Routage inter-VLAN dans un contexte cloud

Philippe Latu
philippe.latu(at)inetdoc.net

https://www.inetdoc.net

Résumé

Le routage inter-VLAN est très largement utilisé dans l'interconnexion entre les réseaux Ethernet. Les manipulations présentées dans ces travaux pratiques illustrent l'interconnexion entre un réseau d'hébergement de type Cloud IAAS (Infrastructure As A Service) et un réseau de conteneurs LXD raccordés à l'aide de la technologie MACVLAN.

On introduit aussi un premier niveau de filtrage induit par le recours à la traduction d'adresses entre les deux réseaux interconnectés.

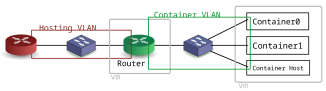


Table des matières

1. Copyright et Licence

1.1. Méta-information

Ce document est écrit avec DocBook XML sur un système Debian GNU/Linux. Il est disponible en version imprimable au format PDF : `interco.inter-vlan-cloud.qa.pdf`.

1.2. Conventions typographiques

Tous les exemples d'exécution des commandes sont précédés d'une invite utilisateur ou prompt spécifique au niveau des droits utilisateurs nécessaires sur le système.

- Toute commande précédée de l'invite `$` ne nécessite aucun privilège particulier et peut être utilisée au niveau utilisateur simple.

- Toute commande précédée de l'invite # nécessite les privilèges du super-utilisateur.

2. Topologies logiques et virtuelles

Les définitions importantes sur les réseaux locaux virtuels et le routage associé sont présentées dans l'article Routage Inter-VLAN

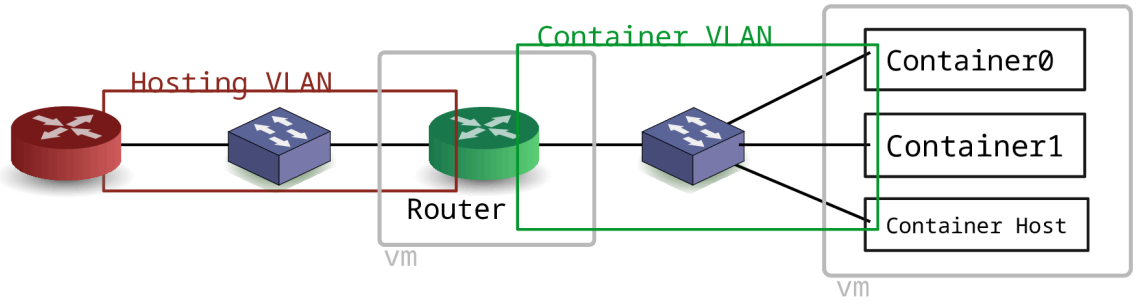
On rappelle simplement que la notion de réseau local virtuel ou VLAN permet de constituer des groupes logiques dans les réseaux Ethernet au niveau liaison de la modélisation. Lors du raccordement entre les équipements (commutateurs, routeurs, serveurs), certaines liaisons doivent véhiculer le trafic de plusieurs réseaux locaux virtuels (VLANs). Ces liaisons sont baptisées trunks dans le jargon. Pour distinguer le trafic appartenant à chaque réseau local, on ajoute à la trame une balise définie par le standard IEEE 802.1Q. C'est cette étiquetage de trame qui permet la distribution des domaines de diffusion entre plusieurs équipements physiques distincts.

On atteint ainsi un objectif très important. Il est possible de concevoir une topologie logique de réseau totalement indépendante de la topologie physique.

Réseau virtuel ou pas, il ne faut pas oublier les éléments suivants sur la segmentation des réseaux locaux.

- Une interface de commutateur délimite un domaine de collision.
- Une interface de routeur délimite à la fois un domaine de collision et un domaine de diffusion.

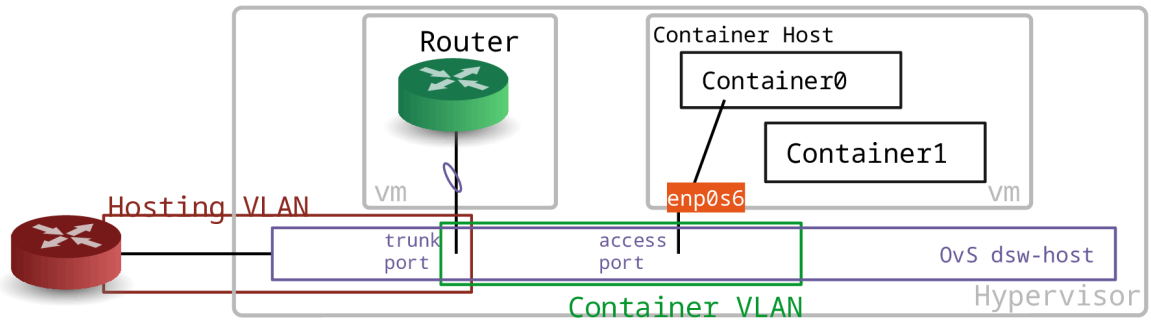
La représentation de la topologie logique ci-dessous montre que le routeur de couleur verte assure l'interconnexion entre un réseau d'infrastructure appelé Hosting VLAN et un réseau de conteneurs appelé Container VLAN. Les deux rectangles en gris “matérialisent“ les machines virtuelles qui sont utilisées pour les manipulations.



Topologie logique

La représentation de la topologie vue sous l'angle de l'hébergement sur un système hôte hyperviseur montre que les deux VLANs sont présents sur le commutateur virtuel de couche distribution appelé dsw-host. Ce commutateur appartient au système hôte. Il assure le raccordement entre les réseaux physiques et virtualisés. On retrouve le routeur de couleur verte raccordé avec un lien unique sur lequel le trafic des deux VLANs doit transiter.

Côté conteneurs, seule l'interface `enp0s6` est raccordé via un lien en mode accès. La technologie macVLAN permet d'utiliser plusieurs adresses MAC sur une même interface réseau.



Topologie hébergée

3. Plan d'adressage des réseaux d'hébergement et de conteneurs

Attention ! Les adresses de passerelle côté hébergement sont déjà implantées dans l'infrastructure de travaux pratiques tandis que les adresses de passerelle côté conteneurs sont à implanter sur le routeur.

Tableau 1. Affectations des adresses de passerelle - Groupe 1

Planète (deux machines virtuelles)	Passerelle - Hosting VLAN	VLAN & Passerelle - Container VLAN	
Christophsis	`192.168.37.9/29` `2001:678:3fc:133::1/64`	400	`203.0.113.1/24` `fda0:7a62:190::1/64`
Corellia	`172.22.9.33/29` `2001:678:3fc:151::1/64`	401	`203.0.113.1/24` `fda0:7a62:191::1/64`
Delaya	`10.31.0.193/26` `2001:678:3fc:136::1/64`	402	`203.0.113.1/24` `fda0:7a62:192::1/64`
Kashyyyk	`10.30.5.129/26` `2001:678:3fc:131::1/64`	403	`203.0.113.1/24` `fda0:7a62:193::1/64`
Korriban	`10.9.10.129/26` `2001:678:3fc:14f::1/64`	404	`203.0.113.1/24` `fda0:7a62:194::1/64`
Kessel	`10.30.6.65/28` `2001:678:3fc:132::1/64`	405	`203.0.113.1/24` `fda0:7a62:195::1/64`
Mygeeto	`10.9.15.17/29` `2001:678:3fc:150::1/64`	406	`203.0.113.1/24` `fda0:7a62:196::1/64`
Nelvaan	`10.31.1.145/28` `2001:678:3fc:137::1/64`	407	`203.0.113.1/24` `fda0:7a62:197::1/64`
Rattatak	`192.168.10.81/28` `2001:678:3fc:145::1/64`	408	`203.0.113.1/24` `fda0:7a62:198::1/64`
Saleucami	`192.168.12.17/29` `2001:678:3fc:138::1/64`	409	`203.0.113.1/24` `fda0:7a62:199::1/64`
Taris	`10.8.11.9/29` `2001:678:3fc:14b::1/64`	410	`203.0.113.1/24` `fda0:7a62:19a::1/64`
Teth	`10.0.10.33/27` `2001:678:3fc:13c::1/64`	411	`203.0.113.1/24` `fda0:7a62:19b::1/64`
Utapau	`172.21.12.17/29` `2001:678:3fc:14c::1/64`	412	`203.0.113.1/24` `fda0:7a62:19c::1/64`
Yavin	`172.19.9.65/26` `2001:678:3fc:13b::1/64`	413	`203.0.113.1/24` `fda0:7a62:19d::1/64`

Tableau 2. Affectations des adresses de passerelle - Groupe 2

Planète (deux machines virtuelles)	Passerelle - Hosting VLAN	VLAN & Passerelle - Container VLAN	
Alderaan	`172.17.64.129/25` `2001:678:3fc:64::1/64`	220	`203.0.113.1/24` `fda0:7a62:dc::1/64`
Bespin	`172.20.135.65/28` `2001:678:3fc:87::1/64`	221	`203.0.113.1/24` `fda0:7a62:dd::1/64`
Centares	`172.18.4.1/22` `2001:678:3fc:65::1/64`	222	`203.0.113.1/24` `fda0:7a62:de::1/64`

Planète (deux machines virtuelles)	Passerelle - Hosting VLAN	VLAN & Passerelle - Container VLAN	
Coruscant	`172.20.136.81/28` `2001:678:3fc:88::1/64`	223	`203.0.113.1/24` `fda0:7a62:df::1/64`
Dagobah	`10.3.2.1/23` `2001:678:3fc:66::1/64`	224	`203.0.113.1/24` `fda0:7a62:e0::1/64`
Endor	`172.24.132.17/28` `2001:678:3fc:84::1/64`	225	`203.0.113.1/24` `fda0:7a62:e1::1/64`
Felucia	`10.6.8.1/23` `2001:678:3fc:69::1/64`	226	`203.0.113.1/24` `fda0:7a62:e2::1/64`
Geonosis	`172.20.131.33/29` `2001:678:3fc:83::1/64`	227	`203.0.113.1/24` `fda0:7a62:e3::1/64`
Hoth	`10.7.10.1/23` `2001:678:3fc:6a::1/64`	228	`203.0.113.1/24` `fda0:7a62:e4::1/64`
Jakku	`172.20.130.25/29` `2001:678:3fc:82::1/64`	229	`203.0.113.1/24` `fda0:7a62:e5::1/64`
Kamino	`192.168.107.1/25` `2001:678:3fc:6b::1/64`	230	`203.0.113.1/24` `fda0:7a62:e6::1/64`
Mustafar	`192.168.110.129/25` `2001:678:3fc:6e::1/64`	231	`203.0.113.1/24` `fda0:7a62:e7::1/64`
Naboo	`192.168.122.1/28` `2001:678:3fc:7a::1/64`	232	`203.0.113.1/24` `fda0:7a62:e8::1/64`
Tatooine	`172.19.115.193/26` `2001:678:3fc:73::1/64`	233	`203.0.113.1/24` `fda0:7a62:e9::1/64`

4. Raccordement au commutateur de distribution

Dans cette section, on étudie le raccordement des deux machines virtuelles au commutateur de distribution sur le système hôte.

Q1. Comment contrôler la configuration des ports du commutateur de distribution sur le système hôte ?

Le commutateur virtuel implanté sur le système hôte est géré par Open vSwitch. On fait donc appel à la commande ovs-vsctl pour afficher la configuration des ports. Le mot clé dans le cas de cette question est `vlan_mode`.

- Pour le port de raccordement du routeur, on obtient :

```
$ sudo ovs-vsctl list port tap100 | grep vlan_mode
vlan_mode           : trunk
```

- Pour le port de raccordement du serveur de conteneur, on obtient :

```
$ sudo ovs-vsctl list port tap1 | grep vlan_mode
vlan_mode           : access
```

Q2. Comment contrôler le numéro de VLAN attribué au port en mode accès du commutateur de distribution sur le système hôte ?

On reprend la même commande que dans la question précédente avec le mot clé `tag`.

```
$ sudo ovs-vsctl list port tap1 | grep tag
tag                 : 430
```

Q3. Comment affecter le numéro de VLAN attribué au port en mode accès du commutateur de distribution sur le système hôte ?

On reprend à nouveau la même commande avec l'option `set`.

```
$ sudo ovs-vsctl set port tap1 tag=430
```

Les valeurs données dans l'exemple ci-dessus sont à changer suivant les attributions de la section Plan d'adressage des réseaux d'hébergement et de conteneurs.

Q4. Comment s'assurer que le port du commutateur est bien configuré à chaque nouveau lancement de machine virtuelle ?

On place les commandes de configuration dans une section dédiée du script de lancement. Voici deux exemples de script de lancement :

```
#!/bin/bash
RAM=1024

echo "Lancement VM rôle routeur"
CORDON_TRUNK=100

sudo ovs-vsctl set port tap$CORDON_TRUNK vlan_mode=trunk

$HOME/vm/scripts/ovs-startup.sh routeur.qcow2 $RAM $CORDON_TRUNK
```

```
#!/bin/bash
RAM=1024

echo "Lancement VM rôle serveur de conteneurs"
CORDON_ACCES=1
VLAN_ACCES=430

sudo ovs-vsctl set port tap$CORDON_ACCES vlan_mode=access
sudo ovs-vsctl set port tap$CORDON_ACCES tag=$VLAN_ACCES

$HOME/vm/scripts/ovs-startup.sh serveur.qcow2 $RAM $CORDON_ACCES
```

Les numéros de port et de VLAN donnés dans les exemples ci-dessus sont à changer suivant le contexte.

5. Rôle routeur

Dans cette section, on étudie la machine virtuelle qui joue le rôle de routeur entre le réseau d'hébergement et un réseau de conteneurs. Pour traiter les questions, il est nécessaire de mettre en œuvre une maquette avec un adressage indépendant de la liste de la section Plan d'adressage des réseaux d'hébergement et de conteneurs. Voici les choix effectués pour la maquette.

Tableau 3. Plan d'adressage des réseaux de la maquette

Réseau	Numéro VLAN	Adresses de passerelles	Numéro interface tap
Hébergement VLAN rouge	300	`10.141.0.161/27` `2001:678:3fc:12c::1/64`	100
Services VLAN vert	430	`192.0.2.1/24` `fda0:7a62:1ae::1/64`	1

5.1. Configuration des interfaces du routeur

Une fois la machine virtuelle routeur lancée, les premières étapes consistent à lui attribuer un nouveau nom et à configurer les interfaces réseau pour joindre les hôtes voisins.

Q5. Comment changer le nom de la machine virtuelle ?

Il faut éditer les deux fichiers `/etc/hosts` et `/etc/hostname` en remplaçant le nom de l'image maître vm0 par le nom voulu. Il est ensuite nécessaire de redémarrer pour que le nouveau nom soit pris en compte par tous les outils du système.

```
etu@vm0:~$ sudo sed -i 's/vm0/rtr/g' /etc/hosts
[sudo] Mot de passe de etu :
etu@vm0:~$ sudo sed -i 's/vm0/rtr/g' /etc/hostname
sudo: impossible de résoudre l'hôte vm0: Échec temporaire dans la résolution du nom
etu@vm0:~$ sudo reboot
```

Q6. Comment appliquer les configurations réseau IPv4 et IPv6 à partir de l'unique interface du routeur ?

Consulter les pages de manuels du fichier de configuration système à l'aide de la commande man interfaces.

Il existe plusieurs possibilités pour configurer une interface réseau. Dans le contexte de ces manipulations, on utilise le fichier de configuration fourni par la distribution Debian GNU/Linux : `/etc/network/interfaces`.

La configuration de base fournie avec l'image maître suppose que l'interface obtienne un bail DHCP pour la partie IPv4 et une configuration automatique via SLAAC pour la partie IPv6. Cette configuration par défaut doit être éditée et remplacée. Il faut configurer trois interfaces.

- L'interface principale doit être placée en mode manuel (manual). Elle doit être activée/désactivée au niveau de la couche liaison.
- Une interface doit être créée pour le réseau d'hébergement avec le numéro de VLAN désigné dans le tableau de la section Plan d'adressage des réseaux d'hébergement et de conteneurs. Cette interface doit désigner les passerelles IPv4 et IPv6 de façon à joindre l'Internet.
- Une interface doit être créée pour le réseau des conteneurs avec, là encore, le bon numéro de VLAN. Les adresses IPv4 et IPv6 de cette interface deviendront les passerelles du serveur et des conteneurs.

Voici une copie du fichier `/etc/network/interfaces` de la maquette.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s6
iface enp0s6 inet manual
 up ip link set dev $IFACE up
 down ip link set dev $IFACE down

auto enp0s6.300
iface enp0s6.300 inet static
 address 10.141.0.162/27
 gateway 10.141.0.161
 dns-nameserver 9.9.9.9

iface enp0s6.300 inet6 static
 address 2001:678:3fc:12c::2/64
 gateway 2001:678:3fc:12c::1

auto enp0s6.430
iface enp0s6.430 inet static
 address 192.0.2.1/24

iface enp0s6.430 inet6 static
 address fda0:7a62:1ae::1/64
```

Une fois le fichier de configuration en place, il est préférable de redémarrer la machine virtuelle de façon à vérifier que la configuration des interfaces est bien appliquée après chaque réinitialisation.

Q7. Quels sont les tests de connectivité réalisables après application de la nouvelle configuration des interfaces réseau ?

Relever l'état des trois interfaces et procédez aux tests en respectant les couches de la modélisation.

La commande ip addr ls permet de relever l'état de la configuration pour chaque interface.

```
$ ip addr ls | grep state
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
2: enp0s6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
3: enp0s6.300@enp0s6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
4: enp0s6.430@enp0s6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
```

Sans la confirmation que la configuration du serveur de conteneurs est prête, c'est du côté hébergement et accès Internet qu'il faut orienter les tests. Classiquemt, on cherche à joindre la passerelle en premier puis l'Internet ensuite via des requêtes ICMP. Enfin, on effectue un test de couche application avec une requête DNS.

```
$ ping -q -c2 10.141.0.161
PING 10.141.0.161 (10.141.0.161) 56(84) bytes of data.

--- 10.141.0.161 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.990/1.188/1.387/0.198 ms
$ ping -q -c2 9.9.9.9
PING 9.9.9.9 (9.9.9.9) 56(84) bytes of data.

--- 9.9.9.9 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 12.218/12.227/12.237/0.009 ms
```

```
$ ping -q -c2 2001:678:3fc:12c::1
PING 2001:678:3fc:12c::1(2001:678:3fc:12c::1) 56 data bytes

--- 2001:678:3fc:12c::1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.023/1.164/1.306/0.141 ms
$ ping -q -c2 2620:fe::fe
PING 2620:fe::fe(2620:fe::fe) 56 data bytes

--- 2620:fe::fe ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 39.999/40.391/40.784/0.392 ms
```

```
$ host quad9.net
quad9.net has address 216.21.3.77
quad9.net has IPv6 address 2620:0:871:9000::77
quad9.net mail is handled by 20 mx2.quad9.net.
quad9.net mail is handled by 100 keriomail.pch.net.
quad9.net mail is handled by 5 mx1.quad9.net.
```

5.2. Activation de la fonction routage

Sans modification de la configuration par défaut, un système GNU/Linux n'assure pas la fonction de routage du trafic d'une interface réseau à une autre.

L'activation du routage correspond à un réglage de paramètres du sous-système réseau du noyau Linux. L'outil qui permet de consulter et modifier les réglages de paramètre sur le noyau est appelé sysctl. Son fichier de configuration principal est `/etc/sysctl.conf`.

Q8. Comment activer le routage dans le sous-système réseau du noyau Linuxe ?

Utiliser la commande sysctl pour effectuer des recherches et identifier les paramètres utiles. Par exemple : `$ sudo sysctl -a -r ".*forward.*"`.

Le fichier `/etc/sysctl.conf` contient des commentaires qui guident facilement vers les bons paramètres.

Attention ! Il ne faut pas oublier d'appliquer les nouvelles valeurs des paramètres de configuration.

Voici un extrait du fichier `/etc/sysctl.conf` du routeur de la maquette après édition.

```
$ egrep -v '(^#|^$)' /etc/sysctl.conf
net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.all.rp_filter=1
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
net.ipv4.conf.all.log_martians = 1
```

Voici une copie d'écran de l'application des nouveaux paramètres.

```
$ sudo sysctl --system
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
kernel.pid_max = 4194304
* Applying /etc/sysctl.d/99-sysctl.conf ...
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.log_martians = 1
* Applying /usr/lib/sysctl.d/protect-links.conf ...
fs.protected_fifos = 1
fs.protected_hardlinks = 1
fs.protected_regular = 2
fs.protected_symlinks = 1
* Applying /etc/sysctl.conf ...
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.log_martians = 1
```

Q9. Quelles sont les conditions à réunir pour tester le fonctionnement du routage ?

Rechercher comment utiliser l'analyseur réseau tshark pour caractériser l'acheminement du trafic d'un réseau à l'autre.

Le plan d'adressage prévoit d'utiliser des préfixes ayant une portée locale pour les réseaux de conteneurs. Il n'est donc pas possible de passer par une requête ICMP pour caractériser l'accès aux réseaux distants. En effet, l'adresse source n'est pas reconnue par l'hôte distant et les routeurs de l'Internet ne disposent d'aucune solution pour joindre le réseau des conteneurs.

Voici un extrait de capture qui montre que le serveur de conteneur cherche à joindre un hôte sur l'Internet sans succès. Cette capture étant réalisée sur l'interface réseau côté hébergement, elle montre que le trafic est bien echeminé d'un réseau à l'autre.

```
$ tshark -i enp0s6.300
Capturing on 'enp0s6.300'
    1 0.000000000    192.0.2.2 → 9.9.9.9      DNS 81 Standard query 0xbdab A 1.debian.pool.ntp.org
    2 0.000056361    192.0.2.2 → 9.9.9.9      DNS 81 Standard query 0xab92 AAAA 1.debian.pool.ntp.org
```

5.3. Activation de la traduction d'adresses

Le résultat de la question ci-dessus montre que les hôtes situés dans le réseau des conteneurs ne peuvent pas joindre l'Internet puisque les préfixes réseau utilisés ont une portée limitée.

Q10. Quels sont les paquets qui fournissent les outils de gestion de la traduction d'adresses ?

Rechercher les paquets relatifs au filtrage et à la gestion des règles de pare-feux.

Sur les systèmes GNU/Linux, le système de pare-feux comprend une partie "espace utilisateur" appelée iptables et une partie "noyau" appelée netfilter.

C'est la partie "espace utilisateur" qui nous intéresse ici.

```
$ aptitude search iptables
p   arno-iptables-firewall          - single- and multi-homed firewall script wi
p   golang-github-coreos-go-iptable - Go bindings for iptables utility
i   iptables                        - administration tools for packet filtering
p   iptables-converter              - convert iptables-commands from a file to i
p   iptables-converter-doc          - convert iptables-commands from a file to i
p   iptables-netflow-dkms           - iptables target which generates netflows
p   iptables-persistent             - boot-time loader for netfilter rules, ipta
p   libiptables-chainmgr-perl       - Perl extension for manipulating iptables p
p   libiptables-parse-perl          - Perl extension for parsing iptables firewa
p   python-iptables-doc             - documentation for the python-iptables libr
p   python3-iptables                - Python bindings for iptables (Python 3 int
```

On voit que le paquet iptables est déjà installé et qu'il ne manque que la gestion de la sauvegarde des règles de filtrage et traduction d'adresses.

```
$ sudo apt install iptables-persistent
```

Q11. Quelles sont les règles à appliquer pour assurer une traduction des adresses sources en sortie sur le réseau hébergement ?

Rechercher dans les pages de manuel de la commande iptables.

C'est la cible `MASQUERADE` qui nous intéresse. Voici un exemple de règles de traduction des adresses sources pour la maquette.

```
$ sudo iptables -t nat -A POSTROUTING -o enp0s6.300 -j MASQUERADE
$ sudo sh -c "iptables-save >/etc/iptables/rules.v4"
```

```
$ sudo ip6tables -t nat -A POSTROUTING -o enp0s6.300 -j MASQUERADE
$ sudo sh -c "ip6tables-save >/etc/iptables/rules.v6"
```

Q12. Comment caractériser le fonctionnement de la traduction d'adresses sources ?

Rechercher dans les pages de manuel de la commande iptables les options d'affichage du décompte du trafic traité.

Voici un exemple d'affichage pour le trafic IPv4 uniquement.

```
$ sudo iptables -vnL -t nat
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    8   598 MASQUERADE  all  --  *      enp0s6.300  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

5.4. Activation de la configuration IPv6 automatique pour le réseau de conteneurs

Pour que les hôtes du réseau de conteneurs obtiennent automatiquement une configuration IPv6, il faut que le routeur assure les annonces auprès de ces voisins. Un moyen simple pour assurer la configuration SLAAC des hôtes voisins du routeur consiste à utiliser le paquet radvd.

On débute par l'installation de ce paquet.

```
$ sudo apt install radvd
...
Préparation du dépaquetage de .../radvd_1%3a2.17-2+b1_amd64.deb ...
Dépaquetage de radvd (1:2.17-2+b1) ...
Paramétrage de radvd (1:2.17-2+b1) ...
Job for radvd.service failed because the control process exited with error code.
See "systemctl status radvd.service" and "journalctl -xe" for details.
invoke-rc.d: initscript radvd, action "start" failed.
● radvd.service - Router advertisement daemon for IPv6
     Loaded: loaded (/lib/systemd/system/radvd.service; disabled; vendor preset: enabled)
     Active: failed (Result: exit-code) since Sun 2020-09-13 22:32:10 CEST; 17ms ago
       Docs: man:radvd(8)
    Process: 2814 ExecStartPre=/usr/sbin/radvd --logmethod stderr_clean --configtest (code=exited, status=1/FAILURE)
```

On voit que le lancement du service a échoué.

Q13. Comment configurer le service radvd pour publier les annonces côté conteneurs ?

Rechercher les options utiles dans les pages de manuel du service : `man radvd.conf`.

Voici une copie du fichier de configuration `/etc/radvd.conf` de la maquette.

```
interface enp0s6.430
{
        AdvSendAdvert on;

        prefix fda0:7a62:1ae::/64
        {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr on;
        };

        RDNSS 2620:fe::fe
        {
        };
};
```

Attention ! Une fois le fichier créé, il ne faut pas oublier de redémarrer le service et de contrôler l'état de son fonctionnement.

```
$ sudo systemctl enable radvd
$ sudo systemctl restart radvd
$ systemctl status radvd
● radvd.service - Router advertisement daemon for IPv6
     Loaded: loaded (/lib/systemd/system/radvd.service; disabled; vendor preset: enabled)
     Active: active (running) since Sun 2020-09-13 22:39:34 CEST; 5s ago
       Docs: man:radvd(8)
    Process: 2890 ExecStartPre=/usr/sbin/radvd --logmethod stderr_clean --configtest (code=exited, status=0/SUC
    Process: 2891 ExecStart=/usr/sbin/radvd --logmethod stderr_clean (code=exited, status=0/SUCCESS)
   Main PID: 2892 (radvd)
      Tasks: 2 (limit: 1142)
     Memory: 1.3M
     CGroup: /system.slice/radvd.service
             ├─2892 /usr/sbin/radvd --logmethod stderr_clean
             └─2893 /usr/sbin/radvd --logmethod stderr_clean

sept. 13 22:39:34 rtr systemd[1]: Starting Router advertisement daemon for IPv6...
sept. 13 22:39:34 rtr radvd[2890]: config file, /etc/radvd.conf, syntax ok
sept. 13 22:39:34 rtr radvd[2891]: version 2.17 started
sept. 13 22:39:34 rtr systemd[1]: Started Router advertisement daemon for IPv6.
```

Enfin, le résultat doit se retrouver sur la configuration réseau de l'interface du serveur de conteneurs.

```
etu@srvr:~$ ip addr ls dev enp0s6
2: enp0s6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether b0:ad:ca:fe:00:01 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.2/24 brd 192.0.2.255 scope global enp0s6
       valid_lft forever preferred_lft forever
    inet6 fda0:7a62:1ae:0:b2ad:caff:fefe:1/64 scope global dynamic mngtmpaddr
       valid_lft 86124sec preferred_lft 14124sec
    inet6 fe80::b2ad:caff:fefe:1/64 scope link
       valid_lft forever preferred_lft forever
```

6. Rôle serveur de conteneurs

6.1. Configuration des interfaces du routeur

Une fois la machine virtuelle serveur de conteneurs lancée, les premières étapes consistent à lui attribuer un nouveau nom et à configurer les interfaces réseau pour joindre le routeur voisin et l'Internet.

Q14. Comment changer le nom de la machine virtuelle ?

Il faut éditer les deux fichiers `/etc/hosts` et `/etc/hostname` en remplaçant le nom de l'image maître vm0 par le nom voulu. Il est ensuite nécessaire de redémarrer pour que le nouveau nom soit pris en compte par tous les outils du système.

```
etu@vm0:~$ sudo sed -i 's/vm0/srvr/g' /etc/hosts
etu@vm0:~$ sudo sed -i 's/vm0/srvr/g' /etc/hostname
sudo: impossible de résoudre l'hôte vm0: Échec temporaire dans la résolution du nom
etu@vm0:~$ sudo reboot
```

Q15. Comment appliquer la configuration réseau IPv4 et IPv6 de l'interface du serveur ?

Consulter les pages de manuels du fichier de configuration système à l'aide de la commande man interfaces.

Il existe plusieurs possibilités pour configurer une interface réseau. Dans le contexte de ces manipulations, on utilise le fichier de configuration fourni par la distribution Debian GNU/Linux : `/etc/network/interfaces`.

La configuration de base fournie avec l'image maître suppose que l'interface obtienne un bail DHCP pour la partie IPv4 et une configuration automatique via SLAAC pour la partie IPv6.

La configuration IPv4 par défaut doit être éditée et remplacée par une configuration statique tandis que la configuration IPv6 doit toujours se faire automatiquement via SLAAC.

Voici une copie du fichier `/etc/network/interfaces` de la maquette.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s6
iface enp0s6 inet static
        address 192.0.2.2/24
        gateway 192.0.2.1
        dns-nameserver 9.9.9.9
```

6.2. Installation du gestionnaire de conteneurs LXD

Sur l'hôte serveur, la gestion des conteneurs est confiée à LXD. Pour des raisons de rapidité de mise en œuvre, on choisit de passer par le gestionnaire de paquets snapd pour l'installation des outils.

Q16. Comment installer le gestionnaire de paquets snap sur une distribution Debian GNU/Linux ?

Effectuer une recherche dans les paquets fournis via APT.

Il existe tout simplement un paquet appelé snapd.

```
$ sudo apt install snapd
```

Q17. Comment installer le gestionnaire de conteneurs LXD ?

Rechercher dans la liste des snaps.

Le snap s'appelle tout simplement lxd.

```
$ sudo snap install lxd
2020-09-13T22:59:46+02:00 INFO Waiting for automatic snapd restart...
Warning: /snap/bin was not found in your $PATH. If you've not restarted your session since you
         installed snapd, try doing that. Please see https://forum.snapcraft.io/t/9469 for more
         details.

lxd 4.4 from Canonical✓ installed
```

On peut lister les snaps installés.

```
$ snap list
Name    Version   Rev    Tracking       Publisher   Notes
core18  20200724  1885   latest/stable  canonical✓  base
lxd     4.4       16926  latest/stable  canonical✓  -
snapd   2.45.3.1  8790   latest/stable  canonical✓  snapd
```

Q18. Comment faire pour que l'utilisateur normal `etu` ait la capacité à gérer les conteneurs ?

Rechercher le nom du groupe système correspondant à l'utilisation des outils LXD.

Il faut que l'utilisteur normal appartienne au groupe système `lxd` pour qu'il est tous les droits sur la gestion des conteneurs.

```
$ sudo adduser etu lxd
```

Attention ! il faut se déconnecter/reconnecter pour bénéficier de la nouvelle attribution de groupe. On peut utiliser la commande groups pour vérifier le résultats.

```
$ groups
etu adm cdrom floppy sudo audio dip video plugdev staff netdev lxd
```

6.3. Configuration du gestionnaire de conteneurs LXD

Q19. Quelle est l'instruction de configuration initiale du gestionnaire LXD ?

Utiliser l'aide de la commande lxd.

C'est l'instruction `lxd init` qui nous intéresse.

Voici une copie d'écran de son exécution.

```
$ lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Name of the storage backend to use (ceph, btrfs, dir, lvm) [default=btrfs]:
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty disk or partition? (yes/no) [default=no]:
Size in GB of the new loop device (1GB minimum) [default=13GB]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]: no
Would you like to configure LXD to use an existing bridge or host interface? (yes/no) [default=no]: yes
Name of the existing bridge or host interface: enp0s6
Would you like LXD to be available over the network? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]: yes
config: {}
networks: []
storage_pools:
- config:
    size: 13GB
  description: ""
  name: default
  driver: btrfs
profiles:
- config: {}
  description: ""
  devices:
    eth0:
      name: eth0
      nictype: macvlan
      parent: enp0s6
      type: nic
    root:
      path: /
      pool: default
      type: disk
  name: default
cluster: null
```

Q20. Quelle est l'instruction qui permet d'afficher le profil par défaut des conteneur ?

Rechercher dans les options de la commande lxc.

Voici un exemple d'exécution.

```
$ lxc profile show default
To start your first instance, try: lxc launch ubuntu:18.04

config: {}
description: Default LXD profile
devices:
  eth0:
    name: eth0
    nictype: macvlan
    parent: enp0s6
    type: nic
  root:
    path: /
    pool: default
    type: disk
name: default
used_by: []
```

Q21. Quelle est l'instruction de lancement d'un conteneur ?

Rechercher dans les options de la commande lxc.

Tester son exécution avec un conteneur de type `debian/bullseye`.

Voici un exemple d'exécution.

```
$ lxc launch images:debian/bullseye container0
Creating container0
Starting container0
$ lxc ls
+------------+---------+------+-----------------------------------------------+-----------+-----------+
|    NAME    |  STATE  | IPV4 |                     IPV6                      |   TYPE    | SNAPSHOTS |
+------------+---------+------+-----------------------------------------------+-----------+-----------+
| container0 | RUNNING |      | fda0:7a62:1ae:0:216:3eff:fe22:6075 (eth0) | CONTAINER | 0         |
+------------+---------+------+-----------------------------------------------+-----------+-----------+
```