Compte rendus Activité NAT PAT

Mise en place d'un raspberry PI

1) Changer l'interface réseau du raspberry en 192.168.10.1/24, DNS 172.16.255.254, passerelle avec l'interface graphique de celui-ci.

2) Vérification de l'installation et du fonctionnement du serveur ssh en avec la commande systemctl status ssh.

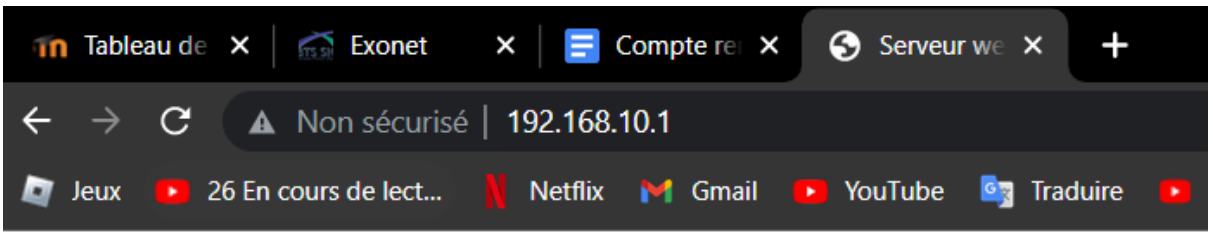
3) Installation du serveur web avec Apache avec la commande sudo apt install appache2

```
pi@raspberrypi:~ $ sudo apt install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
apache2 est déjà la version la plus récente (2.4.54-1~deb11u1).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  libfuse2 libva-wayland2
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 122 non mis à jour.
pi@raspberrypi:~ $
```

Pour tester mon serveur web je dois d'abord créer un fichier html avec la commande nano var/www/html/index.html

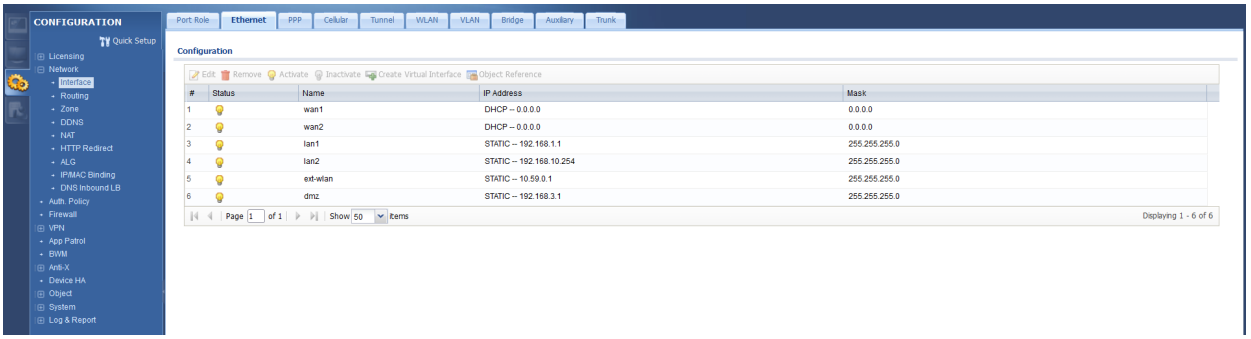
```
 GNU nano 5.4                                  /var/www/html/index.html
<!DOCTYPE html>
<html>
<head>
<title> Serveur web </title>
<meta charset="UTF-8">
</head>

<body>
<h1>Martin Noah </h1>
</body>
</html>
```

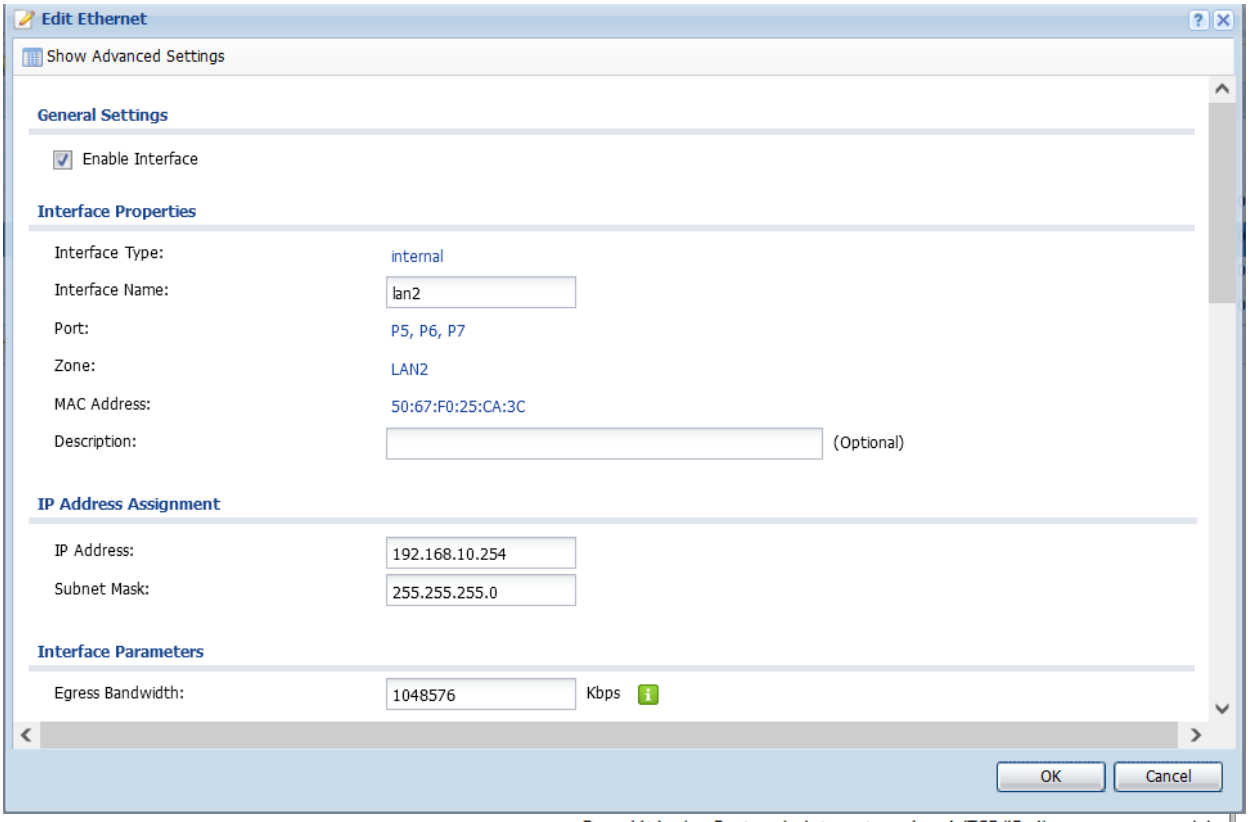
j'utilise un navigateur web et tape l'adresse 192.168.10.1



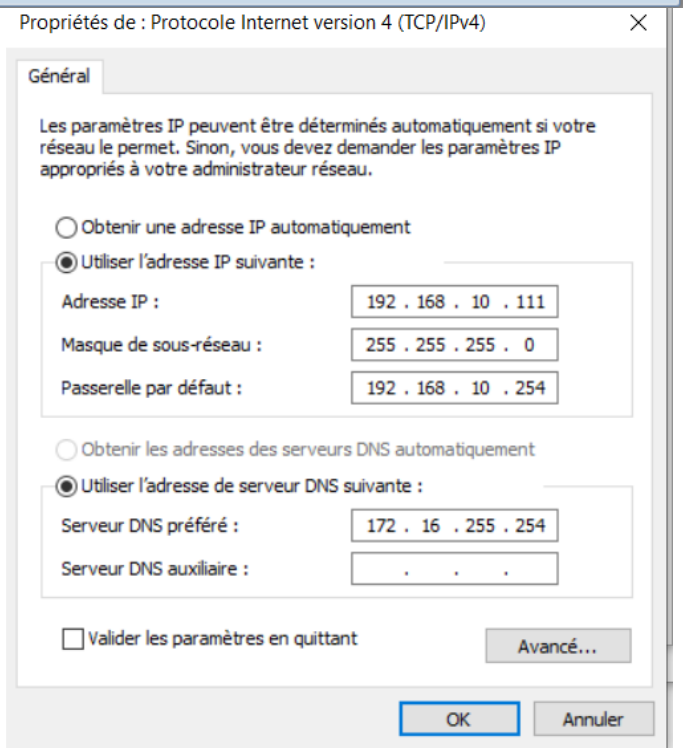
Martin Noah

Configuration du routeur en .254





Configuration ip statique du client



Page documentation pour configurer le routeur en NAT.

5.6 How to Allow Public Access to a Web Server

This is an example of making an HTTP (web) server in the DMZ zone accessible from the Internet (the WAN zone). In this example you have public IP address 1.1.1.1 that you will use on the WAN interface and map to the HTTP server's private IP address of 192.168.3.7.

Figure 45 Public Server Example Network Topology

5.6.1 Configure NAT

Create a NAT rule to send HTTP traffic coming to WAN IP address 1.1.1.1 to the HTTP server's private IP address of 192.168.3.7.

1 Click **Configuration > Network > NAT > Add > Create New Object > Address** and create an IPv4 host address object named DMZ_HTTP for the HTTP server's private IP address of 192.168.3.7. Repeat to create a host address object named Public_HTTP_Server_IP for the public WAN IP address 1.1.1.1.

2 Configure the NAT rule.

For the **Incoming Interface** select the WAN interface.

Set the **Original IP** to the **Public_HTTP_Server_IP** object and the **Mapped IP** to the **DMZ_HTTP** object.

HTTP traffic and the HTTP server in this example both use TCP port 80. So you set the **Port Mapping Type** to **Port**, the **Protocol Type** to **TCP**, and the original and mapped ports to 80.

Keep **Enable NAT Loopback** selected to allow users connected to other interfaces to access the HTTP server.

Diagnostic du réseau.

1) Installation des logiciels nmap, tcpdump, netcat

sudo apt install …

Analyse du trafic.

1) Scan des ports ouvert depuis le post client LAN avec nmap

nmap -p- 192.168.10.1

Voici l'ensemble des ports ouvert :

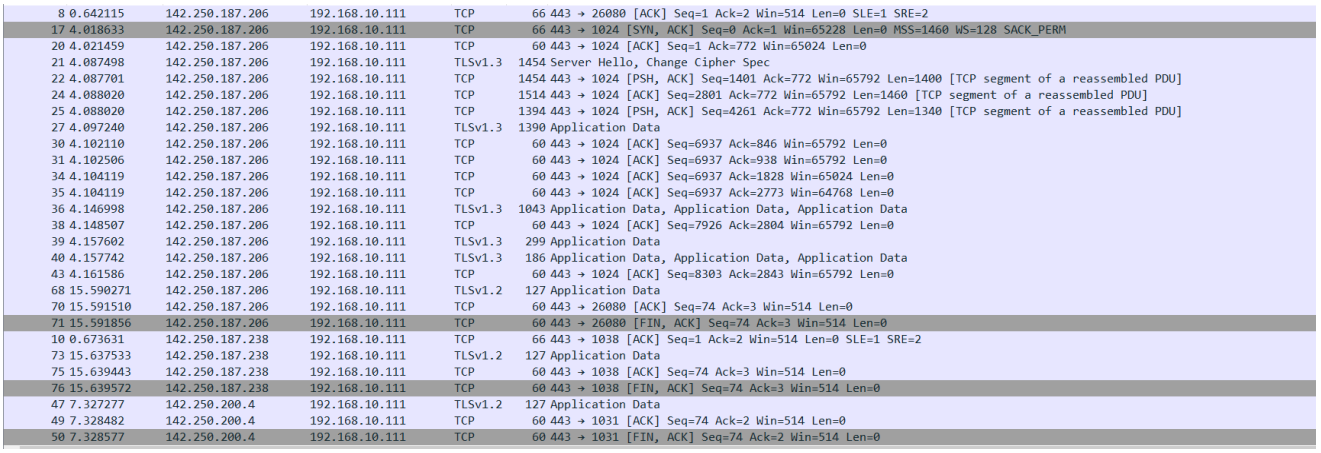
```
pi@raspberrypi:~ $ nmap -p- 192.168.10.1
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-24 09:19 CET
Nmap scan report for 192.168.10.1
Host is up (0.0015s latency).
Not shown: 65530 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
5900/tcp open  vnc

Nmap done: 1 IP address (1 host up) scanned in 12.48 seconds
```

Commande pour se connecter au raspberry : ssh pi@192.168.10.1 par la suite on nous demande le mot de passe pour se connecter à la session. Le mot de passe est raspberry

Analyse de la commande #tcpdump -vv -w dump-server-file.pcap dst 192.168.10.1 and dst port 80

Les options -vv demandent une sortie détaillée et une verbosité élevée.

Analyse du trafic local avec wireshark :



8	0.642115	142.250.187.206	192.168.10.111	TCP	66 443 → 26080 [ACK] Seq=1 Ack=2 Win=514 Len=0 SLE=1 SRE=2
17	4.018633	142.250.187.206	192.168.10.111	TCP	66 443 → 1024 [SYN, ACK] Seq=0 Ack=1 Win=65228 Len=0 MSS=1460 WS=128 SACK_PERM
20	4.021459	142.250.187.206	192.168.10.111	TCP	60 443 → 1024 [ACK] Seq=1 Ack=772 Win=65024 Len=0
21	4.087498	142.250.187.206	192.168.10.111	TLSv1.3	1454 Server Hello, Change Cipher Spec
22	4.087701	142.250.187.206	192.168.10.111	TCP	1454 443 → 1024 [PSH, ACK] Seq=1401 Ack=772 Win=65792 Len=1400 [TCP segment of a reassembled PDU]
24	4.088020	142.250.187.206	192.168.10.111	TCP	1514 443 → 1024 [ACK] Seq=2801 Ack=772 Win=65792 Len=1460 [TCP segment of a reassembled PDU]
25	4.088020	142.250.187.206	192.168.10.111	TCP	1394 443 → 1024 [PSH, ACK] Seq=4261 Ack=772 Win=65792 Len=1340 [TCP segment of a reassembled PDU]
27	4.097240	142.250.187.206	192.168.10.111	TLSv1.3	1390 Application Data
30	4.102110	142.250.187.206	192.168.10.111	TCP	60 443 → 1024 [ACK] Seq=6937 Ack=846 Win=65792 Len=0
31	4.102506	142.250.187.206	192.168.10.111	TCP	60 443 → 1024 [ACK] Seq=6937 Ack=938 Win=65792 Len=0
34	4.104119	142.250.187.206	192.168.10.111	TCP	60 443 → 1024 [ACK] Seq=6937 Ack=1828 Win=65024 Len=0
35	4.104119	142.250.187.206	192.168.10.111	TCP	60 443 → 1024 [ACK] Seq=6937 Ack=2773 Win=64768 Len=0
36	4.146998	142.250.187.206	192.168.10.111	TLSv1.3	1043 Application Data, Application Data, Application Data
38	4.148507	142.250.187.206	192.168.10.111	TCP	60 443 → 1024 [ACK] Seq=7926 Ack=2804 Win=65792 Len=0
39	4.157602	142.250.187.206	192.168.10.111	TLSv1.3	299 Application Data
40	4.157742	142.250.187.206	192.168.10.111	TLSv1.3	186 Application Data, Application Data, Application Data
43	4.161586	142.250.187.206	192.168.10.111	TCP	60 443 → 1024 [ACK] Seq=8303 Ack=2843 Win=65792 Len=0
68	15.590271	142.250.187.206	192.168.10.111	TLSv1.2	127 Application Data
70	15.591510	142.250.187.206	192.168.10.111	TCP	60 443 → 26080 [ACK] Seq=74 Ack=3 Win=514 Len=0
71	15.591856	142.250.187.206	192.168.10.111	TCP	60 443 → 26080 [FIN, ACK] Seq=74 Ack=3 Win=514 Len=0
10	0.673631	142.250.187.238	192.168.10.111	TCP	66 443 → 1038 [ACK] Seq=1 Ack=2 Win=514 Len=0 SLE=1 SRE=2
73	15.637533	142.250.187.238	192.168.10.111	TLSv1.2	127 Application Data
75	15.639443	142.250.187.238	192.168.10.111	TCP	60 443 → 1038 [ACK] Seq=74 Ack=3 Win=514 Len=0
76	15.639572	142.250.187.238	192.168.10.111	TCP	60 443 → 1038 [FIN, ACK] Seq=74 Ack=3 Win=514 Len=0
47	7.327277	142.250.200.4	192.168.10.111	TLSv1.2	127 Application Data
49	7.328482	142.250.200.4	192.168.10.111	TCP	60 443 → 1031 [ACK] Seq=74 Ack=2 Win=514 Len=0
50	7.328577	142.250.200.4	192.168.10.111	TCP	60 443 → 1031 [FIN, ACK] Seq=74 Ack=2 Win=514 Len=0

printf "GET / HTTP/1.0\r\n\r\n" | nc 192.168.10.1 80 | head

↑	↑	↑
écrit la chaîne de caractère qui suit	envois la sortie nf sur le port 80	affiche uniquement les 10 premières lignes d'un fichier

Résultat de la commande : printf "GET / HTTP/1.0\r\n\r\n" | nc 192.168.10.1 80 | head :

```
pi@raspberrypi:~ $ printf "GET / HTTP/1.0\r\n\r\n" | nc 192.168.10.1 80 | head
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 13:35:59 GMT
Server: Apache/2.4.54 (Debian)
Last-Modified: Mon, 20 Mar 2023 14:49:10 GMT
ETag: "89-5f756079a8a2c"
Accept-Ranges: bytes
Content-Length: 137
Vary: Accept-Encoding
Connection: close
Content-Type: text/html
pi@raspberrypi:~ $
```