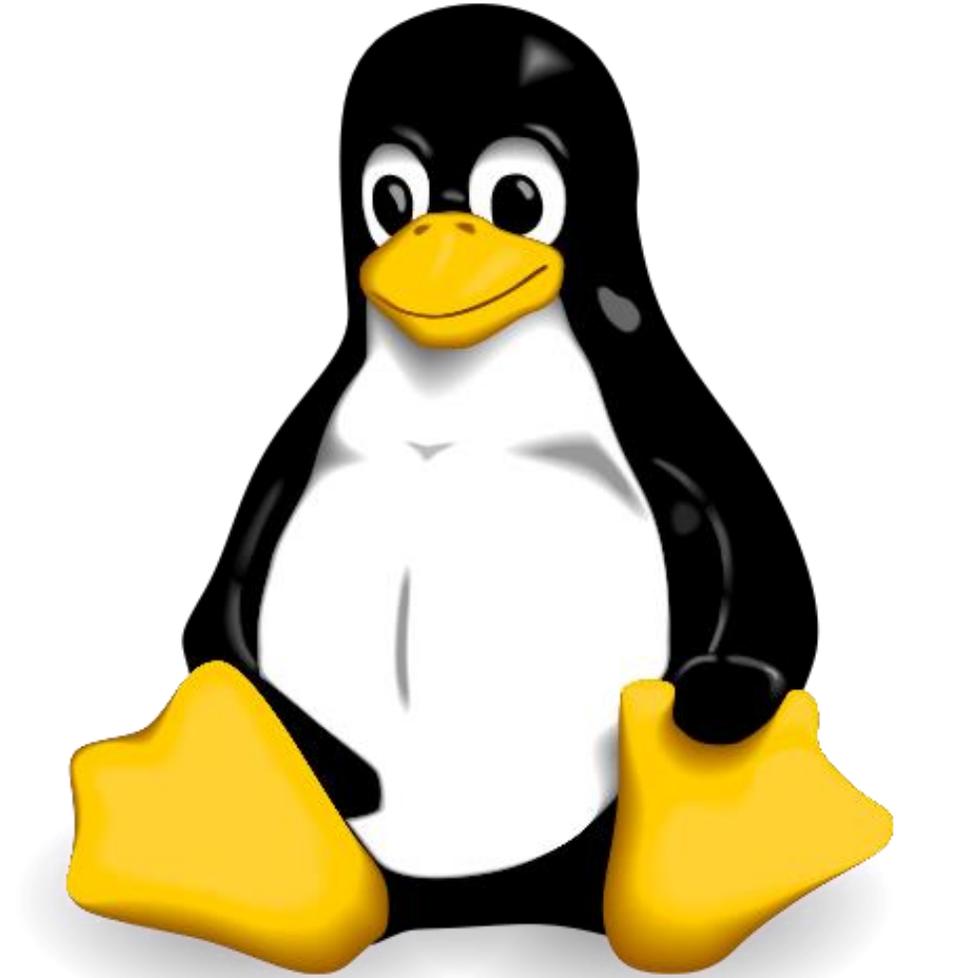


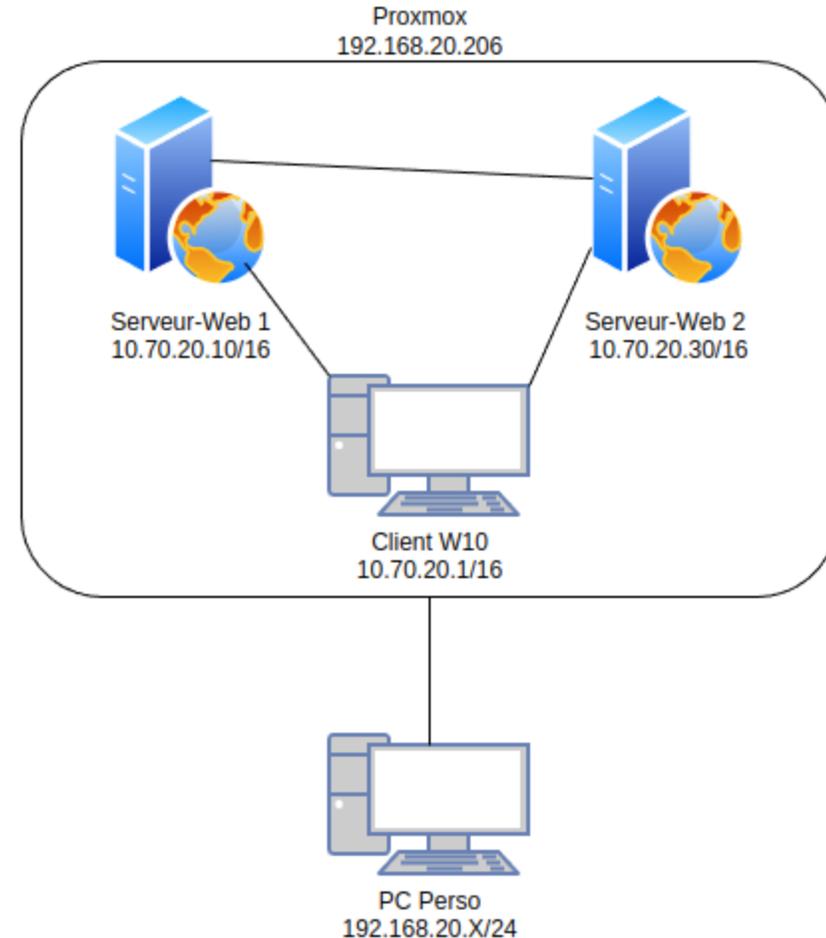
TP – B2 - Heartbeat



Mise en place de notre infrastructure

Pour ce TP, nous aurons :

- 1 VM Windows 10 qui nous servira de client,
- 2 VM Debian 12 sur lesquelles seront héberger nos pages web.



Configuration et paramétrage des serveurs Web

- Test de communication entre nos machines :

```
C:\Users\sio>ping 10.70.20.10

Pinging 10.70.20.10 with 32 bytes of data:
Reply from 10.70.20.10: bytes=32 time<1ms TTL=64

Ping statistics for 10.70.20.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\sio>ping 10.70.20.30

Pinging 10.70.20.30 with 32 bytes of data:
Reply from 10.70.20.30: bytes=32 time<1ms TTL=64

Ping statistics for 10.70.20.30:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
ot@debian12:~# ping 10.70.20.30
PING 10.70.20.30 (10.70.20.30) 56(84) bytes of data.
64 bytes from 10.70.20.30: icmp_seq=1 ttl=64 time=0.376 ms
64 bytes from 10.70.20.30: icmp_seq=2 ttl=64 time=0.169 ms
64 bytes from 10.70.20.30: icmp_seq=3 ttl=64 time=0.198 ms
```

```
root@debian12:~# ping 10.70.20.10
PING 10.70.20.10 (10.70.20.10) 56(84) bytes of data.
64 bytes from 10.70.20.10: icmp_seq=1 ttl=64 time=0.182 ms
64 bytes from 10.70.20.10: icmp_seq=2 ttl=64 time=0.278 ms
64 bytes from 10.70.20.10: icmp_seq=3 ttl=64 time=0.162 ms
```

```
root@debian12:~# ping 10.70.20.1
PING 10.70.20.1 (10.70.20.1) 56(84) bytes of data.
64 bytes from 10.70.20.1: icmp_seq=1 ttl=128 time=0.300 ms
64 bytes from 10.70.20.1: icmp_seq=2 ttl=128 time=0.353 ms
64 bytes from 10.70.20.1: icmp_seq=3 ttl=128 time=0.284 ms
64 bytes from 10.70.20.1: icmp_seq=4 ttl=128 time=0.316 ms
```

Création et paramétrage des serveurs Web

- Modifications des noms des deux serveurs web avec la commande **nano /etc/hostname**,
- Nos serveurs seront renommés par Serveur-1Baptiste & Serveur-2Baptiste

```
GNU nano 7.2
127.0.0.1    localhost
127.0.1.1    Serveur-2Baptiste
10.70.20.10  Serveur-1Baptiste
```

```
GNU nano 7.2
127.0.0.1    localhost
127.0.1.1    Serveur-1Baptiste
10.70.20.30  Serveur-2Baptiste
```

Installation et configuration de Heartbeat

- Nous allons nous remettre sur le réseau public pour avoir accès à Internet, nous installons Heartbeat avec la commande **apt-get install heartbeat**.
- Puis nous rebasculons en réseau interne après avoir fini l'installation.
- Modifications du fichier hosts de nos serveurs, afin de leur attribuer leurs propres adresses.

```
root@Serveur-1Baptiste:~# ping Serveur-1Baptiste
PING Serveur-1Baptiste (127.0.1.1) 56(84) bytes of data.
64 bytes from Serveur-1Baptiste (127.0.1.1): icmp_seq=1 ttl=64 time=0.042 ms
64 bytes from Serveur-1Baptiste (127.0.1.1): icmp_seq=2 ttl=64 time=0.039 ms
64 bytes from Serveur-1Baptiste (127.0.1.1): icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from Serveur-1Baptiste (127.0.1.1): icmp_seq=4 ttl=64 time=0.058 ms
^C
--- Serveur-1Baptiste ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3050ms
rtt min/avg/max/mdev = 0.039/0.047/0.058/0.007 ms
root@Serveur-1Baptiste:~#
```

```
root@Serveur-2Baptiste:/etc/ha.d# ping Serveur-2Baptiste
PING Serveur-2Baptiste (127.0.1.1) 56(84) bytes of data.
64 bytes from Serveur-2Baptiste (127.0.1.1): icmp_seq=1 ttl=64 time=0.070 ms
64 bytes from Serveur-2Baptiste (127.0.1.1): icmp_seq=2 ttl=64 time=0.037 ms
64 bytes from Serveur-2Baptiste (127.0.1.1): icmp_seq=3 ttl=64 time=0.038 ms
^C
--- Serveur-2Baptiste ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2046ms
rtt min/avg/max/mdev = 0.037/0.048/0.070/0.015 ms
root@Serveur-2Baptiste:/etc/ha.d#
```

Mise en place du service

Création des 3 fichiers de configuration dans /etc/ha.d :

- Bash : touch ha.cf && authkeys && haresources

```
root@debian12:/etc/ha.d# touch ha.cf
root@debian12:/etc/ha.d# touch Authkeys
root@debian12:/etc/ha.d# touch haresources
root@debian12:/etc/ha.d# ls -a
.  ..  Authkeys  ha.cf  harc  haresources  rc.d  README.config  resource.d  shellfuncs
```

Mise en place du service

Une fois les 3 fichiers créés, nous allons les compléter:

- ha.cf :
- haresources :
- authkeys :

Nous allons accorder les permissions root au fichier avec la commande :

- **chmod 600 /etc/ha.d/authkeys**
- Dans le fichier authkeys, nous ajoutons les informations suivantes afin que les 2 serveurs communiquent.
- Cette opération est à répéter sur les deux serveurs.

```
GNU nano 7.2
logfile /var/log/ha-log
logfacility local0
keepalive 2
deadtime 30
initdead 120
bcast ens18
udpport 694
auto_failback on
node Serveur-1Baptiste
node Serveur-2Baptiste
```

```
GNU nano 7.2
Serveur-1Baptiste IPaddr::10.70.20.10/16/ens18
```

```
GNU nano 7.2
auth 2
2 sha1 test-ha
```

Mise en place du service

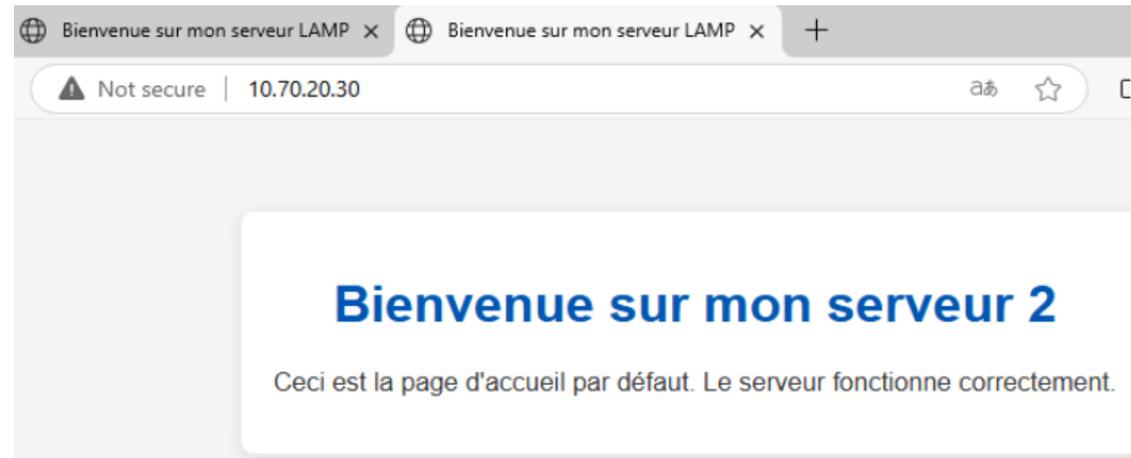
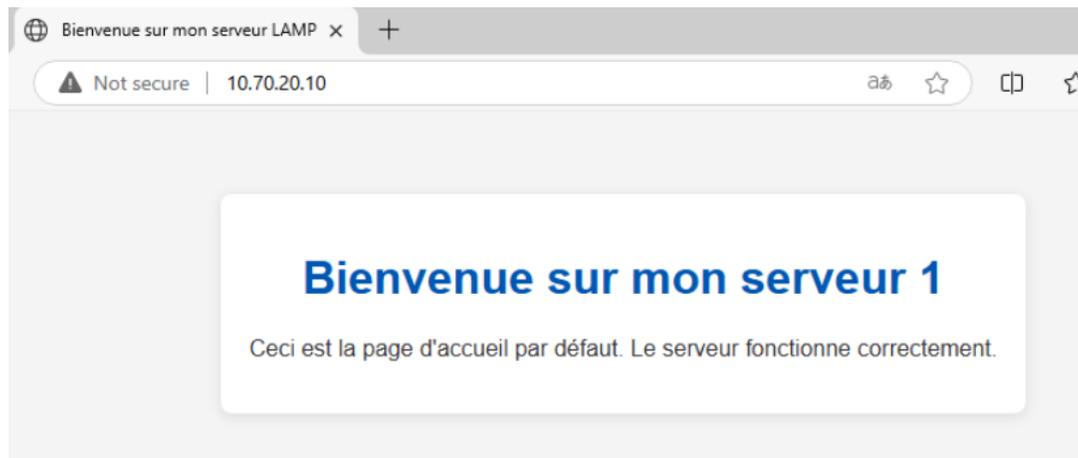
Enfin, on lance notre service heartbeat : **service heartbeat start**

- On vérifie que notre serveur est bien lancé : **service heartbeat status**

```
root@Serveur-2Baptiste:~# service heartbeat start
root@Serveur-2Baptiste:~# service heartbeat status
• heartbeat.service - Heartbeat High Availability Cluster Communication and Membership
  Loaded: loaded (/lib/systemd/system/heartbeat.service; enabled; preset: enabled)
  Active: active (running) since Thu 2025-03-13 11:49:55 CET; 54s ago
  Docs: man:heartbeat(8)
        http://www.linux-ha.org/wiki/Documentation
```

Test du cluster

Après avoir lancé notre service heartbeat, nous allons pouvoir accéder aux pages web sur notre client Windows 10.



Test du cluster

- Enfin nous allons couper volontairement notre serveur 1, notre serveur 2 prendra le relais avec comme IP affichée celle du serveur 1.
- # **service heartbeat stop** pour l'arrêter et # **service heartbeat start** pour le redémarrer.

