Voici une version corrigée et cohérente du TP 7 sur la mise en œuvre automatisée de VXLAN pour la virtualisation réseau :

TP 7 : Mise en œuvre automatisée de VXLAN pour la virtualisation réseau

Objectif :

L'objectif de ce TP est de comprendre et de configurer un réseau virtualisé à l'aide de **VXLAN** (Virtual Extensible LAN), permettant ainsi la communication entre des machines virtuelles réparties sur différents sous-réseaux. Cette configuration simule un environnement où des hôtes distants peuvent échanger des données comme s'ils appartenaient au même réseau local, malgré des segments physiques différents.

Outils :

- **Linux avec support VXLAN** (Alpine Linux)
- **2 machines virtuelles (KVM)** :
  - **Client**
  - **Cible**

Contexte :

Le **VXLAN** est une technologie permettant d'étendre les réseaux locaux virtuels (VLAN) au-delà des limites physiques du réseau local. Cela se fait en encapsulant les paquets de niveau 2 (Ethernet) dans des paquets de niveau 3 (IP/UDP), créant ainsi des réseaux superposés (overlay networks). Dans ce TP, nous utiliserons VXLAN pour interconnecter deux machines virtuelles situées sur des sous-réseaux distincts et leur permettre de communiquer comme si elles faisaient partie du même réseau.

Plan du TP :

1. Prérequis et installation des outils au sein des hôtes

Vérification du support VXLAN sur Linux : VXLAN est pris en charge nativement par le noyau Linux. Pour vérifier que le module **vxlan** est chargé, utilisez la commande suivante :

```
sudo modprobe vxlan
```

Pour que le module soit chargé automatiquement au démarrage, ajoutez-le à `/etc/modules` :

```
echo "vxlan" | sudo tee -a /etc/modules
```

Installation des outils réseau nécessaires : Si certaines commandes comme `ip` ne sont pas disponibles, assurez-vous que les outils réseau nécessaires (tels que **iproute2**) sont installés. Utilisez la commande suivante pour les installer :

```
sudo sudo apt-get install iproute2 iputils net-tools
```

2. Architecture du réseau VXLAN

Topologie : Dans ce TP, nous utiliserons une topologie simple avec deux machines virtuelles : une appelée **Client** et l'autre **Cible**. Le but est de configurer un tunnel VXLAN pour permettre la communication entre ces deux machines virtuelles, même si elles se trouvent sur des sous-réseaux différents.

Si vous ne disposez pas d'un environnement Cloud comme CloudLab, vous pouvez simuler cette configuration avec deux ordinateurs portables ou des machines physiques connectées à un réseau local.

Objectif : L'objectif est d'interconnecter le **Client** et la **Cible** via un tunnel **VXLAN**, permettant ainsi leur communication malgré des sous-réseaux distincts.

3. Configuration des bridges et des interfaces VXLAN Créer un **VTEP** (VXLAN Tunnel Endpoint) sous Linux est relativement simple et se fait en plusieurs étapes :

a. **Création du tunnel VXLAN** : Pour créer un tunnel VXLAN, utilisez la commande suivante. Remplacez `<vxlan_name>`, `<vxlan_id>`, `<remote_ip>`, `<destination_port>`, et `<remove_nic_interface>` par les valeurs appropriées. L'adresse IP distante est celle du VTEP de l'autre machine.

```
sudo ip link add <vxlan_name> type vxlan id <vxlan_id> remote <remote_ip> dstport <destinati
```

b. **Activation de l'interface VXLAN** : Une fois le tunnel VXLAN créé, vous devez activer l'interface :

```
sudo ip link set <vxlan_name> up
```

c. **Attacher le tunnel VXLAN au bridge virtuel** : Ensuite, vous associez l'interface VXLAN à un bridge virtuel pour permettre la communication avec d'autres interfaces réseau, y compris les machines virtuelles. Utilisez la commande `brctl addif` pour ajouter l'interface VXLAN au bridge.

```
sudo brctl addif <interface_principale> <interface_secondaire>
```

Remplacez `<interface_principale>` par le nom de votre interface réseau principale (par exemple, `eth0`), et `<interface_secondaire>` par l'interface virtuelle (par exemple, `vxlan0`).

4. Orchestration avec Ansible et Tofu

- Modifiez les fichiers **hosts**, **infra.yml**, **node0/main.tf**, et **node1/main.tf**.
- Utilisez le script **run.sh** pour déployer l'infrastructure.
- Veillez à vérifier le plan d'adressage et les différents rôles des éléments réseau (cartes réseau, switch virtuel, etc.) avant de déployer.

5. Tests de connectivité et segmentation de réseau Une fois la configuration de base terminée, il est important de tester la connectivité entre les hôtes sur le réseau VXLAN.

a. **Vérification de la connectivité entre les machines** : Depuis le **Client**, testez la connectivité vers la **Cible** en utilisant la commande `ping` ou `ip route`. Cela permettra de vérifier que les paquets circulent correctement à travers le tunnel VXLAN.

Exemple de commande :

```
ping <ip_cible>
```

b. **Vérification de l'encapsulation VXLAN** : Sur la **Cible**, vous pouvez utiliser **tcpdump** pour capturer et analyser le trafic VXLAN entrant. Cela permet de vérifier que les paquets sont correctement encapsulés dans des paquets VXLAN. Utilisez la commande suivante pour écouter sur l'interface VXLAN :

```
sudo tcpdump -i <interface_vxlan>
```

Vous devriez voir des paquets UDP encapsulés portant les informations de VXLAN dans l'en-tête.

---

6. Débriefing et questions de réflexion Après avoir effectué la configuration et les tests, réfléchissez aux points suivants pour mieux comprendre l'architecture VXLAN et ses avantages :

- **Quelle est la configuration d'un VXLAN dans un environnement multi-hôtes réel ?**
- **Quels sont les avantages de VXLAN par rapport aux VLAN traditionnels ?**
- **Comment le VXLAN permet-il d'étendre les réseaux virtuels au-delà des limites physiques ?**
- **Pourquoi VXLAN est-il considéré comme un domaine de diffusion ?**
- **Quelles précautions faut-il prendre lors de la configuration d'un réseau VXLAN ?**
