



## Table des matières

1) Environnement du TP et notation.....	1
1.1) Objectifs du TP.....	1
1.2) Organisation, recommandation et notation du TP.....	2
2) Installation sous votre Debian.....	2
2.1) Installation de VmWare Workstation.....	2
2.2) Installation de l'ova GNS3 dans VMWare Workstation.....	2
3) Mise en place d'un projet didactique sous GNS3.....	3
3.1) Schéma à réaliser :.....	3
3.2) Rappel de commandes Cisco de bases :.....	3
3.3) Intégration de machines virtuelles , de containers Docker et d'une ap-pliance de connexion dans GNS3.....	10
4) Création de machines virtuelles avec Multipass.....	12
5) Utilisation de docker.....	12
5.1) Installation de docker sous Linux.....	12

## 1) Environnement du TP et notation

### 1.1) Objectifs du TP

L'objectif de ce TP est de vous montrer comment fonctionne GNS3 et accessoirement de vous remettre en tête des notions de réseaux de première année de DUT ou de BTS. GNS3 est une interface graphique en Python faisant le lien avec des émulateurs ( dynamips, virtualbox, vmware , docker) qui permet de créer des environnements comprenant des routeurs , des machines virtuelles serveurs et des pseudos PC. Contrairement à Packet Tracer qui simule , GNS3 permet d'avoir un routeur avec l'ensemble de ses capacités et une connectivité internet. Il vous sera utile pour monter des POC ( Proof Of Concept) , des maquettes pour vos projets ou plus simplement pour vous former. GNS3 est plus limité côté switch que packet tracer néanmoins il existe certains "workaround" permettant de contourner ces limitations. Les IOS ne sont pas libres de droits, L'IUT est Académie Cisco mais vous ne pouvez pas utiliser l'IOS en dehors de l'I.U.T. .

## **1.2) Organisation, recommandation et notation du TP**

Vous travaillerez individuellement sur votre PC. Vous ferez valider vos réalisations lors de trois check-points ( installation, réalisation du premier schéma , réalisation du second schéma ).Sauvegardez votre projet GNS3 régulièrement.Vous éviterez ainsi de tout recommencer. Vous réaliserez aussi un compte rendu succinct ( copies d'écran , configurations, quelques explications sur les points importants ) que vous remettrez sur moodle à la fin des quatre heures.Merci de penser à mettre l'adressage sur vos schémas (Classe d'adresse sur le lien et octets significatifs sur les matériels).Vous devez montrer à l'enseignant que la maquette fonctionne ( ping et traceroute sont vos amis).Attention à l'orthographe et la grammaire qui seront pris en compte dans la notation

## **2) Installation sous votre Debian**

### **2.1) Installation de VmWare Workstation**

```
wget http://store.iutbeziers.fr/vmware/vmware-workstation.bundle
```

Cela permet de remettre une licence de 30j sur VmWare

### **2.2) Installation de l'ova GNS3 dans VMWare Workstation**

A partir d'ici on a eu un problème avec docker donc nous avons dû le réinstaller en suivant le tuto d'install sur le site de docker.

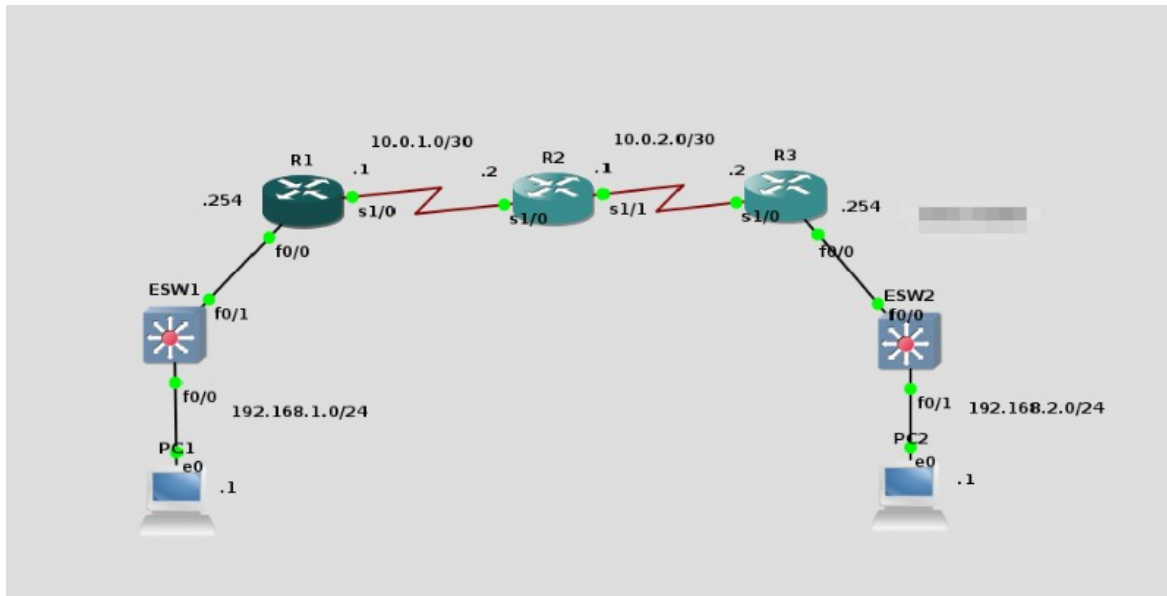
Une fois installé on donne les droits sur docker à test avec la commande suivante :

```
usermod -aG docker test
```

Maintenant nous pouvons poser des pc registry.iutbeziers dans GNS3

### **3) Mise en place d'un projet didactique sous GNS3**

#### **3.1) Schéma à réaliser :**



#### **3.2) Rappel de commandes Cisco de bases :**

2) Je dois configurer les adresses réseaux comme sur le schéma au dessus

3) Je test que R1 peut ping R2 :

```
R1#ping 10.0.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
16/19/20 ms
R1#ping 10.0.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
16/19/20 ms
```

R2 peut ping R3 :

```
R2#ping 10.0.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
60/74/100 ms
R2#
*Mar  1 00:25:12.743: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Serial1/1, changed state to up
R2#ping 10.0.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
48/60/72 ms
```

4) Maintenant je visualise les interfaces de mes routeurs en mode privilégié :

Les interfaces détaillées :

```
FastEthernet0/0 is up, line protocol is up
  Hardware is Gt96k FE, address is c201.0efc.0000 (bia
c201.0efc.0000)
  Internet address is 192.168.1.254/24
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00

Serial1/0 is up, line protocol is up
  Hardware is M4T
  Internet address is 10.0.1.1/30
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
```

Je ne m'intéresse qu'à ces deux interfaces sur le routeur 1 car ce sont celles que j'utilise. On y voit bien qu'elles sont en état "up" et leur adresse IP avec le masque de sous réseau (/24 et /30)

Maintenant je regarde la configuration IP des interfaces :

```
R1#show ip interface brief
Interface          IP-Address      OK? Method Status
Protocol
FastEthernet0/0    192.168.1.254   YES NVRAM  up
up
FastEthernet0/1    unassigned      YES NVRAM  up
up
Serial1/0          10.0.1.1        YES NVRAM  up
up
Serial1/1          unassigned      YES NVRAM
administratively down down
Serial1/2          unassigned      YES NVRAM
administratively down down
Serial1/3          unassigned      YES NVRAM
administratively down down
```

Maintenant la table de routage :

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user
static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/30 is subnetted, 1 subnets
C      10.0.1.0 is directly connected, Serial1/0
C      192.168.1.0/24 is directly connected, FastEthernet0/0
```

5) Je crée maintenant des adresses de loopback sur chaque routeur :

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interf
R1(config)#interface loo
R1(config)#interface loopback 0
R1(config-if)#ip adres
```

```
*Mar 1 00:24:19.643: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Loopback0, changed state to up
R1(config-if)#ip address 1.1.1.1 255.255.255.255
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#exit
R1#
*Mar 1 00:26:02.023: %SYS-5-CONFIG_I: Configured from console by
console
```

```
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int
R2(config)#interface loo
R2(config)#interface loopback 0
R2(config-if)#ip addre
*Mar 1 00:11:51.235: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Loopback0, changed state to up
R2(config-if)#ip address 2.2.2.2 255.255.255.255
R2(config-if)#no shut
R2(config-if)#exit
```

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#inter
R3(config)#interface loo
R3(config)#interface loopback 0
R3(config-if)#ip a
*Mar 1 00:12:07.463: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Loopback0, changed state to up
R3(config-if)#ip address 3.3.3.3 255.255.255.255
R3(config-if)#no shut
R3(config-if)#exit
```

6) Je visualise une nouvelle fois la table de routage :

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
IS level-2
```

```
    ia - IS-IS inter area, * - candidate default, U - per-user
static route
    o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
C    1.1.1.1 is directly connected, Loopback0
    10.0.0.0/30 is subnetted, 1 subnets
C    10.0.1.0 is directly connected, Serial1/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
```

Je vois que l'interface de loopback a bien été connectée

7) Le C devant chaque table de routage signifie que l'interface est bien connectée avec une adresse IP et un masque de sous réseau. Il existe d'autres possibilités que C (Connected) devant chaque table de routage. Par exemple R pour le protocole RIP ou R pour l'OSPF. Cela permet de désigner le type de route. Voici les différents codes :

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user
static route
       o - ODR, P - periodic downloaded static route
```

8) Je test un ping de R1 vers R2 :

```
R1#ping 10.0.1.2 source Serial 1/0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.2, timeout is 2 seconds:
Packet sent with a source address of 10.0.1.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
8/11/12 ms
```

9) Maintenant je test un ping de R1 vers R2 en passant par l'interface loopback :

```
R1#ping 10.0.1.2 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
.....
Success rate is 0 percent (0/5)
R1#ping 10.0.1.2 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
.....
Success rate is 0 percent (0/5)
```

Je dois aussi définir des routes statiques sur mes routeurs pour que par exemple le routeur 1 arrive à trouver le routeur 3. Pour se faire je dois faire la commande suivante :

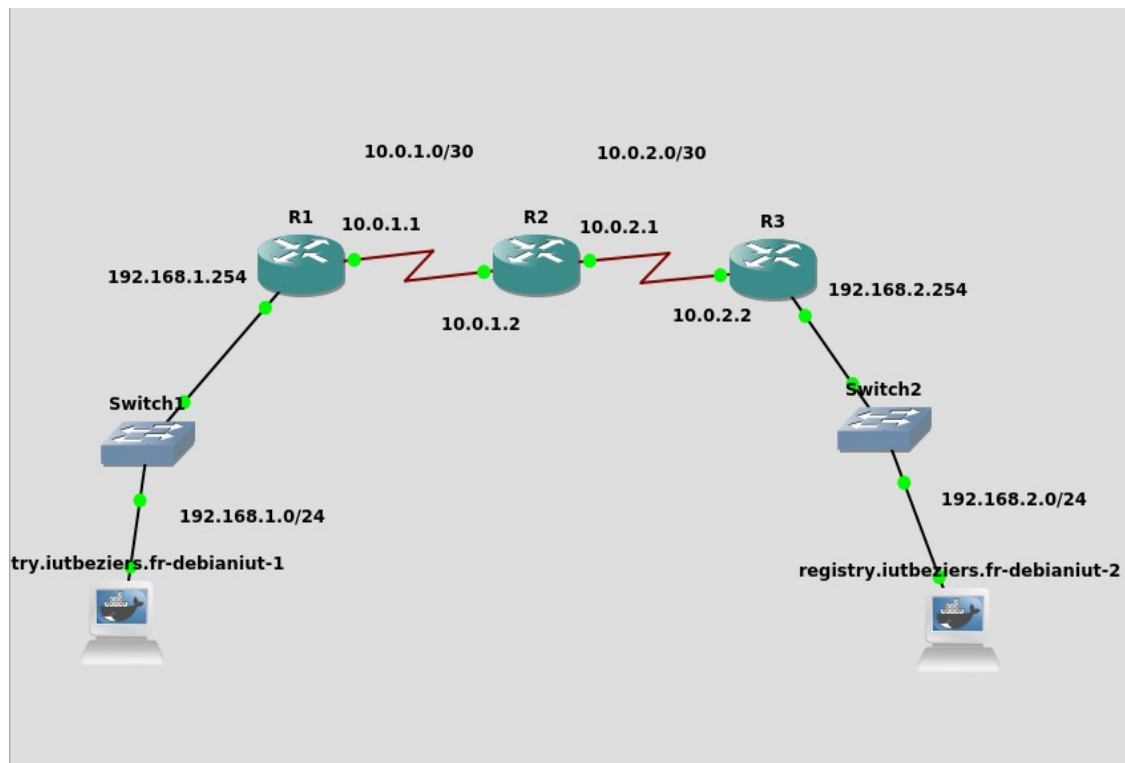
```
ip route @réseau_cible @masque @routeur_à_passer
```

Dans l'exemple du routeur 1 ça donne :

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip route 192.168.2.0 255.255.255.0 10.0.2.2
R1(config)#ip route 10.0.2.0 255.255.255.252 10.0.1.2
R1(config)#exit
```



Voici donc mon réseau :



Nous avons ici la preuve que tout mon réseau fonctionne. Je suis sur le pc du réseau 192.168.1.1 et je ping le pc du réseau 192.168.2.1 :

```
/ # ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=61 time=32.7 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=61 time=47.0 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=61 time=41.10 ms
^C
--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 5ms
rtt min/avg/max/mdev = 32.737/40.574/47.020/5.918 ms
```

### **3.3) Intégration de machines virtuelles , de containers Docker et d'une ap-pliance de connexion dans GNS3**

Tout d'abord je commence par installer les paquets qu'il faut pour pouvoir placer le NAT dans GNS3 :

```
test@202-19:~$ sudo apt install libvirt-clients libvirt-daemon-system virtinst  
test@202-19:~$ sudo virsh net-start default
```

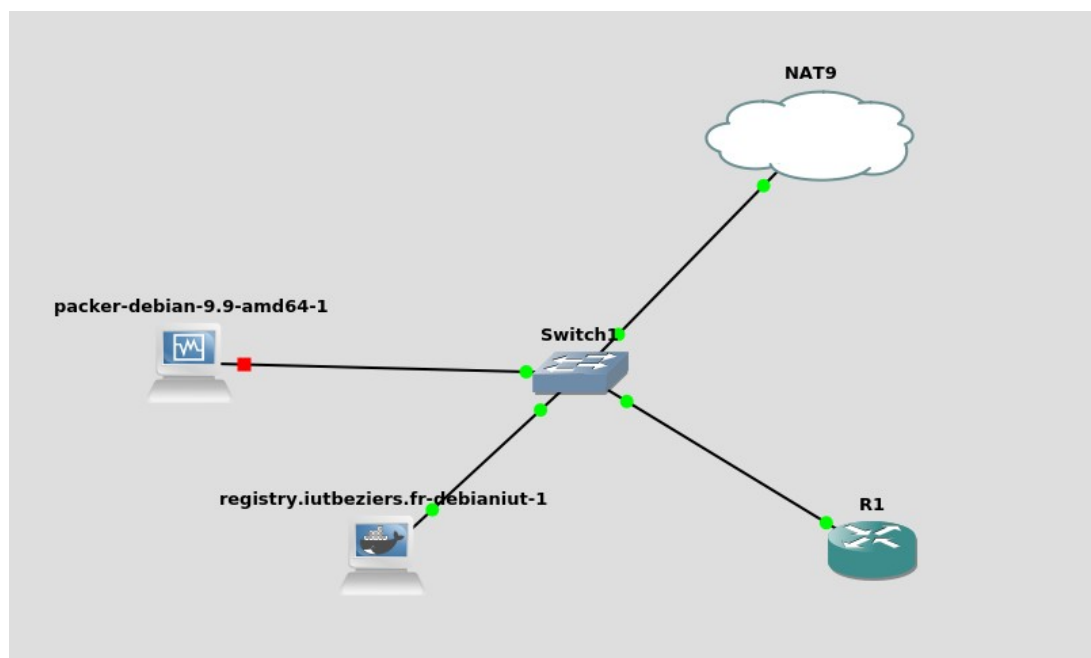
Maintenant je peux placer le NAT dans GNS3

Je fais ensuite la création d'une VM debian 9. Je vais chercher le fichier .ova sur le site de l'iut et je la place dans VmWare. Je modifie la Ram de la VM sur 1Go.

Maintenant dans le menu préférences de GNS3 je vais sur VmWare et j'y ajoute ma vm debian 9.

Je peux maintenant la placer dans GNS3

Mon schéma est le suivant :



Je configure la VM, le container et le routeur pour qu'ils fassent du dhcp.

Une fois le dhcp activé mes machines auront une adresse IP automatiquement allouée

Je peux voir que je peux ping google sur le container docker :

```
/ # ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=6.79 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=8.58 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 6.794/7.685/8.576/0.891 ms
```

Configuration du dhcp sur le routeur :

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interf
R1(config)#interface Fa
R1(config)#interface FastEthernet 0/0
R1(config-if)#ip address dhcp
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
*Mar  1 00:31:46.011: %LINK-3-UPDOWN: Interface FastEthernet0/0,
changed state to up
*Mar  1 00:31:47.011: %LINEPROTO-5-UPDOWN: Line protocol on
Interface FastEthernet0/0, changed state to up
R1(config)#exit

R1#show ip interface brief
Interface                IP-Address      OK? Method Status
Protocol
FastEthernet0/0          192.168.122.5   YES DHCP    up
up
```

Je peux ping google via le routeur à partir de maintenant :

```
R1#ping 8.8.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
60/61/64 ms
```

## 4) Création de machines virtuelles avec Multipass

Commande pour installer une machine virtuelle Ubuntu 20 avec Multipass :

```
root@202-19:~# multipass launch -n Ubuntu 20.04 -c 2 -m 2G

root@202-19:~# multipass list
Name                State      IPv4             Image
Ubuntu20            Running    10.232.222.70    Ubuntu
20.04 LTS
```

```
root@202-19:~# multipass stop Ubuntu20
root@202-19:~# multipass delete Ubuntu20
root@202-19:~# multipass purge
```

Ces trois commandes me permettent de supprimer complètement la machine virtuelle de multipass

## 5) Utilisation de docker

### 5.1) Installation de docker sous Linux

1) La version de docker est 19.03.13

```
test@202-19:~$ docker -v
Docker version 19.03.13, build 4484c46d9d
```

2) On lance la commande :

```
test@202-19:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working
correctly.
```

La commande dit que l'installation a l'air de fonctionner

c) Pour se faire le client docker a contacté son daemon

il a pris l'image depuis docker-hub

Le daemon a crée un container avec cette image et il a renvoyé la réponse au client docker, donc sur notre console

C'est le fichier "scratch" qui a permis de créer l'image "hello-world"

3) Je cherche les images debian avec la commande suivante :

```
test@202-19:~$ docker search --filter=is-official=true debian
NAME                DESCRIPTION
STARS              OFFICIAL          AUTOMATED
ubuntu              Ubuntu is a Debian-based Linux operating sys...
11381               [OK]
debian              Debian is a Linux distribution that's compos...
3610                [OK]
```

Il trouve aussi ubuntu car il est basé sur Debian

Je cherche aussi les images busybox :

```
test@202-19:~$ docker search --filter=is-official=true busybox
NAME                DESCRIPTION          STARS
OFFICIAL            AUTOMATED
busybox             Busybox base image.  2008          [OK]
```

4) Je crée le container grâce à l'image debian :

```
test@202-19:~$ docker run -d debian
Unable to find image 'debian:latest' locally
latest: Pulling from library/debian
```

5) Le container est bien vivant car on peut voir le temps auquel il a été créé et le temps de fonction (ici 2h)

```
test@202-19:~$ docker ps
CONTAINER ID        IMAGE
COMMAND            CREATED            STATUS
PORTS              NAMES
bace97efacea       registry.iutbeziers.fr/debianiut:latest
"/gns3/init.sh /bin/..." 2 hours ago       Up 2 hours
dreamy_roentgen
```

6) Je relance un docker run en lui donnant un argument en plus :

```
test@202-19:~$ docker run -d bash -c "trap : TERM INT ; sleep
infinity & wait"
Unable to find image 'bash:latest' locally
latest: Pulling from library/bash
df20fa9351a1: Already exists
b89aa7cc56cc: Pull complete
344bb7aec2e6: Pull complete
```

7) Je stop le container grâce à son ID :

```
test@202-19:~$ docker stop bace97efacea
bace97efacea
```

8) Je supprime le container et je regarde qu'il n'est plus ici :

```
test@202-19:~$ docker rm bace97efacea
bace97efacea
test@202-19:~$ docker ps
CONTAINER ID        IMAGE               COMMAND
CREATED            STATUS              PORTS              NAMES
51f65bc5ba08        bash               "docker-entrypoint.s..."  5
minutes ago        Up 5 minutes
intelligent_lichterman
```