

# Projets intégrateurs

## 1. Objectifs pédagogiques

Les objectifs de ses projets sont multiples :

- vous familiariser avec les outils les plus couramment utilisés dans la formation, notamment avec les hyperviseurs VMware, le gestionnaire de version (Git), les outils de développement, les outils de gestion systèmes et réseaux ;
- mettre en pratique les notions de développement logiciel, de systèmes et de réseaux, acquises durant les cours de bases scientifiques ou technologiques du semestre 5 (et vos formations antérieures) ;
- vous exercer à la prise d'autonomie, et à la gestion de votre temps.

Pour atteindre ses objectifs, vous aurez plusieurs mini-projets à effectuer. La validation de ses mini-projets s'effectuera individuellement durant tout le semestre 5. Vous disposerez de plusieurs séances dédiées aux projets totalisant 22h (date butoir le 26 janvier 2026).

## 2. Contraintes générales pour les projets

Les contraintes générales sont les suivantes :

- Vous utiliserez l'hyperviseur VMware ou VirtualBox pour la création et la gestion des machines virtuelles.

VMware est un outil professionnel. Il est plus performant et plus complexe que VirtualBox. Son usage est conseillé avec des outils tel que GNS3 (outils de virtualisation de réseaux et de systèmes utilisé dans le parcours SR).

Vous pouvez les télécharger sur Internet, et les utiliser sans limitation :

- Pour Linux, et windows :
  - VMware Workstation Pro 17.x for Personnal Use (Linux), ou VirtualBox 7.2.2.
- Pour Mac :
  - VMware Fusion 13 for Personnal Use ou VirtualBox 7.2.2.
  - UTM pour les ordinateurs à processeur ARM (pertes de performances, et possibles erreurs).
- Vous développerez avec le langage C, et utiliserez les sockets pour la programmation réseau ;
- Vous devrez utiliser un gestionnaire de version Git ;

- Vous effectuerez les mini-projets seuls (pas de groupe) ;
- Vous pourrez utiliser ChatGPT ou tout autre outil similaire.

### 3. Projets

Documenter chaque mini-Project

Il regardera chaque machine de Prod (faire des snapshots)

1	<p>VyOS : le routeur virtuel. + Switch Virtual</p> <p>Vyos est un système d'exploitation Linux basé sur la distribution Debian. VyOS dispose des mêmes services qu'un routeur matériel. S'il est installé sur un ordinateur, il permet de convertir cet ordinateur en routeur physique. S'il est installé dans une machine virtuelle, il permet d'obtenir un routeur virtuel. Il fonctionne en mode console.</p> <p>Vous apprendrez à installer VyOS et à configurer un routeur virtuelle.</p>	
1.1		<p>Créer une machine virtuelle intégrant le système VyOS. Le système doit être installé sur le disque virtuel de la machine virtuel.</p> <p>Nous vous fournissons la version 1.4.3. <u>Cette version ne peut être utilisée que dans un cadre éducatif.</u></p>
1.2		Intégrer automatiquement le clavier azerty.
1.3		<p>Homogénéiser automatiquement les noms des interfaces réseaux des machines virtuelles clonées :</p> <ul style="list-style-type: none"> <li>• Si une seule carte réseau virtuelle est rattachée à la machine virtuelle clonée, l'interface réseau devra se nommer eth0.</li> <li>• Si deux cartes réseaux virtuelles sont rattachées à la machine virtuelle clonée, les interfaces réseaux devront se nommer eth0 et eth1.</li> <li>• etc...</li> </ul>
2	<p>Alpine : le terminal en mode graphique.</p> <p>Alpine est une distribution Linux graphique très légère. Cette légèreté permet de faire fonctionner en parallèle plusieurs machines virtuelles Alpine sur un ordinateur portable.</p> <p>Vous apprendrez à installer Alpine, à le configurer, à l'optimiser, à l'utiliser comme terminal graphique ou console, à utiliser des outils systèmes et à configurer Grub.</p>	
2.1		<p>Créer une machine virtuelle intégrant le système Alpine.</p> <p>Télécharger et utiliser le fichier iso 64 bits version 3.22.2 (téléchargeable sur le site <a href="https://alpinelinux.org/">https://alpinelinux.org/</a>). Télécharger le fichier iso le plus adapté aux exigences des projets.</p>
2.2		Installer et configurer grub afin de pouvoir démarrer en mode graphique ou

		en mode console.
2.3		Intégrer à minima <i>openssh</i> (suite logicielle de sécurisation de transmissions), <i>filezilla</i> (client <i>FTP</i> ), un navigateur Web, <i>tcpdump</i> (analyseur de trafic en mode console) <i>wireshark</i> (analyseur de trafic en mode graphique) et <i>putty</i> (client pour connexion lien série, <i>telnet</i> et <i>SSH</i> ).
2.4		Installer les services et modules <i>VMware tools</i> (intégration automatique de la souris, et de la configuration automatique de la résolution d'écran de la machine virtuelle).
2.5		Optimiser le système, afin que la taille du disque dur supportant le système soit minimale (le système doit être opérationnel). La commande <i>df</i> permet d'identifier la taille disque utilisée par le système.
2.6		Optimiser le système, afin que la taille mémoire (RAM) supportant le système soit minimale (le système doit être opérationnel). La commande <i>free</i> permet d'identifier la taille mémoire utilisée par le système.
3		<b>MicroCore : le terminal en mode console.</b>
		MicroCore est une distribution Linux en mode console extrêmement légère. Sa légèreté permet de faire fonctionner en parallèle une dizaine de machines virtuelles MicroCore sur un simple ordinateur portable. Une autre particularité de MicroCore est d'être un système non persistant (les modifications du système sont perdues au redémarrage).
		Vous apprendrez à installer MicroCore, à configurer un système non persistant, à l'optimiser, à l'utiliser comme console virtuelle, à utiliser des outils systèmes et à configurer Grub.
3.1		Créer une machine virtuelle intégrant le système MicroCore en mode console. Télécharger et utiliser le fichier iso de la version 6.2 (téléchargeable sur le site <a href="http://tinycorelinux.net/">http://tinycorelinux.net/</a> ). Télécharger les fichier iso le plus adapté aux exigences des projets.
3.2		Intégrer automatiquement le clavier azerty.
3.3		Configurer les dossiers utilisateurs (/home) et logiciels (/opt) afin qu'ils deviennent persistant.
3.4		Intégrer à minima IPv6, les commandes Linux <i>ip</i> (gestion des adresses IP et de la table de routage) et <i>tcpdump</i> (analyseur de trafic).
4		<b><i>ifshow</i> : la commande pour lister les interfaces réseaux de la machine locale.</b>

## Pas d'appel sys à la commande IP

Possible de dire que ça fonctionne à maintien

`ifshow` est une commande en ligne à développer en langage C sous Linux. Elle affiche la liste des interfaces réseaux de la machine locale et leurs caractéristiques. Le comportement de la commande dépend des paramètres qui lui sont associées :

- `ifshow -i ifname`: affiche la liste des préfixes d'adresses IPv4 et IPv6 associées à l'interface nommée `ifname`. (un préfixe s'écrit sous la forme d.d.d.d/p pour IPv4 et d:d:d:d:d:d:p pour IPv6) ;
- `ifshow -a`: affiche la liste des noms des interfaces réseaux, et leurs préfixes d'adresses IPv4 et IPv6.

Vous apprendrez à développer une petite application, sous la forme d'une commande en ligne, récupérant des informations systèmes, et à intégrer cette application dans divers environnements.

4.1	Développer la commande <code>ifshow</code> sur la machine hôte.
4.2	Intégrer la commande <code>ifshow</code> à la machine virtuelle VyOS du mini-projet 1.
4.3	Intégrer la commande <code>ifshow</code> à la machine virtuelle Alpine du mini-projet 2.
4.4	Intégrer la commande <code>ifshow</code> à la machine virtuelle MicroCore du mini-projet 3.

5	NetLab: le réseau virtuel
NetLab est un réseau virtuel (voir figure 1) constitué de machines virtuelles conçues lors des mini-projet 1, 2 et 3. Les adresses des réseaux, et des interfaces réseaux, sont définies dans les tableaux 1 et 2. Vous devez respecter rigoureusement les informations contenues dans ces tableaux.  Vous apprendrez à configurer les équipements d'un petit réseau virtuel (utiliser VyOS comme routeur).	

5.1	Configurer les équipements, afin que tous les terminaux puissent communiquer entre eux en IPv4 (avec protocole de routage statique ou dynamique).
5.2	Configurer les équipements, afin que tous les terminaux puissent communiquer entre eux en IPv6 (avec protocole de routage statique ou dynamique). Les interfaces réseaux des équipements terminaux sont configurés automatiquement (configuration automatique des adresses).

6	<code>ifnetshow</code> : la commande pour lister les interfaces réseaux d'une machine distante.
<code>ifnetshow</code> est une commande en ligne à développer en langage C sous Linux. Elle affiche la liste des interfaces réseaux d'une machine distante et leurs caractéristiques. Le comportement de la commande dépend des paramètres qui	

## Réutiliser le code de ifshow (pas de lib !)

Il faut que le réseau met lab fonctionne

lui sont associées :

- `ifnetshow -n addr -i ifname`: affiche la liste des préfixes d'adresses IPv4 et IPv6 associées à l'interface nommée `ifname` de la machine ayant une interface réseau d'adresse IP `addr`;
- `ifnetshow -n addr -a`: affiche la liste des noms des interfaces réseaux, et leurs préfixes d'adresses IPv4 et IPv6 de la machine ayant une interface réseau d'adresse IP `addr`.

`Ifnetshow` nécessite le développement d'un agent à installer sur les machines distantes. L'agent devrait être persistant (il ne devrait pas être exécuté après chaque appel de `ifnetshow`).

Vous apprendrez à développer une application communicante basée sur les sockets, en mode client/serveur (`ifnetshow` étant le client, et l'agent le serveur). Vous utiliserez le réseau du mini-projet 5 pour valider le fonctionnement de `ifnetshow`. Vous réutiliserez impérativement le code de `ifshow`. Vous apprendrez à structurer votre programme pour le réutiliser.

6.1	Développer la commande <code>ifnetshow</code> et son agent (persistant ou non) sur le système hôte. Utiliser le système de socket pour établir les communications entre les machines.
6.2	Intégrer la commande <code>ifnetshow</code> et son agent (persistant ou non) au système VyOS du mini-projet 1.
6.3	Intégrer la commande <code>ifnetshow</code> et son agent (persistant ou non) au système Alpine du mini-projet 2.
6.4	Intégrer la commande <code>ifnetshow</code> et son agent (persistant ou non) au système MicroCore du mini-projet 3.

7	<code>neighborshow</code> : la commande pour lister l'ensemble des machines voisines.
	<code>neighborshow</code> est une commande en ligne à développer en langage C sous Linux. Elle affiche la liste des noms (sans doublon) des machines voisines (machines d'un même réseau physique) sans connaissance préalable de leurs adresses respectives. <code>neighborshow</code> nécessite le développement d'un agent à installer sur les machines distantes. L'agent devrait être persistant (il ne devrait pas être exécuté après chaque appel de <code>neighborshow</code> ).
7.1	Développer la commande <code>neighborshow</code> sur le système hôte. Utiliser le système de socket pour établir les communications entre les machines. La liste des noms est dynamique. Les noms correspondent au nom système (voir commande <code>linuxhostname</code> ).

7.2	Intégrer la commande <code>neighborshow</code> aux systèmes VyOS, Alpine et MicroCore.
7.3	Développer une extension de <code>neighborshow</code> : <ul style="list-style-type: none"> <li>• <code>neighborshow -hop n</code> : affiche la liste des équipements accessibles en <i>n</i> sauts (si n=1, correspond à <code>neighborshow</code> sans paramètre).</li> </ul>

## 4. Évaluations

Durant le semestre 5, nous définirons des séances durant lesquelles nous validerons les différents projets que vous aurez effectués. Assurez vous bien que vos projets respectent toutes les consignes.

## 5. Annexe : Le réseau NetLab

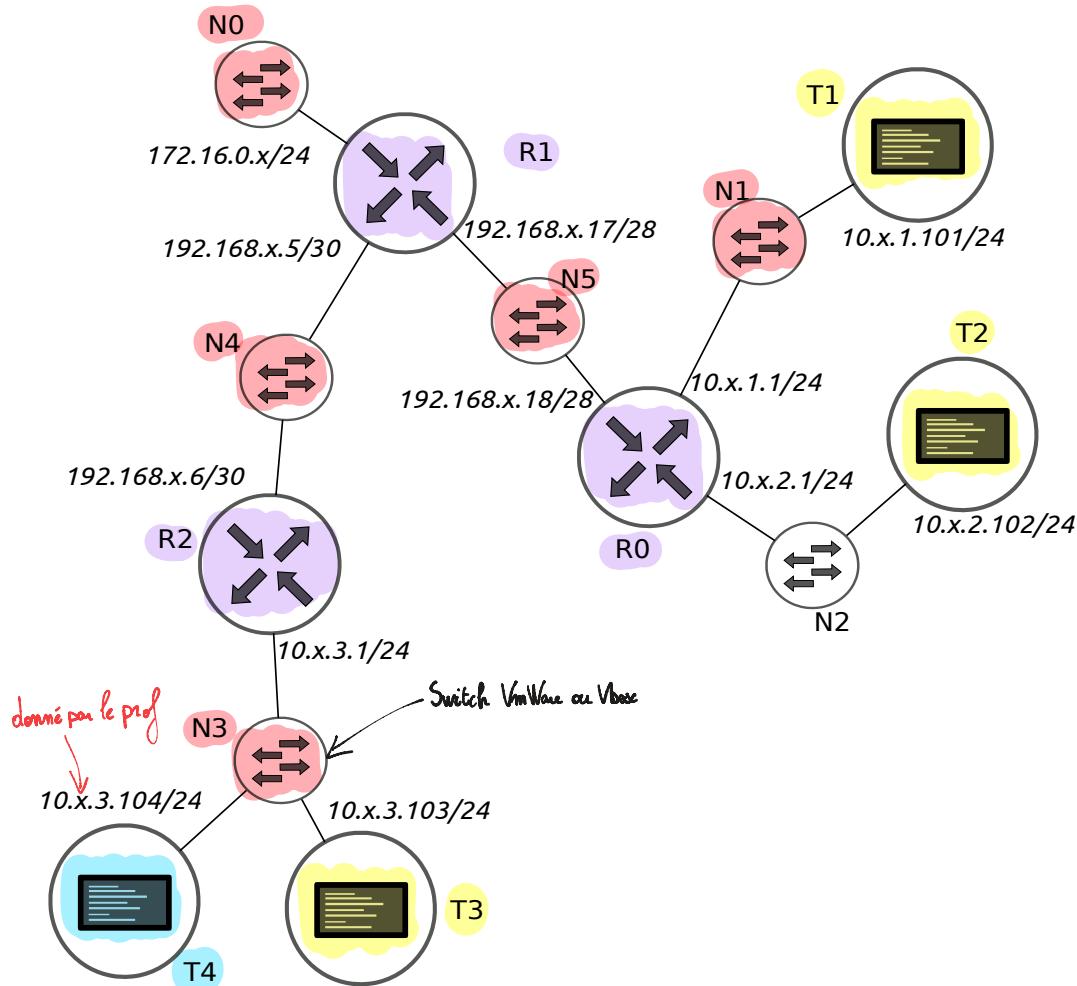


Figure 1: Réseau NetLab

La figure 1 présente le réseau NetLab. Il est constitué des équipements suivants :

- T1, T2 et T3 des terminaux virtuels MicroCore ;
- T4 un terminal virtuel Alpine ;

- R0, R1 et R2 des routeurs virtuels VyOS ;
- N0, N1, N2, N3, N4 et N5 des commutateurs virtuels (intégrés aux hyperviseurs).

Le tableau 1 définit les adresses des réseaux, selon le numéro individuel x. Ce numéro vous sera délivré lors d'une séance de cours.

Le tableau 2 définit les adresses d'interface des équipements. Les adresses d'interface IPv6 des équipements terminaux ne sont pas spécifiées, car leurs configurations doivent être automatiques.

Réseaux	Adresses réseaux
N0	172.16.0.0/24 2002:16:0:0::/64
N4	192.168.x.4/30 3FFE:0:x:4::/64
N5	192.168.x.16/28 3FFE:0:x:16::/64

Réseaux	Adresses réseaux
N1	10.x.1.0/24 2001:0:x:1::/64
N2	10.x.2.0/24 2001:0:x:2::/64
N3	10.x.3.0/24 2001:0:x:3::/64

Tableau 1: Adresses des réseaux de NetLab

	Interfaces	Adresses interfaces
R0	N1	10.x.1.1/24 2001:0:x:1::1/64
	N2	10.x.2.1/24 2001:0:x:2::1/64
	N5	192.168.x.18/28 3FFE:0:x:16::2/64
R2	N3	10.x.3.1/24 2001:0:x:3::/64
	N4	192.168.x.6/30 3FFE:0:x:4::2/64
T1	N1	10.x.1.101/24
T2	N2	10.x.2.102/24

	Interfaces	Adresses interfaces
R1	N0	172.16.0.x/24 2002:16:0:0::x/64
	N4	192.168.x.5/30 3FFE:0:x:4::1/64
	N5	192.168.x.17/28 3FFE:0:x:16::1/64
T3	N3	10.x.3.103/24
T4	N3	10.x.3.104/24

Tableau 2: Adresses des interfaces réseaux des équipements de NetLab