aperçu de commandes d'administration réseau

- Création: lagrenouille
- Objet : du tuto réseau, commandes
- Niveau requis :
  débutant, avisé
- Commentaires : *Contexte d'utilisation du sujet du tuto.* Fix Me!
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.
- Suivi :
  à-tester-à-corriger, à-placer

petit rappel de ce qu'est le fichier hosts

Le fichier hosts dans le fichier /etc/hosts vous verrez la table qui sert de conversion des noms en adresse ip Le fichier hosts est « l'ancêtre » du service DNS. Il servait à indiquer, sur chaque machine, la correspondance entre le nom d'hôte et l'adresse IP.

Le fichier resolv.conf contient le nom de domaine et la configuration de la résolution de nom DNS, que vous allez interroger. Une fois configuré.

Si vous avez installé le paquet resolvconf, ce fichier est mis à jour utomatiquement depuis des informations qui se trouve dans le fichier /etc/network/interfaces. * Le paquet etc/resolvconf gère le contenu du fichier « /etc/resolv.conf servant à la résolution des noms en fonction du type de connexion utilisé et en récupérant les informations à différents endroits statiques ou dynamiques (clients ppp, dhcp ou autres).

petit rappel pour trouver son ip public

```
wget -q http://checkip.dyndns.org -O- | cut -d: -f2 | cut -d\< -f1
```

```
curl -4 ifconfig.me
```

```
wget -qO- http://ipecho.net/plain ; echo
```

```
host -v monserveur.org
```

ou en utilisant un dns d'orange

```
dig +short @80.10.246.132 monserveur.org
```

voir ceux de la FDN

```
dig +short @80.67.169.12 monserveur.org  (ou 80.67.169.40 )
```

Trouver son adresse ipv6 avec curl, bien que, “ip -a” la donnera aussi

```
curl https://api64.ipify.org
```

```
curl -s http://checkipv6.dyndns.org | cut -c77-111
```

```
curl 6.ifconfig.pro
```

celle ci donne ipv4 et ipv6

```
hostname -I
```

ifconfig et iproute2

iproute2 est destiné à remplacer toute une suite d'outils réseau appelés “net-tools” qui étaient anciennement utilisés pour les tâches de configuration d'interfaces réseau, tables de routage, et gestion de table ARP.

ifconfig n'est donc plus installé par défaut

```
apt install iproute2 iproute2-doc
```

```
quelques commandes iproute
```

ifconfig –> ip addr, ip link

```
  route --> ip route
  arp --> ip neigh
  vconfig --> ip link
  iptunnel --> ip tunnel
  ipmaddr --> ip maddr
  netstat --> ss
  ifconfig eth1 10.0.0.1/24 --> ip addr add 10.0.0.1/24 dev eth1
```

Description	net-tools	iproute2

Table ARP	arp -na	ip neighbor
Afficher les interfaces	ifconfig	ip link ou ip l
Afficher toutes les interfaces	ifconfig -a	ip addr show ou ip a
Monter une interface	ifconfig eth0 up	ip link set enp2s0 up
Afficher la table de routage	netstat -r	ip route ou ip r
Afficher la table de routage	route -n	ip route show
Ajouter une route	route add	ip route add
Supprimer une route	route del	ip route del
Afficher l'aide	ifconfig –help	ip help

```
ip neigh
192.168.1.10 dev enp2s0 lladdr 18:c0:4d:c5:ac:9d DELAY
192.168.1.17 dev enp2s0 lladdr 56:4a:53:80:cc:e0 STALE
192.168.1.1 dev enp2s0 lladdr 08:87:c6:b4:a1:50 REACHABLE
fe80::a87:c6ff:feb4:a150 dev enp2s0 lladdr 08:87:c6:b4:a1:50 router DELAY
2a01:cb19:83e9:5500:a87:c6ff:feb4:a150 dev enp2s0 lladdr 08:87:c6:b4:a1:50
router STALE
```

`voir plus bas la commande ss`

voir tuto DF: https://debian-facile.org/doc:reseau:ip-publique

arp

le protocole arp, de net-tools - `devient ip neighbor avec iproute2`

arp est un protocole permettant de déterminer une adresse MAC en fonction d'une adresse IP.

```
 arp -a
jeanne.home (192.168.1.24) at 80:3f:5d:10:84:61 [ether] on enp3s0
? (192.168.1.22) at (incomplète) on enp3s0
jeanne.home (192.168.1.10) at 80:3f:5d:10:84:61 [ether] on enp3s0
livebox.home (192.168.1.1) at 08:3e:5d:9c:8a:ee [ether] on enp3s0
funambule.org (192.168.1.15) at 74:d0:2b:13:6b:57 [ether] on enp3s0
<br />
#  arp -v
Adresse                  TypeMap AdresseMat          Indicateurs
Iface
jeanne.home              ether   80:3f:5d:10:84:61   C
enp3s0
192.168.1.22                     (incomplete)
enp3s0
jeanne.home              ether   80:3f:5d:10:84:61   C
enp3s0
livebox.home             ether   08:3e:5d:9c:8a:ee   C
enp3s0
funambule.org            ether   74:d0:2b:13:6b:57   C
```

```
enp3s0
Entrées: 5  Ignorées: 0    Trouvées: 5
```

L'Address Resolution Protocol (ARP, protocole de résolution d'adresse) est un protocole utilisé pour traduire une adresse de protocole de couche réseau (typiquement une adresse IPv4) en une adresse de protocole de couche de liaison (typiquement une adresse MAC).

Le protocole ARP a un rôle phare parmi les protocoles de la couche Internet de la suite TCP/IP, car il permet de connaître l'adresse physique d'une carte réseau correspondant à une adresse IP, c'est pour cela qu'il s'appelle Protocole de résolution d'adresse (en anglais ARP signifie Address Resolution Protocol). Chaque machine connectée au réseau possède un numéro d'identification de 48 bits. Ce numéro est un numéro unique qui est fixé dès la fabrication de la carte en usine. Toutefois la communication sur Internet ne se fait pas directement à partir de ce numéro (car il faudrait modifier l'adressage des ordinateurs à chaque fois que l'on change une carte réseau)

mais à partir d'une adresse dite logique attribuée par un organisme : l'adresse IP. Ainsi, pour faire correspondre les adresses physiques aux adresses logiques, le protocole ARP interroge les machines du réseau pour connaître leur adresse physique, puis crée une table de correspondance entre les adresses logiques et les adresses physiques dans une mémoire cache.

L'ARP ou "Address Resolution Protocol" est un protocole qui se situe sur la couche 3 du modèle OSI. On l'assimile parfois à un protocole de couche 2 et demi car il assure la liaison entre le protocole IP qui utilise les adresses IP pour construire ses paquets et les trames Ethernet qui elles utilisent les adresse MAC. En plus simple, c'est un protocole qui permet de retrouver un adresse MAC à partir d'une adresse IP. Le protocole RARP

Le protocole RARP (Reverse Address Resolution Protocol) est beaucoup moins utilisé, il signifie Protocole ARP inversé, il s'agit donc d'une sorte d'annuaire inversé des adresses logiques et physiques.

En réalité le protocole RARP est essentiellement utilisé pour les stations de travail n'ayant pas de disque dur et souhaitant connaître leur adresse physique…

Le protocole RARP permet à une station de connaître son adresse IP à partir d'une table de correspondance entre adresse MAC (adresse physique) et adresses IP hébergée par une passerelle (gateway) située sur le même réseau local (LAN).

Pour cela il faut que l'administrateur paramètre le gateway (routeur) avec la table de correspondance des adresses MAC/IP. En effet, à la différence de ARP ce protocole est statique. Il faut donc que la table de correspondance soit toujours à jour pour permettre la connexion de nouvelles cartes réseau. note 1 :Normalement non. Un constructeur qui fabrique des cartes réseau va acheter des adresses MAC, ou plus exactement des morceaux d'adresses MAC. Les trois premiers octets de l'adresse représentent le constructeur.

Ainsi, quand un constructeur veut produire les cartes, il achète trois octets qui lui permettront de donner des adresses à ses cartes. Par exemple, j'achète la suite de trois octets : 00:01:02. Toutes les cartes réseau que je vais produire vont commencer par ces trois octets, par exemple : 00:01:02:00:00:01 ; puis : 00:01:02:00:00:02 ; etc.

Si je choisis toujours les trois derniers octets différents pour les cartes que je produis, je suis sûr qu'aucune autre carte réseau n'aura la même adresse MAC, car je suis le seul à posséder les trois

premiers octets 00:01:02 et j'ai fait attention à ce que les trois derniers ne soient pas identiques.

Récapitulons :

L'adresse MAC est l'adresse d'une carte réseau.

Elle est unique au monde pour chaque carte.

Elle est codée sur 6 octets (soit 48 bits).

Grâce à l'adresse MAC, je suis donc capable d'envoyer des informations à la carte réseau d'une machine !

Une adresse MAC spéciale

Parmi les adresses MAC, il y en a une particulière, c'est l'adresse dans laquelle tous les bits sont à 1, ce qui donne ff:ff:ff:ff:ff:ff. Cette adresse est appelée l'adresse de broadcast.

L'adresse de broadcast est une adresse universelle qui identifie n'importe quelle carte réseau. Elle me permet ainsi d'envoyer un message à toutes les cartes réseaux des machines présentes sur mon réseau, en une seule fois.Toute machine qui reçoit une trame qui a, comme adresse MAC de destination, l'adresse de broadcast, considère que la trame lui est destinée.

CIDR : qu'est-ce que le classless interdomain routing ?

`Introduction aux systèmes d'adressage.`

Le système d'adressage par classes fonctionne selon le même principe :

les adresses IP sont rangées par classes et dans chacune d'elles se trouvent des plages. Si une entreprise demandait des adresses pour cent ordinateurs, on choisirait la classe lui offrant ce nombre d'adresses et on lui offrirait des adresses IP issues de cette classe.

Le problème de ce système d'adressage est le pourcentage assez élevé de perte d'adresses. Nous avons vu que toutes les adresses IP de la classe A, par exemple, nous permettaient d'obtenir 16 777 214 adresses IP par réseau en utilisant les masques par défaut. Cela dit, l'entreprise qui voudrait une adresse IP pour un réseau de 10 000 hôtes aurait quand même 16 767 214 d'adresses en surplus. Quelle perte !

L'adressage sans classes (ou adressage CIDR = Classless Inter Domain Routing) est le système de gestion et d'allocation d'adresses IP le plus utilisé aujourd'hui. Le but de ce nouveau système s'articule autour de deux points :

Économiser les adresses IP. Et Faciliter le routage.

par CIDR comprenez « routage effectué entre domaines qui n'utilisent pas les classes ». On comprend alors que le réseau Internet est fondé sur ce système d'adressage. Logique, quand on y pense… Sinon, comment un système d'adressage par classes aurait-il pu supporter plus de 2 milliards d'internautes ? Depuis les années quatre-vingt-dix, nous n'aurions plus d'adresses IP disponibles

En anglais, les adresses IP utilisant l'adressage CIDR sont appelées classless adresses par opposition aux classful adresses, qui désignent celles qui utilisent l'adressage par classes. Habituez-vous à ce vocabulaire qui est très présent dans les documentations en anglais. pourquoi ce nouveau système a été créé. Soit l'adresse 192.168.10.0/23.

À ce stade, vous êtes censés savoir que le nombre après le slash (/) équivaut au nombre de bits masqués. Si vous avez encore des difficultés, nous vous recommandons la relecture de la sous-partie sur la notation du masque..

Dans une adresse IP, c'est la partie gauche qui correspond à l'identité du réseau

la partie réseau de l'adresse est 192.168

la partie droite soit 1.15

comment on détermine ou comment on affecte ces deux là :

la façon d'identifier la partie réseau et la partie hôte, c'est de connaître le nombre de bits réservés à la partie réseau, et c'est ce que l'on appelle le masque de sous-réseau.

Sauf qu'il y a deux manière de représenté ce masque, soit en notation décimale à point (comme l'adresse IP), soit en notation CIDR (celle que l'on doit privilégier). La notation CIDR à l'avantage de nous informé directement du nombre de bits engagés dans le masque !

Ainsi, un masque 255.255.255.0 correspond à /24, soit 24 sur 32 bits réservés à la partie réseau.

Pour un masque 255.255.255.240, c'est moins évident au premier coup d'œil de comprendre qu'il s'agit du masque /28, et qu'il ne reste donc que 4 bits pour l'adressage des machines de ce sous-réseau.. . Dans mon adresse 192.168.1.15 puisque le masque de sous-réseau est /24.

la partie réseau est 192.168.1.0 et non pas 192.168*

Il ne me reste donc que 8 bits pour adresser les machines de ce sous-réseau.

Ma machine a dont l'identifiant 15 dans le sous-réseau 192.168.1.0/24 (192.168.1.* .

Et quand à savoir comment est attribuée cet identifiant sur ce sous-réseau, et bien, l'attribution est libre. Il n'y a aucune règle.

On peux attribué n'importe quelle identifiant de 1 à 254 à ta machine qui porte actuellement l'identifiant 15, et même un identifiant déjà utilisé par une autre machine sur le même sous-réseau (mais cela entraîne inévitablement certains problèmes…).

D'où l'usage du protocole DHCP, pour s'assurer que l'attribution des identifiants machines suis certaines règles.

outils de communications sécurisées

ssh

SSH, ou Secure Socket Shell, est un protocole réseau qui permet aux administrateurs d'accéder et de gerer à distance à un ordinateur (serveur), en toute sécurité. c'un protocole de communication (programme informatique) Il permet la connexion d'une machine distante (serveur) via une liaison sécurisée dans le but de transférer des fichiers en toute sécurité.

Le protocole SSH permet de sécuriser les transferts de fichier, notamment via la commande SCP et SFTP

vérifier l'installation d'OpenSSH

```
ls -lha /etc/ssh/sshd_config
-rw-r--r-- 1 root root 3,3K 27 mars   2022 /etc/ssh/sshd_config
```

le serveur est-il en fonctionnement

```
systemctl status sshd
```

pour infos: tapez en console

```
apt-cache show openssh-client

apt-cache show openssh-server
```

ou

```
dpkg -l | grep ssh
```

A l'installation de debian la question est posé pur installer ssh.

Si rien n'est installé

```
apt install openssh-client  openssh-server  ssh-askpass task-ssh-server
```

Par défaut, le service SSH écoute sur le port 22.

```
netstat -tnplv | grep ssh
[sudo] Mot de passe de lagrenouille :
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
1234/sshd: /usr/sbi
tcp6       0      0 :::22                   :::*                    LISTEN
1234/sshd: /usr/sbi
```

```
ss -lntp | grep 22
LISTEN 0      128          0.0.0.0:22         0.0.0.0:*
LISTEN 0      224        127.0.0.1:5432       0.0.0.0:*
LISTEN 0      128             [::]:22            [::]:*
LISTEN 0      224            [::1]:5432          [::]:*
```

-l permet de ne lister que les ports en écoute

-n permet d'afficher les ports de manière numérique

-t permet de ne lister que les ports TCP

-p permet enfin de lister les processus derrière chaque ports

lors de la première connexion ssh, il sera créer un fichier known_hosts dans .ssh

ls -lha .ssh rw-r–r– 1 user user 6,1K 17 mai 16:43 known_hosts

pour envoyer un fichier

```
scp fichier.odt user@server.org
```

pour envoyer un repertoire

```
scp -rdv repertoire user@server.org
```

Création et utilisation de clé

```
ssh-keygen -t rsa
```

Le programme va te demander un nom de fichier, par convention c'est « id_rsa » qui est utilisé, mais tu peux nommer ta paire de clés comme tu le souhaites. Ensuite tu dois entrer une passphrase, une sorte de mot de passe au cas tu as plusieurs utilisateurs sur ta machine, pour ne pas qu'ils puissent utiliser tes propres clés.

Le programme va te demander un nom de fichier, par convention c'est « id_rsa » qui est utilisé, mais tu peux nommer ta paire de clés comme tu le souhaites. Ensuite tu dois entrer une passphrase, une sorte de mot de passe au cas tu as plusieurs utilisateurs sur ta machine, pour ne pas qu'ils puissent utiliser tes propres clés. l'installateur va vous demander où enregistrer la pair de clef et une « passphrase », laissez la vide et appuyer simplement sur entrer à chaque fois, jusqu'à ce que le « fingerprint » apparaisse. Vos fichiers « /root/.ssh/id_rsa » (clef privée) et « /root/.ssh/id_rsa.pub » (clef publique) ont été créées !

our identification has been saved in /Users/nouslesdevs/.ssh/test_rsa.

Your public key has been saved in /Users/nouslesdevs/.ssh/test_rsa.pub.

The key fingerprint is:

SHA256:nATZTHGyAok9HUGPlcwC9FAflYTjVkNW/5tYr8CMOVQ xxx@nousxxxxxx

The key's randomart image is:

```
+---[RSA 4096]----+
|    +=**X*B*o.    |
|   . +++*O*+  .   |
|      .+o=o .E .  |
|       +o. .   . |
|       .S .    ..|
|          . =  o +|
|           + +. o.|
|             . . . |
|               .  |
+----[SHA256]-----+
```

Transfert de la clef publique

Maintenant, nous allons partager notre clef publique et ainsi a prendre à notre serveur à accepter les connexions sans mot de passe de notre autre serveur. Pour cela, rien de plus simple, une petite ligne de commande :

$ root@server1:~# ssh-copy-id -i ~/.ssh/id_rsa.pub root@server

Il vous demandera le mot de passe root de votre serveur 2 pour la dernière fois ! Faites de même dans le sens inverse pour accepter les connections de Serveur2 sur Server1.

connectez vous pour travailler sur le serveur

```
ssh  user@serveur2.org
```

ou

```
ssh  user@ip_public
```

sur mon lan

```
ssh lagrenouille@192.168.1.12
```

S’il ne vous demande pas de mot de passe, c’est que tout marche ! Effectuez le même test dans l’autre sens pour vérifier que la manipulation est bien symétrique. Vous venez de créer un tunnel SSH entre vos deux serveurs et ainsi de vous affranchir de vos mots de passes.

Créer le fichier authorized_keys

```
nano  .ssh/authorized_keys
```

```
cat <votre_fichier_de_clés_publiques> >> ~/.ssh/authorized_keys
```

sur le serveur, vous devez mettre la clé publique dans le ~/.ssh/authorized_keysfichier.

Le fichier “authorized_keys” ce trouve sur le serveur, dans le répertoire personnel du compte utilisateur qui reçoit des connexions distante en “ssh” protégé avec une clé privé.

L'emplacement du fichier est “~/.ssh/authorized_keys” ou l'adresse en entier “/home/[nom_utilisateur]/.ssh/authorized_keys”. le mot de pass est demandé

Pour se connecter

ssh-add ~/.ssh/id_rsa

paraphrase cle rsa

xxxxxxxxxxx

un confirmation vous sera affiché sur la console.

vous pouvez maintenant vous connecter

```
ssh -i ~/.ssh/id_rsa toto@monserveur.fr
```

si le port externe du serveur distant est configuré à 2222 pour externe

```
ssh -p 2222 -i ~/.ssh/id_rsa toto@monserveur.fr
```

suivant le nombre de clés utilisés, votre .ssh ressemblera à peu prés à ça:

```
ls -lha .ssh
total 48K
drwxr-xr-x  2 momo momo 4,0K 28 déc.  14:20 .
drwxr-xr-x 65 momo momo  20K 25 mai   21:00 ..
-rw-r--r--  1 momo momo    0 28 déc.  12:55 autorized_key
-rw-------  1 momo momo 1,9K 24 oct.   2021 grenouille
-rw-r--r--  1 momo momo  393 24 oct.   2021 grenouille.pub
-rw-------  1 momo momo 2,6K 24 oct.   2021 id_rsa1
-rw-r--r--  1 momo momo  575 24 oct.   2021 id_rsa1.pub
-rw-r--r--  1 momo momo 6,1K 17 mai   16:43 known_hosts
```

Voir man ssh pour plus d'infos

Lire la doc fail2ban

lire vos fichiers de conf dans etc

```
cat /etc/ssh/sshd_config
```

```
cat /etc/ssh/ssh_config
```

rsync

rsync :pour “remote synchronisation”..(synchroniser les fichiers éloignés)

Fréquemment utilisé pour faire des sauvegardes.

les principales options de la commande rsync

-v –verbose	mode verbeux
-q –quiet	moins loquace, Cette option diminue la quantité d'information affichée lors du transfert, les messages du serveur distant notamment sont supprimés. Cette option est utile lorsque vous appelez rsync à partir de cron.
-c –checksum	utilise la somme de contrôle, pas la date ni la taille,Ceci force l'expéditeur à faire une somme de contrôle 128-bit MD4 de tous les fichiers avant le transfert. La somme de contrôle est ensuite explicitement vérifiée à la réception et tous les fichiers du même nom qui existent déjà et ont la même somme de contrôle et la même taille sur le système de réception sont ignorés. Cette option peut être assez lente.
-a –archive	mode archivage; identique à -rlptgoD (pas -H)

-r –recursive	visite récursive des répertoires
–delete	efface les fichiers qui n'existent pas chez l'émetteur
–delete-before	efface avant le transfert (par défaut)
–delete-during	efface au cours du transfert, pas avant
–delete-after	efface après transfert, pas avant
–delete-excluded	efface également les fichiers exclus côté réception
–ignore-errors	efface même s'il y a eu des erreurs E/S
–force	
–max-delete=NUM	n'efface pas plus de NUM fichiers
–max-size=TAILLE	ne transfert le fichiers plus gros que TAILLE
–partial	conserve les fichiers partiellement transférés
–partial-dir=RÉP	place les fichiers partiellement transférés dans RÉP
–delay-updates	ne remplace les fichiers mis à jour qu'à la fin
–numeric-ids	ne remplace pas les uid/gid par des noms d'utilisateur/groupe
–timeout=DURÉE	fixe la durée d'attente E/S en secondes

Voir le man pour plus d'options

exemple d'une sauvegarde régulière:

```
rsync -av --partial --progress --exclude=cache --exclude=cache-css --
exclude=cache-gd2 --exclude=cache-js --exclude=cache-texte --exclude=cache-
vignettes u896754042@home454121550.1and1-data.host: /media/momo/chemin-vers-
DD_externe/archives-histoires2//sauvegarde-19-mai-2023/
```

https://debian-facile.org/doc:systeme:rsync:backup

https://technique.arscenic.org/transfert-de-donnees-entre/article/rsync-synchronisation-distant-de

filezilla/sftp

FileZilla Client est un client FTP, FTPS et SFTP, développé sous la licence publique générale GNU

```
apt install filezilla
```

FileZilla est un client FTP (File Transfert Protocol) qui vous permet de charger et télécharger des fichiers sur un serveur distant, notamment les éléments d’un site web chez un hébergeur.

filezilla c'est:

Gestionnaire des connexions

Connexion par protocole SSH (SFTP)

File d'attente

Répertoire de liens déjà visités

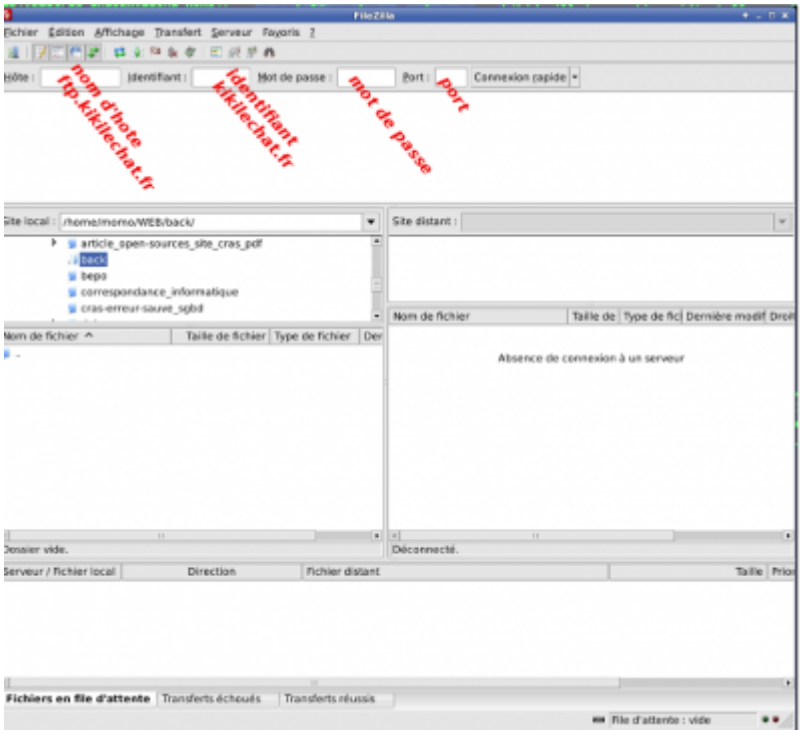
Compression des données en cours de chargement, ce qui permet d'accélérer la vitesse de transfert

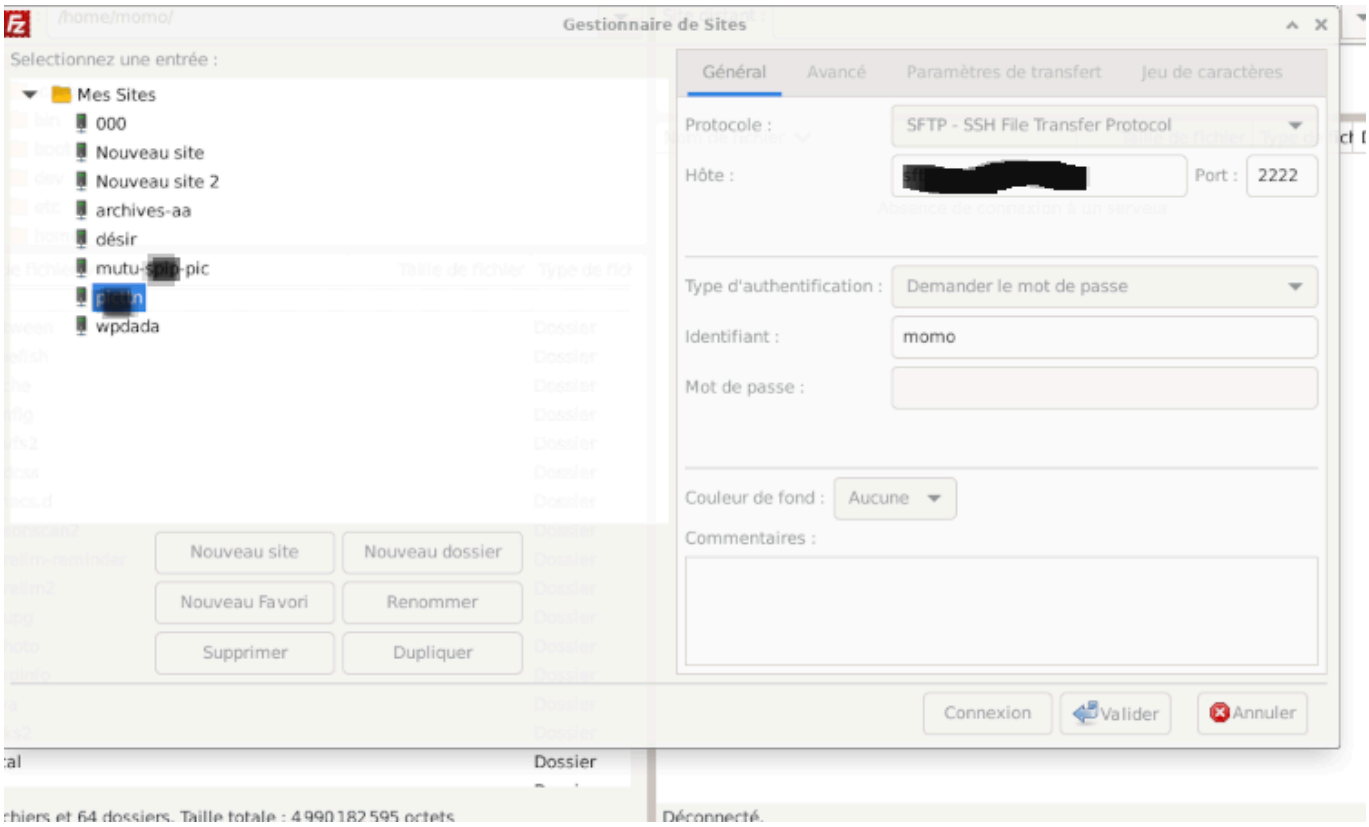
Doubles fenêtres paramétrables « répertoire local/ordinateur distant ».

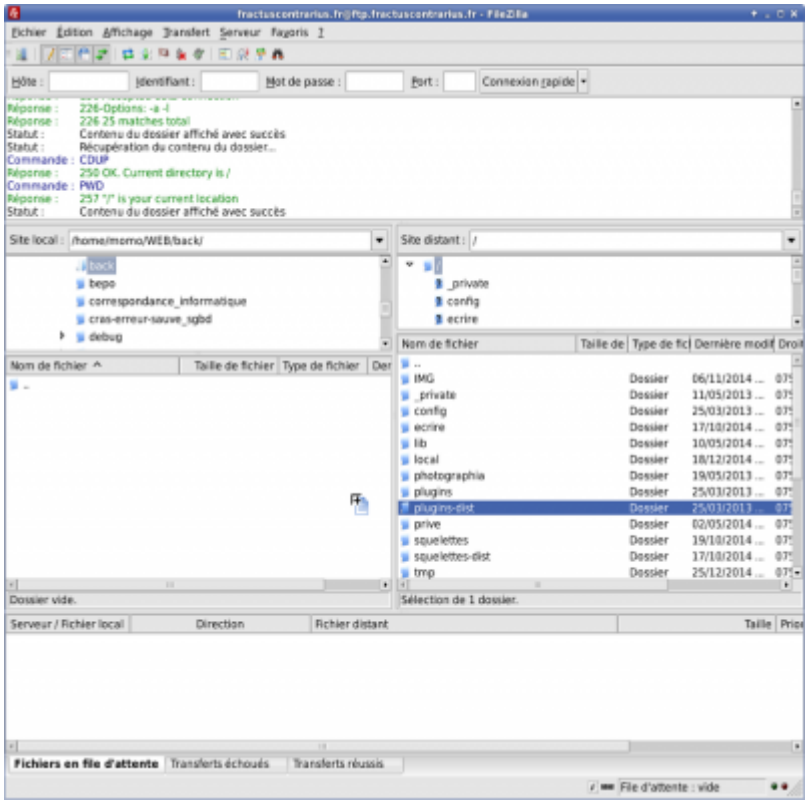
DONNÉ PAR VOTRE HÉBERGEUR :

– FTP – Hôte : ftp.kikilechat.fr ou l'adresse IP – Nom d'utilisateur : kikilechaton – Mot de passe : 05RouleEnBossag12

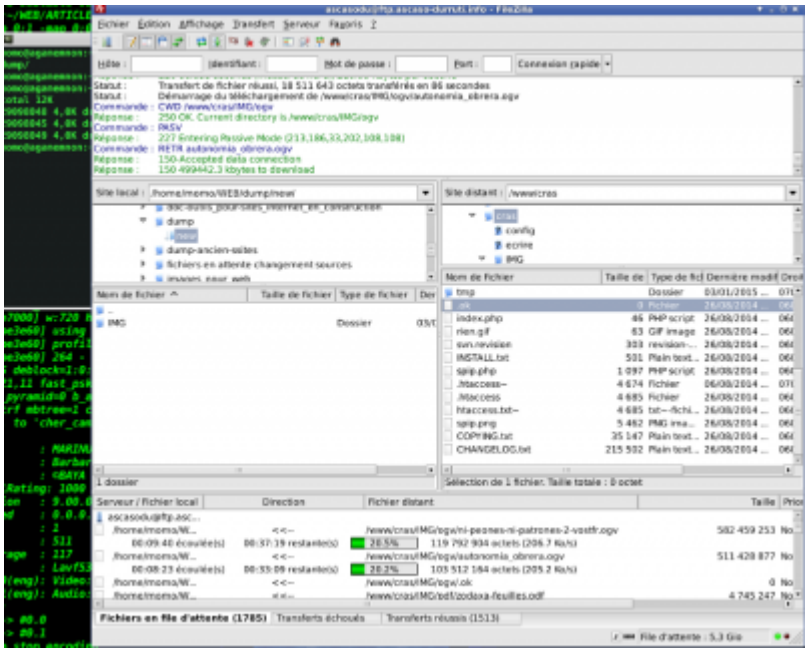
Ci dessous Filezilla tel que vous le verrez à l'ouverture







transfert sftp en cours (ici une mise à jour du cms spip)



commandes réseau

ping:

ping-(Packet Internet Groper) cette commande permet de tester l'accessibilité d'une autre machine à travers un réseau IP. (utilise le protocole ICMP) • cet outils vérifie la connectivité d'un ordinateur à internet. exemple :

```
ping debian-facile.org (89.234.146.138) 56(84) bytes of data.
64 bytes from stolon.debian-facile.org (89.234.146.138): icmp_seq=1 ttl=54
time=28.9 ms
64 bytes from stolon.debian-facile.org (89.234.146.138): icmp_seq=2 ttl=54
time=28.2 ms
64 bytes from stolon.debian-facile.org (89.234.146.138): icmp_seq=3 ttl=54
time=28.0 ms
64 bytes from stolon.debian-facile.org (89.234.146.138): icmp_seq=4 ttl=54
time=28.4 ms
64 bytes from stolon.debian-facile.org (89.234.146.138): icmp_seq=5 ttl=54
time=28.2 ms
^C
--- debian-facile.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 28.003/28.337/28.928/0.318 ms
```

nmap

(“Network Mapper”) est un outil open source d'exploration réseau et d'audit de sécurité.

En plus de la table des ports intéressants, Nmap peut aussi fournir de plus amples informations sur les cibles comme les noms DNS (reverse DNS), deviner les systèmes d'exploitation utilisés, obtenir le type de matériel ou les adresses MAC.surveille les hôtes et les services actifs.

Liste des options Nmap

Options	Commandes
-exclude	Exclure des hôtes du scan
-n	Désactiver la résolution DNS
-open	Afficher que les ports ouverts
-oN	Enregistrer le résultat du scan dans un fichier au formate texte
-oX	Enregistrer le résultat du scan dans un fichier au formate XML
-p	Spécifier les ports réseaux à scanner
-Pn	Désactiver la découverte d’hôte
-r	Analyser les ports consécutivement
-sT	Faire un scan de port TCP
-sU	Faire un scan de port UDP
-sV	Trouver les versions du service
-script	Utilise un script interne à nmap pour scan de vulnérabilité, bruteforce, etc
-v	Mode bavard
-vv	Mode trés bavard

scanner votre LAN

```
nmap -T4 -sP 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2023-05-23 11:17 CEST
```

```
Nmap scan report for lan.home (192.168.1.1)
Host is up (0.00052s latency).
MAC Address: 08:87:C6:B4:A1:50 (Ingram Micro Services)
Nmap scan report for arthur.home (192.168.1.10)
Host is up (0.0015s latency).
MAC Address: 18:C0:4D:C5:AC:9D (Giga-byte Technology)
Nmap scan report for five.home (192.168.1.17)
Host is up (0.10s latency).
MAC Address: 56:4A:53:80:CC:E0 (Unknown)
Nmap scan report for lebug-3.home (192.168.1.15)
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 5.21 seconds
```

scanner un sous-réseau

```
 nmap 192.168.1.*
Starting Nmap 7.80 ( https://nmap.org ) at 2023-05-23 11:39 CEST
Nmap scan report for lan.home (192.168.1.1)
Host is up (0.0011s latency).
Not shown: 992 filtered ports
PORT    STATE  SERVICE
53/tcp  open   domain
80/tcp  open   http
113/tcp closed ident
135/tcp closed msrpc
139/tcp open   netbios-ssn
443/tcp open   https
445/tcp open   microsoft-ds
631/tcp open   ipp
MAC Address: 08:87:C6:B4:A1:50 (Ingram Micro Services)
```

nmap votre serveur

```
nmap rastacouère.org
Starting Nmap 7.80 ( https://nmap.org ) at 2023-05-23 11:16 CEST
Nmap scan report for rastacouère.org (127.0.0.1)
Host is up (0.000013s latency).
rDNS record for 127.0.0.1: localhost
Not shown: 993 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
443/tcp  open  https
3306/tcp open  mysql
5432/tcp open  postgresql
8081/tcp open  blackice-icecap
```

en mode très bavard

```
nmap -vv rastacouère.org
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2023-05-23 11:41 CEST
Initiating SYN Stealth Scan at 11:41
Scanning funambule.org (127.0.0.1) [1000 ports]
Discovered open port 80/tcp on 127.0.0.1
Discovered open port 3306/tcp on 127.0.0.1
Discovered open port 25/tcp on 127.0.0.1
Discovered open port 443/tcp on 127.0.0.1
Discovered open port 22/tcp on 127.0.0.1
Discovered open port 8081/tcp on 127.0.0.1
Discovered open port 5432/tcp on 127.0.0.1
Completed SYN Stealth Scan at 11:41, 0.06s elapsed (1000 total ports)
Nmap scan report for funambule.org (127.0.0.1)
Host is up, received localhost-response (0.000013s latency).
rDNS record for 127.0.0.1: localhost
Scanned at 2023-05-23 11:41:03 CEST for 0s
Not shown: 993 closed ports
Reason: 993 resets
PORT     STATE SERVICE         REASON
22/tcp   open  ssh             syn-ack ttl 64
25/tcp   open  smtp            syn-ack ttl 64
80/tcp   open  http            syn-ack ttl 64
443/tcp  open  https           syn-ack ttl 64
3306/tcp open  mysql           syn-ack ttl 64
5432/tcp open  postgresql      syn-ack ttl 64
8081/tcp open  blackice-icecap syn-ack ttl 64

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
           Raw packets sent: 1000 (44.000KB) | Rcvd: 2007 (84.308KB)
```

```
nmap -e enp2s0 rastacouère.org
Starting Nmap 7.80 ( https://nmap.org ) at 2023-05-23 11:32 CEST
Note: Host seems down. If it is really up, but blocking our ping probes, try
-Pn
Note : L'hôte semble en panne. S'il est réellement actif, mais qu'il bloque
nos sondes ping, essayez -Pn
```

```
 nmap -Pn enp2s0 rastacouère.org
Starting Nmap 7.80 ( https://nmap.org ) at 2023-05-23 11:34 CEST
Failed to resolve "enp2s0".
Nmap scan report for rastacouère.org (127.0.0.1)
Host is up (0.000012s latency).
rDNS record for 127.0.0.1: localhost
Not shown: 993 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
443/tcp  open  https
```

```
3306/tcp open  mysql
5432/tcp open  postgresql
8081/tcp open  blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
```

information sur l'OS

```
 nmap -O 192.168.1.1
Starting Nmap 7.80 ( https://nmap.org ) at 2023-05-23 11:46 CEST
Nmap scan report for lan.home (192.168.1.1)
Host is up (0.0011s latency).
Not shown: 992 filtered ports
PORT    STATE  SERVICE
53/tcp  open   domain
80/tcp  open   http
113/tcp closed ident
135/tcp closed msrpc
139/tcp open   netbios-ssn
443/tcp open   https
445/tcp open   microsoft-ds
631/tcp open   ipp
MAC Address: 08:87:C6:B4:A1:50 (Ingram Micro Services)
Device type: general purpose|media device|storage-misc|firewall
Running (JUST GUESSING): Linux 2.6.X|3.X|4.X (96%), Dish embedded (93%),
Excito embedded (89%), WatchGuard Fireware 11.X (89%), Synology DiskStation
Manager 5.X (88%)
OS CPE: cpe:/o:linux:linux_kernel:2.6.32 cpe:/h:dish:hopper
cpe:/o:linux:linux_kernel:3 cpe:/h:excito:b3 cpe:/o:watchguard:fireware:11.8
cpe:/o:linux:linux_kernel cpe:/a:synology:diskstation_manager:5.1
cpe:/o:linux:linux_kernel:4
Aggressive OS guesses: Linux 2.6.32 (96%), Dish Network Hopper media device
(93%), Linux 2.6.32 - 3.0 (91%), Linux 3.2 - 3.8 (91%), Linux 2.6.32 - 3.10
(90%), Linux 2.6.32 or 3.10 (90%), Linux 3.0 (90%), Excito B3 file server
(Linux 2.6.39) (89%), Linux 2.6.39 (89%), Linux 3.4 (89%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.93 seconds
```

scanner les ports TCP :

```
 nmap -sT 192.168.1.1
```

Scanner tous les ports UDP

```
nmap -sU 192.168.1.1
```

(pour verifier les ports ouvert par vous et aussi par d'autres

```
nmap -v -A localhost
```

Vous pouvez également utiliser Nmap pour lancer une attaque par bruteforce. Là aussi, on utilise l'option -script pour spécifier le type d'attaque.

Lire la doc nmap 😎

netstat

`network statistics`permet aux administrateurs réseau de gérer les équipements du réseau, de superviser et de diagnostiquer des problèmes réseaux et matériels à distance.

`liste les ports ouverts` (avec iproute "ip link show ")

```
netstat -a
```

`Liste de tous les ports tcp`

```
netstat -at
```

`Liste de tous les ports UDP`

```
netstat -ua
```

`Liste uniquement les ports d'écoute` (avec iproute "ss -l "

```
netstat -l
```

`Liste seuls ports tcp en écoute`

```
netstat -lt
```

`Liste écoute uniquement les ports UDP`

```
netstat -lu
```

`Liste seulement les ports d'écoute UNIX`

```
netstat -lx
```

`Voir les statistiques pour chaque protocole`

```
netstat -s
```

`donne des infos assez complete sur l'état du réseau`

```
netstat -laput
```

pour voir en console ce qui entre et sort du pc

```
netstat -ntap
```

pour voir si des ports sont bloqués

```
netstat – alpe
```

Affichage des noms PID et le programme de sortie de netstat utilisant:

```
netstat -p
```

```
netstat -pt
```

```
Netstat -an
```

```
Netstat -c
```

```
netstat - verbose
```

```
netstat -r
```

```
netstat -ap | grep ssh
```

```
Netstat -an | grep ': 80'
```

```
Netstat -dire
```

```
netstat -nr permet de connaître la table de routage construite par ifconfig
```

```
 netstat -nr
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic   MSS Fenêtre irtt
Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG        0 0          0
enp2s0
10.0.3.0        0.0.0.0         255.255.255.0   U         0 0          0
lxcbr0
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0
enp2s0
```

```
pour voir les connexions ssh
```

```
 netstat -anp | grep "sshd"
```

vnstat

```
apt install vnstat
```

```
systemctl status vnstat
● vnstat.service - vnStat network traffic monitor
     Loaded: loaded (/lib/systemd/system/vnstat.service; enabled; vendor
preset: enabled)
     Active: active (running) since Sun 2023-05-21 09:38:04 CEST; 3 days ago
       Docs: man:vnstatd(8)
             man:vnstat(1)
             man:vnstat.conf(5)
   Main PID: 1097 (vnstatd)
      Tasks: 1 (limit: 38351)
     Memory: 2.3M
        CPU: 39.560s
     CGroup: /system.slice/vnstat.service
             └─1097 /usr/sbin/vnstatd -n

Warning: some journal files were not opened due to insufficient permissions.
```

vnstat collectera l'utilisation du réseau en arrière-plan en utilisant un si petit pourcentage de CPU qu'il n'apparaît pas dans la liste des 9 meilleurs processus de conky

Comme vous pouvez le voir vnstat recense immédiatement les interfaces disponibles, et nous donne déjà quelques informations de base.

Nous pouvons voir entre autres si wlan0 est en activité et que enp2s0 n’est pas connectée,ou vice versa…selon que nous sommes connecté en wifi ou en filiaire.

vnstat –help

-q	–query query database
-h	–hours show hours
-d	–days show days
-m	–months show months
-w	–weeks show weeks
-t	–top10 show top10
-s	–hort use short output
-u	–update update database
-i	–iface select interface (default: eth0)
-?	–help short help
-v	–version show version
-tr	–traffic calculate traffic
-ru –rateunit swap configured rate unit	
-l	–live show transfer rate in real time

```
vnstat -i enp2s0
```

testons l'option -h qui va nous indiquer la quantité de trafic heures par heures,

Tx et Rx signifient Transmission et Réception

rx est le trafic reçu

tx est le trafic transféré

```
sudo vnstat -i enp2s0 -h

 enp2s0  /  hourly

        hour        rx      |     tx      |    total    |   avg. rate
    ------------------------+-------------+-------------+---------------
    2023-05-24
        00:00      2,83 MiB |   40,25 MiB |   43,08 MiB |  100,39 kbit/s
        01:00      2,80 MiB |   10,15 MiB |   12,95 MiB |   30,17 kbit/s
        02:00      3,18 MiB |    6,43 MiB |    9,61 MiB |   22,39 kbit/s
        03:00      3,74 MiB |   72,12 MiB |   75,86 MiB |  176,77 kbit/s
        04:00      3,49 MiB |   39,15 MiB |   42,64 MiB |   99,36 kbit/s
        05:00      3,94 MiB |   87,76 MiB |   91,70 MiB |  213,67 kbit/s
        06:00      3,16 MiB |   76,91 MiB |   80,07 MiB |  186,58 kbit/s
        07:00      6,60 MiB |  130,50 MiB |  137,10 MiB |  319,47 kbit/s
        08:00      3,49 MiB |   48,41 MiB |   51,91 MiB |  120,95 kbit/s
        09:00      2,69 MiB |   55,19 MiB |   57,88 MiB |  134,87 kbit/s
        10:00      4,01 MiB |  127,45 MiB |  131,46 MiB |  306,32 kbit/s
        11:00      5,03 MiB |  176,41 MiB |  181,45 MiB |  422,80 kbit/s
        12:00      3,60 MiB |  206,39 MiB |  209,98 MiB |  489,30 kbit/s
        13:00      5,24 MiB |  171,54 MiB |  176,78 MiB |  411,93 kbit/s
        14:00      4,09 MiB |  216,63 MiB |  220,72 MiB |  514,32 kbit/s
        15:00      5,45 MiB |  129,03 MiB |  134,48 MiB |  313,37 kbit/s
        16:00      4,84 MiB |  113,47 MiB |  118,31 MiB |  275,68 kbit/s
        17:00      7,08 MiB |  147,71 MiB |  154,79 MiB |  360,69 kbit/s
        18:00      4,44 MiB |   59,99 MiB |   64,44 MiB |  150,15 kbit/s
        19:00      6,52 MiB |  121,21 MiB |  127,74 MiB |  297,65 kbit/s
        20:00      5,39 MiB |   47,34 MiB |   52,73 MiB |  122,87 kbit/s
        21:00      7,98 MiB |   86,99 MiB |   94,97 MiB |  221,29 kbit/s
        22:00      5,46 MiB |   62,09 MiB |   67,55 MiB |  157,40 kbit/s
        23:00     14,05 MiB |  473,45 MiB |  487,50 MiB |    1,36 Mbit/s
    ------------------------+-------------+-------------+---------------
```

suivre l'évolution du trafic en temps réel :

```
sudo vnstat -i enp2s0 -l
Monitoring enp2s0...    (press CTRL-C to stop)

   rx:       536 bit/s     1 p/s          tx:       744 bit/s     0 p/s^C
```

répertorier toutes les interfaces disponibles

```
sudo vnstat --iflist
Available interfaces: enp3s0 enp2s0 (1000 Mbit) lxcbr0
```

Rechercher le trafic réseau:

```
vnstat -q

                     rx      /      tx      /     total    /   estimated
 enp2s0:
       2023-04      3,73 GiB  /  147,49 GiB  /  151,23 GiB
       2023-05      7,33 GiB  /  160,10 GiB  /  167,43 GiB  /  216,14 GiB
     yesterday    121,75 MiB  /    2,68 GiB  /    2,80 GiB
         today      9,04 MiB  /  374,75 MiB  /  383,80 MiB  /   26,99 GiB

 enp3s0: Not enough data available yet.
```

tcpdump

Le programme tcpdump permet d'analyser les paquets envoyés ou reçus sur une interface réseau.

Voici les options générales de TCPdump :

Flag	et Description
-i <interface>	Écouter une interface réseau spécifique, .e.g. "-i igb0"
-n	N'effectuez pas de résolution DNS inversée sur les adresses IP
-w <filename>	Enregistrez la capture au format pcap dans <nom de fichier>, par exemple "-W /tmp/wan.pcap"
-s	Durée de capture: quantité de données à capturer à partir de chaque image
-c <packets>	Quitter après avoir reçu un nombre spécifique de paquets
-p	Ne mettez pas l'interface en mode promiscuité
-v	Mode Verbose (bavard)
-e	Imprimer l'en-tête de la couche de liaison sur chaque ligne

```
tcpdump -i any
```

Ctrl +C pour stopper

`lister les interfaces réseaux`

```
tcpdump -D
1.enp2s0 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.enp3s0 [Up, Disconnected]
5.lxcbr0 [Up, Disconnected]
6.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
7.nflog (Linux netfilter log (NFLOG) interface) [none]
8.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
9.dbus-system (D-Bus system bus) [none]
10.dbus-session (D-Bus session bus) [none]
```

```
tcpdump icmp -vv -X
```

```
tcpdump: listening on enp2s0, link-type EN10MB (Ethernet), snapshot length
262144 bytes
09:50:07.409815 IP (tos 0x0, ttl 239, id 29297, offset 0, flags [DF], proto
ICMP (1), length 36)
    ec2-16-163-146-171.ap-east-1.compute.amazonaws.com > lebug-3.home: ICMP
echo request, id 10, seq 17458, length 16
    0x0000:  4500 0024 7271 4000 ef01 b461 10a3 92ab  E..$rq@....a....
    0x0010:  c0a8 010f 0800 d2c7 000a 4432 1763 8dcf  ..........D2.c..
    0x0020:  5cce defa 0000 0000 0000 0000 0000       \.............
09:50:07.409858 IP (tos 0x0, ttl 64, id 31077, offset 0, flags [none], proto
ICMP (1), length 36)
    lebug-3.home > ec2-16-163-146-171.ap-east-1.compute.amazonaws.com: ICMP
echo reply, id 10, seq 17458, length 16
    0x0000:  4500 0024 7965 0000 4001 9c6e c0a8 010f  E..$ye..@..n....
    0x0010:  10a3 92ab 0000 dac7 000a 4432 1763 8dcf  ..........D2.c..
    0x0020:  5cce defa                                \...
09:50:07.409888 IP (tos 0x0, ttl 233, id 43556, offset 0, flags [DF], proto
ICMP (1), length 36)
    ec2-13-208-163-151.ap-northeast-3.compute.amazonaws.com > lebug-3.home:
ICMP echo request, id 3, seq 17124, length 16
    0x0000:  4500 0024 aa24 4000 e901 7495 0dd0 a397  E..$.$@...t.....
    0x0010:  c0a8 010f 0800 354c 0003 42e4 1763 8dcf  ......5L..B..c..
    0x0020:  5b73 7f26 0000 0000 0000 0000 0000       [s.&..........
09:50:07.409909 IP (tos 0x0, ttl 64, id 41479, offset 0, flags [none], proto
ICMP (1), length 36)
    lebug-3.home > ec2-13-208-163-151.ap-northeast-3.compute.amazonaws.com:
ICMP echo reply, id 3, seq 17124, length 16
    0x0000:  4500 0024 a207 0000 4001 65b3 c0a8 010f  E..$....@.e.....
    0x0010:  0dd0 a397 0000 3d4c 0003 42e4 1763 8dcf  ......=L..B..c..
    0x0020:  5b73 7f26                                [s.&
09:50:07.477678 IP (tos 0x0, ttl 241, id 64082, offset 0, flags [DF], proto
ICMP (1), length 36)
    ec2-35-180-135-192.eu-west-3.compute.amazonaws.com > lebug-3.home: ICMP
echo request, id 31, seq 24808, length 16
    0x0000:  4500 0024 fa52 4000 f101 225a 23b4 87c0  E..$.R@..."Z#...
    0x0010:  c0a8 010f 0800 d092 001f 60e8 1763 8dcf  ..........`..c..
    0x0020:  65d6 bb5c 0000 0000 0000 0000 0000       e..\..........
09:50:07.477720 IP (tos 0x0, ttl 64, id 44941, offset 0, flags [none], proto
ICMP (1), length 36)
    lebug-3.home > ec2-35-180-135-192.eu-west-3.compute.amazonaws.com: ICMP
echo reply, id 31, seq 24808, length 16
    0x0000:  4500 0024 af8d 0000 4001 5e20 c0a8 010f  E..$....@.^.....
    0x0010:  23b4 87c0 0000 d892 001f 60e8 1763 8dcf  #.........`..c..
    0x0020:  65d6 bb5c                                e..\
```

analyser le trafic réseau sur le port 443

```
tcpdump -i enp2s0: -nn -s0 -v port 443
tcpdump: listening on enp2s0:, link-type EN10MB (Ethernet), snapshot length
262144 bytes
12:01:39.354631 IP (tos 0x0, ttl 119, id 35270, offset 0, flags [none],
```

```
proto TCP (6), length 60)
    66.249.70.134.56747 > 192.168.1.15.443: Flags [S], cksum 0xb029
(correct), seq 722222374, win 65535, options [mss 1412,sackOK,TS val
3607899689 ecr 0,nop,wscale 8], length 0
12:01:39.354699 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP
(6), length 60)
    192.168.1.15.443 > 66.249.70.134.56747: Flags [S.], cksum 0x4b65
(incorrect -> 0x1916), seq 4037143695, ack 722222375, win 65160, options
[mss 1460,sackOK,TS val 3277584316 ecr 3607899689,nop,wscale 7], length 0
12:01:39.457808 IP (tos 0x0, ttl 119, id 35271, offset 0, flags [none],
proto TCP (6), length 52)
    66.249.70.134.56747 > 192.168.1.15.443: Flags [.], cksum 0x4503
(correct), ack 1, win 256, options [nop,nop,TS val 3607899793 ecr
3277584316], length 0
12:01:39.457868 IP (tos 0x0, ttl 119, id 35272, offset 0, flags [none],
proto TCP (6), length 569)
    66.249.70.134.56747 > 192.168.1.15.443: Flags [P.], cksum 0x2b7b
(correct), seq 1:518, ack 1, win 256, options [nop,nop,TS val 3607899793 ecr
3277584316], length 517
12:01:39.457927 IP (tos 0x0, ttl 64, id 30156, offset 0, flags [DF], proto
TCP (6), length 52)
    192.168.1.15.443 > 66.249.70.134.56747: Flags [.], cksum 0x4b5d
(incorrect -> 0x419d), ack 518, win 506, options [nop,nop,TS val 3277584419
ecr 3607899793], length 0
12:01:39.464314 IP (tos 0x0, ttl 64, id 30157, offset 0, flags [DF], proto
TCP (6), length 2852)
    192.168.1.15.443 > 66.249.70.134.56747: Flags [P.], cksum 0x564d
(incorrect -> 0x0165), seq 1:2801, ack 518, win 506, options [nop,nop,TS val
3277584425 ecr 3607899793], length 2800
12:01:39.464330 IP (tos 0x0, ttl 64, id 30159, offset 0, flags [DF], proto
TCP (6), length 1348)
    192.168.1.15.443 > 66.249.70.134.56747: Flags [P.], cksum 0x506d
(incorrect -> 0x32fc), seq 2801:4097, ack 518, win 506, options [nop,nop,TS
val 3277584425 ecr 3607899793], length 1296
12:01:39.464410 IP (tos 0x0, ttl 64, id 30160, offset 0, flags [DF], proto
TCP (6), length 245)
    192.168.1.15.443 > 66.249.70.134.56747: Flags [P.], cksum 0x4c1e
(incorrect -> 0x0305), seq 4097:4290, ack 518, win 506, options [nop,nop,TS
val 3277584425 ecr 3607899793], length 193
12:01:39.566125 IP (tos 0x0, ttl 119, id 35273, offset 0, flags [none],
proto TCP (6), length 52)
    66.249.70.134.56747 > 192.168.1.15.443: Flags [.], cksum 0x3ca1
(correct), ack 1401, win 267, options [nop,nop,TS val 3607899902 ecr
3277584425], length 0 ..............arrêt par contrôl + c
```

si openvpn est installé

```
tcpdump -i tun0
```

mode verbeux avec -v et -vv

Analyser uniquement le port 80 de la source 192.168.1.14

tcpdump src 192.168.1.14 and port 80

ip

récupérer l'adresse IP (lan) de sa machine

```
ip a
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: enp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
state DOWN group default qlen 1000
    link/ether 74:d0:2b:11:23:f4 brd ff:ff:ff:ff:ff:ff
3: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 74:d0:2b:13:6b:57 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.15/24 brd 192.168.1.255 scope global dynamic
noprefixroute enp2s0
       valid_lft 53384sec preferred_lft 53384sec
    inet6 2a01:cb19:83e9:5500:e824:ece0:eb46:ca53/64 scope global temporary
dynamic
       valid_lft 86389sec preferred_lft 589sec
    inet6 2a01:cb19:83e9:5500:cec8:5495:6552:b3c/64 scope global temporary
deprecated dynamic
       valid_lft 86389sec preferred_lft 0sec
    inet6 2a01:cb19:83e9:5500:6e73:4f34:a0f0:f0f4/64 scope global temporary
deprecated dynamic
       valid_lft 86389sec preferred_lft 0sec
    inet6 2a01:cb19:83e9:5500:76d0:2bff:fe13:6b57/64 scope global dynamic
mngtmpaddr noprefixroute
       valid_lft 86389sec preferred_lft 589sec
    inet6 fe80::76d0:2bff:fe13:6b57/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: lxcbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state
DOWN group default qlen 1000
    link/ether 00:16:3e:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.1/24 brd 10.0.3.255 scope global lxcbr0
       valid_lft forever preferred_lft forever
```

récupérer que les adresses IPv4 et ipv6 avec ces commandes:

```
ip -4 a
```

```
ip -4 a
```

```
ip a show enp2s0
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 18:c0:4d:c5:ac:9d brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global dynamic enp2s0
       valid_lft 65081sec preferred_lft 65081sec
    inet6 2a01:cb19:83e9:5500:1ac0:4dff:fec5:ac9d/64 scope global dynamic
mngtmpaddr
       valid_lft 86362sec preferred_lft 562sec
    inet6 fe80::1ac0:4dff:fec5:ac9d/64 scope link
       valid_lft forever preferred_lft forever
```

récupérer les liens up

```
ip link ls up
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
state DOWN mode DEFAULT group default qlen 1000
    link/ether 74:d0:2b:11:23:f4 brd ff:ff:ff:ff:ff:ff
3: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP mode DEFAULT group default qlen 1000
    link/ether 74:d0:2b:13:6b:57 brd ff:ff:ff:ff:ff:ff
4: lxcbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state
DOWN mode DEFAULT group default qlen 1000
    link/ether 00:16:3e:00:00:00 brd ff:ff:ff:ff:ff:ff
```

```
 ip -4 -o addr show
1: lo    inet 127.0.0.1/8 scope host lo\       valid_lft forever
preferred_lft forever
3: enp2s0    inet 192.168.1.15/24 brd 192.168.1.255 scope global dynamic
noprefixroute enp2s0\       valid_lft 47038sec preferred_lft 47038sec
4: lxcbr0    inet 10.0.3.1/24 brd 10.0.3.255 scope global lxcbr0\
valid_lft forever preferred_lft forever
```

ajouter ou supprimer des routes avec

ip route add et ip route dell

la commande “ifconfig” est maintenant “ ip -s -h -a link

ss

la commande ss consiste à afficher toutes les connexions

```
 ss|less
```

afficher que les connexions TCP, UDP

```
ss -t
```

afficher les échanges non connectés (c’est-à-dire les sessions UDP)

```
ss -ua
```

Pour afficherles sockets TCP à l’écoute :

```
ss -ltn
```

Pour afficher les sockets UDP à l’écoute :

```
ss -lun
```

afficher le nom du processus (ainsi que son PID associé)

```
ss -ltp
```

afficher seulement les sockets en écoute

```
ss -l
```

route

La commande route, tout comme ifconfig sert à la fois à connaître l’état de la table de routage de l’hôte et à configurer de nouvelles routes au besoin.

la commande de net-tools **route -n** de devient avec iproute **ip route show**, elle affiche la table de routage qui réside dans le noyau

```
 route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use
Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    100    0        0
enp2s0
10.0.3.0        0.0.0.0         255.255.255.0   U     0      0        0
lxcbr0
192.168.1.0     0.0.0.0         255.255.255.0   U     100    0        0
```

```
enp2s0
```

```
ip route show
default via 192.168.1.1 dev enp2s0 proto dhcp metric 100
10.0.3.0/24 dev lxcbr0 proto kernel scope link src 10.0.3.1 linkdown
192.168.1.0/24 dev enp2s0 proto kernel scope link src 192.168.1.15 metric
100
```

curl

curl permet de tenter des connexion en plusieurs protocoles, HTTP, FTP, IMAP, LDAP, POP3, SCP, SFTP, SMB, SMTP, pour ne citer qu'eux.

Curl est utilisé dans les lignes de commande ou les scripts pour transférer des données. Il est également utilisé dans les voitures, les téléviseurs, les routeurs, les imprimantes, les équipements audio, les téléphones mobiles, les tablettes, les caisses de settop, les lecteurs multimédias et constitue l'épine dorsale du transfert d'Internet pour des milliers d'applications logicielles qui touchent quotidiennement des milliards d'êtres humains .

trouver son ip public avec curl, en simple user comme en root

```
curl -4 ifconfig.me
```

essayez aussi:ipv6

```
wget -qO- http://ipecho.net/plain ; echo
```

```
curl ipv4.icanhazip.com ou curl ipv6.icanhazip.com
```

```
curl ipv4.icanhazip.com   ou  curl ipv6.icanhazip.com
```

curl v4.ident.me ou curl v6.ident.me

Il est possible de télécharger un fichier ou une page web

```
curl http://www.ploufplouf.fr
```

pour avoir le code source de la page

```
curl http://www.ploufplouf.fr > zem.html
```

les possibilités de curl sont nombreuses: envoyer des mails

télécharger et afficher en console la météo du jour

```
curl -s wttr.in/Toulouse| head -37
```

```
curl wttr.in/Berlin?lang=de
```

```
curl wttr.in/Moon
```

wget

```
apt search wget
wget/stable,now 1.21-1+deb11u1 amd64  [installé]
  récupération de fichiers sur le réseau

wget2/stable 1.99.1-2.2 amd64
  téléchargeur récursif de fichier de site web

wget2-dev/stable 1.99.1-2.2 amd64
  development file for libwget2
```

wget [options] [url]

```
wget
https://cdimage.debian.org/debian-cd/current/i386/iso-cd/debian-11.7.0-i386-
netinst.iso
```

```
wget https://www.samba.org/samba/ftp/samba-latest.tar.gz
```

mode recursif

```
wget -r https://rastacouère.le-chat.org
```

le monitoring réseau

en ligne de commande

etherape

```
apt-get install etherape
```

etherape : est un logiciel libre qui permet de surveiller un réseau informatique, il est muni d'une interface graphique qui permet de visualiser ce qui se passe sur un réseau (local et/ou relié à internet).

Chaque transfert de données est représenté par un trait ainsi qu'un disque de couleur au point d'origine. Les protocoles sont représentés par des couleurs différentes et plus le transfert n'est important plus le disque et le trait sont grands.

EtherApe fait visualiser les transferts par IP de destination ou bien par ports TCP. Il est possible

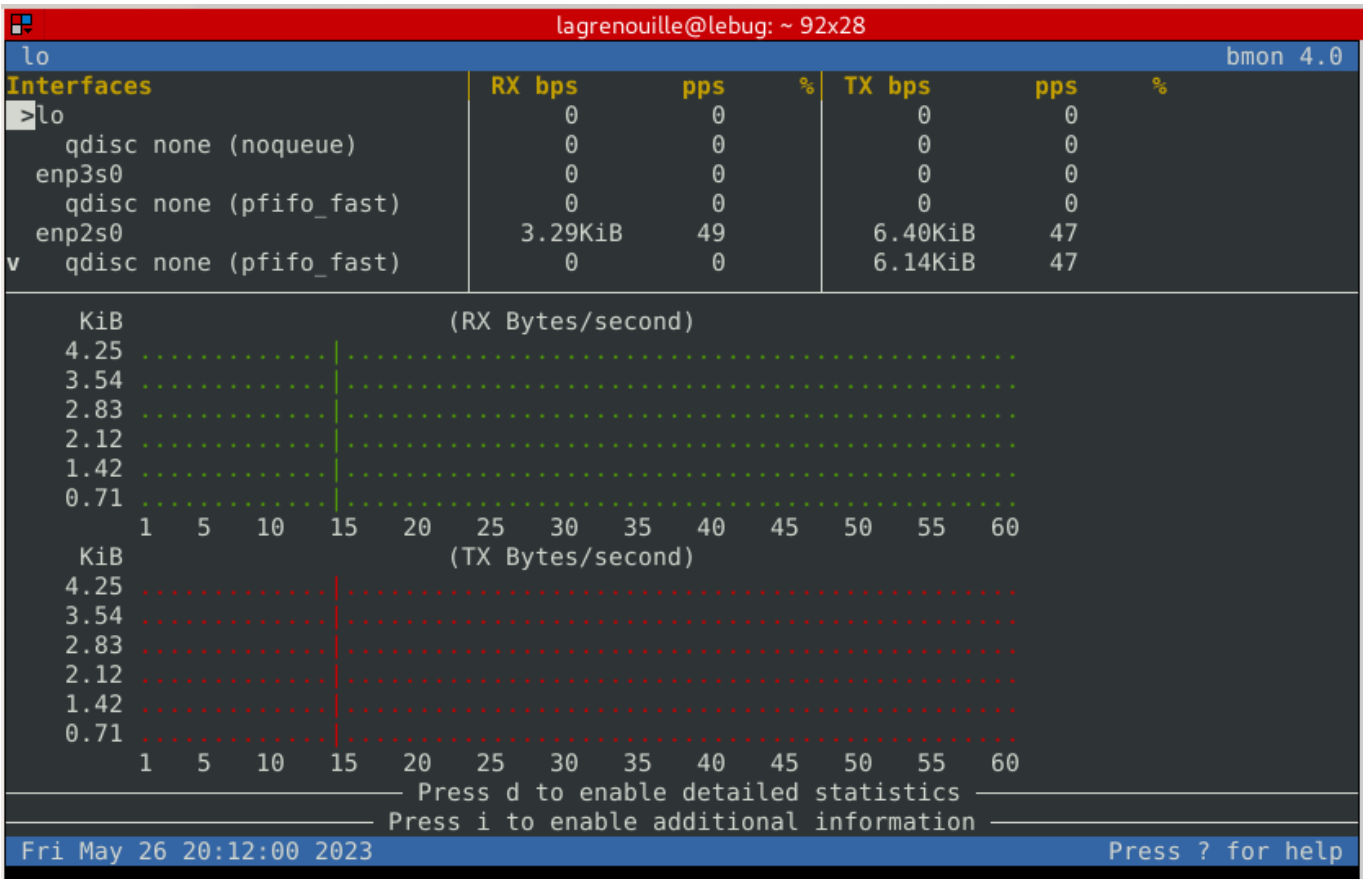
d’enregistrer les activités du réseau afin de les étudier. La destination des transferts d’informations sont affichées soit par son adresse IP soit par l’appellation courante (utilisation d’un serveur DNS).

L’utilisateur peut obtenir des informations supplémentaires sur le transfert (port, origine et destination, taille, date…) s’il clique sur le trait marquant. On peut configurer EtherApe afin de ne visualiser qu’une partie du trafic (par exemple le trafic vers internet seul).

Il est possible d'enregistrer les activités du réseau afin de les étudier après

bmon

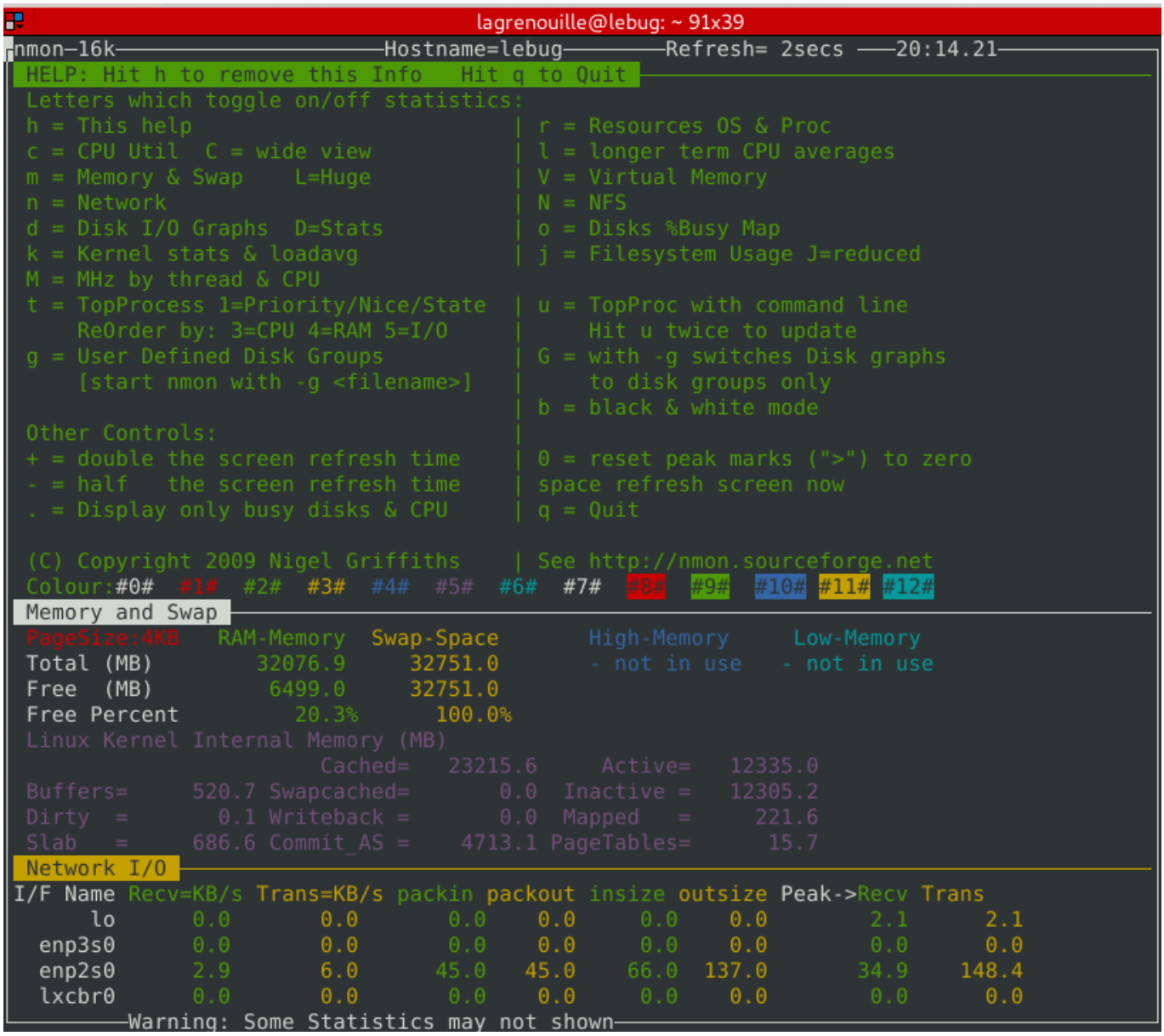
bmon pour une surveillance réseau et bande passante



nmon propose plus de choix

nmon est un outils de surveillance, permettant de monitorer les ressources physiques des machines qui sont sous Linux

nmon peut générer un fichier nmon (ressemblant un peu à du CSV



lsof

lsof est un utilitaire puissant disponible pour les systèmes Linux et Unix qui signifie littéralement «liste (de) fichiers ouverts». il scrute tous les processus en cours d'execution

il existe de nombreuses options, voir man lsof

```
lsof | less
COMMAND      PID    TID TASKCMD               USER   FD      TYPE
DEVICE  SIZE/OFF       NODE NAME
systemd        1                              root  cwd       DIR
8,2      4096          2 /
systemd        1                              root  rtd       DIR
8,2      4096          2 /
systemd        1                              root  txt       REG
8,2   1739200    1192250 /usr/lib/systemd/systemd
```

```
systemd        1                                root  mem       REG
8,2    149576    1178176 /usr/lib/x86_64-linux-gnu/libgpg-error.so.0.29
.0
systemd        1                                root  mem       REG
8,2   3081088    1179815 /usr/lib/x86_64-linux-gnu/libcrypto.so.1.1
systemd        1                                root  mem       REG
8,2     26984    1177679 /usr/lib/x86_64-linux-gnu/libcap-ng.so.0.0.0
systemd        1                                root  mem       REG
8,2    617128    1186396 /usr/lib/x86_64-linux-gnu/libpcre2-8.so.0.10.1
systemd        1                                root  mem       REG
8,2    149520    1182016 /usr/lib/x86_64-linux-gnu/libpthread-2.31.so
```

```
 lsof /var/log/messages
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF   NODE NAME
rsyslogd 1001 root   12w   REG    8,2   167064 550781 /var/log/messages
```

```
lsof -u  lagrenouille
COMMAND      PID         USER   FD      TYPE             DEVICE SIZE/OFF
NODE NAME
systemd     2542 lagrenouille  cwd       DIR                8,2     4096
2 /
systemd     2542 lagrenouille  rtd       DIR                8,2     4096
2 /
systemd     2542 lagrenouille  txt       REG                8,2  1739200
1192250 /usr/lib/systemd/systemd
systemd     2542 lagrenouille  mem       REG                8,2   149576
1178176 /usr/lib/x86_64-linux-gnu/libgpg-error.so.0.29.0
systemd     2542 lagrenouille  mem       REG                8,2  3081088
1179815 /usr/lib/x86_64-linux-gnu/libcrypto.so.1.1
systemd     2542 lagrenouille  mem       REG                8,2    26984
1177679 /usr/lib/x86_64-linux-gnu/libcap-ng.so.0.0.0
```

https://debian-facile.org/doc:systeme:lsof

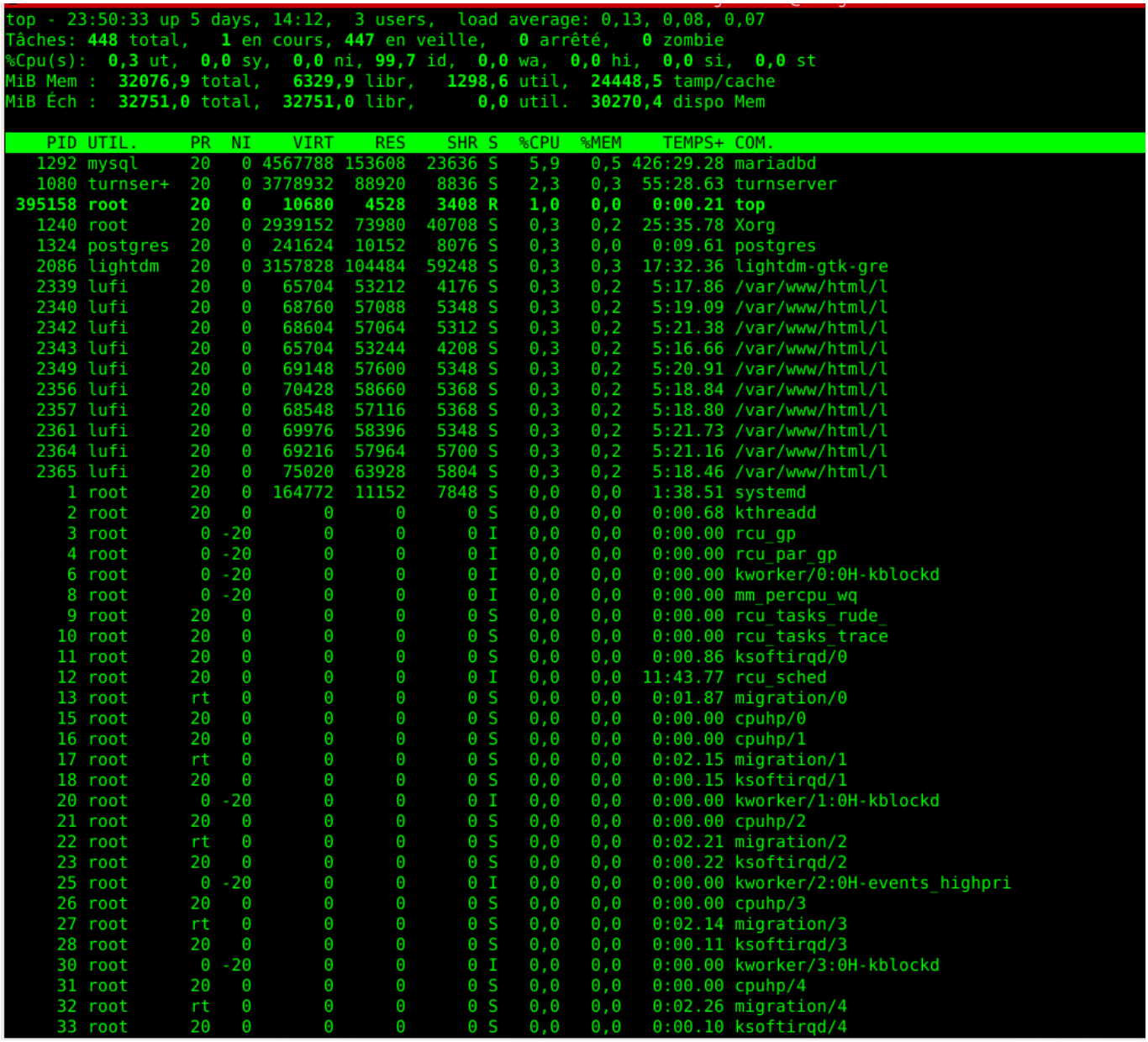
top

affiche les activités du système:

https://debian-facile.org/doc:système:top

La commande `iftop` fait pratiquement la même chose que TOP sauf que là elle surveille le réseau.

https://debian-facile.org/doc:reseau:iftop

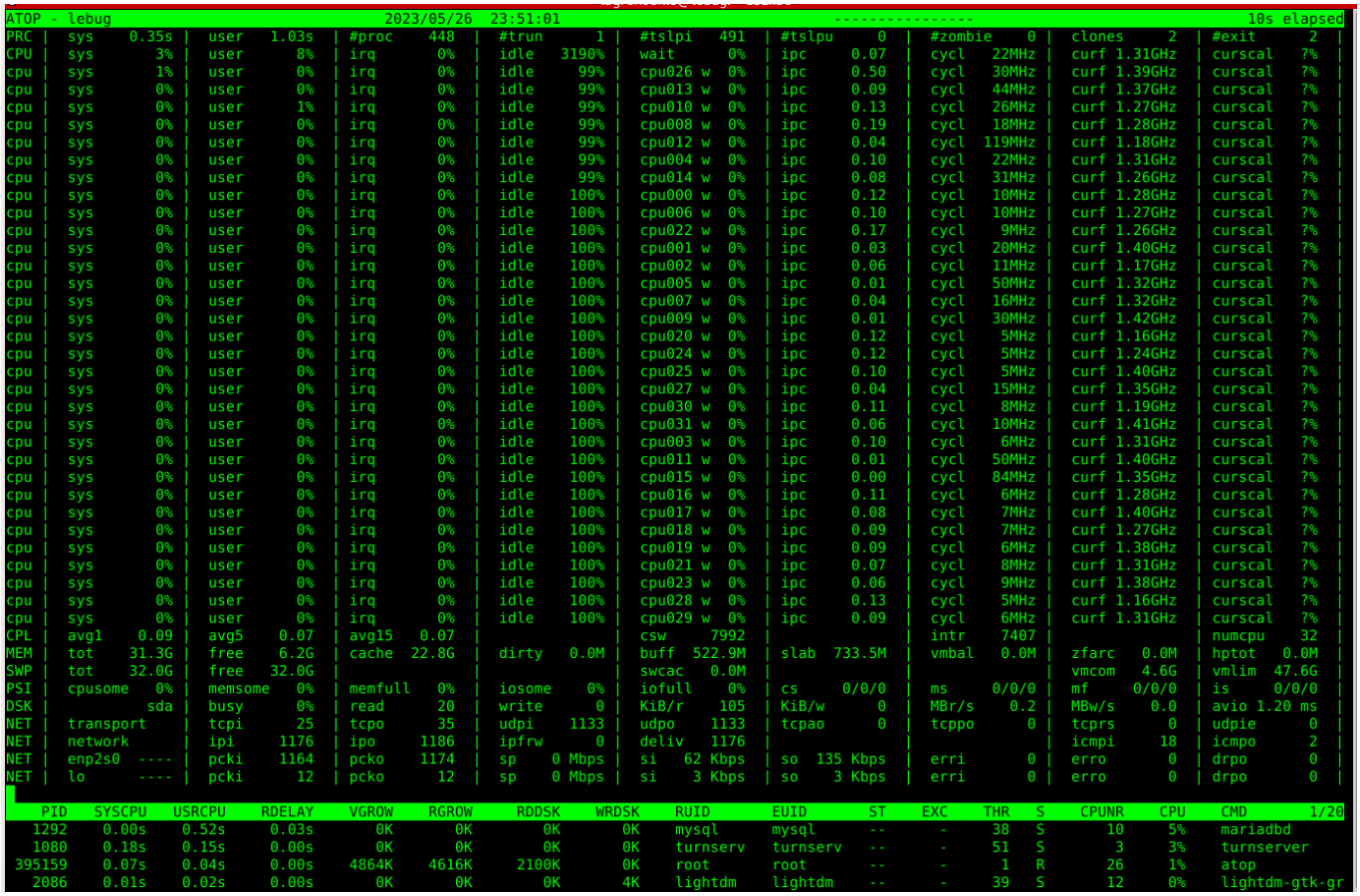


atop

atop est un outil en ligne de commandes interactif pour la supervision de performance sur des systèmes basés sur Linux Vous pouvez récupérer des capacités d'utilisation pour le CPU, la consommation de mémoire et les I/O disque, pour chaque processus et thread.

L'outil atop reste actif en tant que service d’arrière-plan tout en enregistrant les statistiques

Voir man atop https://linux.die.net/man/1/atop



snort

référence et infos prises sur le site :https://all-it-network.com/snort/

Snort est basé sur la bibliothèque de capture de paquets (libpcap). Libpcap est un outil largement utilisé dans les renifleurs de trafic d'adresses de protocole de contrôle de transmission/protocole Internet, les chercheurs et les analyseurs de contenu pour l'enregistrement des paquets, l'analyse du trafic en temps réel, l'analyse des protocoles et la correspondance du contenu.

Mode système de prévention des intrusions.

En tant que système de prévention des intrusions réseau open source, Snort surveille le trafic réseau et le compare à un ensemble de règles Snort défini par l'utilisateur. Il s'agit de la fonction la plus importante de Snort.

NIDS (Network Intrusion Detection System) Il capture tout le trafic du réseau (sniffer) en temps réel. Il se base sur des règles qui lui ont été définies pour pouvoir détecter des comportements suspects. Il sert à détecter un comportement anormal sur le réseau.

HIDS (Host Intrusion Detection System) sert à détecter un comportement anormal sur une machine. Il collecte les informations qui lui sont envoyées par les équipements. Il utilise les signatures ou le comportement. Un agent est installé sur chacune des machines. Un HIDS va ensuite envoyer les informations au HIDS qui va analyser les signatures et les comportements.

IDS hybride qui permet de détecter les intrusions sur les hôtes et sur le réseau.

IDS/IPS aspirent tout le trafic, y compris personnelles de vos utilisateurs si vous en avez, vont être

analysées par l'IDS/IPS. attention aux lois en vigueurs.

Snort peut fonctionner dans 3 modes différents:

1) Sniffer: permet d'observer les paquets reçus

2) Log de paquet: pour archiver les logs du réseau

3) IDS: génération d'alerte en fonction des comportements du réseau

Avant de commencer l'installation de SNORT, vous devez avoir installé : apache, mysql-client, php-mysql, mod_php

```
apt-get install snort
```

il faut renseigner l'interface sur laquelle l'outil écoutera le réseau..

enp3s0, enp2s0, enx803f5d108461, eth .....

`ip addr show` ou ip a ou `ifconfig -a` vous affichent vos interfaces

Snort ne permet pas d'envoyer de mail directement, et c'est dommage.😎

.

Si vous lancez cette commande, chaque intervention sur votre LAN provoquera un Warning, ping etc..

remplacez enp2s0 par votre interface

```
snort -A console -i enp2s0 -u snort -c /etc/snort/snort.conf
```

voin le man pour plus d'infos, j'ai pas trop approfondi cette commande 🤨

Wireshark

Wireshark analyse les paquets qui transitent sur le réseau.

Il capture chaque paquet entrant ou sortant d'une interface réseau et les affiche dans un fichier texte .

Il est utilisé pour le dépannage des réseaux, l'analyse, le développement de logiciels et de protocoles de communication

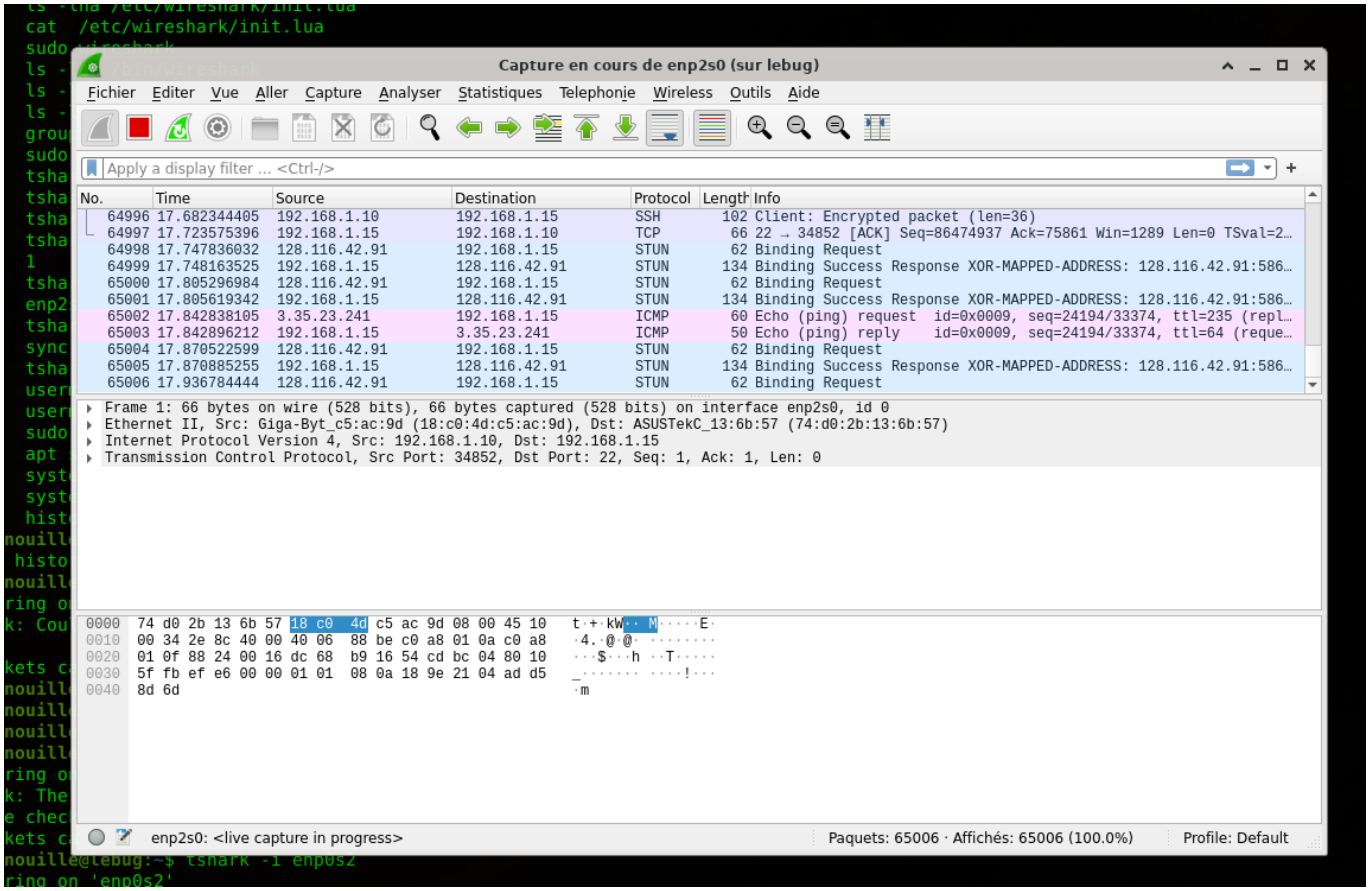
apt install wireshark wireshark-qt tshark

comme j'ai répondu non à l'installation sur l'attribution des droits pour user je reviens sur ce choix

```
<code user>sudo dpkg-reconfigure wireshark-common
```

je réponds "oui"</code>

```
sudo chmod +x /usr/bin/dumpcap
```



explication des commande, pris sur :https://www.it-connect.fr/decouverte-de-linterface-de-wireshark/

Num	le numéro de paquet en sachant que le 1er paquet capturé à le numéro 1 et ainsi de suite
Time	le temps écoulé entre le moment où Wireshark a capturé le paquet et le moment où l'on a démarré la capture (en secondes par défaut)
Source	adresse IP source qui a envoyé le paquet
Destination	adresse IP qui va recevoir ou a reçu le paquet
Protocol	protocole utilisé (DNS, TCP, TLS, SSH....)
Length	taille du paquet (entête protocolaire + données transportées)
Info	sur le paquet comme le port TCP, la requête applicative....

Outils en CLI _ Des Commandes DNS
https://debian-facile.org/utilisateurs:lagrenouille:tutos:quelques-commandes-dns#utilisation

adminer https://debian-facile.org/doc:reseau:serveur:adminer

IRC doc:reseau:irc-fichiers