

Mise en place du BGP interne/externe

Installation du Cisco 8200Catalyst

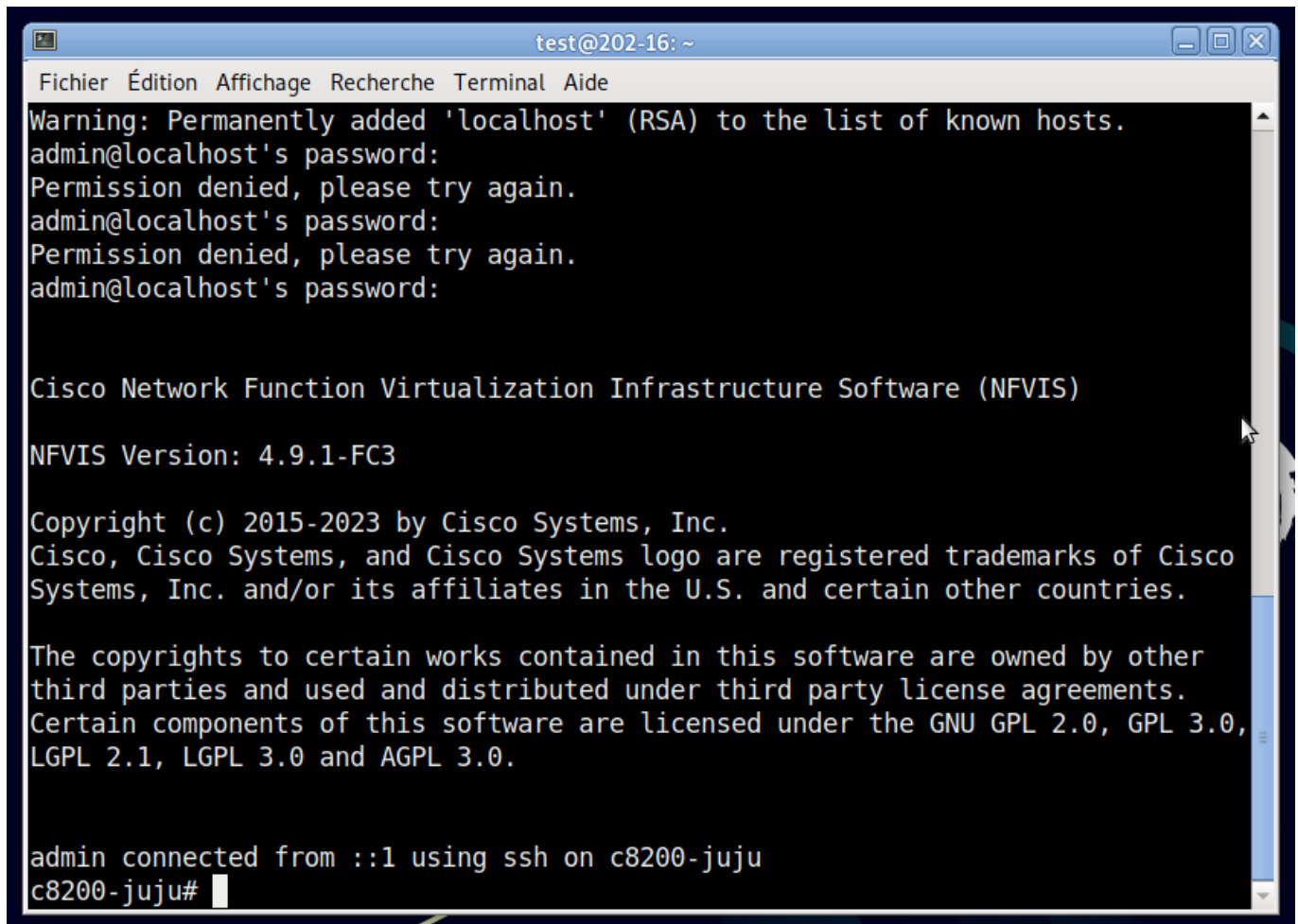
Branchement au réseau

Connexion du port serie à mon ordinateur pour la configuration. Je m'y connecte via le logiciel [minicom](#) que je règle pour pouvoir configurer le routeur :

- Port série : /dev/ttyUSB0
- Débit/Parité/Bits : 9600 8N1

```
+-----+
| A -          Port série : /dev/ttyUSB0
| B - Emplacement fichier verr. : /var/lock
| C -   Prog. d'appel entrant :
| D -   Prog. d'appel sortant :
| E -          Débit/Parité/Bits : 9600 8N1
| F - Contrôle de flux matériel : Oui
| G - Contrôle de flux logiciel : Non
| H -   RS485 Enable      : No
| I -   RS485 Rts On Send  : No
| J -   RS485 Rts After Send : No
| K -   RS485 Rx During Tx  : No
| L -   RS485 Terminate Bus : No
| M - RS485 Delay Rts Before: 0
| N - RS485 Delay Rts After : 0
|
|   Changer quel réglage ? ☐
+-----+
```

Je sauvegarde la configuration dans le menu avec l'option [Enregistrer conf. sous dfl](#) et lance la connexion en appuyant sur entrée, j'ai remarqué que lors de la première connexion en série cela démarre le routeur entièrement, sinon il tourne physiquement mais pas logiciellement. Une fois le démarrage fait il faut se connecter à l'interface de configuration en administrateur [Login : admin](#) et [MDP : Root123#](#)



```
test@202-16: ~
Fichier  Édition  Affichage  Recherche  Terminal  Aide
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
admin@localhost's password:
Permission denied, please try again.
admin@localhost's password:
Permission denied, please try again.
admin@localhost's password:

Cisco Network Function Virtualization Infrastructure Software (NFWIS)

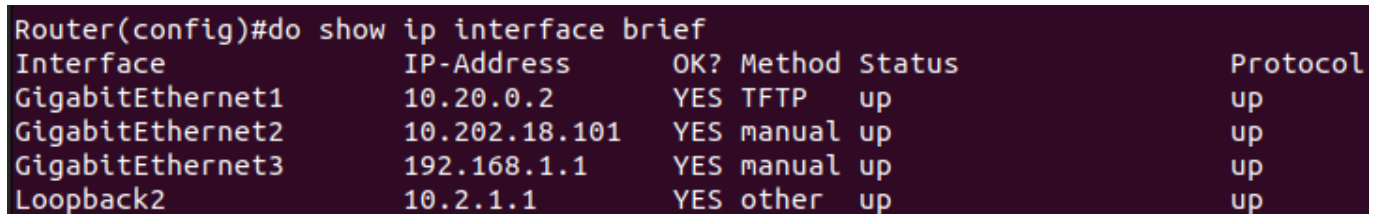
NFWIS Version: 4.9.1-FC3

Copyright (c) 2015-2023 by Cisco Systems, Inc.
Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco
Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

The copyrights to certain works contained in this software are owned by other
third parties and used and distributed under third party license agreements.
Certain components of this software are licensed under the GNU GPL 2.0, GPL 3.0,
LGPL 2.1, LGPL 3.0 and AGPL 3.0.

admin connected from ::1 using ssh on c8200-juju
c8200-juju#
```

Je configure ensuite mon router en lui attribuant une adresse statique sur le réseau, je lui ai affecter la 10.202.18.100. Visible en faisant un `do show run ip interface` :



```
Router(config)#do show ip interface brief
Interface      IP-Address      OK? Method Status      Protocol
GigabitEthernet1  10.20.0.2       YES TFTP  up          up
GigabitEthernet2  10.202.18.101   YES manual up          up
GigabitEthernet3  192.168.1.1     YES manual up          up
Loopback2        10.2.1.1        YES other  up          up
```

Je connecte le routeur au réseau de la salle via la prise Ethernet physique 0/2 et y accède via l'interface web à l'IP cité précédemment. Après connexion avec l'identifiant administrateur, j'arrive sur la page de configuration du router via le web :

Mise à niveau

The screenshot shows the NFVIS dashboard with the following sections:

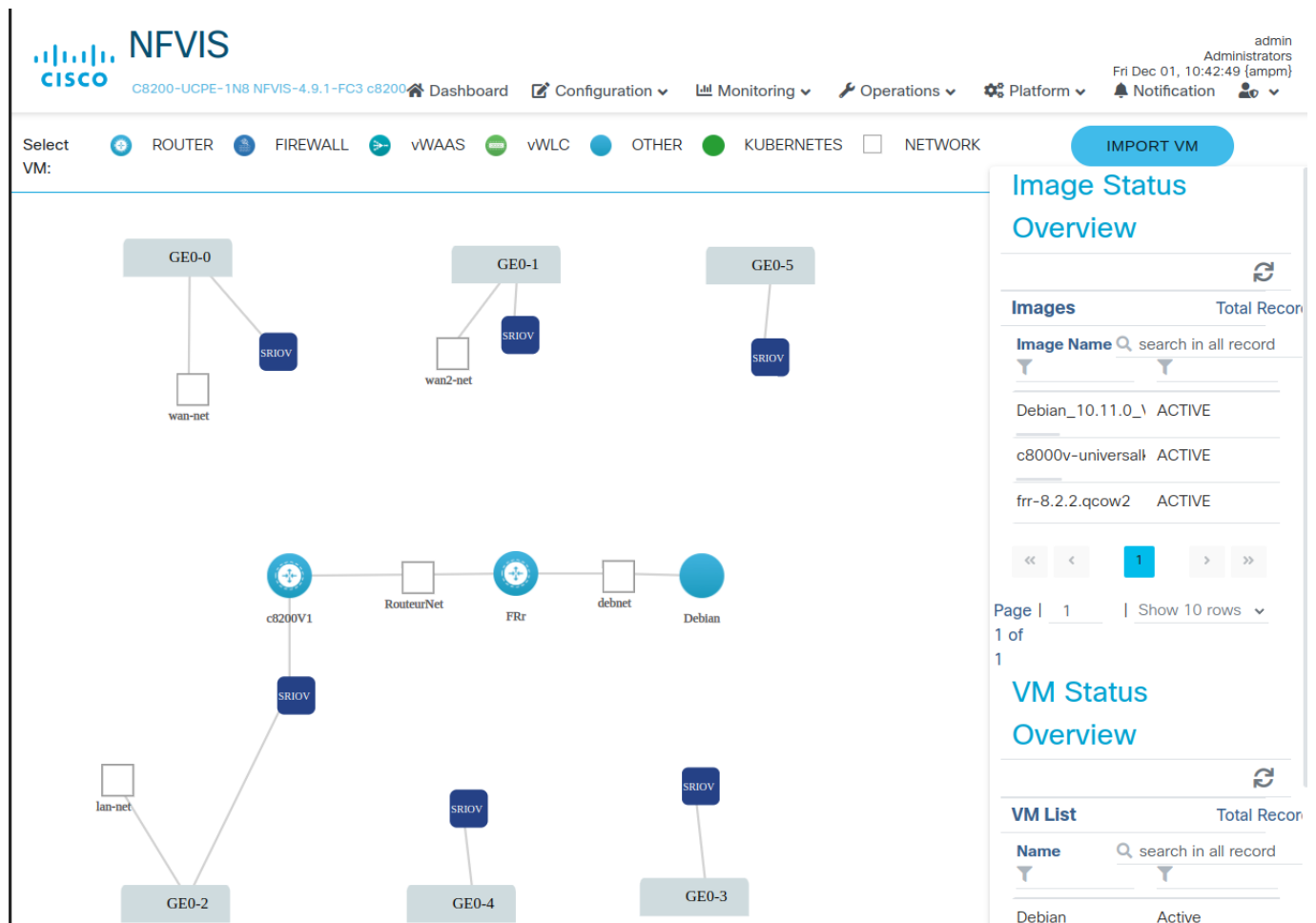
- System Details:**
 - System Status: Healthy
 - PID: C8200-UCPE-1N8
 - Serial Number: FGL2526LG94
 - BIOS-Version: C8200-UCPE_1.04.103020201614
- Resource Utilization:**
 - System CPU Utilization: 0 %
 - Storage Utilization: 0 %
- VM Details:**
 - Total VM: 0
 - Active VM: 0
 - Failed VM: 0
 - Shutdown VM: 0
 - Transitioning VM: 0
- Resource Allocation:**
 - Total Cores: System Occupied: Allocated Cores: Allocated %:
 - Total Memory: System Memory: Allocated Memory: Allocated %:
 - Values: GB, GB, 4 GB, NaN%
- Interface Status:**
 - Fetching Interface Status...
- Failed Notifications:**
 - No Failed Notifications.

J'ai directement commencé par le mettre à jour, via le menu **Opérations** -> **Upgrade**. Les mises à jour doivent être effectuées dans une montée de versions progressive et il faut pas faire une mise à jour de plus de N+3 pour éviter tout bug. J'ai donc suivi cet ordre : 4.5.1 (Version de base) -> 4.6.2 (Format .nfvispkg) -> 4.7.1 (Format .nfvispkg) -> 4.9.1 (Format .iso)

Importation des images routeur nécessaire

J'importe les images des routeurs, .qcow2 du FRR et le .tar.gz du Cisco C8200v. Je crée des bridges réseaux non connectés à des interfaces pour pouvoir les interconnecter entre eux. J'importe aussi une image .ova de Debian pour pouvoir effectuer des tests dans le réseau interne. Pour les importer il faut aller dans le menu **Configuration** -> **Virtual Machine** -> **Image Repository**.

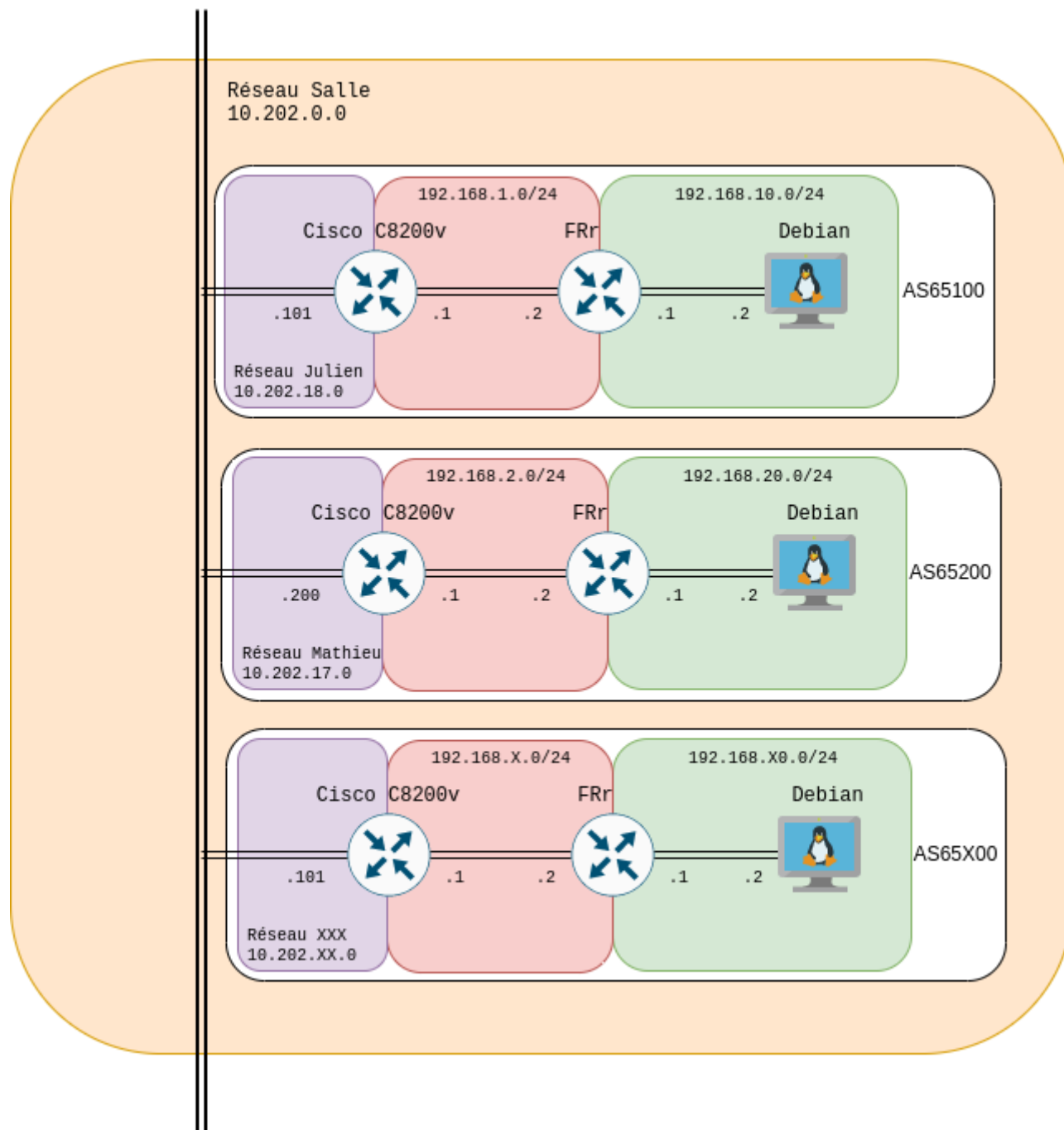
Une fois importées il faut les déployer sur le menu **Configuration** -> **Deploy**.



Mise en place du réseau interne

Configuration des routeurs

Shéma de mon réseau générale :



Je vais reproduire ce schéma en commençant par le cisco, puis le Frr en listant toutes les commandes utilisées et leurs fonctions.

Configuration du Cisco

Connexion en ssh grâce à la commande : `ssh -o HostKeyAlgorithms=+ssh-rsa admin1@10.202.18.101`

```
Router#enable
Router#configure terminal
#Entre dans la partie configuration du router#
-----
Router(config)#interface GigabitEthernet2
#Entre dans la configuration de l'interface GigabitEthernet2#
Router(config-if)#ip address 10.202.18.101 255.255.0.0
```

```
#Affectation de l'IP à l'interface#
Router(config-if)#no shut
#Active l'interface#
Router(config-if)#end
#Sort du mode configuration#
Router#wr mem
#Sauvegarde la configuration même après un redémarrage#
-----
Router#conf t
Router(config)#interface GigabitEthernet3
#Entre dans la configuration de l'interface GigabitEthernet3#
Router(config-if)#ip address 192.168.1.1 255.255.255.0
#Affectation de l'IP à l'interface#
Router(config-if)#no shut
#Activation de l'interface#
Router(config-if)#end
-----
Router#conf t
Router(config)#ip route 0.0.0.0 0.0.0.0 10.202.255.254
#Ajout de la passerelle de la salle pour avoir un accès internet depuis le
cisco#
Router(config)#end
Router#wr mem
-----
Router#conf t
Router(config)#router bgp 65100
#Active et entre dans la configuration de BGP pour l'AS 65100#
Router(config-router)#network 192.168.1.0
#Réseau qui font partie de notre AS#
Router(config-router)#redistribute connected
#Redistribut dans l'AS les routes connecté#
Router(config-router)#neighbor 192.168.1.2
#Déclare un réseau voisins de l'AS faisant aussi du BGP#
Router(config-router)#end
#Sort du mode configuration#
Router#wr mem
#Sauvergarde la configuration#
```

Configuration du FRr

Connexion au FRr à partir du WebUi accessible dans le menu **Configuration** -> **Virtual Machine** -> **Manage** puis à gauche du tableau cliquer sur l'icône Terminal. Les identifiants pour s'y connecter sont **Login : root** et **MDP : root**.

```
Router#enable
Router#configure terminal
#Entre dans le partie configuration du router#
-----
Router(config)#interface eth0
#Entre dans la configuration de l'interface eth0#
Router(config-if)#ip address 192.168.1.2 255.255.255.0
```

```
#Affectation de l'IP à l'interface#
Router(config-if)#no shut
#Active l'interface#
Router(config-if)#end
#Sort du mode configuration#
Router#wr mem
#Sauvergarde la configuration même après un redémarrage#
-----
Router#conf t
Router(config)#interface eth1
#Entre dans la configuration de l'interface eth1#
Router(config-if)#ip address 192.168.10.2 255.255.255.0
#Affectation de l'IP à l'interface#
Router(config-if)#no shut
#Activation de l'interface#
Router(config-if)#end
-----
Router#conf t
Router(config)#router bgp 65100
#Active et entre dans la configuration de BGP pour l'AS 65100#
Router(config-router)#network 192.168.1.0
#Réseau qui font partie de notre AS#
Router(config-router)#network 192.168.10.0
#Réseau qui font partie de notre AS#
Router(config-router)#neighbor 192.168.1.1 remote-as 65100
#Déclare un réseau voisins de l'AS faisant aussi du BGP#
Router(config-router)#end
#Sort du mode configuration#
Router#wr mem
#Sauvergarde la configuration#
```

Je configure aussi le debian en IP statique 192.168.10.2.

Je vérifie ma configuration en effectuant un ping entre le cisco et le debian :

```
debian@debian10:~$ ping 10.202.18.101
PING 10.202.18.101 (10.202.18.101) 56(84) bytes of data.
64 bytes from 10.202.18.101: icmp_seq=1 ttl=254 time=0.864 ms
64 bytes from 10.202.18.101: icmp_seq=2 ttl=254 time=0.446 ms
^C
--- 10.202.18.101 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 0.446/0.655/0.864/0.209 ms
```

Puis un traceroute depuis le cisco afin de m'assurer du chemin emprunté :

```
Router#traceroute 192.168.10.2
Type escape sequence to abort.
Tracing the route to 192.168.10.2
VRF info: (vrf in name/id, vrf out name/id)
 1 192.168.1.2 1 msec 0 msec 0 msec
 2 192.168.10.2 1 msec 1 msec 1 msec
```

Mise en place du réseau externe

Configuration des routeurs

Je vais maintenant rajouter dans bgp le réseau de Mathieu, le 10.202.17.0, AS65200(Visible sur le schéma plus haut).

Je rajoute dans la configuration du cisco les lignes suivantes :

```
Router#conf t
Router(config)#router bgp 65100
Router(config-router)#neighbor 10.202.17.200 remote-as 65200
#Déclare un réseau voisins de l'AS faisant aussi du BGP#
Router(config-router)#end
#Sort du mode configuration#
Router#wr mem
#Sauvergarde la configuration#
```

Et voilà le réseau de Mathieu est interconnecté à mon réseau. Mathieu doit de même effectuer cette commande de son côté.

Test et preuves

Je teste de le ping et j'effectue un traceroute vers sa debian qui est la machine situé le plus loins dans le réseau:

```
debian@debian10:~$ ping 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_seq=1 ttl=60 time=1.61 ms
64 bytes from 192.168.20.2: icmp_seq=2 ttl=60 time=0.812 ms
64 bytes from 192.168.20.2: icmp_seq=3 ttl=60 time=0.898 ms
^C
--- 192.168.20.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 5ms
rtt min/avg/max/mdev = 0.812/1.107/1.612/0.359 ms
```

```
debian@debian10:~$ traceroute 192.168.20.2
traceroute to 192.168.20.2 (192.168.20.2), 30 hops max, 60 byte packets
 1  192.168.10.1 (192.168.10.1)  0.224 ms  0.179 ms  0.196 ms
 2  192.168.1.1 (192.168.1.1)  0.553 ms  0.518 ms  0.508 ms
 3  10.202.17.200 (10.202.17.200)  0.637 ms  0.619 ms  0.624 ms
 4  192.168.2.2 (192.168.2.2)  1.051 ms  1.008 ms  1.003 ms
 5  192.168.20.2 (192.168.20.2)  0.968 ms  0.931 ms  0.923 ms
```

Affichage des routes et configuration BGP de mon Cisco et FRr :

- BGP :

	Network	Next Hop	Metric	LocPrf	Weight	Path
*	10.202.0.0/16	10.202.17.200	0			0 65200 ?
*>		0.0.0.0	0		32768	?
* i	192.168.1.0	192.168.1.2	0	100	0	i
*>		0.0.0.0	0		32768	i
*>	192.168.2.0	10.202.17.200	0		0	65200 i
*>i	192.168.10.0	192.168.1.2	0	100	0	i
*>	192.168.20.0	10.202.17.200			0	65200 i

- Route :

```

Gateway of last resort is 10.202.255.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.202.255.254
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C      10.202.0.0/16 is directly connected, GigabitEthernet2
L      10.202.18.101/32 is directly connected, GigabitEthernet2
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.1.0/24 is directly connected, GigabitEthernet3
L      192.168.1.1/32 is directly connected, GigabitEthernet3
B      192.168.2.0/24 [20/0] via 10.202.17.200, 2d08h
B      192.168.10.0/24 [200/0] via 192.168.1.2, 2d06h
B      192.168.20.0/24 [20/0] via 10.202.17.200, 2d06h

```

Conf FRr et route :

- BGP :

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>i	10.202.0.0/16	192.168.1.1	0	100	0	?
* i	192.168.1.0/24	192.168.1.1	0	100	0	i
*>		0.0.0.0	0		32768	i
*>i	192.168.2.0/24	10.202.17.200	0	100	0	65200 i
*>	192.168.10.0/24	0.0.0.0	0		32768	i
*>i	192.168.20.0/24	10.202.17.200	0	100	0	65200 i

Displayed 5 routes and 6 total paths

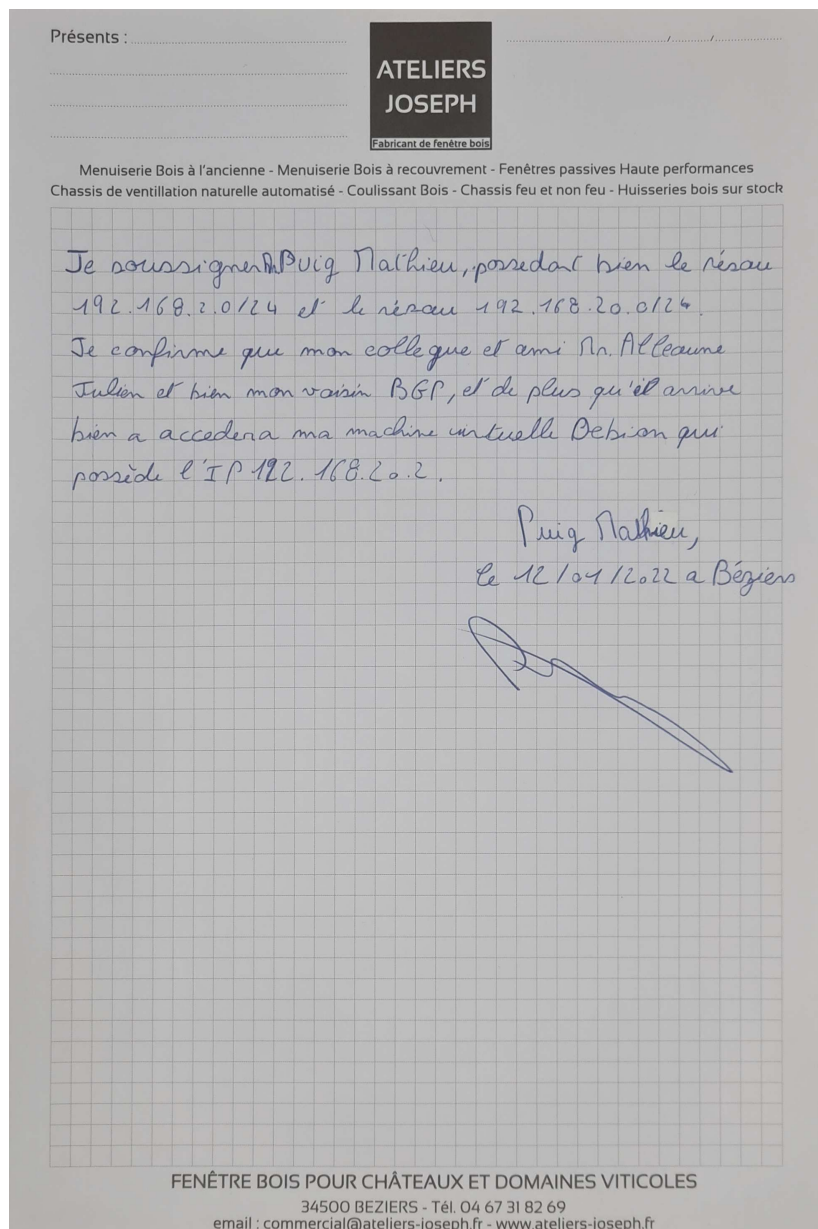
- Route :

```

B>* 10.202.0.0/16 [200/0] via 192.168.1.1, eth0, weight 1, 2d08h40m
C>* 192.168.1.0/24 is directly connected, eth0, 2d08h40m
B> 192.168.2.0/24 [200/0] via 10.202.17.200 (recursive), weight 1, 2d08h03m
   * via 192.168.1.1, eth0, weight 1, 2d08h03m
C>* 192.168.10.0/24 is directly connected, eth1, 2d07h56m
B> 192.168.20.0/24 [200/0] via 10.202.17.200 (recursive), weight 1, 2d06h13m
   * via 192.168.1.1, eth0, weight 1, 2d06h13m

```

Tout fonctionne parfaitement. Nous n'avons pas pu affecter d'autres personnes à notre réseau BGP car si les personnes avaient une mauvaise configuration BGP ils cassaient l'entièreté du réseau en créant une boucle infinie.



Auto Formation grâce au TP Cisco et les supports de COURS

Configuration en utilisant NetConf

Je regarde si mes daemons sont démarrés sur le router c8200v :

```
show platform software yang-management process
```

Pour allumer au besoin dans le routeur les daemons :

```
netconf-yang
```

Sur nos routeurs il est important de rajouter cette ligne à la première balise à chaque requête sinon les versions ne sont pas bonnes et font planter la requêtes :

```
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
```

Connection et affichage de la configuration filtrée et préttifiée du 8200v :

```
from ncclient import manager

m = manager.connect (
    host="10.202.18.101",
    port=830,
    username="admin1",
    password="Root123#",
    hostkey_verify=False
)

netconf_filter = """
<filter xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native"/>
</filter>
"""

netconf_reply = m.get_config(source="running", filter=netconf_filter)
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
```

Sortie du code python :

```
<?xml version="1.0" ?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:052eac15-e33a-4206-972d-8f2537bd049e">
  <data>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
      <version>17.6</version>
      <boot-start-marker/>
      <boot-end-marker/>
      <memory>
        <free>
          <low-watermark>
            <processor>68484</processor>
          </low-watermark>
        </free>
      </memory>
      <call-home>
        ...
        ...
        <first>0</first>
        <last>4</last>
```

```

        <login>
            <local/>
        </login>
        <transport>
            <input>
                <input>ssh</input>
            </input>
        </transport>
    </vty>
</line>
<diagnostic xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-
diagnostics">
    <bootup>
        <level>minimal</level>
    </bootup>
</diagnostic>
</native>
</data>
</rpc-reply>

```

Création d'une nouvelle interface loopback2 sur le 8200v :

```

from ncclient import manager
import xml.dom.minidom
m = manager.connect (
host="10.202.18.101",
port=830,
username="admin1",
password="Root123#",
hostkey_verify=False,
#device_params={'name': 'iosxe'},
#allow_agent=False,
#look_for_keys=False
)

netconf_newloop = """
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
<interface>
<Loopback>
<name>2</name>
<description>Deuxieme tourne en rond</description>
<ip>
<address>
<primary>
<address>10.2.2.2</address>
<mask>255.255.255.0 </mask>
</primary>
</address>
</ip>

```

```
</Loopback>
</interface>
</native>
</config>
"""

netconf_reply = m.edit_config(target="running", config=netconf_newloop)
print(xml.dom.minidom.parseString(netconf_reply.xml).toprettyxml())
```

Je lance la commande `sh run loopback2` sur le routeur pour m'assurer de sa création :

```
Router#sh run interface loopback2
Building configuration...

Current configuration : 100 bytes
!
interface Loopback2
  description Deuxieme tourne en rond
  ip address 10.2.2.2 255.255.255.0
end
```

Configuration en utilisant Restconf

Je regarde si mes daemon sont démarré sur le router c8200v :

```
show platform software yang-management process
```

Pour allumer au besoins dans le routeur les daemons :

```
restconf
```

Affichage des interfaces du cisco 8200v à l'aide d'un restconf-get :

```
import json
import requests
requests.packages.urllib3.disable_warnings()

api_url = "https://10.202.18.101/restconf/data/ietf-interfaces:interfaces"

headers = { "Accept": "application/yang-data+json",
            "Content-type": "application/yang-data+json"
          }
```

```
basicauth = ("admin1", "Root123#")

resp = requests.get(api_url, auth=basicauth, headers=headers, verify=False)

print(resp)

response_json = resp.json()

print(response_json)
print (json.dumps (response_json, indent=4))
```

En sortie de la commande j'ai la liste de mes interfaces dont la loopback2 que j'ai créé précédemment grâce à Netconf :

```
<Response [200]>
{"ietf-interfaces:interfaces": {
  "interface": [
    {
      "name": "GigabitEthernet1",
      "type": "iana-if-type:ethernetCsmacd",
      "enabled": true,
      "ietf-ip:ipv4": {
        "address": [
          {
            "ip": "10.20.0.2",
            "netmask": "255.255.255.0"
          }
        ]
      },
      "ietf-ip:ipv6": {}
    },
    {
      "name": "GigabitEthernet2",
      "type": "iana-if-type:ethernetCsmacd",
      "enabled": true,
      "ietf-ip:ipv4": {
        "address": [
          {
            "ip": "10.202.18.101",
            "netmask": "255.255.0.0"
          }
        ]
      },
      "ietf-ip:ipv6": {}
    },
    {
      "name": "GigabitEthernet3",
      "type": "iana-if-type:ethernetCsmacd",
      "enabled": true,
      "ietf-ip:ipv4": {
```

```

        "address": [
            {
                "ip": "192.168.1.1",
                "netmask": "255.255.255.0"
            }
        ]
    },
    "ietf-ip:ipv6": {}
},
{
    "name": "Loopback2",
    "description": "Deuxieme tourne en rond",
    "type": "iana-if-type:softwareLoopback",
    "enabled": true,
    "ietf-ip:ipv4": {
        "address": [
            {
                "ip": "10.2.2.2",
                "netmask": "255.255.255.0"
            }
        ]
    },
    "ietf-ip:ipv6": {}
}
]
}
}

```

Création de la loopback1 sur le cisco 8200v à l'aide d'un restconf-put :

```

import json
import requests
requests.packages.urllib3.disable_warnings()

api_url = "https://10.202.18.101/restconf/data/ietf-
interfaces:interfaces/interface=Loopback2"

headers = { "Accept": "application/yang-data+json",
"Content-type":"application/yang-data+json"
}

basicauth = ("admin1", "Root123#")

YangConfig = {
"ietf-interfaces:interface": {
"name": "Loopback1",
"description": "Ma première RESTCONF route de retour",
"type": "iana-if-type:softwareLoopback",
"enabled": True,
"ietf-ip:ipv4": {
"address": [

```

```
{
  "ip": "10.1.1.1",
  "netmask": "255.255.255.0"
}
],
"ietf-ip:ipv6": {}
}
}
```

```
resp = requests.put(api_url, data=json.dumps(YangConfig), auth=basicauth,
headers=headers, verify=False)

if(resp.status_code >= 200 and resp.status_code <= 299):
    print("STATUS OK: {}".format(resp.status_code))
else:
    print("Error. Code d'état : {} \n Message d'erreur : {}".format(resp.status_code, resp.json()))
```

En sortie de la commande j'obtiens :

STATUS OK: 204

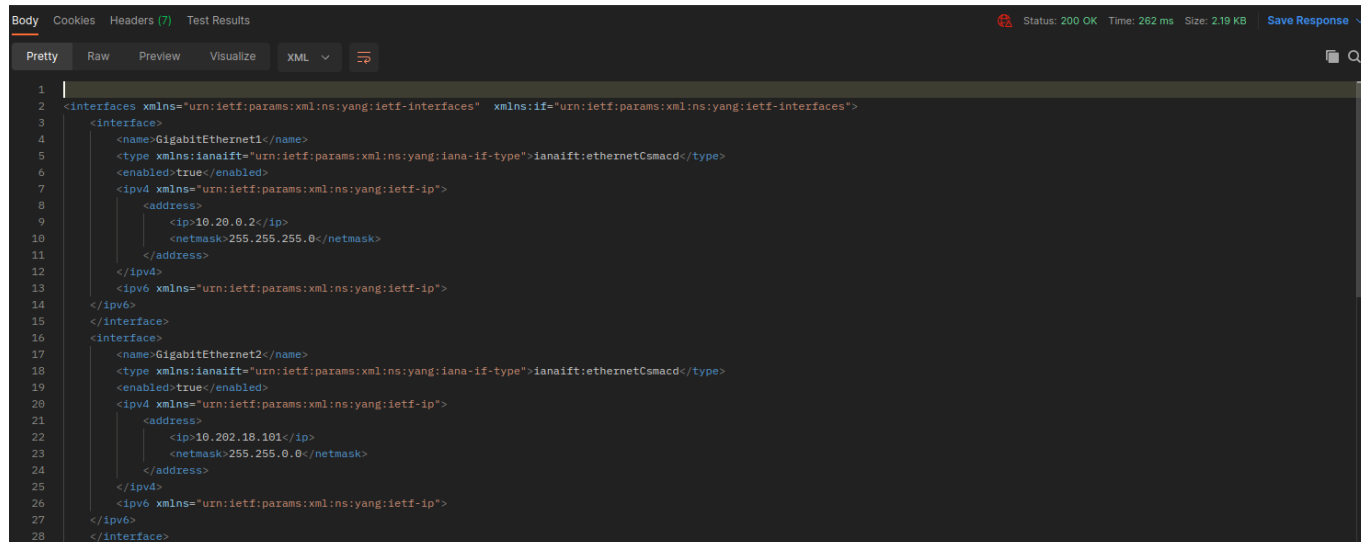
Qui confirme le bon fonctionnement de la commande.

Configuration avec Postman

Je le configure comme conseiller dans le TP 8.3.7 Partie 3, et met bien mon ip dans le lien, je procède ensuite à un **get** de mes interfaces :

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Authorization ⓘ	Basic YWRtaW4xOIJvb3QxMjMj	
<input checked="" type="checkbox"/> Postman-Token ⓘ	<calculated when request is sent>	➤ Go to settings
<input checked="" type="checkbox"/> Host ⓘ	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent ⓘ	PostmanRuntime/7.30.0	
<input checked="" type="checkbox"/> Accept ⓘ	*/*	
<input checked="" type="checkbox"/> Accept-Encoding ⓘ	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection ⓘ	keep-alive	
<input checked="" type="checkbox"/> Content-Type	application/yang-data+json.	
Key	Value	Description

Je reçois une réponse positive avec les interfaces de mon routeur :



SNMP

Activer le protocole snmp sur le routeur cisco 8200v Virtuel :

```

Router#enable
Router#configure terminal
-----
Router(config)#snmp-server community public RO
#Active le serveur SNMP publique en Read-Only#

```

De plus j'installe de mon côté les paquets `snmp` et `snmp-mibs-downloader` afin d'avoir un affichage plus lisible.

Résultat d'une requête sur le cisco 8200v Virtuelle :

```

snmpwalk -v1 -c public 10.202.18.101 -Cp -Ct | grep bgp
#Requête SNMP version 1, communauté publique, à l'adresse 10.202.180.101,
avec les options nombre de variable et temps de la commande#

```

Voici une partie de la réponse :

```

BGP4-MIB::bgp4PathAttrAggregatorAddr.192.168.1.0.24.192.168.1.2 =
IpAddress: 0.0.0.0
BGP4-MIB::bgp4PathAttrAggregatorAddr.192.168.2.0.24.10.202.17.200 =
IpAddress: 0.0.0.0
BGP4-MIB::bgp4PathAttrAggregatorAddr.192.168.10.0.24.192.168.1.2 =
IpAddress: 0.0.0.0
BGP4-MIB::bgp4PathAttrAggregatorAddr.192.168.20.0.24.10.202.17.200 =
IpAddress: 0.0.0.0
BGP4-MIB::bgp4PathAttrCalcLocalPref.10.202.0.0.16.0.0.0.0 = INTEGER: 100
BGP4-MIB::bgp4PathAttrCalcLocalPref.10.202.0.0.16.10.202.17.200 = INTEGER:

```

```
100
BGP4-MIB::bgp4PathAttrCalcLocalPref.192.168.1.0.24.0.0.0.0 = INTEGER: 100

Total traversal time = 4.228723 seconds
```

Debugage du tp cisco

TP 8,3,6 Étape 3, a :

Handshake corrigé :

Il faut rajouter « > » à la fin de la deuxième ligne du hello.

```
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
  <capability>urn:ietf:params:netconf:base:1.0</capability>
</capabilities>
</hello>
```

Partie 4 Étape 1 b :

Enlever l'ipv6 : pour réparer les erreurs.

TP 8,3,7 Partie 7 Étape 2, d :

Il faut remplacer les simples quotes double par des doubles quotes, qui encadre netmask et le masque réseau.

```
"address": [
  {
    "ip": "192.168.56.101",
    "netmask": "255.255.255.0"
  }
]
```

TP 8,3,7 Partie 6 Étape 3, a :

Enlever les espace en trop ou les mauvaises traductions facilement visibles et sur plupart part des commandes comme par exemples celle-ci :

Avant correction :

```
resp = requests.put (api_url, data=json.dumps (YangConfig),  
auth=basicauth, en-têtes=, verify=false)
```

Après correction :

```
resp = requests.put(api_url, data=json.dumps(YangConfig), auth=basicauth,  
headers=headers, verify=false)
```

TP 8,3,7 Étape 3, b :

Changer les ' par des " pour pouvoir afficher le contenu sans problème.

Avant :

```
else:  
print('Error. Code d'état : {} \ nMessage d'erreur : {} '.format  
(resp.status_code, resp.json ()))
```

Après:

```
else:  
print("Error. Code d'état : {} \n Message d'erreur : {}  
".format(resp.status_code, resp.json()))
```