexion un identifiant et un mot de passe,
- Moins sécurisé qu'un système par clef asymétrique.

Authentification par clef asymétrique

- Le **client** envoie au serveur une requête d'authentification par clé asymétrique qui contient le module de la clé à utiliser,
- Le **serveur** recherche une correspondance pour ce module dans le fichier des clés autorisés **~/.ssh/authorized_keys**,
  - Dans le cas où une correspondance n'est pas trouvée, le serveur met fin à la communication,
  - Dans le cas contraire le serveur génère une chaîne aléatoire de 256 bits appelée un **challenge** et la chiffre avec la **clé publique du client**,
- Le **client** reçoit le challenge et le décrypte avec la partie privée de sa clé. Il combine le challenge avec l'identifiant de session et chiffre le résultat. Ensuite il envoie le résultat chiffré au serveur.
- Le **serveur** génère le même haché et le compare avec celui reçu du client. Si les deux hachés sont identiques, l'authentification est réussie.

Configuration du Serveur

La configuration du serveur s'effectue dans le fichier **/etc/ssh/sshd_config** :

```
root@debian11:~# cat /etc/ssh/sshd_config
#       $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $
```

```
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
```

```
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
```

```
# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication.  Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
UsePAM yes

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
```

```
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem       sftp    /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
```

Pour ôter les lignes de commentaires dans ce fichier, utilisez la commande suivante :

```
root@debian11:~# cd /tmp ; grep -E -v '^(#|$)'  /etc/ssh/sshd_config > sshd_config
root@debian11:/tmp# cat sshd_config
Include /etc/ssh/sshd_config.d/*.conf
PermitRootLogin yes
ChallengeResponseAuthentication no
UsePAM yes
X11Forwarding yes
PrintMotd no
AcceptEnv LANG LC_*
Subsystem       sftp    /usr/lib/openssh/sftp-server
```

Pour sécuriser le serveur ssh, ajoutez ou modifiez les directives suivantes :

```
AllowGroups adm
Banner /etc/issue.net
HostbasedAuthentication no
IgnoreRhosts yes
LoginGraceTime 60
LogLevel INFO
PermitEmptyPasswords no
PermitRootLogin no
PrintLastLog yes
Protocol 2
StrictModes yes
X11Forwarding no
```

Votre fichier ressemblera à celui-ci :

```
root@debian11:/tmp# vi sshd_config
root@debian11:/tmp# cat sshd_config
AllowGroups adm
Banner /etc/issue.net
HostbasedAuthentication no
IgnoreRhosts yes
LoginGraceTime 60
LogLevel INFO
PermitEmptyPasswords no
PermitRootLogin no
PrintLastLog yes
Protocol 2
StrictModes yes
X11Forwarding no
Include /etc/ssh/sshd_config.d/*.conf
ChallengeResponseAuthentication no
UsePAM yes
PrintMotd no
AcceptEnv LANG LC_*
```

```
Subsystem       sftp    /usr/lib/openssh/sftp-server
```

Renommez le fichier **/etc/ssh/sshd_config** en **/etc/ssh/sshd_config.old** :

```
[root@centos11 tmp]# cp /etc/ssh/sshd_config /etc/ssh/sshd_config.old
```

Copiez le fichier **/tmp/sshd_config** vers **/etc/ssh/** :

```
root@debian11:/tmp# cp /tmp/sshd_config /etc/ssh
```

Redémarrez le service sshd :

```
root@debian11:/tmp# systemctl restart sshd
root@debian11:/tmp# systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2022-05-03 11:01:24 CEST; 7s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 4885 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 4888 (sshd)
      Tasks: 1 (limit: 4632)
     Memory: 1.1M
        CPU: 24ms
     CGroup: /system.slice/ssh.service
             └─4888 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

May 03 11:01:24 debian11.ittraining.loc sshd[4888]: Server listening on 0.0.0.0 port >
May 03 11:01:24 debian11.ittraining.loc systemd[1]: Stopping OpenBSD Secure Shell ser>
May 03 11:01:24 debian11.ittraining.loc sshd[4888]: Server listening on :: port 22.
May 03 11:01:24 debian11.ittraining.loc systemd[1]: ssh.service: Succeeded.
May 03 11:01:24 debian11.ittraining.loc systemd[1]: Stopped OpenBSD Secure Shell serv>
May 03 11:01:24 debian11.ittraining.loc systemd[1]: Starting OpenBSD Secure Shell ser>
May 03 11:01:24 debian11.ittraining.loc systemd[1]: Started OpenBSD Secure Shell serv>
```

```
lines 1-20/20 (END)
[q]
```

Mettez l'utilisateur **trainee** dans le groupe **adm** :

```
root@debian11:/tmp# groups trainee
trainee : trainee cdrom floppy audio dip src video plugdev netdev lpadmin scanner
root@debian11:/tmp# usermod -aG adm trainee
root@debian11:/tmp# groups trainee
trainee : trainee adm cdrom floppy audio dip src video plugdev netdev lpadmin scanner
```

Pour générer les clefs du serveur, saisissez la commande suivante en tant que **root**. Notez que la passphrase doit être **vide**.

```
root@debian11:/tmp# ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa): /etc/ssh/ssh_host_dsa_key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_dsa_key
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub
The key fingerprint is:
SHA256:emwF3Bq/2H+JGk5dNXgKBLTtbuC04byZmFJDxkUxso4 root@debian11.ittraining.loc
The key's randomart image is:
+---[DSA 1024]----+
|       .o*o.     |
|       .oo=   .  |
|      ..=..o . o.|
|      o+ =. . o..|
|     Eo.S+o. ..  |
|       ==+=o .   |
|      o *==o.. . |
|     . ooo=o. o  |
|      .o +o...   |
+----[SHA256]-----+
```

```
root@debian11:/tmp# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /etc/ssh/ssh_host_rsa_key
/etc/ssh/ssh_host_rsa_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_rsa_key
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub
The key fingerprint is:
SHA256:xrkpZ6lfF1hQIZEYJuH1L+Z2QGhQCf8Xwt9HPwTuT7Y root@debian11.ittraining.loc
The key's randomart image is:
+---[RSA 3072]----+
|      =+=+++oo   |
|     . *o+.o. .  |
|      . + = o. ..|
|       o + *.o...|
|        S * =..+o|
|       . * + .+.o|
|      . * + o  E |
|       = o o     |
|      ...        |
+----[SHA256]-----+
root@debian11:/tmp# ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/root/.ssh/id_ecdsa): /etc/ssh/ssh_host_ecdsa_key
/etc/ssh/ssh_host_ecdsa_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_ecdsa_key
Your public key has been saved in /etc/ssh/ssh_host_ecdsa_key.pub
The key fingerprint is:
SHA256:38O9lqa1AHvviceNEbQ1AOUMspkYBpIpLlu/U05ymqo root@debian11.ittraining.loc
```

```
The key's randomart image is:
+---[ECDSA 256]---+
|   .o..o . ooo.  |
| . o. . o = + ...|
|. .    . +   + o.|
|...           o  |
|.o .    S .    . |
|.   o +  . = ..  |
|     X    o *.o= |
|    = .    . *B+.|
|E... .      o**  |
+----[SHA256]-----+
root@debian11:/tmp# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519): /etc/ssh/ssh_host_ed25519_key
/etc/ssh/ssh_host_ed25519_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_ed25519_key
Your public key has been saved in /etc/ssh/ssh_host_ed25519_key.pub
The key fingerprint is:
SHA256:f8arQ5MBRGNJoj4eARYapvxf/MLxFFMZcKf1eLkgeow root@debian11.ittraining.loc
The key's randomart image is:
+--[ED25519 256]--+
|..+.  .+*o.+oo   |
|++ . . oo.o.+ o .|
|o.  o    + o o + |
|  .. ..   B . o .|
|   .+  +SE =   . |
|   ..oo =.=.     |
|    .. o +..+    |
|       . .o .    |
|         .o.     |
```

```
+----[SHA256]-----+
```

Les clefs publiques générées possèdent l'extension **.pub**. Les clefs privées n'ont pas d'extension :

```
root@debian11:/tmp# ls /etc/ssh
moduli        sshd_config.d         ssh_host_ecdsa_key        ssh_host_rsa_key
ssh_config    sshd_config.old       ssh_host_ecdsa_key.pub    ssh_host_rsa_key.pub
ssh_config.d  ssh_host_dsa_key      ssh_host_ed25519_key
sshd_config   ssh_host_dsa_key.pub  ssh_host_ed25519_key.pub
```

Re-démarrez ensuite le service sshd :

```
root@debian11:/tmp# systemctl restart sshd.service
root@debian11:/tmp# systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2022-05-03 11:30:09 CEST; 8s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 4942 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 4943 (sshd)
      Tasks: 1 (limit: 4632)
     Memory: 1.1M
        CPU: 24ms
     CGroup: /system.slice/ssh.service
             └─4943 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

May 03 11:30:09 debian11.ittraining.loc systemd[1]: Starting OpenBSD Secure Shell ser>
May 03 11:30:09 debian11.ittraining.loc sshd[4943]: Server listening on 0.0.0.0 port >
May 03 11:30:09 debian11.ittraining.loc sshd[4943]: Server listening on :: port 22.
May 03 11:30:09 debian11.ittraining.loc systemd[1]: Started OpenBSD Secure Shell serv>
lines 1-17/17 (END)
[q]
```

Configuration du Client

Saisissez maintenant les commandes suivantes en tant que **trainee** :

Important - Lors de la génération des clefs, la passphrase doit être **vide**.

```
root@debian11:/tmp# exit
logout
trainee@debian11:~$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_dsa):
Created directory '/home/trainee/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_dsa
Your public key has been saved in /home/trainee/.ssh/id_dsa.pub
The key fingerprint is:
SHA256:WijVQNwc9klBZxW4R8ZY5LwZK08/FaZCkWP0yKNj+20 trainee@debian11.ittraining.loc
The key's randomart image is:
+---[DSA 1024]----+
|     ooooo=++B=. |
|      .+oo.B*o+  |
|      . . +=.=+o |
|     . .  o o +=.|
|    . . S+ ..o= .|
|     . o. o .+ ..|
|      .  .    ...|
|          . .E  .|
|           ...   |
+----[SHA256]-----+
```

```
trainee@debian11:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_rsa
Your public key has been saved in /home/trainee/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:r8id4Px97b9t7GMybe7OTaQU5lgaRjsTK/EAyMFdgjo trainee@debian11.ittraining.loc
The key's randomart image is:
+---[RSA 3072]----+
|     o.=oo+ o    |
|      = .. = +   |
|     .    . O +  |
|    E      o X . |
|     .  S   o o .|
|         .   . o |
|      .   .  .o..|
|     + + +  .oo*=|
|      =.= .. .XXB|
+----[SHA256]-----+
trainee@debian11:~$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_ecdsa
Your public key has been saved in /home/trainee/.ssh/id_ecdsa.pub
The key fingerprint is:
SHA256:IL44+ZzExDxBFCt9nZtMgCQB/iuq9UtJC6uq/RhrPvg trainee@debian11.ittraining.loc
The key's randomart image is:
+---[ECDSA 256]---+
|..o+=o.          |
|.  +.. o .       |
```

```
| .. = o +        |
|  .= + + o       |
|  ..B   S        |
|   O.=           |
| oB.B            |
|o++@ .           |
|X+EoB.           |
+----[SHA256]-----+
trainee@debian11:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_ed25519
Your public key has been saved in /home/trainee/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:yQ9mtIx1nK7D1vSZjkkbofXHZsTDp5P0rywiJIMX35M trainee@debian11.ittraining.loc
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|         . .     |
|        o +      |
|      .* =   o   |
|     ..oS.=.  * .|
|    . ++oOE+ * * |
|     . +* =.= O .|
|       ..o.*.= ..|
|         .+...o..|
+----[SHA256]-----+
```

Les clés générées seront placées dans le répertoire **~/.ssh/** :

```
trainee@debian11:~$ ls .ssh
id_dsa      id_ecdsa      id_ed25519      id_rsa
```

```
id_dsa.pub  id_ecdsa.pub  id_ed25519.pub  id_rsa.pub
```

Tunnels SSH

Le protocole SSH peut être utilisé pour sécuriser les protocoles tels telnet, pop3 etc.. En effet, on peut créer un *tunnel* SSH dans lequel passe les communications du protocole non-sécurisé.

La commande pour créer un tunnel ssh prend la forme suivante :

```
ssh -N -f compte@hôte -Lport-local:localhost:port_distant
```

Dans votre cas, vous allez créer un tunnel dans votre propre VM entre le port 15023 et le port 23 :

```
trainee@debian11:~$ su -
Password: fenestros
root@debian11:~# ssh -N -f trainee@localhost -L15023:localhost:23
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:38O9lqa1AHvviceNEbQ1AOUMspkYBpIpLlu/U05ymqo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Debian GNU/Linux 11
trainee@localhost's password: trainee
```

Installez maintenant le serveur telnet :

```
root@debian11:~# apt -y install telnetd
```

Vérifiez que le service **inetd** est démarré :

```
root@debian11:~# systemctl status inetd
● inetd.service - Internet superserver
     Loaded: loaded (/lib/systemd/system/inetd.service; enabled; vendor preset: enabl>
```

```
     Active: active (running) since Tue 2022-05-03 11:55:27 CEST; 44s ago
       Docs: man:inetd(8)
   Main PID: 5110 (inetd)
      Tasks: 1 (limit: 4632)
     Memory: 576.0K
        CPU: 7ms
     CGroup: /system.slice/inetd.service
             └─5110 /usr/sbin/inetd

May 03 11:55:27 debian11.ittraining.loc systemd[1]: Starting Internet superserver...
May 03 11:55:27 debian11.ittraining.loc systemd[1]: Started Internet superserver.
```

Connectez-vous ensuite via telnet sur le port 15023, vous constaterez que votre connexion n'aboutit pas :

```
root@debian11:~# telnet localhost 15023
Trying ::1...
Connected to localhost.
Escape character is '^]'.
Debian GNU/Linux 11
debian11.ittraining.loc login: trainee
Password: trainee
Linux debian11.ittraining.loc 5.17.0-1-amd64 #1 SMP PREEMPT Debian 5.17.3-1 (2022-04-18) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May  3 08:57:59 CEST 2022 from 10.0.2.1 on pts/0

trainee@debian11:~$ whoami
trainee
```

```
trainee@debian11:~$ pwd
/home/trainee

trainee@debian11:~$ exit
logout
Connection closed by foreign host.
```

Important - Notez bien que votre communication telnet passe par le tunnel SSH.

3.5 - SCP

Présentation

La commande **scp** est le successeur et la remplaçante de la commande **rcp** de la famille des commandes **remote**. Il permet de faire des transferts sécurisés à partir d'une machine distante :

```
$ scp compte@numero_ip(nom_de_machine):/chemin_distant/fichier_distant /chemin_local/fichier_local
```

ou vers une machine distante :

```
$ scp /chemin_local/fichier_local compte@numero_ip(nom_de_machine):/chemin_distant/fichier_distant
```

Utilisation

Nous allons maintenant utiliser **scp** pour chercher un fichier sur le «serveur» :

Créez le fichier **/home/trainee/scp_test** :

```
[trainee@centos8 ~]$ touch scp-test
[trainee@centos8 ~]$ exit
logout
Connection closed by foreign host.
[root@centos8 ~]#
```

Récupérez le fichier **scp_test** en utilisant scp :

```
[root@centos8 ~]# scp trainee@127.0.0.1:/home/trainee/scp-test .
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:Q7T/CP0SLiMbMAIgVzTuEHegYS/spPE5zzQchCHD5Vw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '127.0.0.1' (ECDSA) to the list of known hosts.
\S
Kernel \r on an \m
trainee@127.0.0.1's password: trainee
scp-test
100%    0     0.0KB/s   00:00

[root@centos8 ~]# ls -l
total 32
-rw-------. 1 root root 1358 Jun 16 06:40 anaconda-ks.cfg
drwxr-xr-x. 3 root root   21 Jun 16 06:39 home
-rw-r--r--. 1 root root 1749 Aug 24 11:20 I2TCH.asc
-rw-r--r--. 1 root root 1853 Jun 16 06:54 initial-setup-ks.cfg
-rw-r--r--. 1 root root   31 Aug 24 11:22 message.txt
-rw-r--r--. 1 root root  561 Aug 24 11:32 message.txt.asc
-rw-r--r--. 1 root root  367 Aug 24 11:30 message.txt.gpg
-rw-r--r--. 1 root root  329 Aug 24 11:23 message.txt.sig
-rw-r--r--. 1 root root    0 Aug 30 03:55 scp-test
-rw-r--r--. 1 root root   46 Aug 29 06:22 wget_file.txt
```

3.6 - Mise en Place des Clefs Asymétriques

Il convient maintenant de se connecter sur le «serveur» en utilisant ssh et vérifiez la présence du répertoire ~/.ssh :

```
[root@centos8 ~]# ssh -l trainee 127.0.0.1
\S
Kernel \r on an \m
trainee@127.0.0.1's password: trainee
Activate the web console with: systemctl enable --now cockpit.socket

[trainee@centos8 ~]$ ls -la | grep .ssh
drwx------.  2 trainee trainee      4096 Aug 30 02:26 .ssh
```

Important - Si le dossier distant .ssh n'existe pas dans le répertoire personnel de l'utilisateur connecté, il faut le créer avec des permissions de 700. Dans votre cas, puisque votre machine joue le rôle de serveur **et** du client, le dossier /home/trainee/.ssh existe **déjà**.

Ensuite, il convient de transférer le fichier local **.ssh/id_ecdsa.pub** du «client» vers le «serveur» en le renommant en **authorized_keys** :

```
[trainee@centos8 ~]$ exit
logout
Connection to 127.0.0.1 closed.

[root@centos8 ~]# exit
logout

[trainee@centos8 ~]$ scp .ssh/id_ecdsa.pub trainee@127.0.0.1:/home/trainee/.ssh/authorized_keys
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:Q7T/CP0SLiMbMAIgVzTuEHegYS/spPE5zzQchCHD5Vw.
```

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '127.0.0.1' (ECDSA) to the list of known hosts.
\S
Kernel \r on an \m
trainee@127.0.0.1's password: trainee
id_ecdsa.pub
100%  192   497.6KB/s   00:00
```

Connectez-vous via telnet :

```
[trainee@centos8 ~]$ ssh -l trainee localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:Q7T/CP0SLiMbMAIgVzTuEHegYS/spPE5zzQchCHD5Vw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
\S
Kernel \r on an \m
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Mon Aug 30 03:57:14 2021 from 127.0.0.1
[trainee@centos8 ~]$
```

Important - Lors de la connexion au serveur, l'authentification utilise le couple de clefs asymétrique au format ecdsa et aucun mot de passe n'est requis.

Insérez maintenant les clefs publiques restantes dans le fichier .ssh/authorized_keys :

```
[trainee@centos8 ~]$ cd .ssh
[trainee@centos8 .ssh]$ ls
authorized_keys  id_dsa  id_dsa.pub  id_ecdsa  id_ecdsa.pub  id_ed25519  id_ed25519.pub  id_rsa  id_rsa.pub
known_hosts
```

```
[trainee@centos8 .ssh]$ cat authorized_keys
ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHDrzSXP+Ecxf/sQ18VwCRNm7rrSrrsaJmuIw/RgTH5puKF5E+Yy15cvAAKBX
pJPxUmrOaOyhab84PevV7XSHcI= trainee@centos8.ittraining.loc

[trainee@centos8 .ssh]$ cat id_rsa.pub >> authorized_keys
[trainee@centos8 .ssh]$ cat id_dsa.pub >> authorized_keys
[trainee@centos8 .ssh]$ cat id_ed25519.pub >> authorized_keys

[trainee@centos8 .ssh]$ cat authorized_keys
ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHDrzSXP+Ecxf/sQ18VwCRNm7rrSrrsaJmuIw/RgTH5puKF5E+Yy15cvAAKBX
pJPxUmrOaOyhab84PevV7XSHcI= trainee@centos8.ittraining.loc
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQD3ZSMn/GIAHtaDFc6ZNnKJam9hzq8TxqMN5IopUr8QhwODyPadbB+FgH4r50qTux4ubwr1BlymgIdqRVWy3
2mE15M8tdtKc3j8DNMpUwPGEh+s/PT7GW+3E3shoyPvpLc1kKaKXKGl/JwfCK/8IYsubk2BmiiJYkzLECotPlaaxm4w1K0AtlnZQuLHt1HK3/rHCh
xo2o2w1t59/QwNcMLiKve1Z+zQ1POKo8VJ/DDrf90y2QWC28ejUs/ZjP6f6C4bn5Jjol4TbHls3ArMsbU6C1Ev5jqbzZ0kmognQ2CnRjeNM51YdFo
6nRsLoPQKpeLRMBpT/87HK1bPUOBVCyhXxSkqkVhMlgg8tgcD/MRlBaUVFoZ1wQ0L26Fe+q7c2Oykj54pdXCAK2ZTpCXGfhd/FxrfqGw7cSeKlGX4
QUzHMjMk2oHFC9h30BUk1gGN21KJPTh7/S1OZVrnCc3GUi5fXPvEpral0IU0sws+j03dj0sWm5ICQFKRkmZN11HCyT0=
trainee@centos8.ittraining.loc
ssh-dss
AAAAB3NzaC1kc3MAAACBALIdwEEqHrMWSUdzARm9ldsZK9ebbtZShtmwgdjphOk77fxymK0y6wV7QEmLL25LOcLb12uZ1F0LtRt/t2oqgrwqk3vUS
pCPLr09AXpcD/nxL9kc+rUxHyl6u1mHtyfCVLCPSvavCMR8TaA8egVMk3EwGRfHTiuDOKi7Iwus7gXPAAAAFQDHEQPGVRI7gVYKzCT6nrjDsQQ6jw
AAAIEAhhhH7fEjdldASXY0qTWkCvcs3cfK9/Ff315zByn47O02y9Vdo3QG5nOr1Oo8fc2xEkIBNmFr8Rr2g60cpvEev5hy4XZ1ghxnQ53iwKuiS72
ZATwhD6bZBrsiH0k1Et25gRcj5KCvDe/jHhbxCxsCuHUH2qvWsQNVwztE7hD0sxkAAACAQ8Dkpy8zXj7jW8o1txxf2W6J4r2+1lPldymA45ywZokN
4SCwvXlpPAuyBt0/HiU0R2PI9aqOAMosCLcy9WmnSwLQ2Z7QcD2i3XlAih2+1q9NJP22sPT3jSK9UZcdRjoZ/eNiz84sXZucNape32tFxjvcV4txo
bH/vD53q8g63fA= trainee@centos8.ittraining.loc
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOfFQULLU8IZyKiSU63D2Zz6yGLqyHcBHnCRdSR9JSmc trainee@centos8.ittraining.loc
```

3.7 - Services réseaux

Quand un client émet une demande de connexion vers une application réseau sur un serveur, il utilise un socket attaché à un port local **supérieur à 1023**, alloué d'une manière dynamique. La requête contient le port de destination sur le serveur. Certaines applications serveurs se gèrent toutes

seules, ce qui est la cas par exemple d'**apache2**. Par contre d'autres sont gérées par le service **xinetd**.

inetd

Le programme inetd est configuré via le fichier **/etc/inetd.conf** :

```
root@debian11:~# cat /etc/inetd.conf
# /etc/inetd.conf:  see inetd(8) for further informations.
#
# Internet superserver configuration database
#
#
# Lines starting with "#:LABEL:" or "#<off>#" should not
# be changed unless you know what you are doing!
#
# If you want to disable an entry so it isn't touched during
# package updates just comment it out with a single '#' character.
#
# Packages should modify this file by using update-inetd(8)
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
#:INTERNAL: Internal services
#discard                stream  tcp     nowait  root    internal
#discard                dgram   udp     wait    root    internal
#daytime                stream  tcp     nowait  root    internal
#time           stream  tcp     nowait  root    internal

#:STANDARD: These are standard services.
telnet          stream  tcp     nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.telnetd

#:BSD: Shell, login, exec and talk are BSD protocols.
```

```
#:MAIL: Mail, news and uucp services.

#:INFO: Info services

#:BOOT: TFTP service is provided primarily for booting.  Most sites
#       run this only on machines acting as "boot servers."

#:RPC: RPC based services

#:HAM-RADIO: amateur-radio services

#:OTHER: Other services
```

Les lignes de configuration des serveurs ressemblent à :

```
telnet          stream  tcp     nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.telnetd
```

Le premier champs de la ligne identifie le nom du port qui identifie l'application. Inetd lit ensuite le fichier **/etc/services** pour déduire le numéro de port concerné et se met à l'écoute de celui-ci.

Le deuxième et le troisième champs définissent le type de protocole, à savoir:

- stream tcp pour le tcp
- dgram udp pour l'udp

Le quatrième champs prend un de deux valeurs:

- nowait
  - indique qu'il y aura un serveur par client
- wait
  - indique qu'il y aura un seul serveur pour l'ensemble des clients

Le cinquième champs indique l'identité sous laquelle sera exécuté le serveur.

Le sixième champs indique l'exécutable. Dans ce cas c'est **/usr/sbin/tcpd**.

Le septième champs indique les arguments de l'application serveur dont l'argument **0** est le nom de l'application.

TCP Wrapper

Lors de l'utilisation d'inetd, **TCP Wrapper** est utilisé pour contrôler l'accès à des services réseaux grâce à des **ACL** :

```
telnet          stream  tcp     nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.telnetd
```

Quand une requête arrive pour le serveur telnet, inetd active le wrapper **tcpd** au lieu d'activer **/usr/sbin/in.telnetd** directement.

tcpd met à jour un journal et vérifie si le client a le droit d'utiliser le service concerné. Les ACL se trouvent dans deux fichiers:

```
root@debian11:~# cat /etc/hosts.allow
# /etc/hosts.allow: list of hosts that are allowed to access the system.
#                   See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: LOCAL @some_netgroup
#             ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#

root@debian11:~# cat /etc/hosts.deny
# /etc/hosts.deny: list of hosts that are _not_ allowed to access the system.
#                  See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: some.host.name, .some.domain
#             ALL EXCEPT in.fingerd: other.host.name, .other.domain
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
```

```
# The PARANOID wildcard matches any host whose name does not match its
# address.
#
# You may wish to enable this to ensure any programs that don't
# validate looked up hostnames still leave understandable logs. In past
# versions of Debian this has been the default.
# ALL: PARANOID
```

Il faut noter que si ces fichiers n'existent pas ou sont vides, il n'y a pas de contrôle d'accès.

Le format d'une ligne dans un de ces deux fichiers est:

```
démon: liste_de_clients
```

Par exemple dans le cas de notre serveur telnetd, une ligne dans le fichier **/etc/hosts.allow** similaire à:

```
in.telnetd: 192.168.1.10, .fenestros.com
```

implique que la machine dont le numéro IP est le 192.168.1.10 ainsi que les machines du domaine **fenestros.com** sont autorisées à utiliser le service.

Le mot clef **ALL** peut être utilisé pour indiquer tout. Par exemple, **ALL:ALL** dans le fichier **/etc/host.deny** bloque effectivement toute tentative de connexion à un service inetd sauf pour les ACL inclus dans le fichier /etc/host.allow.

LAB #4 - Le Parefeu Netfilter

4.1 - Présentation

Netfilter est composé de 5 *hooks* :

- NF_IP_PRE_ROUTING
- NF_IP_LOCAL_IN
- NF_IP_LOCAL_OUT

- NF_IP_FORWARD
- NF_IP_POSTROUTING

Ces hooks sont utilisés par deux branches, la première est celle concernée par les paquets qui entrent vers des services locaux :

- NF_IP_PRE_ROUTING > NF_IP_LOCAL_IN > NF_IP_LOCAL_OUT > NF_IP_POSTROUTING

tandis que la deuxième concerne les paquets qui traversent la passerelle:

- NF_IP_PRE_ROUTING > NF_IP_FORWARD > NF_IP_POSTROUTING

Si IPTABLES a été compilé en tant que module, son utilisation nécessite le chargement de plusieurs modules supplémentaires en fonction de la situation:

- iptable_filter
- iptable_mangle
- iptable_net
- etc

Netfilter est organisé en **tables**. La commande **iptables** de netfilter permet d'insérer des **policies** dans les **chaines**:

- La table **FILTER**
  - La chaîne INPUT
    - Concerne les paquets entrants
      - Policies: ACCEPT, DROP, REJECT
  - La chaîne OUTPUT
    - Concerne les paquets sortants
      - Policies: ACCEPT, DROP, REJECT
  - La chaîne FORWARD
    - Concerne les paquets traversant le par-feu.
      - Policies: ACCEPT, DROP, REJECT

Si aucune table n'est précisée, c'est la table FILTER qui s'applique par défaut.

- La table **NAT**
  - La chaîne PREROUTING

        - Permet de faire la translation d'adresse de destination
            - Cibles: SNAT, DNAT, MASQUERADE
    - La chaîne POSTROUTING
        - Permet de faire la translation d'adresse de la source
            - Cibles: SNAT, DNAT, MASQUERADE
    - Le cas spécifique OUTPUT
        - Permet la modification de la destination des paquets générés localement

- La table **MANGLE**
    - Permet le marquage de paquets générés localement (OUTPUT) et entrants (PREROUTING)

Les **policies** sont:

- ACCEPT
    - Permet d'accepter le paquet concerné
- DROP
    - Permet de rejeter le paquet concerné sans générer un message d'erreur
- REJECT
    - Permet de rejeter le paquet concerné en générant une message d'erreur

Les **cibles** sont:

- SNAT
    - Permet de modifier l'adresse source du paquet concerné
- DNAT
    - Permet de modifier l'adresse de destination du paquet concerné
- MASQUERADE
    - Permet de remplacer l'adresse IP privée de l'expéditeur par un socket public de la passerelle.

4.2 - La Configuration de Netfilter par firewalld

firewalld est à Netfilter ce que NetworkManager est au réseau. firewalld utilise des **zones** - des jeux de règles pré-définis dans lesquels sont placés les interfaces :

- **trusted** - un réseau fiable. Dans ce cas tous les ports sont autorisés,
- **work**, **home**, **internal** - un réseau partiellement fiable. Dans ce cas quelques ports sont autorisés,
- **dmz**, **public**, **external** - un réseau non fiable. Dans ce cas peu de ports sont autorisés,
- **block**, **drop** - tout est interdit. La zone drop n'envoie pas de messages d'erreurs.

Important - Une interface ne peut être que dans une zone à la fois tandis que plusieurs interfaces peuvent être dans la même zone.

Sous Debian 11, firewalld n'est pas installé par défaut :

```
root@debian11:~# apt-get -y install firewalld
```

Le service firewalld est déjà lancé et activé :

```
root@debian11:~# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/lib/systemd/system/firewalld.service; enabled; vendor preset: e>
     Active: active (running) since Tue 2022-05-03 15:17:03 CEST; 11s ago
       Docs: man:firewalld(1)
   Main PID: 5695 (firewalld)
      Tasks: 2 (limit: 4632)
     Memory: 30.2M
        CPU: 619ms
     CGroup: /system.slice/firewalld.service
             └─5695 /usr/bin/python3 /usr/sbin/firewalld --nofork --nopid

May 03 15:17:02 debian11.ittraining.loc systemd[1]: Starting firewalld - dynamic fire>
May 03 15:17:03 debian11.ittraining.loc systemd[1]: Started firewalld - dynamic firew>
lines 1-13/13 (END)
[q]
```

La Configuration de Base de firewalld

La configuration par défaut de firewalld se trouve dans **/usr/lib/firewalld** :

```
root@debian11:~# ls -lR /usr/lib/firewalld/
/usr/lib/firewalld/:
total 32
drwxr-xr-x 2 root root  4096 May  3 15:16 helpers
drwxr-xr-x 2 root root  4096 May  3 15:16 icmptypes
drwxr-xr-x 2 root root  4096 May  3 15:16 ipsets
drwxr-xr-x 2 root root  4096 May  3 15:16 policies
drwxr-xr-x 2 root root 12288 May  3 15:16 services
drwxr-xr-x 2 root root  4096 May  3 15:16 zones

/usr/lib/firewalld/helpers:
total 52
-rw-r--r-- 1 root root 125 Feb  1  2021 amanda.xml
-rw-r--r-- 1 root root 119 Feb  1  2021 ftp.xml
-rw-r--r-- 1 root root  85 Feb  1  2021 h323.xml
-rw-r--r-- 1 root root 134 Feb  1  2021 irc.xml
-rw-r--r-- 1 root root 141 Feb  1  2021 netbios-ns.xml
-rw-r--r-- 1 root root 136 Feb  1  2021 pptp.xml
-rw-r--r-- 1 root root  90 Feb  1  2021 proto-gre.xml
-rw-r--r-- 1 root root 122 Feb  1  2021 Q.931.xml
-rw-r--r-- 1 root root 122 Feb  1  2021 RAS.xml
-rw-r--r-- 1 root root 122 Feb  1  2021 sane.xml
-rw-r--r-- 1 root root 158 Feb  1  2021 sip.xml
-rw-r--r-- 1 root root 135 Feb  1  2021 snmp.xml
-rw-r--r-- 1 root root 120 Feb  1  2021 tftp.xml

/usr/lib/firewalld/icmptypes:
total 180
-rw-r--r-- 1 root root 385 Feb  1  2021 address-unreachable.xml
```

```
-rw-r--r-- 1 root root 258 Feb  1  2021 bad-header.xml
-rw-r--r-- 1 root root 294 Feb  1  2021 beyond-scope.xml
-rw-r--r-- 1 root root 279 Feb  1  2021 communication-prohibited.xml
-rw-r--r-- 1 root root 222 Feb  1  2021 destination-unreachable.xml
-rw-r--r-- 1 root root 173 Feb  1  2021 echo-reply.xml
-rw-r--r-- 1 root root 210 Feb  1  2021 echo-request.xml
-rw-r--r-- 1 root root 261 Feb  1  2021 failed-policy.xml
-rw-r--r-- 1 root root 280 Feb  1  2021 fragmentation-needed.xml
-rw-r--r-- 1 root root 266 Feb  1  2021 host-precedence-violation.xml
-rw-r--r-- 1 root root 257 Feb  1  2021 host-prohibited.xml
-rw-r--r-- 1 root root 242 Feb  1  2021 host-redirect.xml
-rw-r--r-- 1 root root 239 Feb  1  2021 host-unknown.xml
-rw-r--r-- 1 root root 247 Feb  1  2021 host-unreachable.xml
-rw-r--r-- 1 root root 229 Feb  1  2021 ip-header-bad.xml
-rw-r--r-- 1 root root 355 Feb  1  2021 neighbour-advertisement.xml
-rw-r--r-- 1 root root 457 Feb  1  2021 neighbour-solicitation.xml
-rw-r--r-- 1 root root 250 Feb  1  2021 network-prohibited.xml
-rw-r--r-- 1 root root 248 Feb  1  2021 network-redirect.xml
-rw-r--r-- 1 root root 239 Feb  1  2021 network-unknown.xml
-rw-r--r-- 1 root root 247 Feb  1  2021 network-unreachable.xml
-rw-r--r-- 1 root root 239 Feb  1  2021 no-route.xml
-rw-r--r-- 1 root root 328 Feb  1  2021 packet-too-big.xml
-rw-r--r-- 1 root root 225 Feb  1  2021 parameter-problem.xml
-rw-r--r-- 1 root root 233 Feb  1  2021 port-unreachable.xml
-rw-r--r-- 1 root root 256 Feb  1  2021 precedence-cutoff.xml
-rw-r--r-- 1 root root 249 Feb  1  2021 protocol-unreachable.xml
-rw-r--r-- 1 root root 185 Feb  1  2021 redirect.xml
-rw-r--r-- 1 root root 244 Feb  1  2021 reject-route.xml
-rw-r--r-- 1 root root 241 Feb  1  2021 required-option-missing.xml
-rw-r--r-- 1 root root 227 Feb  1  2021 router-advertisement.xml
-rw-r--r-- 1 root root 223 Feb  1  2021 router-solicitation.xml
-rw-r--r-- 1 root root 248 Feb  1  2021 source-quench.xml
-rw-r--r-- 1 root root 236 Feb  1  2021 source-route-failed.xml
-rw-r--r-- 1 root root 253 Feb  1  2021 time-exceeded.xml
```

```
-rw-r--r-- 1 root root 233 Feb  1  2021 timestamp-reply.xml
-rw-r--r-- 1 root root 228 Feb  1  2021 timestamp-request.xml
-rw-r--r-- 1 root root 258 Feb  1  2021 tos-host-redirect.xml
-rw-r--r-- 1 root root 257 Feb  1  2021 tos-host-unreachable.xml
-rw-r--r-- 1 root root 272 Feb  1  2021 tos-network-redirect.xml
-rw-r--r-- 1 root root 269 Feb  1  2021 tos-network-unreachable.xml
-rw-r--r-- 1 root root 293 Feb  1  2021 ttl-zero-during-reassembly.xml
-rw-r--r-- 1 root root 256 Feb  1  2021 ttl-zero-during-transit.xml
-rw-r--r-- 1 root root 259 Feb  1  2021 unknown-header-type.xml
-rw-r--r-- 1 root root 249 Feb  1  2021 unknown-option.xml

/usr/lib/firewalld/ipsets:
total 4
-rw-r--r-- 1 root root 29 Feb  1  2021 README

/usr/lib/firewalld/policies:
total 4
-rw-r--r-- 1 root root 649 Feb  1  2021 allow-host-ipv6.xml

/usr/lib/firewalld/services:
total 700
-rw-r--r-- 1 root root  399 Feb  1  2021 amanda-client.xml
-rw-r--r-- 1 root root  427 Feb  1  2021 amanda-k5-client.xml
-rw-r--r-- 1 root root  283 Feb  1  2021 amqps.xml
-rw-r--r-- 1 root root  273 Feb  1  2021 amqp.xml
-rw-r--r-- 1 root root  285 Feb  1  2021 apcupsd.xml
-rw-r--r-- 1 root root  301 Feb  1  2021 audit.xml
-rw-r--r-- 1 root root  320 Feb  1  2021 bacula-client.xml
-rw-r--r-- 1 root root  346 Feb  1  2021 bacula.xml
-rw-r--r-- 1 root root  429 Feb  1  2021 bb.xml
-rw-r--r-- 1 root root  339 Feb  1  2021 bgp.xml
-rw-r--r-- 1 root root  275 Feb  1  2021 bitcoin-rpc.xml
-rw-r--r-- 1 root root  307 Feb  1  2021 bitcoin-testnet-rpc.xml
-rw-r--r-- 1 root root  281 Feb  1  2021 bitcoin-testnet.xml
```

```
-rw-r--r-- 1 root root  244 Feb  1  2021 bitcoin.xml
-rw-r--r-- 1 root root  410 Feb  1  2021 bittorrent-lsd.xml
-rw-r--r-- 1 root root  294 Feb  1  2021 ceph-mon.xml
-rw-r--r-- 1 root root  329 Feb  1  2021 ceph.xml
-rw-r--r-- 1 root root  168 Feb  1  2021 cfengine.xml
-rw-r--r-- 1 root root  211 Feb  1  2021 cockpit.xml
-rw-r--r-- 1 root root  296 Feb  1  2021 collectd.xml
-rw-r--r-- 1 root root  260 Feb  1  2021 condor-collector.xml
-rw-r--r-- 1 root root  296 Feb  1  2021 ctdb.xml
-rw-r--r-- 1 root root  305 Feb  1  2021 dhcpv6-client.xml
-rw-r--r-- 1 root root  234 Feb  1  2021 dhcpv6.xml
-rw-r--r-- 1 root root  227 Feb  1  2021 dhcp.xml
-rw-r--r-- 1 root root  205 Feb  1  2021 distcc.xml
-rw-r--r-- 1 root root  318 Feb  1  2021 dns-over-tls.xml
-rw-r--r-- 1 root root  346 Feb  1  2021 dns.xml
-rw-r--r-- 1 root root  374 Feb  1  2021 docker-registry.xml
-rw-r--r-- 1 root root  391 Feb  1  2021 docker-swarm.xml
-rw-r--r-- 1 root root  228 Feb  1  2021 dropbox-lansync.xml
-rw-r--r-- 1 root root  338 Feb  1  2021 elasticsearch.xml
-rw-r--r-- 1 root root  304 Feb  1  2021 etcd-client.xml
-rw-r--r-- 1 root root  304 Feb  1  2021 etcd-server.xml
-rw-r--r-- 1 root root  224 Feb  1  2021 finger.xml
-rw-r--r-- 1 root root  270 Feb  1  2021 foreman-proxy.xml
-rw-r--r-- 1 root root  408 Feb  1  2021 foreman.xml
-rw-r--r-- 1 root root  709 Feb  1  2021 freeipa-4.xml
-rw-r--r-- 1 root root  489 Feb  1  2021 freeipa-ldaps.xml
-rw-r--r-- 1 root root  488 Feb  1  2021 freeipa-ldap.xml
-rw-r--r-- 1 root root  242 Feb  1  2021 freeipa-replication.xml
-rw-r--r-- 1 root root  657 Feb  1  2021 freeipa-trust.xml
-rw-r--r-- 1 root root  361 Feb  1  2021 ftp.xml
-rw-r--r-- 1 root root  184 Feb  1  2021 ganglia-client.xml
-rw-r--r-- 1 root root  176 Feb  1  2021 ganglia-master.xml
-rw-r--r-- 1 root root  212 Feb  1  2021 git.xml
-rw-r--r-- 1 root root  218 Feb  1  2021 grafana.xml
```

```
-rw-r--r-- 1 root root  119 Feb  1  2021 gre.xml
-rw-r--r-- 1 root root  608 Feb  1  2021 high-availability.xml
-rw-r--r-- 1 root root  448 Feb  1  2021 https.xml
-rw-r--r-- 1 root root  353 Feb  1  2021 http.xml
-rw-r--r-- 1 root root  372 Feb  1  2021 imaps.xml
-rw-r--r-- 1 root root  327 Feb  1  2021 imap.xml
-rw-r--r-- 1 root root  454 Feb  1  2021 ipp-client.xml
-rw-r--r-- 1 root root  427 Feb  1  2021 ipp.xml
-rw-r--r-- 1 root root  894 Feb  1  2021 ipsec.xml
-rw-r--r-- 1 root root  255 Feb  1  2021 ircs.xml
-rw-r--r-- 1 root root  247 Feb  1  2021 irc.xml
-rw-r--r-- 1 root root  264 Feb  1  2021 iscsi-target.xml
-rw-r--r-- 1 root root  358 Feb  1  2021 isns.xml
-rw-r--r-- 1 root root  213 Feb  1  2021 jenkins.xml
-rw-r--r-- 1 root root  182 Feb  1  2021 kadmin.xml
-rw-r--r-- 1 root root  272 Feb  1  2021 kdeconnect.xml
-rw-r--r-- 1 root root  233 Feb  1  2021 kerberos.xml
-rw-r--r-- 1 root root  384 Feb  1  2021 kibana.xml
-rw-r--r-- 1 root root  249 Feb  1  2021 klogin.xml
-rw-r--r-- 1 root root  221 Feb  1  2021 kpasswd.xml
-rw-r--r-- 1 root root  182 Feb  1  2021 kprop.xml
-rw-r--r-- 1 root root  242 Feb  1  2021 kshell.xml
-rw-r--r-- 1 root root  308 Feb  1  2021 kube-apiserver.xml
-rw-r--r-- 1 root root  232 Feb  1  2021 ldaps.xml
-rw-r--r-- 1 root root  199 Feb  1  2021 ldap.xml
-rw-r--r-- 1 root root  385 Feb  1  2021 libvirt-tls.xml
-rw-r--r-- 1 root root  389 Feb  1  2021 libvirt.xml
-rw-r--r-- 1 root root  269 Feb  1  2021 lightning-network.xml
-rw-r--r-- 1 root root  324 Feb  1  2021 llmnr.xml
-rw-r--r-- 1 root root  349 Feb  1  2021 managesieve.xml
-rw-r--r-- 1 root root  432 Feb  1  2021 matrix.xml
-rw-r--r-- 1 root root  424 Feb  1  2021 mdns.xml
-rw-r--r-- 1 root root  245 Feb  1  2021 memcache.xml
-rw-r--r-- 1 root root  343 Feb  1  2021 minidlna.xml
```

```
-rw-r--r-- 1 root root  237 Feb  1  2021 mongodb.xml
-rw-r--r-- 1 root root  473 Feb  1  2021 mosh.xml
-rw-r--r-- 1 root root  211 Feb  1  2021 mountd.xml
-rw-r--r-- 1 root root  296 Feb  1  2021 mqtt-tls.xml
-rw-r--r-- 1 root root  287 Feb  1  2021 mqtt.xml
-rw-r--r-- 1 root root  170 Feb  1  2021 mssql.xml
-rw-r--r-- 1 root root  190 Feb  1  2021 ms-wbt.xml
-rw-r--r-- 1 root root  242 Feb  1  2021 murmur.xml
-rw-r--r-- 1 root root  171 Feb  1  2021 mysql.xml
-rw-r--r-- 1 root root  250 Feb  1  2021 nbd.xml
-rw-r--r-- 1 root root  342 Feb  1  2021 nfs3.xml
-rw-r--r-- 1 root root  324 Feb  1  2021 nfs.xml
-rw-r--r-- 1 root root  293 Feb  1  2021 nmea-0183.xml
-rw-r--r-- 1 root root  247 Feb  1  2021 nrpe.xml
-rw-r--r-- 1 root root  389 Feb  1  2021 ntp.xml
-rw-r--r-- 1 root root  368 Feb  1  2021 nut.xml
-rw-r--r-- 1 root root  335 Feb  1  2021 openvpn.xml
-rw-r--r-- 1 root root  260 Feb  1  2021 ovirt-imageio.xml
-rw-r--r-- 1 root root  343 Feb  1  2021 ovirt-storageconsole.xml
-rw-r--r-- 1 root root  235 Feb  1  2021 ovirt-vmconsole.xml
-rw-r--r-- 1 root root 1024 Feb  1  2021 plex.xml
-rw-r--r-- 1 root root  433 Feb  1  2021 pmcd.xml
-rw-r--r-- 1 root root  474 Feb  1  2021 pmproxy.xml
-rw-r--r-- 1 root root  544 Feb  1  2021 pmwebapis.xml
-rw-r--r-- 1 root root  460 Feb  1  2021 pmwebapi.xml
-rw-r--r-- 1 root root  357 Feb  1  2021 pop3s.xml
-rw-r--r-- 1 root root  348 Feb  1  2021 pop3.xml
-rw-r--r-- 1 root root  181 Feb  1  2021 postgresql.xml
-rw-r--r-- 1 root root  509 Feb  1  2021 privoxy.xml
-rw-r--r-- 1 root root  213 Feb  1  2021 prometheus.xml
-rw-r--r-- 1 root root  261 Feb  1  2021 proxy-dhcp.xml
-rw-r--r-- 1 root root  424 Feb  1  2021 ptp.xml
-rw-r--r-- 1 root root  414 Feb  1  2021 pulseaudio.xml
-rw-r--r-- 1 root root  297 Feb  1  2021 puppetmaster.xml
```

```
-rw-r--r-- 1 root root  273 Feb  1  2021 quassel.xml
-rw-r--r-- 1 root root  520 Feb  1  2021 radius.xml
-rw-r--r-- 1 root root  183 Feb  1  2021 rdp.xml
-rw-r--r-- 1 root root  212 Feb  1  2021 redis-sentinel.xml
-rw-r--r-- 1 root root  268 Feb  1  2021 redis.xml
-rw-r--r-- 1 root root  381 Feb  1  2021 RH-Satellite-6-capsule.xml
-rw-r--r-- 1 root root  556 Feb  1  2021 RH-Satellite-6.xml
-rw-r--r-- 1 root root  214 Feb  1  2021 rpc-bind.xml
-rw-r--r-- 1 root root  213 Feb  1  2021 rquotad.xml
-rw-r--r-- 1 root root  310 Feb  1  2021 rsh.xml
-rw-r--r-- 1 root root  311 Feb  1  2021 rsyncd.xml
-rw-r--r-- 1 root root  350 Feb  1  2021 rtsp.xml
-rw-r--r-- 1 root root  329 Feb  1  2021 salt-master.xml
-rw-r--r-- 1 root root  371 Feb  1  2021 samba-client.xml
-rw-r--r-- 1 root root 1298 Feb  1  2021 samba-dc.xml
-rw-r--r-- 1 root root  448 Feb  1  2021 samba.xml
-rw-r--r-- 1 root root  324 Feb  1  2021 sane.xml
-rw-r--r-- 1 root root  283 Feb  1  2021 sips.xml
-rw-r--r-- 1 root root  496 Feb  1  2021 sip.xml
-rw-r--r-- 1 root root  299 Feb  1  2021 slp.xml
-rw-r--r-- 1 root root  231 Feb  1  2021 smtp-submission.xml
-rw-r--r-- 1 root root  577 Feb  1  2021 smtps.xml
-rw-r--r-- 1 root root  550 Feb  1  2021 smtp.xml
-rw-r--r-- 1 root root  308 Feb  1  2021 snmptrap.xml
-rw-r--r-- 1 root root  342 Feb  1  2021 snmp.xml
-rw-r--r-- 1 root root  405 Feb  1  2021 spideroak-lansync.xml
-rw-r--r-- 1 root root  275 Feb  1  2021 spotify-sync.xml
-rw-r--r-- 1 root root  173 Feb  1  2021 squid.xml
-rw-r--r-- 1 root root  421 Feb  1  2021 ssdp.xml
-rw-r--r-- 1 root root  463 Feb  1  2021 ssh.xml
-rw-r--r-- 1 root root  631 Feb  1  2021 steam-streaming.xml
-rw-r--r-- 1 root root  287 Feb  1  2021 svdrp.xml
-rw-r--r-- 1 root root  231 Feb  1  2021 svn.xml
-rw-r--r-- 1 root root  297 Feb  1  2021 syncthing-gui.xml
```

```
-rw-r--r-- 1 root root  311 Feb  1  2021 syncthing.xml
-rw-r--r-- 1 root root  496 Feb  1  2021 synergy.xml
-rw-r--r-- 1 root root  444 Feb  1  2021 syslog-tls.xml
-rw-r--r-- 1 root root  329 Feb  1  2021 syslog.xml
-rw-r--r-- 1 root root  393 Feb  1  2021 telnet.xml
-rw-r--r-- 1 root root  252 Feb  1  2021 tentacle.xml
-rw-r--r-- 1 root root  288 Feb  1  2021 tftp-client.xml
-rw-r--r-- 1 root root  424 Feb  1  2021 tftp.xml
-rw-r--r-- 1 root root  221 Feb  1  2021 tile38.xml
-rw-r--r-- 1 root root  336 Feb  1  2021 tinc.xml
-rw-r--r-- 1 root root  771 Feb  1  2021 tor-socks.xml
-rw-r--r-- 1 root root  244 Feb  1  2021 transmission-client.xml
-rw-r--r-- 1 root root  264 Feb  1  2021 upnp-client.xml
-rw-r--r-- 1 root root  593 Feb  1  2021 vdsm.xml
-rw-r--r-- 1 root root  475 Feb  1  2021 vnc-server.xml
-rw-r--r-- 1 root root  310 Feb  1  2021 wbem-https.xml
-rw-r--r-- 1 root root  352 Feb  1  2021 wbem-http.xml
-rw-r--r-- 1 root root  323 Feb  1  2021 wsmans.xml
-rw-r--r-- 1 root root  316 Feb  1  2021 wsman.xml
-rw-r--r-- 1 root root  329 Feb  1  2021 xdmcp.xml
-rw-r--r-- 1 root root  509 Feb  1  2021 xmpp-bosh.xml
-rw-r--r-- 1 root root  488 Feb  1  2021 xmpp-client.xml
-rw-r--r-- 1 root root  264 Feb  1  2021 xmpp-local.xml
-rw-r--r-- 1 root root  545 Feb  1  2021 xmpp-server.xml
-rw-r--r-- 1 root root  314 Feb  1  2021 zabbix-agent.xml
-rw-r--r-- 1 root root  315 Feb  1  2021 zabbix-server.xml

/usr/lib/firewalld/zones:
total 40
-rw-r--r-- 1 root root 299 Feb  1  2021 block.xml
-rw-r--r-- 1 root root 293 Feb  1  2021 dmz.xml
-rw-r--r-- 1 root root 291 Feb  1  2021 drop.xml
-rw-r--r-- 1 root root 304 Feb  1  2021 external.xml
-rw-r--r-- 1 root root 369 Feb  1  2021 home.xml
```

```
-rw-r--r-- 1 root root 384 Feb  1  2021 internal.xml
-rw-r--r-- 1 root root 729 Apr 12  2021 nm-shared.xml
-rw-r--r-- 1 root root 315 Feb  1  2021 public.xml
-rw-r--r-- 1 root root 162 Feb  1  2021 trusted.xml
-rw-r--r-- 1 root root 311 Feb  1  2021 work.xml
```

Ces fichiers sont au format **xml**, par exemple :

```
root@debian11:~# cat /usr/lib/firewalld/zones/home.xml<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Home</short>
  <description>For use in home areas. You mostly trust the other computers on networks to not harm your computer.
Only selected incoming connections are accepted.</description>
  <service name="ssh"/>
  <service name="mdns"/>
  <service name="samba-client"/>
  <service name="dhcpv6-client"/>
</zone>
```

La configuration de firewalld ainsi que les définitions et règles personnalisées se trouvent dans **/etc/firewalld** :

```
root@debian11:~# ls -lR /etc/firewalld/
/etc/firewalld/:
total 32
-rw-r--r-- 1 root root 2745 Feb  1  2021 firewalld.conf
drwxr-xr-x 2 root root 4096 Feb  1  2021 helpers
drwxr-xr-x 2 root root 4096 Feb  1  2021 icmptypes
drwxr-xr-x 2 root root 4096 Feb  1  2021 ipsets
-rw-r--r-- 1 root root  268 Feb  1  2021 lockdown-whitelist.xml
drwxr-xr-x 2 root root 4096 Feb  1  2021 policies
drwxr-xr-x 2 root root 4096 Feb  1  2021 services
drwxr-xr-x 2 root root 4096 Feb  1  2021 zones

/etc/firewalld/helpers:
```

```
total 0

/etc/firewalld/icmptypes:
total 0

/etc/firewalld/ipsets:
total 0

/etc/firewalld/policies:
total 0

/etc/firewalld/services:
total 0

/etc/firewalld/zones:
total 0
```

Le fichier de configuration de firewalld est **/etc/firewalld/firewalld.conf** :

```
root@debian11:~# cat /etc/firewalld/firewalld.conf
# firewalld config file

# default zone
# The default zone used if an empty zone string is used.
# Default: public
DefaultZone=public

# Clean up on exit
# If set to no or false the firewall configuration will not get cleaned up
# on exit or stop of firewalld
# Default: yes
CleanupOnExit=yes

# Lockdown
```

```
# If set to enabled, firewall changes with the D-Bus interface will be limited
# to applications that are listed in the lockdown whitelist.
# The lockdown whitelist file is lockdown-whitelist.xml
# Default: no
Lockdown=no

# IPv6_rpfilter
# Performs a reverse path filter test on a packet for IPv6. If a reply to the
# packet would be sent via the same interface that the packet arrived on, the
# packet will match and be accepted, otherwise dropped.
# The rp_filter for IPv4 is controlled using sysctl.
# Default: yes
IPv6_rpfilter=yes

# IndividualCalls
# Do not use combined -restore calls, but individual calls. This increases the
# time that is needed to apply changes and to start the daemon, but is good for
# debugging.
# Default: no
IndividualCalls=no

# LogDenied
# Add logging rules right before reject and drop rules in the INPUT, FORWARD
# and OUTPUT chains for the default rules and also final reject and drop rules
# in zones. Possible values are: all, unicast, broadcast, multicast and off.
# Default: off
LogDenied=off

# FirewallBackend
# Selects the firewall backend implementation.
# Choices are:
#       - nftables (default)
#       - iptables (iptables, ip6tables, ebtables and ipset)
FirewallBackend=nftables
```

```
# FlushAllOnReload
# Flush all runtime rules on a reload. In previous releases some runtime
# configuration was retained during a reload, namely; interface to zone
# assignment, and direct rules. This was confusing to users. To get the old
# behavior set this to "no".
# Default: yes
FlushAllOnReload=yes

# RFC3964_IPv4
# As per RFC 3964, filter IPv6 traffic with 6to4 destination addresses that
# correspond to IPv4 addresses that should not be routed over the public
# internet.
# Defaults to "yes".
RFC3964_IPv4=yes

# AllowZoneDrifting
# Older versions of firewalld had undocumented behavior known as "zone
# drifting". This allowed packets to ingress multiple zones - this is a
# violation of zone based firewalls. However, some users rely on this behavior
# to have a "catch-all" zone, e.g. the default zone. You can enable this if you
# desire such behavior. It's disabled by default for security reasons.
# Note: If "yes" packets will only drift from source based zones to interface
# based zones (including the default zone). Packets never drift from interface
# based zones to other interfaces based zones (including the default zone).
# Possible values; "yes", "no". Defaults to "no".
AllowZoneDrifting=no
```

La Commande firewall-cmd

firewalld s'appuie sur netfilter. Pour cette raison, l'utilisation de firewall-cmd est incompatible avec l'utilisation des commandes iptables et system-config-firewall.

Important - firewall-cmd est le front-end de firewalld en ligne de commande. Il existe aussi la commande **firewall-config** qui lance un outil de configuration graphique.

Pour obtenir la liste de toutes les zones prédéfinies, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --get-zones
block dmz drop external home internal nm-shared public trusted work
```

Pour obtenir la liste de toutes les services prédéfinis, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule amanda-client amanda-k5-client amqp amqps apcupsd audit bacula bacula-
client bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-mon cfengine
cockpit collectd condor-collector ctdb dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-registry docker-
swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap
freeipa-ldaps freeipa-replication freeipa-trust ftp ganglia-client ganglia-master git grafana gre high-
availability http https imap imaps ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnect
kerberos kibana klogin kpasswd kprop kshell kube-apiserver ldap ldaps libvirt libvirt-tls lightning-network llmnr
managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nfs
nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi
pmwebapis pop3 pop3s postgresql privoxy prometheus proxy-dhcp ptp pulseaudio puppetmaster quassel radius rdp
redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane sip sips slp
smtp smtp-submission smtps snmp snmptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn
syncthing syncthing-gui synergy syslog syslog-tls telnet tentacle tftp tftp-client tile38 tinc tor-socks
transmission-client upnp-client vdsm vnc-server wbem-http wbem-https wsman wsmans xdmcp xmpp-bosh xmpp-client
xmpp-local xmpp-server zabbix-agent zabbix-server
```

Pour obtenir la liste de toutes les types ICMP prédéfinis, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --get-icmptypes
address-unreachable bad-header beyond-scope communication-prohibited destination-unreachable echo-reply echo-
```

```
request failed-policy fragmentation-needed host-precedence-violation host-prohibited host-redirect host-unknown
host-unreachable ip-header-bad neighbour-advertisement neighbour-solicitation network-prohibited network-redirect
network-unknown network-unreachable no-route packet-too-big parameter-problem port-unreachable precedence-cutoff
protocol-unreachable redirect reject-route required-option-missing router-advertisement router-solicitation
source-quench source-route-failed time-exceeded timestamp-reply timestamp-request tos-host-redirect tos-host-
unreachable tos-network-redirect tos-network-unreachable ttl-zero-during-reassembly ttl-zero-during-transit
unknown-header-type unknown-option
```

Pour obtenir la liste des zones de la configuration courante, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --get-active-zones
public
  interfaces: ens18
```

Pour obtenir la liste des zones de la configuration courante pour une interface spécifique, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --get-zone-of-interface=ens18
public
```

Pour obtenir la liste des services autorisés pour la zone public, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=public --list-services
dhcpv6-client ssh
```

Pour obtenir toute la configuration pour la zone public, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=public --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens18
  sources:
  services: dhcpv6-client ssh
  ports:
```

```
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Pour obtenir la liste complète de toutes les zones et leurs configurations, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --list-all-zones
block
  target: %%REJECT%%
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

dmz
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
```

```
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

drop
  target: DROP
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

external
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  forward: no
  masquerade: yes
  forward-ports:
  source-ports:
```

```
  icmp-blocks:
  rich rules:

home
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client mdns samba-client ssh
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

internal
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client mdns samba-client ssh
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

nm-shared
```

```
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcp dns ssh
  ports:
  protocols: icmp ipv6-icmp
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule priority="32767" reject

public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens18
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

trusted
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
```

```
  sources:
  services:
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

work
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Pour changer la zone par défaut de public à work, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --set-default-zone=work
success
root@debian11:~# firewall-cmd --get-active-zones
work
  interfaces: ens18
```

HERE

Pour ajouter l'interface ip_fixe à la zone work, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=work --add-interface=ip_fixe
success
root@debian11:~# firewall-cmd --get-active-zones
work
  interfaces: ens18 ip_fixe
```

Pour supprimer l'interface ip_fixe à la zone work, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=work --remove-interface=ip_fixe
success
root@debian11:~# firewall-cmd --get-active-zones
work
  interfaces: ens18
```

Pour ajouter le service **http** à la zone **work**, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=work --add-service=http
success
root@debian11:~# firewall-cmd --zone=work --list-services
dhcpv6-client http ssh
```

Pour supprimer le service **http** de la zone **work**, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=work --remove-service=http
success
root@debian11:~# firewall-cmd --zone=work --list-services
dhcpv6-client ssh
```

Pour ajouter un nouveau bloc ICMP, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=work --add-icmp-block=echo-reply
success
root@debian11:~# firewall-cmd --zone=work --list-icmp-blocks
echo-reply
```

Pour supprimer un bloc ICMP, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=work --remove-icmp-block=echo-reply
success
root@debian11:~# firewall-cmd --zone=work --list-icmp-blocks

root@debian11:~#
```

Pour ajouter le port 591/tcp à la zone work, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=work --add-port=591/tcp
success
root@debian11:~# firewall-cmd --zone=work --list-ports
591/tcp
```

Pour supprimer le port 591/tcp à la zone work, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=work --remove-port=591/tcp
success
root@debian11:~# firewall-cmd --zone=work --list-ports

root@debian11:~#
```

Pour créer un nouveau service, il convient de :

- copier un fichier existant se trouvant dans le répertoire **/usr/lib/firewalld/services** vers **/etc/firewalld/services**,
- modifier le fichier,
- recharger la configuration de firewalld,
- vérifier que firewalld voit le nouveau service.

Par exemple :

```
root@debian11:~# cp /usr/lib/firewalld/services/http.xml /etc/firewalld/services/filemaker.xml

root@debian11:~# cat /etc/firewalld/services/filemaker.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>WWW (HTTP)</short>
  <description>HTTP is the protocol used to serve Web pages. If you plan to make your Web server publicly
available, enable this option. This option is not required for viewing pages locally or developing Web
pages.</description>
  <port protocol="tcp" port="80"/>
</service>

root@debian11:~# vi /etc/firewalld/services/filemaker.xml

root@debian11:~# cat /etc/firewalld/services/filemaker.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>FileMakerPro</short>
  <description>fichier de service firewalld pour FileMaker Pro</description>
  <port protocol="tcp" port="591"/>
</service>

root@debian11:~# firewall-cmd --reload
success

root@debian11:~# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule amanda-client amanda-k5-client amqp amqps apcupsd audit bacula bacula-
client bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-mon cfengine
cockpit collectd condor-collector ctdb dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-registry docker-
swarm dropbox-lansync elasticsearch etcd-client etcd-server filemaker finger foreman foreman-proxy freeipa-4
freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp ganglia-client ganglia-master git grafana gre
high-availability http https imap imaps ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnect
```

```
kerberos kibana klogin kpasswd kprop kshell kube-apiserver ldap ldaps libvirt libvirt-tls lightning-network llmnr
managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nfs
nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi
pmwebapis pop3 pop3s postgresql privoxy prometheus proxy-dhcp ptp pulseaudio puppetmaster quassel radius rdp
redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane sip sips slp
smtp smtp-submission smtps snmp snmptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn
syncthing syncthing-gui synergy syslog syslog-tls telnet tentacle tftp tftp-client tile38 tinc tor-socks
transmission-client upnp-client vdsm vnc-server wbem-http wbem-https wsman wsmans xdmcp xmpp-bosh xmpp-client
xmpp-local xmpp-server zabbix-agent zabbix-server
```

La Configuration Avancée de firewalld

La configuration de base de firewalld ne permet que la configuration des zones, services, blocs ICMP et les ports non-standard. Cependant firewalld peut également être configuré avec des **Rich Rules** ou ***Règles Riches***. Rich Rules ou Règles Riches évaluent des **critères** pour ensuite entreprendre une **action**.

Les **Critères** sont :

- **source address=“<adresse_IP>“**
- **destination address=”<adresse_IP>“**,
- **rule port port=”<numéro_du_port>“**,
- **service name=<nom_d'un_sevice_prédéfini>**.

Les **Actions** sont :

- **accept**,
- **reject**,
  - une Action reject peut être associée avec un message d'erreur spécifique par la clause **type=”<type_d'erreur>**,
- **drop**.

Saisissez la commande suivante pour ouvrir le port 80 :

```
root@debian11:~# firewall-cmd --add-rich-rule='rule port port="80" protocol="tcp" accept'
```

```
success
```

Important - Notez que la Rich Rule doit être entourée de caractères '.

Important - Notez que la Rich Rule a créé deux règles, une pour IPv4 et une deuxième pour IPv6. Une règle peut être créée pour IPv4 seul en incluant le Critère **family=ipv4**. De la même façon, une règle peut être créée pour IPv6 seul en incluant le Critère **family=ipv6**.

Cette nouvelle règle est écrite en mémoire mais non pas sur disque. Pour l'écrire sur disque dans le fichier zone se trouvant dans **/etc/firewalld**, il faut ajouter l'option **–permanent** :

```
root@debian11:~# firewall-cmd --add-rich-rule='rule port port="80" protocol="tcp" accept' --permanent
success

root@debian11:~# cat /etc/firewalld/zones/work.xml
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Work</short>
  <description>For use in work areas. You mostly trust the other computers on networks to not harm your computer.
Only selected incoming connections are accepted.</description>
  <service name="ssh"/>
  <service name="dhcpv6-client"/>
  <rule>
    <port port="80" protocol="tcp"/>
    <accept/>
  </rule>
</zone>
```

Important - Attention ! La règle ajoutée avec l'option –permanent n'est pas prise en compte imédiatement mais uniquement au prochain redémmarge. Pour qu'une règle soit appliquée immédiatement **et** être écrite sur disque, il faut saisir la commande deux fois dont une avec l'option –permanent et l'autre sans l'option –permanent.

Pour visualiser cette règle dans la configuration de firewalld, il convient de saisir la commande suivante :

```
root@debian11:~# firewall-cmd --reload
success

root@debian11:~# firewall-cmd --zone=work --list-all
work (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens18
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule port port="80" protocol="tcp" accept
```

Notez que la Rich Rule est créée dans la Zone par Défaut. Il est possible de créer une Rich Rule dans une autre zone en utilisant l'option **–zone=<zone>** de la commande firewall-cmd :

```
root@debian11:~# firewall-cmd --zone=public --add-rich-rule='rule port port="80" protocol="tcp" accept'
success

root@debian11:~# firewall-cmd --reload
success

root@debian11:~# firewall-cmd --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

root@debian11:~# firewall-cmd --zone=public --add-rich-rule='rule port port="80" protocol="tcp" accept' --
permanent
success

root@debian11:~# firewall-cmd --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
```

```
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

root@debian11:~# firewall-cmd --reload
success

root@debian11:~# firewall-cmd --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule port port="80" protocol="tcp" accept
```

Pour supprimer une Rich Rule, il faut copier la ligne entière la concernant qui se trouve dans la sortie de la commande **firewall-cmd –list-all-zones** :

```
root@debian11:~# firewall-cmd --zone=public --remove-rich-rule='rule port port="80" protocol="tcp" accept'
success

root@debian11:~# firewall-cmd --zone=public --list-all
```

```
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Le mode Panic de firewalld

Le mode Panic de firewalld permet de bloquer tout le trafic avec une seule commande. Pour connaître l'état du mode Panic, utilisez la commande suivante :

```
root@debian8:~# firewall-cmd --query-panic
no
```

Pour activer le mode Panic, il convient de saisir la commande suivante :

```
# firewall-cmd --panic-on
```

Pour désactiver le mode Panic, il convient de saisir la commande suivante :

```
# firewall-cmd --panic-off
```