

SAE: S2.03 INSTALLATION DE SERVICES RÉSEAUX

PARTIE 1: CRÉATION DU SERVEUR WEB EN LOCAL

SOMMAIRE

I - [NOTICE](#)

II - [DOCUMENTATION](#)

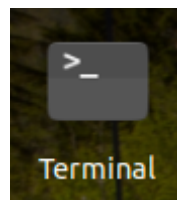
AUTEURS: [IDRISSA SALL](#)

BOUDKDIHT AMINE

Groupe: G1S2

I - NOTICE

Dans les paragraphes qui suivent, vous apprendrez à configurer un serveur web Apache local sur le port 80. Nous utilisons Linux pour arriver au bout du manuel d'utilisateur. Vous n'avez pas besoin d'être fort sur les commandes de Linux, on vous expliquera tout en détail. Alors pour commencer, rien de plus. Ouvrez le terminal de Linux. Ce terminal nous permettra de taper des lignes de commande.

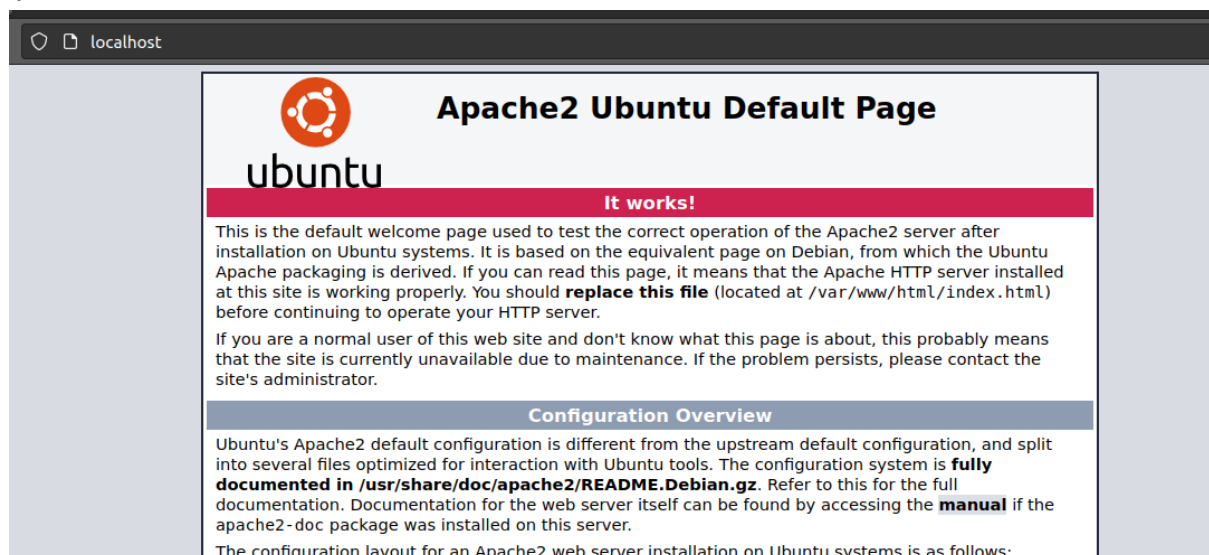


Une fois que vous avez ouvert le terminal, vous allez tomber sur ça

```
idrisa@ubuntu:~$
```

Vous êtes maintenant en ligne de commande. Pour vérifier que cela fonctionne, tapez la commande ls et puis Entrée. Le résultat que vous avez devrait être très familier pour vous. La commande affiche les caractéristiques d'une liste de fichiers.

On entre maintenant dans le vif du sujet. On va voir comment installer Apache2. Pour votre culture générale, Apache est un logiciel de serveur web gratuit et open-source qui alimente environ 46% des sites web à travers le monde. On va installer la dernière version d'apache2. Saisissez en ligne de commande sudo apt update et sudo apt install apache2. Une fois que vous avez fait ça, tous les packages nécessaires, pour créer notre serveur local, sont réunis. Et on peut les tester en se rendant sur notre application de navigateur *Firefox*. Tapez localhost sur la barre de recherche. Voici ce qui est attendu



Si vous voyez cette page, cela signifie qu'Apache a été installé avec succès sur votre serveur. Oui, je sais que vous aimeriez voir votre page au lieu de la page par défaut de Apache. Mais pour pouvoir le faire il faut donc comprendre encore quelques notions. On commence d'abord par configurer Apache. Ce qui veut dire

trouver les fichiers de configuration. Il faut savoir qu'Apache respecte l'arborescence Linux. Et dans l'arborescence Linux c'est le répertoire /etc qui gère les fichiers de configurations du système informatique. Alors oui c'est là-bas qu'on va se rendre pour configurer apache2. En ligne de commande, cd /etc/apache2 et ensuite ls comme je vous ai montré ci-dessus. Notez que la commande cd permet de changer de répertoire courant.

```
idrisa@ubuntu:~$ cd /etc/apache2
idrisa@ubuntu:/etc/apache2$ ls
apache2.conf  conf-available  conf-enabled  envvars  magic  mods-available
mods-enabled  ports.conf  sites-available  sites-enabled
idrisa@ubuntu:/etc/apache2$
```

On peut voir que Apache2 utilise 3 types de configurations:

- les "conf-available": fichiers décrivant une configuration qui s'applique à tous les domaines.
- les "conf-enabled": contient des liens symboliques vers liens configurations, dans conf-available, des autres services activés.
- les "mods-available": ajout de fonctionnalités au serveur HTTP.
- les "mods-enabled": contient des liens symboliques vers liens configurations, dans mods-available, des modules activés.
- les "sites-available": fichiers décrivant vos sites pour un nom de domaine précis.
- les "sites-enabled": contient des liens symboliques vers liens configurations, dans sites-available, des sites activés.

En ligne de commande, cat ports.conf

```
idrisa@ubuntu:/etc/apache2$ cat ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
idrisa@ubuntu:/etc/apache2$
```

Vous vous en doutez certes mais cat affiche le contenu d'un fichier texte. Listen 80 est le numéro de port par défaut d'apache 2. Si on souhaite changer ou ajouter un autre port c'est d'ici qu'il faudra le faire. Si vous souhaitez ajouter le port 8080, il faut ajouter la ligne suivante: Listen 8080 au-dessous de Listen 80. Cependant, il faut souligner que quand vous voulez changer de port, vous préciserez le numéro dans l'URL après le nom du serveur et le symbole " : ". On va maintenant créer notre propre site web. C'est-à-dire modifier la page par défaut d'apache 2 et mettre notre

site web à la place. C'est normal que vous vous demandiez où se trouve cette page par défaut d'apache2. Et comment la modifier. Comme le répertoire /etc, il y a aussi le répertoire /var. Ce dernier gère les données fréquemment utilisées et réécrites. En ligne de commande, cd /var/www et puis ls. Vous avez devant vous un répertoire /html. Pour changer de répertoire, saisissez cd html et puis ls. Vous avez là la page web par défaut de apache2. Pour voir le contenu, rien de plus simple. cat index.html.

```
idrisa@ubuntu:~$ cd /var/www
idrisa@ubuntu:/var/www$ ls
html
idrisa@ubuntu:/var/www$ cd html
idrisa@ubuntu:/var/www/html$ ls
index.html
idrisa@ubuntu:/var/www/html$ cat index.html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2016-11-16
    See: https://launchpad.net/bugs/1288690
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }

  body, html {
```

NOTE ???

C'est le grand moment pour mettre en place votre premier site web.

Ce que nous allons faire c'est de faire tout ça par nous même au lieu de modifier les fichiers par défaut de apache2. Pour cela nous allons commencer par créer un répertoire mkdir /var/www/monsie. La commande mkdir permet de créer un répertoire. Ici nous l'avons nommé "monsie". Vous pouvez nommer votre répertoire selon le nom qui vous convient. Pour vérifier que ce répertoire a été correctement créé, cd /var/www et puis ls.

```
idrisa@ubuntu:~$ sudo mkdir /var/www/monsie
idrisa@ubuntu:~$ cd /var/www/
idrisa@ubuntu:/var/www$ ls
html  monsie
idrisa@ubuntu:/var/www$
```

Si vous avez le résultat ci-dessus, c'est que la création du répertoire a bien fonctionné. Si vous vous déplacez dans ce répertoire, vous verrez qu'il n'y a rien. Il est donc temps de créer votre première page. Pour cela je dois vous faire un petit rappel. UNIX est un système d'exploitation multi-utilisateurs et chaque utilisateur a ses propres données. Donc pour chaque fichier, il y a des droits c'est-à-dire

autorisation pour les utilisateurs. Et c'est la Commande chmod qui permet de modifier les droits. Notez qu'il y a trois types de droit: le droit de lecture, le droit d'écriture et le droit d'exécution. Nous allons donc nous donner tous les droits pour pouvoir modifier nos fichiers.

En ligne de commande sudo chmod 777 -R /var/www.

sudo: vous donne les droits d'administrateur.

777 est un code octal. Cela vous permet d'avoir tous les droits nécessaires.

-R est une option de la commande chmod qui veut dire de manière récursive.

Une fois que vous avez tous les droits, on se dirige vers le répertoire monsite avec cd /var/www/monsite. Nous allons donc créer notre première page avec la commande nano index.html. La commande nano permet d'éditer un fichier. Et si ce fichier n'existait pas avant, alors il est créé automatiquement. Vous pouvez mettre quelques balises html dans le fichier ou bien vous pouvez recopier l'exemple ci-après. Une fois que vous avez fini, sauvegarder le fichier avec "Ctrl + s" et ensuite "Ctrl + x"

Ici par exemple, notre répertoire "monsite" était vide. Mais avec cette commande, il y a maintenant le fichier index.html

```
idrisa@ubuntu:~$ sudo chmod 777 -R /var/www
idrisa@ubuntu:~$ cd /var/www/monsite
idrisa@ubuntu:/var/www/monsite$ nano index.html
idrisa@ubuntu:/var/www/monsite$ ls
index.html
idrisa@ubuntu:/var/www/monsite$ cat index.html
<html>
<head>
  <title>mon site avec apache2 </title>
</head>
  <p>Je suis le site que vous avez créé. je suis en serveur local </p>
</html>
idrisa@ubuntu:/var/www/monsite$
```

Ce qui nous reste maintenant c'est de configurer le VirtualHost. Comme je vous l'ai dit au début, il faut donc se rendre dans /etc/apache2 pour le faire.

En ligne de commande, cd /etc/apache2

```
idrisa@ubuntu:~$ cd /etc/apache2
idrisa@ubuntu:/etc/apache2$ ls
apache2.conf  conf-available  conf-enabled  envvars  magic  mods-available
  mods-enabled  ports.conf  sites-available  sites-enabled
idrisa@ubuntu:/etc/apache2$
```

Si vous vous souvenez je vous avais dit que site-availables: c'est pour fichiers décrivant vos sites pour un nom de domaine précis. On va donc s'y rendre

En ligne de commande, cd site-available

```
idrisa@ubuntu:/etc/apache2$ cd sites-available/
idrisa@ubuntu:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf
idrisa@ubuntu:/etc/apache2/sites-available$
```

000-default.conf est le fichier de configuration par défaut de apache2. Ce qui important dedans c'est le "serverAdmin" où vous devez mettre votre mail pour que

apache2 puisse vous avertir en cas d'erreur et le "DocumentRoot" qui indique le chemin du répertoire de votre page web. Dans notre exemple monsite

```
idrisa@ubuntu:/etc/apache2/sites-available$ cat 000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
```

Pour créer notre fichier nous allons faire en ligne de commande, sudo cp 000-default.conf monsite.conf

La commande cp permet de copier un fichier vers un autre. Si le fichier n'existe pas, il le crée automatiquement. monsite.conf est utilisé pour notre configuration. Ce qu'on va faire maintenant c'est de modifier le "serverAdmin" et "DocumentRoot" avec la commande nano, sudo nano monsite.conf

```
ServerAdmin yourmail@mail.com
DocumentRoot /var/www/monsite
ServerName monsite.com

<Directory "/var/www/monsite" >
    Options +FollowSymLinks
    AllowOverride all
    Require all granted
</Directory>
```

Et vous ajouter en plus le ServerName qui n'est pas venu avec le fichier par défaut. Vous avez maintenant terminé de configurer votre site. Ajouter aussi la balise <directory> comme dans l'exemple. Après ceci, modifier le fichier default-ssl, sudo nano default-ssl.conf. et changer le documentRoot et le serverAdmin.

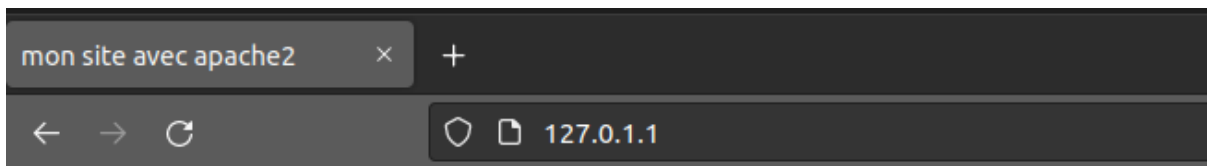
En ligne de commande, sudo a2ensite monsite.conf

```
idrisa@ubuntu:/etc/apache2/sites-available$ sudo a2ensite monsite.conf
Enabling site monsite.
To activate the new configuration, you need to run:
    systemctl reload apache2
idrisa@ubuntu:/etc/apache2/sites-available$ sudo systemctl reload apache2
```

a2ensite permet d'activer la configuration du site.

sudo systemctl reload apache2 est une commande pour recharger la configuration. Vous ne vous en rendez pas compte mais vous venez de mettre votre site sur un serveur web local. Avant de vous dévoiler le résultat, saisissez d'abord sudo systemctl restart apache2 pour relancer apache2.

Pour voir le résultat, retournez sur *Firefox* et tapez sur la barre de recherche 127.0.1.1



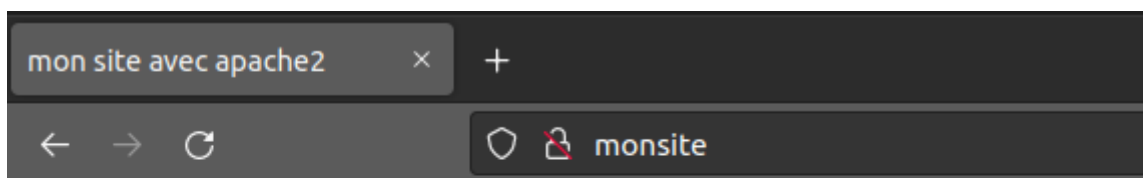
Je suis le site que vous avez créé. je suis en serveur local

Pour aller plus loin, vous pouvez remplacer l'adresse IP par le nom de notre site "monsite". Il faut retourner dans le répertoire /etc et modifier le fichier hosts avec la commande sudo nano hosts. Et vous faites exactement comme j'ai fait, écrire le nom que vous voulez devant l'adresse IP.

```
GNU nano 4.8
127.0.0.1    localhost
127.0.1.1    monsite

# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

Voici le résultat attendu.



Je suis le site que vous avez créé. je suis en serveur local

Maintenant que nous avons vu les configurations de base d'apache 2 ça devient plus facile pour la suite. Si vous avez déjà votre propre site web et que vous voulez le mettre sur le serveur local, vous placez votre dossier ici

```
idrisa@ubuntu:~$ cd /var/www ; ls
html  monsite
idrisa@ubuntu:/var/www$
```

dans le répertoire de /www. veuillez renommer votre première page en index.html Ensuite vous créez votre fichier de configuration comme je vous l'ai montré


```
ldrissa@ubuntu:/var/www$ cd /etc/apache2/sites-available/
ldrissa@ubuntu:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf  monsite.conf
ldrissa@ubuntu:/etc/apache2/sites-available$
```

Il ne reste plus qu'à l'activer avec a2ensite et relancer apache.

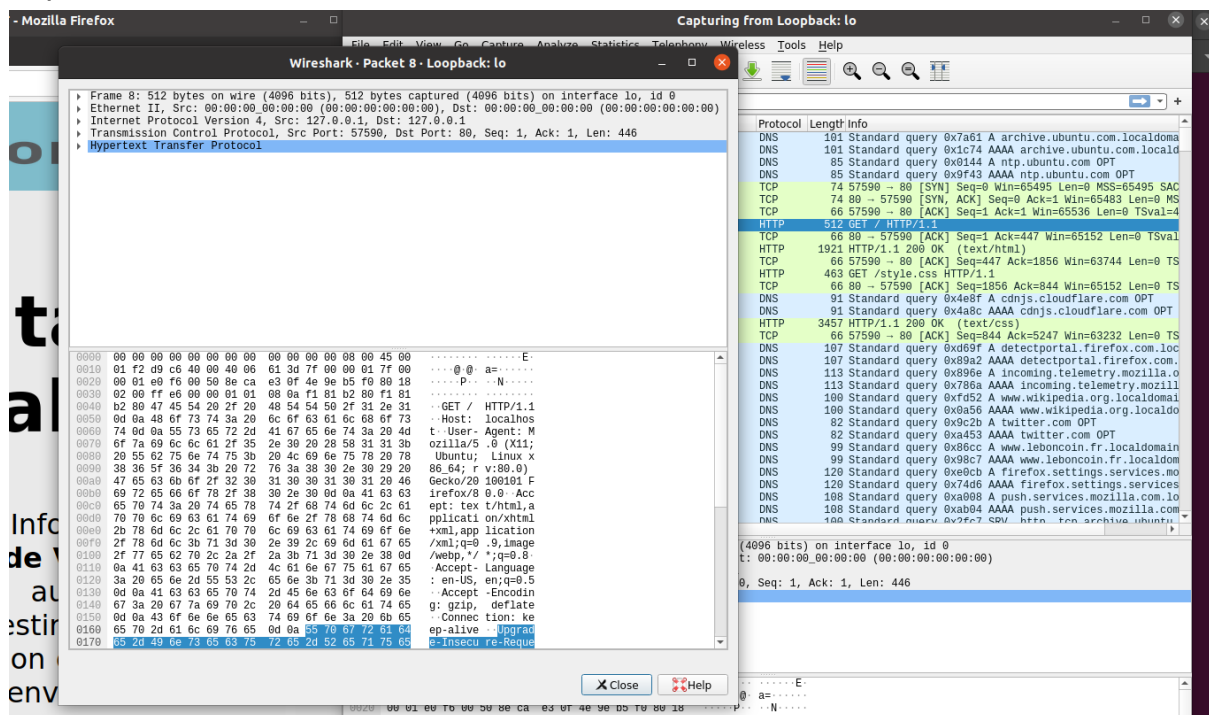
Ce manuel est à présent terminé. Nous avons vu comment configurer un serveur web local. Nous avons installé apache avec la commande sudo apt install apache2. Pour configurer apache2 nous avons fait cd /etc/apache2. Une fois que cela est fait, nous avons activé notre site avec la commande sudo a2ensite monsite.conf. Et pour le résultat nous avons tapé l'adresse IP 127.0.1.1. Nous avons aussi vu dans /etc/hosts comment donner un nom à notre site web.

II - DOCUMENTATION

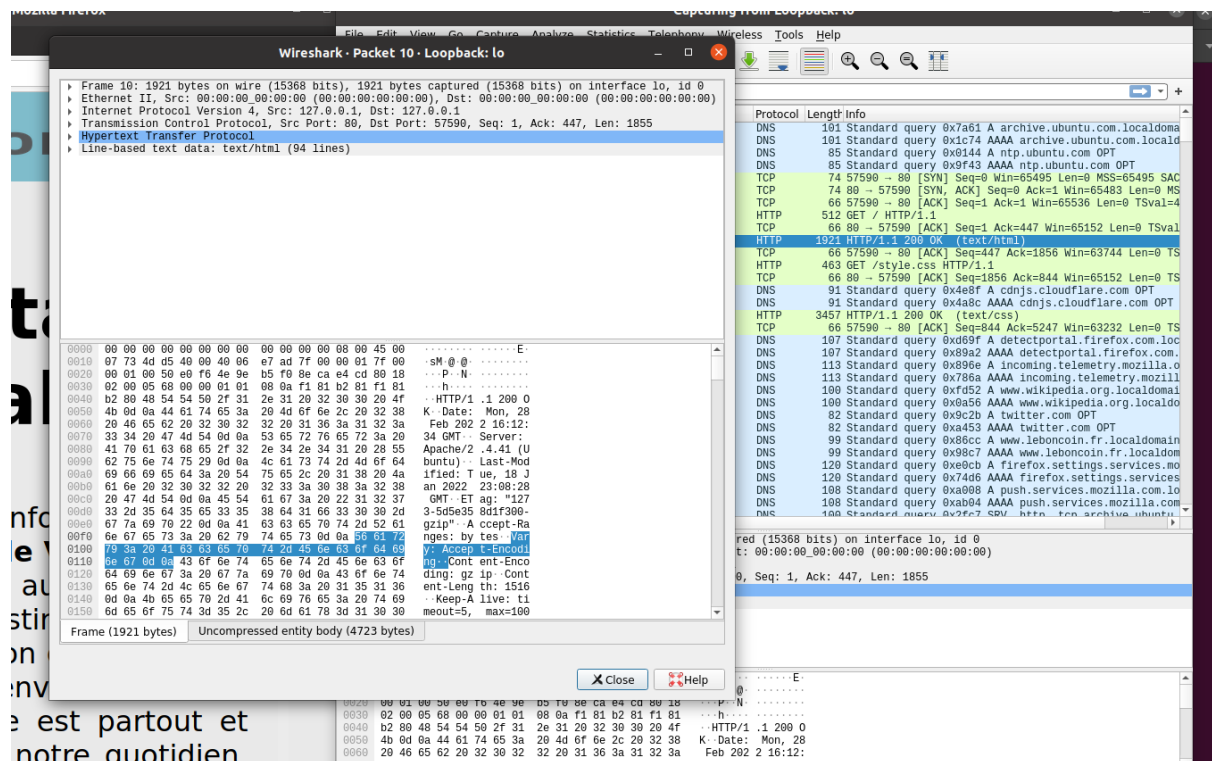
Avec la commande sudo wireshark nous avons réussi à faire des captures de trame sur les ports 80 et 8080.

1- Pour le port 80

Capture GET / HTTP/1.1



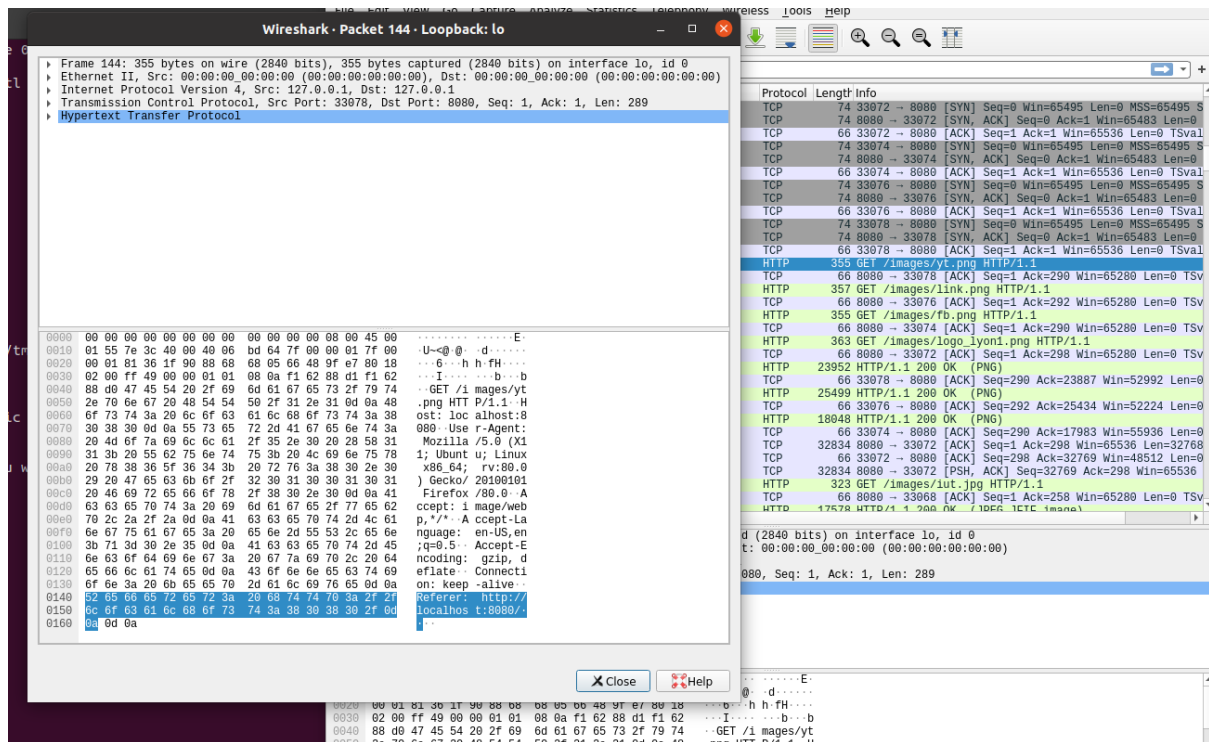
Capture HTTP/1.1 200 ok (text/html)



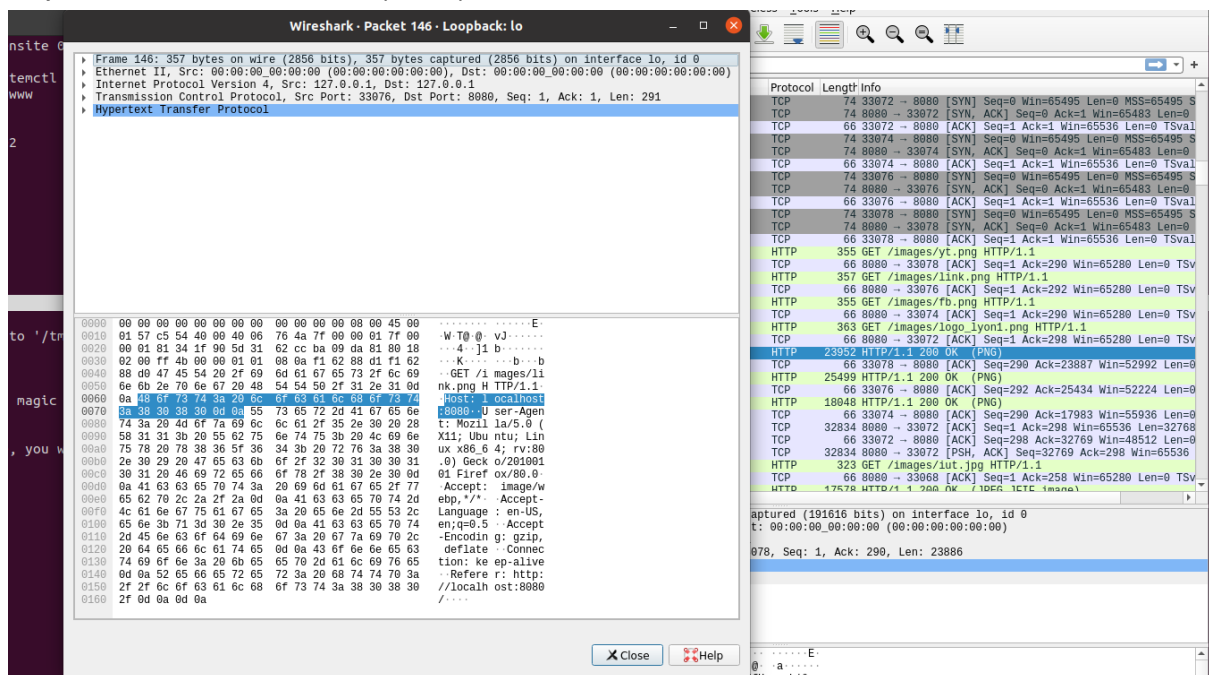
La capture avec GET a une taille de 512 bytes, qui est moins importante que celle de HTTP qui est de 1921. Il n'y a pas de différence sur la première couche OSI(physique), aucune adresse “*trame Ethernet*” (00:00:00:00:00:00). Au niveau du “*paquet IP*” qui est la deuxième couche du modèle OSI, on note pas de différence avec une version 4. L’adresse IP étant 127.0.0.1 pour la source que pour la destination. Sur le “*segment TCP*” (troisième couche OSI), l’adresse Src port est de 80 et Dst port 57599 pour la capture HTTP, c’est l’inverse pour la capture avec GET. Et enfin nous avons le “*message HTTP*” de la couche applications, qui n’est pas le même pour les deux captures.

2- Pour le port 8080

Capture GET /images/yt.png HTTP/1.1



Capture HTTP/1.1 200 ok (PNG)



Comme pour le port 80, le port 8080 présente une différence de taille entre les deux captures au niveau du “Frame”. Aucun changement sur la “trame Ethernet” et identique à celui du port 80. Il en est de même pour les deux au niveau du “paquet IP” avec une version 4 qui est notée. L’adresse IP est toujours 127.0.0.1 pour la Src port et la Dst port. Sur le “segment TCP” (troisième couche OSI), l’adresse Src port est de 33076 et Dst port 8080 pour la capture HTTP, c’est l’inverse pour la capture

avec GET. Et enfin sur la couche application, le "*message HTTP*" est différent pour les deux captures.

Pour les modifications, il y a évidemment les Src port et Dst port qui changent entre le port 8080 et le port 80. Cependant, la taille au niveau du Frame est moins importante dans le port 8080 que celui du port 80.