

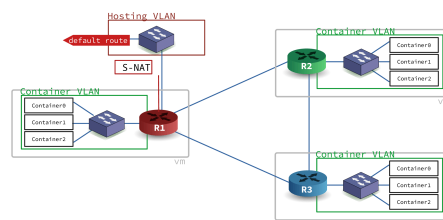
# Introduction au routage dynamique OSPF avec FRRouting

Philippe Latu  
philippe.latu(at)inetdoc.net

<https://www.inetdoc.net>

## Résumé

L'objectif de ce support de travaux pratiques est d'étudier le protocole de routage dynamique OSPF. Cette illustration s'appuie sur une topologie minimale très classique : le triangle. L'originalité consiste à utiliser les VLANs pour marquer la distinction entre une topologie physique de type étoile et une topologie logique en triangle. Cette version du support utilise la suite de démons de routage FRRouting.



## Table des matières

1. Copyright et Licence .....	1
2. Topologie réseau étudiée .....	2
3. Préparer les systèmes pour le routage IPv4 et IPv6 .....	3
4. Valider les communications entre routeurs .....	9
5. Configurer les démons OSPFv2 et OSPFv3 .....	11
6. Publier les routes par défaut via OSPF .....	20
7. Ajouter un réseau de conteneurs à chaque routeur du triangle .....	26
8. Adapter de la métrique de lien au débit .....	29
9. Consulter les documents de référence .....	31
10. Sauvegarder les fichiers de configuration .....	32

## 1. Copyright et Licence

Copyright (c) 2000,2024 Philippe Latu.  
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright (c) 2000,2024 Philippe Latu.  
Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License), version 1.3 ou toute version ultérieure publiée par la Free Software Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture, et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est incluse dans la section intitulée « Licence de Documentation Libre GNU ».

## Méta-information

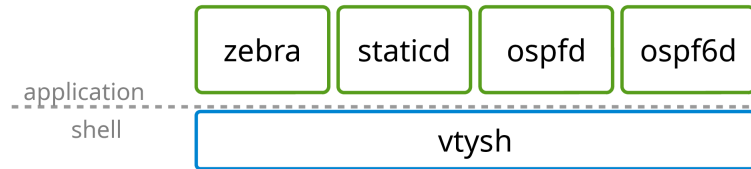
Ce document est écrit avec [DocBook XML](#) sur un système [Debian GNU/Linux](#). Il est disponible en version imprimable au format PDF : [ospf.pdf](#).

Toutes les commandes utilisées dans ce document ne sont pas spécifiques à une version particulière des systèmes UNIX ou GNU/Linux. C'est la distribution Debian GNU/Linux qui est utilisée pour les tests présentés. Voici une liste des paquets contenant les commandes :

- procs - Utilitaires pour le système de fichiers /proc
- iproute2 - Outils de contrôle du trafic et du réseau

- ifupdown - Outils de haut niveau pour configurer les interfaces réseau
- iputils-ping - Outils pour tester l'accessibilité de nœuds réseaux
- frr - BGP/OSPF/RIP routing daemon

La suite de démons de routage FRRouting couvre la totalité des protocoles de routage dynamiques. Le paquet frr fournit autant de processus que de protocoles. Ici, on se concentre sur le protocole OSPF. Voici une représentation de l'organisation des démons actifs dans notre contexte.



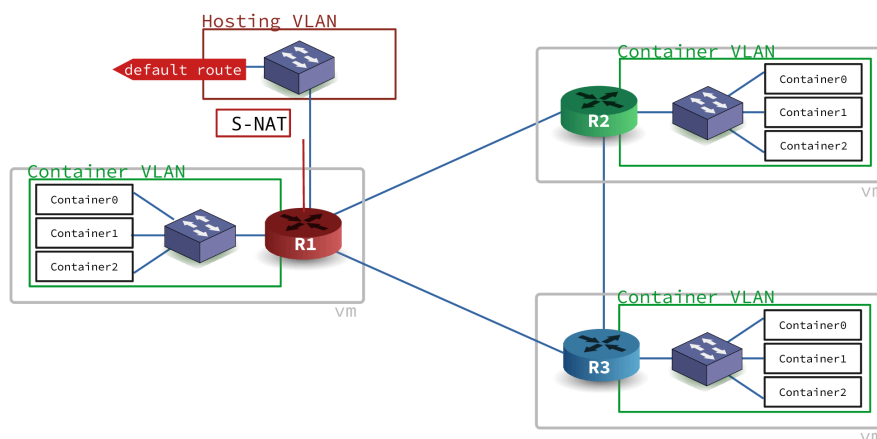
## 2. Topologie réseau étudiée

La topologie réseau étudiée peut être présentée sous deux formes distinctes : logique et physique.

### Topologie logique

On retrouve un grand classique dans l'introduction aux protocoles de routage dynamiques : le triangle dont les trois côtés (liens) sont de type LAN.

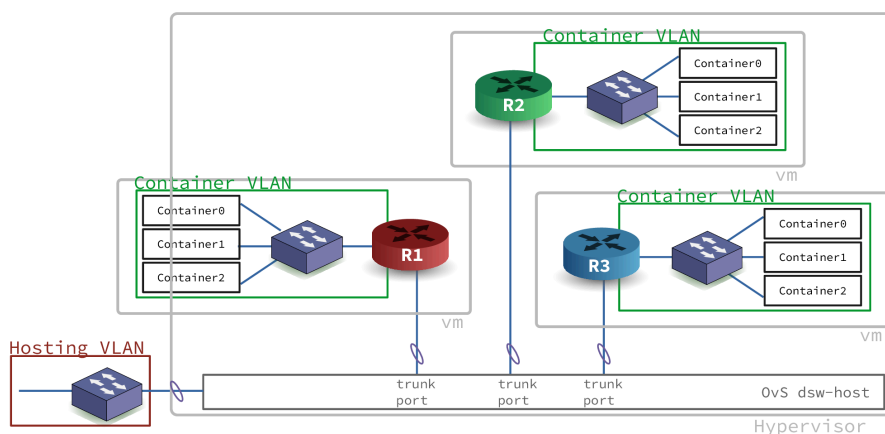
Si on se place sur un sommet du triangle, le côté opposé correspond à un réseau "inconnu" que l'on peut joindre via les routeurs voisins.



Topologie logique en triangle

### Topologie physique

On s'appuie sur le support **Routage Inter-VLAN** pour constituer une topologie physique à base de réseaux locaux virtuels ou VLANs. On fait correspondre à chaque lien de la topologie logique en triangle un numéro de VLAN défini.



## Topologie physique en étoile

Après avoir mis en œuvre la topologie physique en s'appuyant sur le support de la séance de travaux pratiques précédente : **Routage Inter-VLAN**, on implante les démons de routage OSPF sur les trois routeurs R1, R2 et R3.

Ce support se limite à l'étude du routage dynamique à l'intérieur d'une aire unique. La seule «frontière» de communication inter-aire visible est constituée par le lien vers l'Internet. Cette route par défaut sera redistribuée via OSPF par le routeur R1 aux autres routeurs. On verra alors un exemple de route externe dans les bases de données OSPF.

Voici le plan d'adressage de la maquette qui a été utilisée pour rédiger ce document.

Tableau 1. Maquette

Rôle	OSPF router-id	VLAN	Type	Adresses
R1	OSPFv2 : 0.0.4.1 OSPFv3 : 0.0.6.1	360	Passerelle	192.168.104.129/29 fe80:168::1
		480	R1 -> R2	10.48.0.1/29
		481	R1 -> R3	10.48.1.1/29
		10	Vert	10.10.0.1/24 fd14:ca46:3864:a::1/64
R2	OSPFv2 : 0.0.4.2 OSPFv3 : 0.0.6.2	480	R2 -> R1	10.48.0.2/29
		482	R2 -> R3	10.48.2.2/29
		20	Vert	10.20.0.1/24 fd14:ca46:3864:14::1/64
R3	OSPFv2 : 0.0.4.3 OSPFv3 : 0.0.6.3	481	R3 -> R1	10.48.1.3/29
		482	R3 -> R2	10.48.2.3/29
		30	Vert	10.30.0.1/24 fd14:ca46:3864:1e::1/64

### 3. Préparer les systèmes pour le routage IPv4 et IPv6

Les manipulations de ce support s'appuient sur la documentation du projet : [FRRouting User Guide](#)

La première étape consiste à installer les outils sur les trois routeurs, à appliquer une configuration commune et à mettre en place la topologie physique.

1. Si on respecte la topologie physique présentée dans la section précédente, chaque routeur virtuel est raccordé sur un port de commutateur en mode trunk. Il est donc nécessaire de définir une sous-interface réseau appartenant à un VLAN avec adressage automatique pour réaliser les opérations suivantes.

Voici un exemple de fichier `/etc/network/interfaces` avec une définition d'interface temporaire qui permet d'installer des paquets et de préparer la suite de la configuration.

```
# 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 enp0s1
iface enp0s1 inet manual
    up ip link set dev $IFACE up
    down ip link set dev $IFACE down

# ----- TEMPORAIRE
auto enp0s1.20
iface enp0s1.20 inet dhcp

iface enp0s1.20 inet6 auto
```



#### Avertissement

Attention ! L'interface temporaire doit être désactivée dès que la partie préparation est terminée.

2. Pour installer le paquet frr, on doit ajouter un nouveau dépôt au système.

On commence par ajouter la clé de signature des paquets à la configuration du gestionnaire.

```
sudo apt -y install curl
curl -s https://deb.frrouting.org/frr/keys.asc | \
sudo gpg -o /usr/share/keyrings/frr-keyring.gpg --dearmor
```

On crée une nouvelle entrée dans la liste des sources de paquets et on lance la mise à jour du catalogue.

```
echo "deb [signed-by=/usr/share/keyrings/frr-keyring.gpg] \
https://deb.frrouting.org/frr bullseye frr-stable" | \
sudo tee /etc/apt/sources.list.d/frr.list
```

```
sudo apt update
sudo apt -y install frr frr-pythontools
```

```
apt search ^frr$
En train de trier... Fait
Recherche en texte intégral... Fait
frr/stable,now 8.3.1-0~deb11u1 amd64 [installé]
  FRRouting suite of internet protocols (BGP, OSPF, IS-IS, ...)
```

Sans configuration particulière, les services zebra et staticd sont lancés. Aucun protocole de routage dynamique n'est activé.

```
systemctl status frr
• frr.service - FRRouting
  Loaded: loaded (/lib/systemd/system/frr.service; enabled; preset: enabled)
  Active: active (running) since Sun 2022-10-23 14:55:02 CEST; 1min 22s ago
  Docs: https://frrouting.readthedocs.io/en/latest/setup.html
  Process: 2369 ExecStart=/usr/lib/frr/frrinit.sh start (code=exited, status=0/SUCCESS)
  Main PID: 2378 (watchfrr)
  Status: "FRR Operational"
  Tasks: 7 (limit: 1114)
  Memory: 9.4M
  CPU: 201ms
  CGroup: /system.slice/frr.service
          └─2378 /usr/lib/frr/watchfrr -d -F traditional zebra staticd
            └─2389 /usr/lib/frr/zebra -d -F traditional -A 127.0.0.1 -s 90000000
              └─2394 /usr/lib/frr/staticd -d -F traditional -A 127.0.0.1

oct. 23 14:55:02 R1 watchfrr[2378]: [ZCJ3S-SPH5S] zebra state -> down : initial connection attempt
oct. 23 14:55:02 R1 watchfrr[2378]: [ZCJ3S-SPH5S] staticd state -> down : initial connection attempt
oct. 23 14:55:02 R1 watchfrr[2378]: [YFT0P-5Q5YX] Forked background command [pid 2379]: /usr/lib/fr
oct. 23 14:55:02 R1 zebra[2389]: [VTVCM-Y2NW3] Configuration Read in Took: 00:00:00
oct. 23 14:55:02 R1 staticd[2394]: [VTVCM-Y2NW3] Configuration Read in Took: 00:00:00
oct. 23 14:55:02 R1 watchfrr[2378]: [QDG3Y-BY5TN] zebra state -> up : connect succeeded
oct. 23 14:55:02 R1 watchfrr[2378]: [QDG3Y-BY5TN] staticd state -> up : connect succeeded
oct. 23 14:55:02 R1 watchfrr[2378]: [KWE5Q-QNGFC] all daemons up, doing startup-complete notify
oct. 23 14:55:02 R1 frrinit.sh[2369]: Started watchfrr.
oct. 23 14:55:02 R1 systemd[1]: Started FRRouting.
```

### 3. Activer le routage IPv4 et IPv6 au niveau noyau.

Il faut éditer le fichier `/etc/sysctl.conf` pour fixer les valeurs des paramètres de configuration du routage. Voir la section Fonctions réseau d'une interface du support [Configuration d'une interface de réseau local](#).

```
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
```

```
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
```

### 4. Pour créer les interfaces associées aux VLANs sur chacun des routeurs R1, R2 et R3, on édite le fichier `/etc/network/interfaces`.

Auparavant, on doit installer le paquet du commutateur virtuel qui sert à raccorder des conteneurs du VLAN "vert" d'hébergement.

```
sudo apt -y install openvswitch-switch
```

Voici un exemple de fichier `/etc/network/interfaces` pour le routeur R1 de la maquette.

```

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s1
iface enp0s1 inet manual
  up ip link set dev $IFACE up
  down ip link set dev $IFACE down

# uplink
auto enp0s1.360
iface enp0s1.360 inet static
  address 192.168.104.130/27
  gateway 192.168.104.129
  dns-nameserver 172.16.0.2

iface enp0s1.360 inet6 static
  address 2001:678:3fc:168::2/64
  gateway fe80:168::1

# R1 -> R2
auto enp0s1.480
iface enp0s1.480 inet static
  address 10.48.0.1/29

# R1 -> R3
auto enp0s1.481
iface enp0s1.481 inet static
  address 10.48.1.1/29

# R1 -> lxd
auto asw-host
iface asw-host inet manual
  ovs_type OVSBridge
  ovs_ports sw-vlan10
  up ip link set dev $IFACE up
  down ip link set dev $IFACE down

allow-asw-host sw-vlan10
iface sw-vlan10 inet static
  ovs_type OVSBridge
  ovs_bridge asw-host
  ovs_options asw-host 10
  address 10.10.0.1/24

iface sw-vlan10 inet6 static
  ovs_type OVSBridge
  ovs_bridge asw-host
  ovs_options asw-host 10
  address fd14:ca46:3864:a::1/64

```

On adapte la liste des interfaces pour les deux autres routeurs R2 et R3 avec les numéros de VLANs concernés.

- Éditer le fichier de configuration de la console “unifiée” vtysh de façon à ce qu'il contienne les informations minimales.

Voici un exemple pour le routeur R1 de la maquette.

```

echo service integrated-vtysh-config | \
sudo tee /etc/frr/vtysh.conf

```

Cet outil offre une console unifiée pour les trois démons : zebra, ospfd et ospf6d.

- Ajouter l'utilisateur `etu` aux groupes `frr` et `frrvty` pour lui donner un accès direct à la console de configuration des démons et aux fichiers de configuration.

```

sudo adduser etu frrvty
sudo adduser etu frr

```

Il ne faut pas oublier de se déconnecter/reconnecter pour bénéficier de la nouvelle attribution de groupe.

- Activer les deux démons relatifs au protocole OSPF dans le fichier de configuration : `/etc/frr/daemons`.

```
sudo sed -i 's/ospfd=no/ospfd=yes/' /etc/frr/daemons
sudo sed -i 's/ospf6d=no/ospf6d=yes/' /etc/frr/daemons
sudo systemctl restart frr
```

On peut alors relancer le service et vérifier que les nouveaux démons de routage dynamique OSPF sont bien activés.

```
systemctl status frr
• frr.service - FRRouting
  Loaded: loaded (/lib/systemd/system/frr.service; enabled; preset: enabled)
  Active: active (running) since Sun 2022-10-23 15:36:35 CEST; 8s ago
  Docs: https://frrouting.readthedocs.io/en/latest/setup.html
  Process: 3356 ExecStart=/usr/lib/frr/frrinit.sh start (code=exited, status=0/SUCCESS)
  Main PID: 3365 (watchfrr)
  Status: "FRR Operational"
  Tasks: 11 (limit: 1114)
  Memory: 16.0M
  CPU: 237ms
  CGroup: /system.slice/frr.service
          └─3365 /usr/lib/frr/watchfrr -d -F traditional zebra ospfd ospf6d staticd
             └─3381 /usr/lib/frr/zebra -d -F traditional -A 127.0.0.1 -s 90000000
                └─3386 /usr/lib/frr/ospfd -d -F traditional -A 127.0.0.1
                   └─3389 /usr/lib/frr/ospf6d -d -F traditional -A ::1
                      └─3392 /usr/lib/frr/staticd -d -F traditional -A 127.0.0.1

oct. 23 15:36:35 R1 ospfd[3386]: [VTVCM-Y2NW3] Configuration Read in Took: 00:00:00
oct. 23 15:36:35 R1 ospf6d[3389]: [VTVCM-Y2NW3] Configuration Read in Took: 00:00:00
oct. 23 15:36:35 R1 staticd[3392]: [VTVCM-Y2NW3] Configuration Read in Took: 00:00:00
oct. 23 15:36:35 R1 watchfrr[3365]: [QDG3Y-BY5TN] zebra state -> up : connect succeeded
oct. 23 15:36:35 R1 watchfrr[3365]: [QDG3Y-BY5TN] ospfd state -> up : connect succeeded
oct. 23 15:36:35 R1 watchfrr[3365]: [QDG3Y-BY5TN] ospf6d state -> up : connect succeeded
oct. 23 15:36:35 R1 watchfrr[3365]: [QDG3Y-BY5TN] staticd state -> up : connect succeeded
oct. 23 15:36:35 R1 watchfrr[3365]: [KWE5Q-QNGFC] all daemons up, doing startup-complete notify
oct. 23 15:36:35 R1 frrinit.sh[3356]: Started watchfrr.
oct. 23 15:36:35 R1 systemd[1]: Started FRRouting.
```

On peut aussi lister les démons actifs à partir de la console du service.

```
vttysh

Hello, this is FRRouting (version 8.3.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

R1# sh daemons
zebra ospfd ospf6d watchfrr staticd
```

- Compléter la configuration des interfaces dans la console unifiée vtysh de façon à fixer la bande passante de chaque interface active à 1Gbps.

Voici un exemple de séquence d'instructions pour configurer les interfaces du routeur rouge de la maquette.

On commence par lister les interfaces actives à partir de la console.

```

R1# sh int brief
Interface      Status  VRF      Addresses
-----
asw-host       up      default
enp0s1         up      default
enp0s1.360     up      default   192.168.104.130/27
               + 2001:678:3fc:168::2/64
enp0s1.480     up      default   10.48.0.1/29
enp0s1.481     up      default   10.48.1.1/29
lo             up      default
ovs-system     down    default
sw-vlan10      up      default   10.10.0.1/24
               + fd14:ca46:3864:a::1/64

```

Pour les interfaces `enp0s1.480` (lien R1 -> R2) et `enp0s1.481` (lien R1 -> R3), on ajoute le paramètre `bandwidth` qui permet de définir arbitrairement le débit binaire d'une interface.

```

R1# conf t
R1(config)# int enp0s1.480
R1(config-if)# bandwidth ?
(1-100000) Bandwidth in megabits
R1(config-if)# bandwidth 100000
R1(config-if)# ^Z
R1# sh int enp0s1.480
Interface enp0s1.480 is up, line protocol is up
  Link ups:      1      last: 2022/10/24 09:25:42.65
  Link downs:    0      last: (never)
  vrf: default
  index 4 metric 0 mtu 1500 speed 4294967295
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  Type: Ethernet
  HWaddr: b8:ad:ca:fe:00:c8
  bandwidth 100000 Mbps
  inet 10.48.0.1/29
  inet6 fe80::baad:caff:fefe:c8/64
  Interface Type Vlan
  Interface Slave Type None
  VLAN Id 480
  protodown: off
  Parent interface: enp0s1

```

On répète la même opération pour l'interface `enp0s1.481`. Ensuite, on peut sauvegarder la configuration.

```

R1# copy run start
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]

```

Une fois la configuration complétée, les paramètres apparaissent dans la configuration du service.



```

R1# sh run
Building configuration...

Current configuration:
!
frr version 8.3.1
frr defaults traditional
hostname R1
log syslog informational
service integrated-vtysh-config
!
interface enp0s1.480
  bandwidth 100000
exit
!
interface enp0s1.481
  bandwidth 100000
exit
!
end

```



#### Note

Contrairement à un routeur «intégré» avec un système d'exploitation dédié, le démon de routage statique n'a pas directement accès aux interfaces matérielles. Or, sur un système GNU/Linux, le débit d'une interface Ethernet filaire peut varier en fonction du débit proposé par le port du commutateur. Sans information spécifique du noyau, l'application «service de routage» n'a aucun moyen de connaître le débit exact de l'interface. C'est la raison pour laquelle il est nécessaire de paramétrer manuellement les débits de chaque interface dans la configuration.



#### Avertissement

Sans configuration manuelle du coût de lien d'une interface, ce paramétrage de la bande passante est essentiel dans le **calcul des métriques** et le fonctionnement du protocole de routage OSPF. Si les calculs de métriques pour les liens actifs sont erronés, le choix des routes à emprunter pour faire transiter le trafic utilisateur entre deux routeurs peut lui aussi être erroné.

Une fois l'ensemble des opérations de cette section réalisées, chaque routeur dispose des outils pour mettre en œuvre la topologie physique et ensuite les protocoles de routage dynamique OSPF.

## 4. Valider les communications entre routeurs

Avant d'aborder le déploiement du protocole de routage dynamique, il est nécessaire de valider le raccordement des routeurs aux commutateurs désignés, les communications entre chaque routeur et la visualisation des tables de routage pour les interfaces réseau configurées.

Q1. Quelles sont les tests à effectuer pour vérifier l'état des différents «côtés» de la topologie triangle ?

Les interfaces ont déjà été configurées via le fichier `/etc/network/interfaces` de chaque routeur. Il faut afficher la table de routage de chaque routeur puis la table des voisins. Ainsi, on peut contrôler les correspondances entre les adresses de couche réseau et de couche liaison.

Dans le contexte de la maquette on obtient les résultats suivants pour le routeur R1. On se limite à l'affichage des entrées de la table de routage apprises par le noyau avec l'option `proto kernel`.

```

ip route ls proto kernel
10.10.0.0/24 dev sw-vlan10 scope link src 10.10.0.1
10.48.0.0/29 dev enp0s1.480 scope link src 10.48.0.1
10.48.1.0/29 dev enp0s1.481 scope link src 10.48.1.1
192.168.104.128/27 dev enp0s1.360 scope link src 192.168.104.130

```

```
ip -6 route ls proto kernel
::1 dev lo metric 256 pref medium
2001:678:3fc:168::/64 dev enp0s1.360 metric 256 pref medium
fd14:ca46:3864:a::/64 dev sw-vlan10 metric 256 pref medium
fe80::/64 dev enp0s1 metric 256 pref medium
fe80::/64 dev enp0s1.360 metric 256 pref medium
fe80::/64 dev enp0s1.480 metric 256 pref medium
fe80::/64 dev enp0s1.481 metric 256 pref medium
fe80::/64 dev sw-vlan10 metric 256 pref medium
fe80::/64 dev asw-host metric 256 pref medium
```

Pour les entrées relatives au voisinage réseau, on se limite aussi à l'affichage des voisins directs des interfaces du routeur.

On repère dans la copie d'écran ci-dessous les adresses des deux autres routeurs : R2 et R3.

```
ip neigh ls | grep 'enp0s1\.'
```

**10.48.0.2 dev enp0s1.480 lladdr b8:ad:ca:fe:00:c9 STALE**  
192.168.104.129 dev enp0s1.360 lladdr 80:6a:00:dc:67:53 STALE  
**10.48.1.3 dev enp0s1.481 lladdr b8:ad:ca:fe:00:ca STALE**  
fe80::baad:caff:fefe:c9 dev enp0s1.480 lladdr b8:ad:ca:fe:00:c9 STALE  
fe80:168::1 dev enp0s1.360 lladdr 80:6a:00:dc:67:53 router STALE  
fe80::baad:caff:fefe:ca dev enp0s1.481 lladdr b8:ad:ca:fe:00:ca STALE

Q2. Quelles sont les opérations à effectuer pour valider les communications IPv4 et IPv6 entre chacun des routeurs ?

Lancer les tests ICMP usuels entre chaque routeur sur chaque lien actif.

Exemple entre R1 et R2 ; toujours dans le contexte de la maquette.

- Requête IPv4 de R1 vers R2 sur le VLAN 480.

```
ping -qc2 10.48.0.2
PING 10.48.0.2 (10.48.0.2) 56(84) bytes of data.

--- 10.48.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.860/2.512/4.165/1.652 ms
```

Aucun paquet n'a été perdu. Le routeur R2 est bien joignable depuis R1.

- Requête IPv6 multicast depuis R1 sur le VLAN 480.

```
ping -c2 ff02::1%enp0s1.480
PING ff02::1%enp0s1.480(ff02::1%enp0s1.480) 56 data bytes
64 bytes from fe80::baad:caff:fefe:c8%enp0s1.480: icmp_seq=1 ttl=64 time=0.138 ms
64 bytes from fe80::baad:caff:fefe:c9%enp0s1.480: icmp_seq=1 ttl=64 time=4.28 ms
64 bytes from fe80::baad:caff:fefe:c8%enp0s1.480: icmp_seq=2 ttl=64 time=0.064 ms

--- ff02::1%enp0s1.480 ping statistics ---
2 packets transmitted, 2 received, +1 duplicates, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.064/1.495/4.284/1.972 ms
```

L'identification des adresses IPv6 de lien local montrent que fe80::baad:caff:fefe:c8 correspond à R1 et que fe80::baad:caff:fefe:c9 correspond à R2.

L'opération est à répéter sur chaque lien entre deux routeurs reliés sur le même VLAN.

Q3. Quelles sont les opérations à effectuer pour visualiser les tables de routage IPv4 et IPv6 existantes d'un routeur au niveau de la console unifiée vtysh ?

Afficher les tables de routage à partir de la console vtysh avec les commandes du système Cisco™ IOS show ip route et show ipv6 route.

Dans le contexte de la maquette, on obtient les résultats suivants pour le routeur R2.

```
R1# sh ip route connected
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

C>* 10.20.0.0/24 is directly connected, sw-vlan20, 01:24:23
C>* 10.48.0.0/29 is directly connected, enp0s1.480, 01:24:24
C>* 10.48.2.0/29 is directly connected, enp0s1.482, 01:24:23
```

```
R1# sh ipv6 route connected
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
       v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

C>* fd14:ca46:3864:14::/64 is directly connected, sw-vlan20, 01:25:11
C * fe80::/64 is directly connected, asw-host, 01:25:09
C * fe80::/64 is directly connected, sw-vlan20, 01:25:11
C * fe80::/64 is directly connected, enp0s1, 01:25:11
C * fe80::/64 is directly connected, enp0s1.480, 01:25:11
C>* fe80::/64 is directly connected, enp0s1.482, 01:25:11
```

## 5. Configurer les démons OSPFv2 et OSPFv3

Dans cette section, on introduit les premières commandes de configuration du protocole de routage dynamique OSPF qui permettent d'activer le protocole puis d'ajouter des entrées de réseau dans la base de données de ce protocole.

Q4. Comment peut-on contrôler que le protocole OSPF est actif ou non sur un routeur ?

Une fois la console vtysh ouverte, lancer les commandes de visualisation de l'état du protocole listées ci-dessous. Ces commandes peuvent être lancées sur chacun des trois routeurs.

```
show ip ospf
show ipv6 ospf6
show daemons
```

Dans les informations données dans la copie d'écran ci-dessous, il apparaît qu'aucune configuration du protocole de routage dynamique n'a été activée.

```
R1# sh ip ospf
R1# sh ipv6 ospf6
% OSPFv3 instance not found
R1# sh daemons
zebra ospfd ospf6d watchfrr staticd
```

Q5. Quelles sont les opérations à effectuer pour activer les protocoles de routage OSPFv2 et OSPFv3 ? Comment affecter manuellement l'identifiant du routeur ?

### Avertissement

Les identifiants à utiliser lors de la séance de travaux pratiques sont donnés dans les tableaux des plans d'adressage. Voir [Section 2, « Topologie réseau étudiée »](#).

La liste des commandes utiles en mode configuration dans la console vtysh est la suivante.

```
router ospf
router ospf6
ospf router-id X.X.X.X
ospf6 router-id X.X.X.X
log detail
```

Toujours à partir de la console vtysh, on accède au mode configuration à l'aide de la commande `conf t`. Voici un exemple de séquence sur le troisième routeur.

```
R1# conf t
R1(config)# router ospf
R1(config-router)# ospf router-id 0.0.4.1
R1(config-router)# log detail
R1(config-router)# ^Z
```

```
R1# conf t
R1(config)# router ospf6
R1(config-ospf6)# ospf6 router-id 0.0.6.1
R1(config-ospf6)# log detail
R1(config-ospf6)# ^Z
```

```
R1# sh run ospfd
Building configuration...

Current configuration:
!
frr version 8.3.1
frr defaults traditional
hostname R1
log syslog informational
service integrated-vtysh-config
!
router ospf
  ospf router-id 0.0.4.1
  log-adjacency-changes detail
exit
!
end
```

```
R1# sh run ospf6d
Building configuration...

Current configuration:
!
frr version 8.3.1
frr defaults traditional
hostname R1
log syslog informational
service integrated-vtysh-config
!
router ospf6
  ospf6 router-id 0.0.6.1
  log-adjacency-changes detail
exit
!
end
```

Le choix de codage des identifiants OSPF a pour but d'éviter une confusion avec les adresses des réseaux actifs sur chaque routeur.

Si on reprend l'instruction de la question précédente, on obtient l'état de chacun des démons de protocole de routage dynamique.

```
R3# sh ip ospf
OSPF Routing Process, Router ID: 0.0.4.3
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is disabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 0 millise(c)s
Minimum hold time between consecutive SPF(s) 50 millise(c)s
Maximum hold time between consecutive SPF(s) 5000 millise(c)s
Hold time multiplier is currently 1
SPF algorithm has not been run
SPF timer is inactive
LSA minimum interval 5000 msec(s)
LSA minimum arrival 1000 msec(s)
Write Multiplier set to 20
Refresh timer 10 sec(s)
Maximum multiple paths(ECMP) supported 256
Administrative distance 110
Number of external LSA 0. Checksum Sum 0x00000000
Number of opaque AS LSA 0. Checksum Sum 0x00000000
Number of areas attached to this router: 0
All adjacency changes are logged
```

```
R3# sh ipv6 ospf6
OSPFv3 Routing Process (0) with Router-ID 0.0.6.3
Running 00:01:52
LSA minimum arrival 1000 msec(s)
Maximum-paths 256
Administrative distance 110
Initial SPF scheduling delay 0 millise(c)s
Minimum hold time between consecutive SPF(s) 50 millise(c)s
Maximum hold time between consecutive SPF(s) 5000 millise(c)s
Hold time multiplier is currently 1
SPF algorithm has not been run
SPF timer is inactive
Number of AS scoped LSAs is 0
Number of areas in this router is 0
Authentication Sequence number info
Higher sequence no 0, Lower sequence no 0
All adjacency changes are logged
```

- Q6. Quelles sont les opérations à effectuer pour activer les protocoles de routage OSPFv2 et OSPFv3 pour les réseaux d'interconnexion de chaque routeur ?

Il faut activer le protocole de routage dynamique sur chaque interface de la topologie qui participe à la construction du triangle.

La liste des commandes utiles en mode console et en mode configuration dans vtysh est la suivante.

```
show ip route connected
show ip route ospf
show ipv6 route connected
show ipv6 route ospf
ip ospf area 0
ipv6 ospf6 area 0
```

Voici un exemple de séquence d'instructions pour le routeur R1-rouge.

On commence par lister les entrées marquées C ou connected de la table de routage IPv4 de façon à reconnaître les deux côtés de la topologie triangle connus du "sommet" R1-rouge.

```
R1# sh ip route connected
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

C>* 10.10.0.0/24 is directly connected, sw-vlan10, 02:28:24
C>* 10.48.0.0/29 is directly connected, enp0s1.480, 02:28:24
C>* 10.48.1.0/29 is directly connected, enp0s1.481, 02:28:24
C>* 192.168.104.128/27 is directly connected, enp0s1.360, 02:28:26
```

À partir de cette identification, on doit activer le protocole OSPFv2 pour les deux interfaces : enp0s1.480 et enp0s1.481.

```
R1# conf t
R1(config)# int enp0s1.480
R1(config-if)# ip ospf area 0
R1(config-if)# int enp0s1.481
R1(config-if)# ip ospf area 0
R1(config-if)# ^Z
```

On reprend le même processus pour le protocole OSPFv3. Voici une copie des entrées connectées de la table de routage IPv6.

```
R1# sh ipv6 route connected
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
       v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

C>* 2001:678:3fc:168::/64 is directly connected, enp0s1.360, 02:31:04
C>* fd14:ca46:3864:a::/64 is directly connected, sw-vlan10, 02:31:02
C * fe80::/64 is directly connected, asw-host, 02:31:00
C * fe80::/64 is directly connected, enp0s1.481, 02:31:02
C * fe80::/64 is directly connected, sw-vlan10, 02:31:02
C * fe80::/64 is directly connected, enp0s1.480, 02:31:03
C * fe80::/64 is directly connected, enp0s1, 02:31:04
C>* fe80::/64 is directly connected, enp0s1.360, 02:31:04
```

On active le protocole OSPFv3 sur les mêmes interfaces.

```
R1# conf t
R1(config)# int enp0s1.480
R1(config-if)# ipv6 ospf6 area 0
R1(config-if)# int enp0s1.481
R1(config-if)# ipv6 ospf6 area 0
R1(config-if)# ^Z
```

Q7. Comment visualiser l'état des interfaces actives pour chaque processus de protocole de routage dynamique OSPFv2 ou OSPFv3 ?

Les interfaces sont dites actives pour les protocoles OSPFv2 ou OSPFv3 dès qu'elles ont été ajoutées aux processus de routage dynamique en précisant l'aire à laquelle elles appartiennent.

La liste des commandes utiles dans la console vtysh est la suivante.

```
show ip ospf interface
show ipv6 ospf6 interface
```

En reprenant l'exemple du routeur R1-rouge, on obtient les résultats suivants.

```

R1# sh ip ospf interface
enp0s1.480 is up
  ifindex 4, MTU 1500 bytes, BW 1000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.48.0.1/29, Broadcast 10.48.0.7, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 0.0.4.1, Network Type BROADCAST, Cost: 100
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 0.0.4.1 Interface Address 10.48.0.1/29
  No backup designated router on this network
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 9.342s
  Neighbor Count is 0, Adjacent neighbor count is 0
enp0s1.481 is up
  ifindex 5, MTU 1500 bytes, BW 1000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.48.1.1/29, Broadcast 10.48.1.7, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 0.0.4.1, Network Type BROADCAST, Cost: 100
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 0.0.4.1 Interface Address 10.48.1.1/29
  No backup designated router on this network
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 5.464s
  Neighbor Count is 0, Adjacent neighbor count is 0

```

La copie d'écran ci-dessus permet d'identifier les éléments suivants :

- L'indicateur is up confirme que l'interface est bien active pour le protocole de routage.
- L'identifiant de routeur correspond bien à celui défini dans le processus OSPFv2 et le type de réseau du protocole indique un réseau de diffusion Ethernet.
- Les paquets Hello sont bien émis toutes les 10 secondes pour un réseau de diffusion.

On reprend la même démarche pour le protocole OSPFv3. On extrait de liste complète les interfaces des deux côtés du triangle.

```

R1# sh ipv6 ospf6 interface enp0s1.480
enp0s1.480 is up, type BROADCAST
  Interface ID: 4
  Internet Address:
    inet : 10.48.0.1/29
    inet6: fe80::baad:caff:fefe:c8/64
  Instance ID 0, Interface MTU 1500 (autodetect: 1500)
  MTU mismatch detection: enabled
  Area ID 0.0.0.0, Cost 100
  State DR, Transmit Delay 1 sec, Priority 1
  Timer intervals configured:
    Hello 10(1.398), Dead 40, Retransmit 5
  DR: 0.0.6.1 BDR: 0.0.0.0
  Number of I/F scoped LSAs is 1
    0 Pending LSAs for LSUpdate in Time 00:00:00 [thread off]
    0 Pending LSAs for LSAck in Time 00:00:00 [thread off]
  Authentication Trailer is disabled

```

Q8. Comment vérifier que l'identifiant de routeur a correctement été attribué ?

À partir des résultats des questions précédentes, rechercher l'information demandée.

Quelque soit la version du protocole OSPF, l'identifiant de routeur est toujours codé sous la forme d'une adresse IPv4.

```
R1# sh ip ospf database
```

**OSPF Router with ID (0.0.4.1)**

Router Link States (Area 0.0.0.0)

Link ID	ADV Router	Age	Seq#	CkSum	Link count
0.0.4.1	0.0.4.1	640	0x80000004	0xf411 2	

```
R1# sh ipv6 ospf6 database
```

Area Scoped Link State Database (Area 0)

Type	LSId	AdvRouter	Age	SeqNum	Payload
------	------	-----------	-----	--------	---------

I/F Scoped Link State Database (I/F enp0s1.480 in Area 0)

Type	LSId	AdvRouter	Age	SeqNum	Payload
Lnk	0.0.0.4	<u>0.0.6.1</u>	542	80000001	fe80::baad:caff:fefe:c8

I/F Scoped Link State Database (I/F enp0s1.481 in Area 0)

Type	LSId	AdvRouter	Age	SeqNum	Payload
Lnk	0.0.0.5	<u>0.0.6.1</u>	538	80000001	fe80::baad:caff:fefe:c8

Q9. Comment identifier le type de réseau d'une interface ?

À partir des résultats des questions précédentes, rechercher l'information demandée.

Comme on utilise des liens Ethernet dans ce contexte de travaux pratiques, le type le plus important est le réseau de diffusion ou BROADCAST.

```
R1# sh ip ospf interface enp0s1.480
```

```
enp0s1.480 is up
```

```
  ifindex 4, MTU 1500 bytes, BW 1000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
```

```
  Internet Address 10.48.0.1/29, Broadcast 10.48.0.7, Area 0.0.0.0
```

```
  MTU mismatch detection: enabled
```

```
  Router ID 0.0.4.1, Network Type BROADCAST, Cost: 100
```

```
  Transmit Delay is 1 sec, State DR, Priority 1
```

```
  Designated Router (ID) 0.0.4.1 Interface Address 10.48.0.1/29
```

```
  No backup designated router on this network
```

```
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
```

```
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
```

```
    Hello due in 2.643s
```

```
  Neighbor Count is 0, Adjacent neighbor count is 0
```

```
R1# sh ipv6 ospf6 interface enp0s1.480
```

```
enp0s1.480 is up, type BROADCAST
```

```
  Interface ID: 4
```

```
  Internet Address:
```

```
    inet : 10.48.0.1/29
```

```
    inet6: fe80::baad:caff:fefe:c8/64
```

```
  Instance ID 0, Interface MTU 1500 (autodetect: 1500)
```

```
  MTU mismatch detection: enabled
```

```
  Area ID 0.0.0.0, Cost 100
```

```
  State DR, Transmit Delay 1 sec, Priority 1
```

```
  Timer intervals configured:
```

```
    Hello 10(3.490), Dead 40, Retransmit 5
```

```
  DR: 0.0.6.1 BDR: 0.0.0.0
```

```
  Number of I/F scoped LSAs is 1
```

```
    0 Pending LSAs for LSupdate in Time 00:00:00 [thread off]
```

```
    0 Pending LSAs for LSack in Time 00:00:00 [thread off]
```

```
  Authentication Trailer is disabled
```

Q10. Comment obtenir la liste du ou des routeurs voisins pour chaque processus de protocole de routage dynamique OSPFv2 ou OSPFv3 ?

Dès qu'une interface est active, il y a émission de paquets HELLO et si un routeur avec un démon OSPF envoie aussi des paquets HELLO dans le même VLAN, les deux routeurs cherchent à former une adjacence.



La liste des commandes utiles dans la console vtysh est la suivante.

```
show ip ospf neighbor
show ipv6 ospf6 neighbor
```

Toujours à partir du routeur R1, voici un exemple de liste de routeurs OSPF voisins dans laquelle on reconnaît les identifiants des routeurs R2 et R3.

```
R1# sh ip ospf neighbor
```

Neighbor ID	Pri	State	Up Time	Dead Time	Address	Interface	RXmtL	Rqs
<u>0.0.4.2</u>	1	Full/Backup	1m06s	32.660s	10.48.0.2	enp0s1.480:10.48.0.1	0	0
<u>0.0.4.3</u>	1	Full/Backup	2m10s	38.523s	10.48.1.3	enp0s1.481:10.48.1.1	0	0

```
R1# sh ipv6 ospf6 neighbor
```

Neighbor ID	Pri	DeadTime	State/IfState	Duration	I/F[State]
<u>0.0.6.2</u>	1	00:00:37	Full/BDR	00:02:44	enp0s1.480[DR]
<u>0.0.6.3</u>	1	00:00:33	Full/BDR	00:03:50	enp0s1.481[DR]

Q11. Comment identifier le rôle des différentes interfaces des routeurs pour chacun des liens du triangle de la topologie logique ?

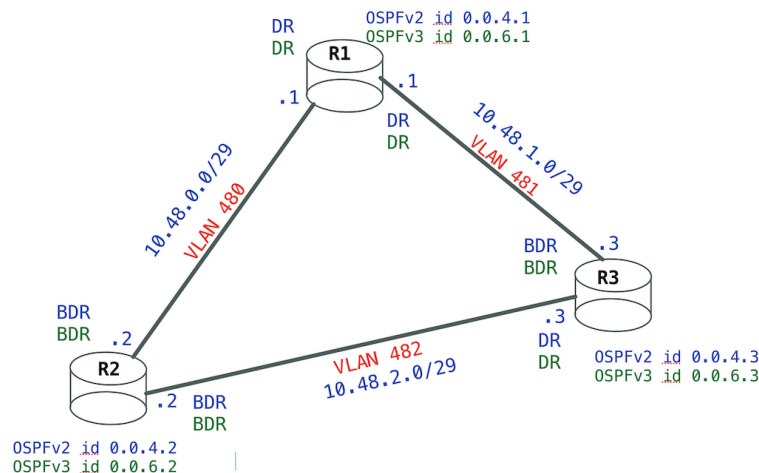


#### Avertissement

La réponse à cette question suppose que les démons OSPF des trois routeurs de la topologie logique en triangle aient convergé. On doit repérer l'état Full pour les listes de routeurs voisins.

De plus, la réponse varie en fonction de l'ordre d'activation des démons OSPF des différents routeurs. En effet, un routeur peut être élu routeur désigné (DR) en l'absence de routeurs voisins. Cette élection n'est pas remise en cause tant qu'il n'y pas de changement d'état de lien.

À partir des résultats des questions précédentes sur les interfaces actives, il est possible de compléter le schéma de la topologie étudiée avec l'état des interfaces pour chacun des trois liens.



Sur un même réseau de diffusion, il est possible de trouver plusieurs routeurs OSPF. Établir une relation de voisinage et procéder aux échanges de bases de données topologiques entre chaque routeur revient à constituer un réseau de relations complètement maillé. À chaque nouveau calcul de topologie, ce réseau complètement maillé est inefficace. C'est la raison pour laquelle la notion de routeur référent ou Designated Router a été introduite. Lors d'un recalcul de topologie, tous les routeurs s'adressent au routeur référent qui correspond au cœur d'un réseau en topologie étoile.

Dans le contexte de la topologie triangle étudiée, il y a bien élection d'un routeur référent et d'un routeur référent de secours. Cependant, comme il n'y a que deux routeurs par domaine de diffusion ou VLAN, on ne peut pas caractériser l'utilité de cette élection.

Q12. Quels sont les réseaux IPv4 et IPv6 présents dans la base calcul du protocole OSPF ?

On cherche à visualiser la liste des préfixes des réseaux connus des deux démons OSPF.

La liste des commandes utiles dans la console vtysh est la suivante.

```
show ip ospf route
show ipv6 ospf6 route
```

Une fois que les trois routeurs de la topologie ont convergé, chaque démon connaît les trois préfixes qui correspondent aux trois côtés du triangle. Un routeur correspond à un sommet du triangle et il doit apprendre le préfixe réseau du côté opposé via ses deux routeurs voisins.

Voici la vue depuis le routeur R1.

```
R1# sh ip ospf route
===== OSPF network routing table =====
N   10.48.0.0/29          [1] area: 0.0.0.0
                        directly attached to enp0s1.480
N   10.48.1.0/29          [1] area: 0.0.0.0
                        directly attached to enp0s1.481
N   10.48.2.0/29          [2] area: 0.0.0.0
                        via 10.48.0.2, enp0s1.480
                        via 10.48.1.3, enp0s1.481
```

Les valeurs notées entre crochets correspondent à la métrique du lien pour joindre le réseau noté à gauche. Depuis la première spécification du protocole OSPF, le calcul de métrique se fait à partir de l'expression :  $10^8 / \text{Bande\_Passante\_du\_lien}$ .

La valeur du numérateur ( $10^8$ ) correspond à un débit de 100Mbps. À l'époque de la rédaction du standard OSPFv2, ce débit a servi de référence. Aujourd'hui, cette valeur est complètement dépassée. C'est la raison pour laquelle on adapte le calcul de métrique en changeant le coefficient du numérateur. Voir la [Section 8, « Adapter de la métrique de lien au débit »](#).

Les deux premiers réseaux de la table sont joignable via un lien Ethernet à 100Mbps ; soit une métrique de 1. Le troisième réseau est joignable via deux liens Ethernet à 100Mbps ; d'où la métrique de 2.

Pour les préfixes IPv6, aucun préfixe n'est présent sachant que les relations de voisinage entre routeurs utilisent obligatoirement les adresses de lien local appartenant au préfixe  $\text{fe80}::/10$ .

```
R1# sh ipv6 ospf6 route
```

Les préfixes des réseaux de conteneurs apparaîtront dès que le protocole de routage aura été activé pour les interfaces SVI.

Q13. Comment visualiser les tables de routage depuis la console vtysh ?

L'affichage demandé illustre les mécanismes de choix entre différentes solutions pour une même destination. Cet affichage est à comparer avec celui demandé à la question suivante.

La liste des commandes utiles dans la console vtysh est la suivante.

```
show ip route
show ipv6 route
```

On reprend à nouveau l'exemple du routeur R1.

```
R1# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure
```

```
K>* 0.0.0.0/0 [0/0] via 192.168.104.129, enp0s1.360 onlink, 04:41:28
C>* 10.10.0.0/24 is directly connected, sw-vlan10, 04:41:26
O 10.48.0.0/29 [110/1] is directly connected, enp0s1.480, weight 1, 00:05:51
C>* 10.48.0.0/29 is directly connected, enp0s1.480, 04:41:26
O 10.48.1.0/29 [110/1] is directly connected, enp0s1.481, weight 1, 00:02:01
C>* 10.48.1.0/29 is directly connected, enp0s1.481, 04:41:26
O>* 10.48.2.0/29 [110/2] via 10.48.0.2, enp0s1.480, weight 1, 00:00:42
   * via 10.48.1.3, enp0s1.481, weight 1, 00:00:42
C>* 192.168.104.128/27 is directly connected, enp0s1.360, 04:41:28
```

```
R1# sh ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
       v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure
```

```
K>* ::/0 [0/1024] via fe80:168::1, enp0s1.360 onlink, 04:42:10
C>* 2001:678:3fc:168::/64 is directly connected, enp0s1.360, 04:42:08
C>* fd14:ca46:3864:a::/64 is directly connected, sw-vlan10, 04:42:06
C * fe80::/64 is directly connected, asw-host, 04:42:04
C * fe80::/64 is directly connected, enp0s1.481, 04:42:06
C * fe80::/64 is directly connected, sw-vlan10, 04:42:06
C * fe80::/64 is directly connected, enp0s1.480, 04:42:07
C * fe80::/64 is directly connected, enp0s1, 04:42:08
C>* fe80::/64 is directly connected, enp0s1.360, 04:42:08
```

- Les entrées marquées avec le caractère \* correspondent aux routes retenues et mémorisées par le sous-système réseau du noyau. Les autres entrées sont placées en réserve au cas où la solution initialement retenue serait en défaut.
- L'entrée notée k correspond à une route apprise depuis le sous-système réseau du noyau.
- Les entrées notées c correspondent à des routes pour lesquelles il existe une interface sur le routeur. Les métriques de ses routes ont la valeur 0. Ce sont les routes les plus prioritaires.
- Les entrées notées o correspondent aux routes apprises via le protocole OSPF. La métrique de ces routes se décompose en deux parties. La valeur figée à 110 définit le niveau de priorité du protocole OSPF (Administrative Distance) relativement aux autres protocoles de routage. Les valeurs notées après le / sont les métriques de liens calculées comme indiqué ci-dessus.

Q14. Comment visualiser les tables de routage au niveau système ?

Utiliser une commande usuelle de visualisation de la table de routage.

```
ip route ls
ip -6 route ls
```

Avec la commande ip, on voit apparaître les «sources» d'alimentation de la table de routage finale du système.

- kernel pour les entrées connues du sous-système réseau du noyau. Ce sont les entrées avec le caractère c dans la console vtsh.
- ospf pour les entrées apprises via le protocole de routage dynamique. Le réseau correspondant au côté opposé au sommet du triangle est appris via OSPF puisque le sous-système réseau du noyau ne le connaît pas.

```
ip route ls
default via 192.168.104.129 dev enp0s1.360 onlink
10.10.0.0/24 dev sw-vlan10 proto kernel scope link src 10.10.0.1
10.48.0.0/29 dev enp0s1.480 proto kernel scope link src 10.48.0.1
10.48.1.0/29 dev enp0s1.481 proto kernel scope link src 10.48.1.1
10.48.2.0/29 nhid 85 proto ospf metric 20
    nexthop via 10.48.1.3 dev enp0s1.481 weight 1
    nexthop via 10.48.0.2 dev enp0s1.480 weight 1
192.168.104.128/27 dev enp0s1.360 proto kernel scope link src 192.168.104.130
```

La table de routage IPv6 ne fait apparaître aucune nouvelle entrée puisque les réseaux de conteneurs desservis par R2 et R3 n'ont pas encore été annoncés à ce stade de la configuration.

```
ip -6 route ls
::1 dev lo proto kernel metric 256 pref medium
2001:678:3fc:168::/64 dev enp0s1.360 proto kernel metric 256 pref medium
fd14:ca46:3864:a::/64 dev sw-vlan10 proto kernel metric 256 pref medium
fe80::/64 dev enp0s1 proto kernel metric 256 pref medium
fe80::/64 dev enp0s1.360 proto kernel metric 256 pref medium
fe80::/64 dev enp0s1.480 proto kernel metric 256 pref medium
fe80::/64 dev enp0s1.481 proto kernel metric 256 pref medium
fe80::/64 dev sw-vlan10 proto kernel metric 256 pref medium
fe80::/64 dev asw-host proto kernel metric 256 pref medium
default via fe80:168::1 dev enp0s1.360 metric 1024 onlink pref medium
```

## 6. Publier les routes par défaut via OSPF

Dans la topologie logique étudiée, le routeur R1 dispose d'un lien montant vers l'Internet. On peut donc considérer que ce lien est la route par défaut vers tous les réseaux non connus de l'aire OSPF contenant les trois routeurs.

Il est possible de publier une route par défaut via le protocole OSPF depuis le routeur R1 vers les routeurs R2 et R3.

Voici, pour mémoire, une copie des bases de données OSPFv2 et OSPFv3 avant la mise en place de la publication de route par défaut. On reconnaît les LSAs (Link State Advertisement) de type 1 et 2 qui correspondent respectivement aux annonces de routeurs et de réseaux.

```
R1# sh ip ospf database

    OSPF Router with ID (0.0.4.1)

        Router Link States (Area 0.0.0.0)

Link ID        ADV Router    Age Seq#         CkSum Link count
0.0.4.1        0.0.4.1       450 0x800000012 0xa296 2
0.0.4.2        0.0.4.2       372 0x800000008 0x2912 2
0.0.4.3        0.0.4.3       416 0x800000008 0x57dd 2

        Net Link States (Area 0.0.0.0)

Link ID        ADV Router    Age Seq#         CkSum
10.48.0.1     0.0.4.1       56 0x800000002 0xe232
10.48.1.1     0.0.4.1       86 0x800000002 0xe52d
10.48.2.3     0.0.4.3       35 0x800000002 0xc448
```

```
R1# sh ipv6 ospf6 database
```

```
Area Scoped Link State Database (Area 0)
```

Type	LSID	AdvRouter	Age	SeqNum	Payload
Rtr	0.0.0.0	0.0.6.1	497	8000000b	0.0.6.1/0.0.0.4
Rtr	0.0.0.0	0.0.6.1	497	8000000b	0.0.6.1/0.0.0.5
Rtr	0.0.0.0	0.0.6.2	419	80000005	0.0.6.1/0.0.0.4
Rtr	0.0.0.0	0.0.6.2	419	80000005	0.0.6.3/0.0.0.4
Rtr	0.0.0.0	0.0.6.3	463	80000005	0.0.6.1/0.0.0.5
Rtr	0.0.0.0	0.0.6.3	463	80000005	0.0.6.3/0.0.0.4
Net	0.0.0.4	0.0.6.1	1789	80000001	0.0.6.1
Net	0.0.0.4	0.0.6.1	1789	80000001	0.0.6.2
Net	0.0.0.5	0.0.6.1	54	80000002	0.0.6.1
Net	0.0.0.5	0.0.6.1	54	80000002	0.0.6.3
Net	0.0.0.4	0.0.6.3	1781	80000001	0.0.6.3
Net	0.0.0.4	0.0.6.3	1781	80000001	0.0.6.2

```
I/F Scoped Link State Database (I/F enp0s1.480 in Area 0)
```

Type	LSID	AdvRouter	Age	SeqNum	Payload
Lnk	0.0.0.4	0.0.6.1	930	80000005	fe80::baad:caff:fefe:c8
Lnk	0.0.0.3	0.0.6.2	1798	80000001	fe80::baad:caff:fefe:c9

```
I/F Scoped Link State Database (I/F enp0s1.481 in Area 0)
```

Type	LSID	AdvRouter	Age	SeqNum	Payload
Lnk	0.0.0.5	0.0.6.1	926	80000005	fe80::baad:caff:fefe:c8
Lnk	0.0.0.3	0.0.6.3	61	80000002	fe80::baad:caff:fefe:ca

Q15. Quelle est la condition préalable à respecter pour que le routeur R1 soit en mesure de publier une route par défaut via le protocole de routage OSPF ?

À partir des tables de routage relevées dans la [Section 5, « Configurer les démons OSPFv2 et OSPFv3 »](#), repérer le routeur qui dispose d'un accès vers un réseau qui n'appartient pas à la topologie triangle.

```
ip route ls default
ip -6 route ls default
sh ip route kernel
sh ipv6 route kernel
```

Une route par défaut doit exister avant d'être injectée dans une aire OSPF. Dans notre cas, une route statique par défaut suffit à respecter la condition préalable.

Sur la maquette, on valide la présence des routes par défaut à l'aide la commande ip au niveau système.

```
ip route ls default
default via 192.168.104.129 dev enp0s1.360 onlink
```

```
ip -6 route ls default
default via fe80::168::1 dev enp0s1.360 metric 1024 onlink pref medium
```

Au niveau de la console vtysh, ces même routes correspondent aux entrées marquées k pour kernel.

```
R1# sh ip route kernel
Codes: K - kernel route, C - connected, S - static, R - RIP,
O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
f - OpenFabric,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup
t - trapped, o - offload failure
```

```
K>* 0.0.0.0/0 [0/0] via 192.168.104.129, enp0s1.360 onlink, 04:50:31
```

```
R1# sh ipv6 route kernel
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
       v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

K>* ::/0 [0/1024] via fe80:168::1, enp0s1.360 onlink, 04:51:23
```

Q16. Quelle est l'instruction à utiliser pour publier une route par défaut via le protocole de routage OSPFv2 ?

Parmi toutes les méthodes de redistribution de routes disponibles avec le protocole OSPFv2, il en existe une dédiée à l'injection de route par défaut dans une aire normale. Consulter le guide [FRRouting User Guide](#).

Rechercher le mot clé `redistribution` dans la section OSPF route-map.

L'instruction qui correspond à la redistribution de route par défaut à destination des autres routeurs de l'aire OSPF est la suivante.

`default-information originate`

On doit l'appliquer dans la section `router ospf` de la configuration du routeur R1.

```
R1# conf t
R1(config)# router ospf
R1(config-router)# default-information originate
R1(config-router)# ^Z
```

Une fois cette instruction exécutée, le rôle du routeur R1 change. Il devient Autonomous System Boundary Router ou ASBR. Les bases de données sont complétées avec des LSAs de type 5.

```
R1# sh ip ospf database external

      OSPF Router with ID (0.0.4.1)

          AS External Link States

LS age: 38
Options: 0x2  : *|-|-|-|-|E|-
LS Flags: 0xb
LS Type: AS-external-LSA
Link State ID: 0.0.0.0 (External Network Number)
Advertising Router: 0.0.4.1
LS Seq Number: 80000001
Checksum: 0x259e
Length: 36

Network Mask: /0
Metric Type: 2 (Larger than any link state path)
TOS: 0
Metric: 10
Forward Address: 0.0.0.0
External Route Tag: 0
```

On voit apparaître une nouvelle rubrique baptisée AS External Link States. Ce nouveau rôle pour le routeur R1 apparaît aussi lorsque l'on affiche l'état de l'instance de routage OSPFv2.

```

R1# sh ip ospf
OSPF Routing Process, Router ID: 0.0.4.1
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is disabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 0 millisec(s)
Minimum hold time between consecutive SPFs 50 millisec(s)
Maximum hold time between consecutive SPFs 5000 millisec(s)
Hold time multiplier is currently 1
SPF algorithm last executed 1m50s ago
Last SPF duration 133 usecs
SPF timer is inactive
LSA minimum interval 5000 msec
LSA minimum arrival 1000 msec
Write Multiplier set to 20
Refresh timer 10 secs
Maximum multiple paths(ECMP) supported 256
Administrative distance 110
This router is an ASBR (injecting external routing information)
Number of external LSA 1. Checksum Sum 0x0000259e
Number of opaque AS LSA 0. Checksum Sum 0x00000000
Number of areas attached to this router: 1
All adjacency changes are logged
Area ID: 0.0.0.0 (Backbone)
  Number of interfaces in this area: Total: 2, Active: 2
  Number of fully adjacent neighbors in this area: 2
  Area has no authentication
  SPF algorithm executed 21 times
  Number of LSA 6
  Number of router LSA 3. Checksum Sum 0x0001277e
  Number of network LSA 3. Checksum Sum 0x00028ba7
  Number of summary LSA 0. Checksum Sum 0x00000000
  Number of ASBR summary LSA 0. Checksum Sum 0x00000000
  Number of NSSA LSA 0. Checksum Sum 0x00000000
  Number of opaque link LSA 0. Checksum Sum 0x00000000
  Number of opaque area LSA 0. Checksum Sum 0x00000000

```

Le routeur R1, est maintenant à la frontière entre deux systèmes autonomes. Il est responsable de l'émission des LSAs de type 5 à destination des autres routeurs de l'aire.

Ici, R1 possède une route statique définie au niveau système vers l'Internet. Cette route statique est redistribuée R2 et R3. Cette route apparaît comme une entrée de type E2 dans les tables de routage de ces routeurs.

L'indicateur E2 correspond au type par défaut des routes apprises par le biais de la redistribution. La métrique est un point important à considérer avec les routes de type E2. Ces routes ne présentent que le coût du chemin allant du routeur ASBR vers le réseau de destination ; ce qui ne correspond pas au coût réel du chemin à l'intérieur de l'aire OSPF.

Q17. Quelle est l'instruction à utiliser pour publier une route par défaut via le protocole de routage OSPFv3 ?

Reprendre la même démarche de la question précédente avec le protocole OSPFv3. Consulter le guide [FRRouting User Guide](#).

Rechercher le mot clé redistribution dans la section OSPF6 route-map.

L'instruction est identique pour les deux versions du protocole OSPF.

default-information originate

On doit l'appliquer dans la section router ospf6 de la configuration du routeur R1.

```

R1# conf t
R1(config)# router ospf6
R1(config-ospf6)# default-information originate
R1(config-ospf6)# ^Z

```

Une fois cette instruction exécutée, le rôle du routeur R1 change. Il devient Autonomous System Boundary Router ou ASBR. Les bases de données sont complétées avec des LSAs de type 5.

```
R1# sh ipv6 ospf6 database as-external

      AS Scoped Link State Database

Type LSId          AdvRouter      Age   SeqNum          Payload
ASE  0.0.0.1      0.0.6.1       58   80000001       ::
```

On voit apparaître une nouvelle rubrique baptisée AS Scoped Link State Database. Ce nouveau rôle pour le routeur R1 apparaît aussi lorsque l'on affiche l'état de l'instance de routage OSPFv3.

```
R1# sh ipv6 ospf6
OSPFv3 Routing Process (0) with Router-ID 0.0.6.1
Running 02:40:17
LSA minimum arrival 1000 msec
Maximum-paths 256
Administrative distance 110
Initial SPF scheduling delay 0 millisecond(s)
Minimum hold time between consecutive SPF's 50 millisecond(s)
Maximum hold time between consecutive SPF's 5000 millisecond(s)
Hold time multiplier is currently 1
SPF algorithm last executed 00:01:46 ago, reason R+, R-, A
Last SPF duration 0 sec 118 usec
SPF timer is inactive
Number of AS scoped LSAs is 1
Number of areas in this router is 1
Authentication Sequence number info
  Higher sequence no 0, Lower sequence no 0
All adjacency changes are logged

Area 0
  Number of Area scoped LSAs is 6
  Interface attached to this area: enp0s1.480 enp0s1.481
  SPF last executed 106.592900s ago
```

Q18. Comment la publication de route par défaut apparaît-elle sur les autres routeurs de la topologie triangle ?

Relevez, dans la console vtysh, la métrique de la route par défaut sur les routeurs qui n'ont pas une connexion directe vers l'Internet.

Les instructions à utiliser pour traiter cette question entrent dans la liste suivante.

```
show ip route A.B.C.D/MM
show ipv6 route A:B:C::D/MM
show ip ospf route
show ipv6 ospf6 route
```

En prenant l'exemple du routeur R2, on retrouve les informations suivantes.

- Vue de la table de routage :

```
R2# sh ip route 0.0.0.0
Routing entry for 0.0.0.0/0
  Known via "ospf", distance 110, metric 10, best
  Last update 00:07:26 ago
  * 10.48.0.1, via enp0s1.480, weight 1
```

```
R2# sh ipv6 route ::/0
Routing entry for ::/0
  Known via "ospf6", distance 110, metric 10, best
  Last update 00:04:43 ago
  * fe80::baad:caff:fe80:c8, via enp0s1.480, weight 1
```

- Vue de la base de topologie OSPF :



```
R2# sh ip ospf route
===== OSPF network routing table =====
N   10.48.0.0/29      [1] area: 0.0.0.0
                        directly attached to enp0s1.480
N   10.48.1.0/29      [2] area: 0.0.0.0
                        via 10.48.0.1, enp0s1.480
                        via 10.48.2.3, enp0s1.482
N   10.48.2.0/29      [1] area: 0.0.0.0
                        directly attached to enp0s1.482

===== OSPF router routing table =====
R   0.0.4.1          [1] area: 0.0.0.0, ASBR
                        via 10.48.0.1, enp0s1.480

===== OSPF external routing table =====
N E2 0.0.0.0/0    [1/10] tag: 0
                        via 10.48.0.1, enp0s1.480
```

```
R2# sh ipv6 ospf6 route
*N E2 ::/0          fe80::baad:caff:fefe:c8   enp0s1.480 00:07:05
```

Avec les copies d'écran ci-dessus, on vérifie bien que les routes par défaut ont été apprises via le protocole OSPF.

Q19. Comment assurer la traduction d'adresses source sur l'interface de sortie du routeur R1-rouge vers l'Internet ?

Reprendre la section Rôle routeur du support [Routage inter-VLAN dans un contexte IaaS](#).

Dans le cas de la maquette, c'est l'interface `enp0s1.360` qui assure l'interconnexion avec l'Internet. Voici une copie de l'instruction d'ajout de la règle de traduction des adresses sources et de sa sauvegarde pour IPv4 et IPv6.

```
sudo iptables -t nat -A POSTROUTING -o enp0s1.360 -j MASQUERADE
sudo sh -c "iptables-save >/etc/iptables/rules.v4"
```

```
sudo ip6tables -t nat -A POSTROUTING -o enp0s1.360 -j MASQUERADE
sudo sh -c "ip6tables-save >/etc/iptables/rules.v6"
```

Une fois le jeu de règles en place, on peut vérifier leur utilisation à partir du relevé des compteurs de paquets traités.

```
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 OUTPUT (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
 150 10288 MASQUERADE all -- *      enp0s1.360 0.0.0.0/0 0.0.0.0/0
```

```
sudo ip6tables -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 OUTPUT (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
 57 5083 MASQUERADE all *      enp0s1.360 ::/0      ::/0
```

## 7. Ajouter un réseau de conteneurs à chaque routeur du triangle

Dans cette section, on crée un réseau de conteneur attaché à chaque routeur de la topologie triangle. L'objectif est de peupler les tables de routage en ajoutant un lien OSPF au delà des routeurs.

Une fois ces réseaux en place, il est possible de réaliser des tests de connectivité entre conteneurs et d'optimiser les métriques.

Cette partie reprend le contenu de la section Rôle serveur de conteneurs du support de travaux pratiques [Routage inter-VLAN dans un contexte IaaS](#).

Q20. Quelle est la suite de commande à exécuter pour mettre en place de gestionnaire de conteneurs LXD ?

consulter la section rôle serveur de conteneurs.

Voici un exemple pour le routeur R2.

```
sudo apt install snapd
sudo snap install lxd
sudo adduser etu lxd
```

Après déconnexion - reconnexion, on peut passer à l'étape de définition du profil par défaut pour la création des conteneurs.

```
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 (cephobject, dir, lvm, btrfs, ceph) [default=btrfs]:
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]:
Size in GiB of the new loop device (1GiB minimum) [default=23GiB]:
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]:
Name of the existing bridge or host interface: sw-vlan20
Would you like the LXD server 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]:
```

```
lxc profile device set default eth0 nictype bridged
```

Voici une copie du profil après configuration.

```
lxc profile show default
config: {}
description: Default LXD profile
devices:
  eth0:
    name: eth0
    nictype: bridged
    parent: sw-vlan20
    type: nic
  root:
    path: /
    pool: default
    type: disk
name: default
used_by: []
```

Q21. Comment créer 3 conteneurs avec un adressage IPv4 et IPv6 conforme au plan défini ?

consulter la section rôle serveur de conteneurs.

On commence par la création des 3 conteneurs demandés.

```
for i in {0..2}
do
  lxc launch images:debian/12 c$i
done
```







Rechercher le mot clé référence dans l'index des instructions de configuration. Consulter la documentation [FRRouting User Guide](#).

C'est l'instruction auto-cost reference-bandwidth qui permet de fixer une nouvelle référence de coût de lien en Mbps.

```
R1# conf t
R1(config)# router ospf
R1(config-router)# auto-cost reference-bandwidth
(1-4294967) The reference bandwidth in terms of Mbits per second
R1(config-router)# auto-cost reference-bandwidth 100000
R1(config-router)# ^Z
```

```
R1# conf t
R1(config)# router ospf6
R1(config-ospf6)# auto-cost reference-bandwidth
(1-4294967) The reference bandwidth in terms of Mbits per second
R1(config-ospf6)# auto-cost reference-bandwidth 100000
R1(config-ospf6)# ^Z
```

Voici un extrait de la configuration du routeur R2 après application d'une nouvelle référence à 100000Mbps soit 100Gbps.

```
R2# conf t
R2(config)# router ospf
R2(config-router)# auto-cost reference-bandwidth 100000
R2(config-router)# router ospf6
R2(config-ospf6)# auto-cost reference-bandwidth 100000
R2(config-ospf6)# ^Z
```

Il est indispensable que tous les routeurs de l'aire OSPF aient la même référence de calcul de métrique. La même instruction doit donc être implantée dans la configuration des trois routeurs.

Q27. Comment modifier le débit binaire d'un lien à 10Gbps ?

Normalement, le débit d'un lien est directement extrait des paramètres du composant de l'interface connectée au lien. Dans le cas d'interfaces qui n'ont aucune réalité physique, ce débit peut être attribué arbitrairement par configuration. On doit rechercher dans les options de la console vtysh le moyen d'attribuer une indication de débit aux sous-interfaces de VLANs.

Comme indiqué dans la [Section 3, « Préparer les systèmes pour le routage IPv4 et IPv6 »](#), c'est dans la configuration des interfaces que l'on attribue le débit binaire d'une interface réseau.

```
R3# sh int sw-vlan30
Interface sw-vlan30 is up, line protocol is up
  Link ups:      1      last: 2022/10/24 10:13:11.41
  Link downs:   2      last: 2022/10/24 10:13:11.40
  vrf: default
  index 7 metric 0 mtu 1500 speed 0
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  Type: Ethernet
  HWaddr: a6:91:26:2f:34:42
  inet 10.30.0.1/24
  inet6 fd14:ca46:3864:1e::1/64
  inet6 fe80::a491:26ff:fe2f:3442/64
  Interface Type Other
  Interface Slave Type None
  protodown: off
```

Q28. Comment peut-on identifier le débit d'un lien dans la configuration OSPF ?

Visualiser les paramètres des interfaces réseau depuis la console vtysh. Les commandes suivantes peuvent être lancées sur chacun des trois routeurs.

```
show ip ospf interface
show ipv6 ospf6 interface
```

Voici le résultat obtenu sur le routeur R3-bleu.

```
R3# conf t
R3(config)# int sw-vlan30
R3(config-if)# bandwidth 10000
R3(config-if)# ^Z
```

```
R3# sh ip ospf interface sw-vlan30
sw-vlan30 is up
  ifindex 7, MTU 1500 bytes, BW 10000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.30.0.1/24, Broadcast 10.30.0.255, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 0.0.4.3, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 0.0.4.3 Interface Address 10.30.0.1/24
  No backup designated router on this network
  Multicast group memberships: <None>
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
  No Hellos (Passive interface)
  Neighbor Count is 0, Adjacent neighbor count is 0
```

Q29. Quel est le coût d'accès au réseau de conteneurs attaché au routeur R3 depuis le routeur R2 ? Justifier la valeur de métrique obtenue.

À partir des informations de routage sur R2, faire la somme des métriques de chaque lien entre les deux extrémités en communication.

Il est possible d'obtenir les informations de calcul de métrique à partir de l'affichage d'une route particulière à la console vtysh.

```
R2# sh ip route 10.30.0.0/24
Routing entry for 10.30.0.0/24
  Known via "ospf", distance 110, metric 11, best
  Last update 00:43:47 ago
  * 10.48.2.3, via enp0s1.482, weight 1
```

Dans l'exemple ci-dessus, la métrique d'accès au réseau 10.30.0.0/24 correspond à la somme des métriques de deux liens. La somme donne une métrique de 11.

- Le lien entre R2 et R3 est à 100Gbps, soit une métrique de 1
- Le lien du réseau de conteneurs est à 10Gbps, soit une métrique de 10.

On obtient un résultat identique avec la table de routage IPv6.

```
R2# sh ipv6 route fd14:ca46:3864:1e::/64
Routing entry for fd14:ca46:3864:1e::/64
  Known via "ospf6", distance 110, metric 11, best
  Last update 00:45:59 ago
  * fe80::baad:caff:fefe:ca, via enp0s1.482, weight 1
```

## 9. Consulter les documents de référence

### Configuration d'une interface réseau

Le support [Configuration d'une interface de réseau local](#) présente les opérations de configuration d'une interface réseau et propose quelques manipulations sur les protocoles de la pile TCP/IP

### Introduction au routage inter-VLAN

Le support [Routage Inter-VLAN](#) introduit le principe du routage inter-VLAN ainsi que ses conditions d'utilisation. C'est aussi un support de travaux pratiques dans lequel on n'utilise que du routage statique et de la traduction d'adresses sources (S-NAT) pour acheminer le trafic utilisateur entre les différents réseaux.

### Documentation FRRouting

La page à l'adresse : [FRRouting User Guide](#) fournit toutes les instructions utilisables dans console unifiée vtysh.

Ces commandes sont largement inspirées du système d'exploitation IOS de Cisco™.

## 10. Sauvegarder les fichiers de configuration

---

Voici une copie des fichiers de configuration des trois routeurs une fois toutes les manipulations réalisées.

- Routeur R1.

```
frr version 8.3.1
frr defaults traditional
hostname R1
log syslog informational
service integrated-vtysh-config
!
interface enp0s1.480
  bandwidth 100000
  ip ospf area 0
  ipv6 ospf6 area 0
exit
!
interface enp0s1.481
  bandwidth 100000
  ip ospf area 0
  ipv6 ospf6 area 0
exit
!
interface sw-vlan10
  bandwidth 10000
  ip ospf area 0
  ip ospf passive
  ipv6 ospf6 area 0
  ipv6 ospf6 passive
exit
!
router ospf
  ospf router-id 0.0.4.1
  log-adjacency-changes detail
  default-information originate
exit
!
router ospf6
  ospf6 router-id 0.0.6.1
  log-adjacency-changes detail
  default-information originate
exit
!
```

- Routeur R2.

```
frr version 8.3.1
frr defaults traditional
hostname R2
log syslog informational
service integrated-vtysh-config
!
interface enp0s1.480
  bandwidth 100000
  ip ospf area 0
  ipv6 ospf6 area 0
exit
!
interface enp0s1.482
  bandwidth 100000
  ip ospf area 0
  ipv6 ospf6 area 0
exit
!
interface sw-vlan20
  bandwidth 10000
  ip ospf area 0
  ip ospf passive
```



```
ipv6 ospf6 area 0
ipv6 ospf6 passive
exit
!
router ospf
ospf router-id 0.0.4.2
log-adjacency-changes detail
exit
!
router ospf6
ospf6 router-id 0.0.6.2
log-adjacency-changes detail
exit
!
```

- Routeur R3.

```
frr version 8.3.1
frr defaults traditional
hostname R3
log syslog informational
service integrated-vtysh-config
!
interface enp0s1.481
bandwidth 100000
ip ospf area 0
ipv6 ospf6 area 0
exit
!
interface enp0s1.482
bandwidth 100000
ip ospf area 0
ipv6 ospf6 area 0
exit
!
interface sw-vlan30
bandwidth 10000
ip ospf area 0
ip ospf passive
ipv6 ospf6 area 0
ipv6 ospf6 passive
exit
!
router ospf
ospf router-id 0.0.4.3
log-adjacency-changes detail
exit
!
router ospf6
ospf6 router-id 0.0.6.3
log-adjacency-changes detail
exit
!
```