

Aksel PAULET

RT1 : B1

## SAE 12

---

*S'initier aux réseaux informatiques*

---

### Sommaire :

---

- [Documentation Linux ↔ Windows](#)
- [Installation de la machine virtuel](#)
- [Documentation VIM](#)

### Documentation Linux ↔ Windows

---

- Utilisation de la documentation :

Vous retrouverez en premier lieu la documentation Linux puis celle de Windows. Si vous connaissez une commande Windows vous aurez un lien clickable pour vous amener a la commande Linux et vice versa. Si la documentation suivante ne suffit pas, les commandes "helps" sont: `man { commande }` ( pour Linux ) et `{ commande } ?` ( pour Windows ).

- [Allez directement à la documentation Linux](#)
- [Allez directement à la documentation Windows](#)

### Commande linux :

---

## Attention !

- Si une commande vous résiste vous pouvez toujours écrire devant votre commande " sudo " ce dernier vous permet de temporairement tous faire sur l'ordinateur si votre utilisateur a les droits administrateurs . Attention a ne pas utiliser l'utilisateur "Root", se dernier peut mettre des droits trop importants sur certain fichier si mal utilisé ainsi que lancer une application avec ce dernier peut représenter une faille de sécurité très importante.

Certaines commandes nécessitent d'être installées. si jamais la commande " existe pas " faire : **apt install { nom de la commande } /!\**

- ip { argument } :

---

- ip link { argument } [commande Windows](#)

- ip link show { nom du device } : [commande Windows](#)

Cette commande va pouvoir permettre d'afficher la carte réseau sélectionnée avec son état ainsi que diverses informations.

- **Note :**

*Si toutefois l'on ne met pas le nom du service, on aura l'affichage de toutes les cartes réseaux avec leurs informations.*



- ip link set { argument } { nom du device } : [commande Windows](#)

- up :

Allume la carte réseau



- down :

```
Eteint la carte réseau
```



- `ip address { argument } :`

- **Note :**

*Si toute fois l'on ne met pas le nom du servise l'on aura l'affichage de toute les cartes réseaux avec leur informations.*

- `add { ip/masque } dev { device } :` [commande Windows](#)

```
Ajoute l'adresse donnée.
```



- `flush dev { device } :`

```
Enlève toutes les adresses réseaux de la carte réseau.
```



- `del { ip } dev { device } :`

```
supprime une adresse ip spécifique.
```



- `show { device } :` [commande Windows](#)

```
Permet de voir toute les informations d'une adresse.
```



- `ip neigh { argument }` : [commande Windows](#)

- *utilisation* :

On utilise `ip neigh` pour manager / connaître la table ARP du poste sur lequel vous êtes.

Plusieur stade de liaison possible, les principales :

Stale : Connexion valide mais suspicieuse.

Permanent : Connexion statique, se supprime uniquement manuellement.

Reachable : Connexion valide mais périmable.

Delay : pas de connexion depuis les 30 dernières seconde.

Probe : Une tentative de connexion a eu lieu, pas de réponse.

Failed : Connexion impossible.

- `get { ip } dev { device }` : [commande Windows](#)

Permet de connaître le status de connection de l'ip. Permet également de savoir si elle se trouve dans la table.



- `add { ip } { adresse MAC }` :

Ajoute manuellement une adresse MAC liée à une adresse ip.

- `flush { ip } dev { device }` : [commande Windows](#)

Efface les données liée a l'adresse ip de la table ARP.

Si aucune ip est entrée, supprime tout ce qui est relié au device.



- `ip route { argument }` : [commande Windows](#)

- *Note:*

Lors de l'affectation d'une adresse ip avec son masque, une route vers le réseau est automatiquement créée pour le { device } d'affectation.

On peut toujours en rajouter une avec la commande suivante.

- *utilisation :*

On utilise cette commande pour pouvoir avoir une route vers notre réseau local ainsi que savoir qui est notre passerelle et ainsi avoir accès à Internet.

- `show` : [commande Windows](#)

Cette argument permet de pouvoir voir toutes les routes définies ainsi que celles qui sont par défaut.

Chaque route est reliée à une carte réseau.



- `add { ip }` : [commande Windows](#)

Permet de rajouter une route vers un réseau.



- `add default via { ip } dev { device }` : [commande Windows](#)

Permet d'ajouter une route de sortie par défaut vers le routeur.



- flush { ip } dev { device } [commande Windows](#)

Cela permet de supprimer cette adresse ip des routes connues, si aucune ip est donnée : supprime tout ce qui est liée à la carte réseau.



## • Outils de teste de connection :

---

- mii-tool { nom du device } :

Si câblé en E-thernet cette commande permet de chercher si le câble est bien branché de bout en bout.



- ping { ip } OU { nom de domaine } : [commande Windows](#)

Permet de tester la connection a une machine.

Dans le cas d'une connexion filaire, on peut déduire;  
Si un certain nombre de paquet est perdu que le câble Ethernet est défectueux.

Dans le cas où on arrive à faire un ping vers une adresse ip mais pas vers un nom de domaine:  
Alors, on a un problème de DNS.

Ping utilise le protocole echo-request ICMP



- traceroute { ip } OU { nom de domaine } : [commande Windows](#)

Permet de lister le nombre de saut à effectuer pour rejoindre une machine.  
Renvoie également les noms / ip des routeurs par lequel le paquet passe.  
Traceroute est utilisé principalement via le protocole ICMP pour éviter des échecs liés à des firewalls

```
root@Aksel:~# traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.665 ms  0.610 ms  0.594 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * ^C
root@Aksel:~#
```

Dans cette capture d'écran, on remarque des étoiles sur les lignes précédentes la gateway, c'est dernier sont des erreurs. Probablement dû à un problème de translation d'adresse ( NAT ).

- lshw :

permet de lister les périphériques réseaux connectés à l'ordinateur

## • Fichier de configuration utile à connaître :

---

- DNS : [commande Windows](#)

Le serveur DNS permet de faire le lien entre les adresses ip et les noms symboliques.

Ces derniers étant plus facile à retenir que les adresses ip.

Exemple : 8.8.8.8 = www.google.com

On peut modifier les informations relatives aux serveurs DNS dans le

```
fichier :  
/etc/network/interfaces
```

- DHCLIENT : [commande Windows](#)

Le DHCLIENT permet de faire un adressage dynamique, attribuer automatiquement à la connexion l'adresse ip / la table de routage d'un ordinateur client.  
On peut également faire un paramétrage ciblé par rapport aux adresses mac.

Pour faire intervenir le DHCLIENT il suffit de taper : `dhclient` avec des droits super-utilisateurs.

pour accéder au fichier de configuration :  
`/etc/netplan/`

cette dernière peut avoir des noms variables mais il sera retrouvable avec une tabulation dans ce répertoire.



- droit super-utilisateur :

les droits super-utilisateur alias sudoers, Permettent d'exécuter des commandes qu'un simple utilisateur ne pourrait pas exécuter.  
Notamment pour la modification des adresses.

```
/etc/sudoers
```



## Commande Windows :

---

/!\ Lancez un cmd avec des droits administrateur ( clique droit ) Si jamais cette documentation ne suffit pas, on peut taper dans le cmd: { commande } ?



Afin d'éviter de casser mon ordinateur personnel il y aura uniquement des captures d'écran pour les commandes ne changeant pas les configurations. ( les VM Windows n'étant pas opérationnel sur le store.iutbéziers )

Si la commande recherchée n'est pas ici, chercher du côté de netsh ( regroupant les commandes réseaux sous windows ).

/!\

- netsh { argument } : [commande Linux](#)

---

- interface { ipv4 / ipv6 } show interfaces : [commande Linux](#)

Permet de voir toute les interfaces réseaux / leur état / leur priorité et leur transport d'octet maximal ( MTU ).



- interface set interface { nom de l'interface } {option} : [commande Linux](#)
  - disable

Eteint l'interface donnée

Idx	Mét	MTU	État	Nom
1	75	4294967295	connected	Loopback Pseudo-Interface 1
6	55	1480	connected	Wi-Fi
15	5	1500	disconnected	Ethernet
4	25	1500	disconnected	Connexion au réseau local* 1
14	65	1500	disconnected	Connexion réseau Bluetooth
16	25	1500	disconnected	Connexion au réseau local* 2
18	25	1500	disconnected	Ethernet 2
8	25	1500	connected	VirtualBox Host-Only Network

```
C:\windows\system32>netsh interface set interface "Wi-Fi" disable
```

```
C:\windows\system32>netsh interface ipv4 show interfaces
```

Idx	Mét	MTU	État	Nom
1	75	4294967295	connected	Loopback Pseudo-Interface 1
15	5	1500	disconnected	Ethernet
14	65	1500	disconnected	Connexion réseau Bluetooth
18	5	1500	disconnected	Ethernet 2
8	25	1500	connected	VirtualBox Host-Only Network

## ■ enable

Allume l'interface donnée

Idx	Mét	MTU	État	Nom
1	75	4294967295	connected	Loopback Pseudo-Interface 1
15	5	1500	disconnected	Ethernet
14	65	1500	disconnected	Connexion réseau Bluetooth
18	5	1500	disconnected	Ethernet 2
8	25	1500	connected	VirtualBox Host-Only Network

```
C:\windows\system32>netsh interface set interface "Wi-Fi" enable
```

```
C:\windows\system32>netsh interface ipv4 show interfaces
```

Idx	Mét	MTU	État	Nom
1	75	4294967295	connected	Loopback Pseudo-Interface 1
6	55	1500	connected	Wi-Fi
15	5	1500	disconnected	Ethernet
4	25	1500	disconnected	Connexion au réseau local* 1
14	65	1500	disconnected	Connexion réseau Bluetooth
18	5	1500	disconnected	Ethernet 2
8	25	1500	connected	VirtualBox Host-Only Network

- interface { ipv4 / ipv6 } set address name="{ interface }" static { ip } { masque sur 4octets } {passerelle} : [commande Linux](#)

Ajoute une adresse ipv4 / ipv6 a l'interface donnée

- `ipconfig { argument }:`

---

- sans argument

Permet d'afficher les paramètres réseaux de votre ordinateur.



- `/all` : [commande Linux](#)

Permet d'afficher tous les paramètres réseaux de votre ordinateur.



- `/renew`

prend la dernière configuration dhclient connu



- `/flushdns`

Supprime le cache DNS

- `/release`

renouvèle les ip de la carte réseau

- arp { argument } : [commande Linux](#)

- -s { ip } { mac [ délimiter = - ] }

Ajoute une entrée statique dans la table d'arp.

- -a { argument } OU -g { argument } : [commande Linux](#)

- pas d'argument

Affiche le contenu de la table ARP

```
C:\Users\aksel>arp -a

Interface : 192.168.0.19 --- 0x6
  Adresse Internet      Adresse physique      Type
  192.168.0.25          08-a6-bc-b1-de-5f     dynamique
  192.168.0.26          14-0c-76-43-84-26     dynamique
  192.168.0.31          68-a3-78-27-c9-fa     dynamique
  192.168.0.37          f0-b4-d2-0c-2f-0e     dynamique
  192.168.0.254         14-0c-76-7d-17-bd     dynamique
  192.168.0.255         ff-ff-ff-ff-ff-ff     statique
  224.0.0.22            01-00-5e-00-00-16     statique
  224.0.0.251          01-00-5e-00-00-fb     statique
  224.0.0.252          01-00-5e-00-00-fc     statique
  224.0.2.60            01-00-5e-00-02-3c     statique
  230.0.0.1             01-00-5e-00-00-01     statique
  239.255.255.250       01-00-5e-7f-ff-fa     statique
  255.255.255.255       ff-ff-ff-ff-ff-ff     statique

Interface : 192.168.56.1 --- 0x8
  Adresse Internet      Adresse physique      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     statique
  224.0.0.22            01-00-5e-00-00-16     statique
  224.0.0.251          01-00-5e-00-00-fb     statique
  224.0.0.252          01-00-5e-00-00-fc     statique
  224.0.2.60            01-00-5e-00-02-3c     statique
  230.0.0.1             01-00-5e-00-00-01     statique
  239.255.255.250       01-00-5e-7f-ff-fa     statique
  255.255.255.255       ff-ff-ff-ff-ff-ff     statique
```

- { adresse interface } -N { adresse recherchée } : [commande Linux](#)

Affiche l'adresse ip, la mac ainsi que le type d'adressage de l'adresse demandée.

```
C:\windows\system32>arp -a 192.168.0.25 -N 192.168.0.19

Interface : 192.168.0.19 --- 0x6
    Adresse Internet    Adresse physique    Type
    192.168.0.25        08-a6-bc-b1-de-5f   dynamique

C:\windows\system32>
```

- -d { ip } : [commande Linux](#)

Supprime l'adresse ip ainsi que sa mac.  
Si l'on remplace l'adresse ip par " \* " alors toute la table arp est supprimée

- route { argument } : [commande Linux](#)

---

- Option possible :
  - -f | Efface entièrement les tables de routage
  - -p | A utilisé à l'ajout d'une route, afin de la rendre persistante au redémarrage système.
  - -4 | Force l'utilisation de la notation IPv4
  - -6 | Force l'utilisation de la notation IPv6
- print : [commande Linux](#)

Affiche toutes les routes réseaux de l'ordinateur.



- ADD { adresse réseau } MASK { masque sur 4octets } { via quelle ip } METRIC { valeur de priorité } : [commande Linux](#)

Ajoute une route réseau.

- DELETE { adresse réseau } MASK { masque sur 4octets } { via quelle ip } METRIC { valeur de priorité } : [commande Linux](#)

Supprime une route réseau

## • Outils de teste de connection :

---

- ping { ip } OU { nom de domaine } : [commande Linux](#)

Permet de tester la connection à une machine.

Dans le cas d'une connexion filaire l'on peut déduire;  
Si un certain nombre de paquet est perdu que le câble Ethernet est défectueux.

Dans le cas ou on arrive a faire un ping vers une adresse ip mais pas vers un nom de domaine:  
Alors on a un problème de DNS.

Ping utilise le protocole echo-request ICMP



- tracer { ip } OU { nom de domaine } : [commande Linux](#)

Si on utilise une adresse ipv4 alors on récupèrera les informations routeur en ipv4.

Si l'on utilise un nom de domaine, on aura des adresses ipv6.

Permet de lister le nombre de saut à effectuer pour rejoindre une machine. Renvoie également les noms / ip des routeurs par lequel le paquet passe. tracer est utilisé principalement via le protocole ICMP pour éviter des échecs lié a des firewalls

```

C:\windows\system32>tracert 8.8.8.8

Détermination de l'itinéraire vers dns.google [8.8.8.8]
avec un maximum de 30 sauts :

 1      1 ms      1 ms      1 ms  192.168.0.254
 2     35 ms     36 ms     27 ms  bzg34-1-78-222-154-254.fbx.proxad.net [78.222.154.254]
 3     36 ms     42 ms     47 ms  213.228.31.62
 4     42 ms     39 ms     35 ms  montpellier-49m-1-v804.intf.routers.proxad.net [212.27.50.205]
 5     41 ms     57 ms     52 ms  p11-crs16-1-be1126.intf.routers.proxad.net [194.149.163.213]
 6     59 ms     52 ms     49 ms  194.149.166.62
 7    109 ms     46 ms     85 ms  72.14.211.26
 8     47 ms     50 ms     48 ms  72.14.233.77
 9     51 ms     57 ms     51 ms  142.250.234.41
10     86 ms     50 ms     49 ms  dns.google [8.8.8.8]

Itinéraire déterminé.

C:\windows\system32>tracert www.google.com

Détermination de l'itinéraire vers www.google.com [2a00:1450:4007:80e::2004]
avec un maximum de 30 sauts :

 1      1 ms      1 ms      1 ms  2a01:e34:ede9:a8e0::1
 2      *        *        *      Délai d'attente de la demande dépassé.
 3     76 ms      *        53 ms  2a01:e00:29::5
 4     76 ms     46 ms     54 ms  2001:4860:1:1:0:3022:0:2
 5     56 ms     49 ms     47 ms  2a00:1450:80be::1
 6     50 ms     48 ms     61 ms  2001:4860:0:1::1e54
 7     56 ms     52 ms     56 ms  2001:4860:0:1018::6
 8     42 ms     55 ms     41 ms  2001:4860::c:4000:d9af
 9     52 ms      *        *      2001:4860::9:4000:cd8a
10    1396 ms     46 ms     55 ms  2001:4860:0:1015::1
11     138 ms    135 ms     47 ms  2001:4860:0:1::1af7
12     85 ms     47 ms     86 ms  par10s42-in-x04.1e100.net [2a00:1450:4007:80e::2004]

Itinéraire déterminé.

```

Sans savoir pourquoi à chaque lancement de la commande vers un nom de domaine, le deuxième saut a un délai trop long. Blocage du fournisseur d'accès internet (FAI) ?

## • service extérieur :

---

### ◦ DNS : [commande Linux](#)

- Pour mettre un DNS :
  - netsh interface ipv4 set DNS name = { interface } static { ip } index = { ordre de priorité (1 = premier DNS etc...) }

### ◦ DHCPCLIENT : [commande Linux](#)

- procédure de remise a jour dhclient :

```
netsh interface { ipv4 / ipv6 } set address name="{ interface }" source=dhcp
```

## Installation d'une machine virtuel.

---

Comme machine virtuelle, j'ai choisi sur le site [store.iutbéziers](http://store.iutbéziers) une Ubuntu car cette dernière est celle qui correspond à mes besoins actuels. Ces derniers sont une machine rapidement bootable, une interface en ligne de commande pour faire des tests divers.

- Pour se faire j'ai donc importé depuis Oracle Visual Box mon .ova :



Il n'y a aucune option à changer.

- Une fois la VM importée, l'on peut directement la démarrer:



- On teste la connectivité de la VM avec la commande ping :



Le tour est joué l'on peut utiliser notre machine virtuelle sans crainte !

## Les bases pour utiliser VIM :

---

Il existe plusieurs modes. Pour naviguer entre eux il suffit d'appuyer sur des touches "actions" ou bien de rentrer des commandes directement. Pour quitter un mode, appuyer sur "escape" :

- Normal :
  - Ce dernier n'a pas de particularité, il permet une lecture simple.
- Visual ( activable en appuyant sur "v" ):
  - Mode permettant de faire des sélections multiples en bloc ou bien par ligne.
- Insertion ( activable en appuyant sur "i" ):



- Mode permettant de modifier le texte

Il y a également des commandes / touches spéciales :

- escape puis ":q!" permet de quitter immédiatement l'interpréteur.
- escape puis ":wq!" permet de forcer l'enregistrement et quitter l'interpréteur.
- escape puis ":w" permet d'enregistrer le fichier
- double "y" permet de copier l'élément sélectionné, si aucun élément est sélectionné : copie la ligne.
- double "p" permet de copier l'élément sélectionnée.
- double "z" permet de sauvegardé et forcer la sortie de l'interpréteur.
- double "d" permet d'effacer l'élément sélectionné, si aucun élément est sélectionné : supprime la ligne.
- escape puis "![commande]" permet d'exécuter du bash depuis l'interpréteur.