Gestion du réseau

M1 RÉSEAUX & TÉLÉCOMS – RT0702

OLIVIER FLAUZAC

Ponts réseau

Modes réseau

Sans réseau

« Capture » de carte

Forward de port

Réseau interne

- nat
- réseau privé
- bridge

Pont réseau

Bridge

Dispositif permettant de faire communiquer plusieurs interfaces réseaux

Hub reliant plusieurs interfaces réseaux

Bus d'interconnexion

Exploitation couche 2

Pont physique / pont virtuel

Comment créer un bridge virtuel ?

Bridge ?

Pont entre plusieurs interfaces
- interfaces physiques
- interface virtuelles

Un bridge ne contient pas obligatoirement une interface physique

Un bridge ne contient pas obligatoirement une interface virtuelle

Attention à la terminologie !
- bridge = concentrateur d'interfaces
- bridge = connexion sur le réseau de l'hôte

Création
- au démarrage du système `/etc/network/interfaces` + `bridge-utils`
- en ligne de commande
  - avec les outils `bridge-utils`
  - avec les outils `iproute2`

Bridge au démarrage

Configuration dans `/etc/network/interfaces`

- création du bridge
- ajout des interfaces , positionnement des propriétés
- autorisation de forwarding

/etc/network/interfaces

```
auto lo
iface lo inet loopback
auto br0
iface br0 inet dhcp
    bridge_ports eth0
    bridge_maxwait 0
```

/etc/sysctl.conf

```
net.ipv4.ip_forward=1
```

Rechargement configuration

```
sysctl -p /etc/sysctl.conf
```

Aucune interface dans le bridge !

```
br0       Link encap:Ethernet  HWaddr 00:0c:29:24:0c:6e
          inet adr:192.168.230.139  Bcast:192.168.230.255  Masque:255.255.255.0
          adr inet6: fe80::20c:29ff:fe24:c6e/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:131 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:12172 (11.8 KiB)  TX bytes:10678 (10.4 KiB)

eth0      Link encap:Ethernet  HWaddr 00:0c:29:24:0c:6e
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:132 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:14181 (13.8 KiB)  TX bytes:10732 (10.4 KiB)

lo        .....
```

pas d'adresse sur eth0
l'adresse est sur le bridge

Mise en place de bridge avec iproute

Création du bridge

Configuration des adresses

Enregistrement des interfaces

Gestion du routage

```
root@debLXC:/home/user# ip link add name br type bridge
root@debLXC:/home/user# ip link set eth0 down
root@debLXC:/home/user# ip addr flush dev eth0
root@debLXC:/home/user# ip link set eth0 up
root@debLXC:/home/user# ip addr add 10.0.2.15/24 broadcast 10.0.2.255 dev br
root@debLXC:/home/user# ip link set dev br up
root@debLXC:/home/user# ip link set eth0 master br
root@debLXC:/home/user# bridge link
2: eth0 state UP : <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
          ...master br state forwarding priority 32 cost 4
```

```
root@debLXC:/home/user# ip addr show
1: lo: ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast ...
          ...master br state UP group default qlen 1000
    link/ether 08:00:27:c1:26:3c brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fec1:263c/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: ...
4: br: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue ...
          ...state UP group default
    link/ether 08:00:27:c1:26:3c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global br
       valid_lft forever preferred_lft forever
    inet6 fe80::e8eb:31ff:fee6:7ecf/64 scope link
       valid_lft forever preferred_lft forever
root@debLXC:/home/user# ip route add default via 10.0.2.2
```

LXC et le réseau

Connexions au réseau

Pas de réseau

- `empty`
- juste la boucle locale dans le conteneur
- pas de communication

Physique

- `phys`
- attribution à l'invité d'une carte réseau de l'hôte
- perte de la carte pour l'hôte, et des éléments réseaux associés
- communications identiques avec celles d'origine de l'hôte

VETH

Virtual Ethernet

- `veth`
- intégration d'une connexion Ethernet dans un bridge de l'hôte
  - définition du type de connexion :
    - réseau interne
    - nat
    - full bridge
- liaison entre l'eth du conteneur est le veth dans le bridge de l'hôte
- permet l'isolation des conteneur au niveau réseau
  - des configurations réseau
  - du *forward* de port

LXC - réseau privé

Principe

Mise en place d’un réseau interne

Communication entre les machines du réseau interne

Plusieurs réseaux internes possibles

- Séparation des réseaux

Plan d’adressage à la charge de l’administrateur

Configuration LXC

/var/lib/lxc/ct1/config

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br
lxc.network.veth.pair = br-ct1
lxc.network.ipv4 = 192.168.1.101/24

lxc.rootfs = /var/lib/lxc/ct1/rootfs

# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf

# Container specific configuration
lxc.mount = /var/lib/lxc/ct1/fstab
lxc.utsname = ct1
lxc.arch = amd64
lxc.autodev = 1
lxc.kmsg = 0
```

Script UP

1. Création du bridge
2. Activation du bridge
3. Démarrage des conteneurs
   1. Création des TAP
   2. Intégration des TAP dans le bridge

private-up.sh

```
#!/bin/bash

ip link add name br type bridge
ip link set dev br up

lxc-start -n ct1 -d
lxc-start -n ct2 -d
```

Script DOWN

1. Destruction du bridge
2. Arrêt des conteneurs
   - Destruction des TAP
3. Destruction du bridge

private-down.sh

```
#!/bin/bash


lxc-stop -n ct2

lxc-stop -n ct1

ip link delete dev br
```

Configuration réseau de l'hôte

ip addr show

```
1: lo: …
2: eth0: …
3: eth1: …
9: br: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether fe:75:ee:01:dd:b3 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::7827:67ff:feb4:4727/64 scope link
       valid_lft forever preferred_lft forever
11: br-ct1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br state UP
group default qlen 1000
    link/ether fe:a1:3b:b6:e6:17 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::fca1:3bff:feb6:e617/64 scope link
       valid_lft forever preferred_lft forever
13: br-ct2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br state UP
group default qlen 1000
    link/ether fe:75:ee:01:dd:b3 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::fc75:eeff:fe01:ddb3/64 scope link
       valid_lft forever preferred_lft forever
```

Configuration réseau d'un invité

ip addr show

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
10: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether d2:d4:b9:8b:2e:c7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.101/24 brd 192.168.1.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::d0d4:b9ff:fe8b:2ec7/64 scope link
       valid_lft forever preferred_lft forever
```

Test depuis le conteneur

ping -c 3 192.168.1.102

```
PING 192.168.1.102 (192.168.1.102) 56(84) bytes of data.
64 bytes from 192.168.1.102: icmp_seq=1 ttl=64 time=1.03 ms
64 bytes from 192.168.1.102: icmp_seq=2 ttl=64 time=0.077 ms
64 bytes from 192.168.1.102: icmp_seq=3 ttl=64 time=0.053 ms
```

Attention à l'installation de ping !!!

LXC - Full bridge

Principe

Intégration de l’invité dans le réseau de l’hôte

Mise en place d’un bridge contenant :

- L’interface de sortie vers le réseau
- L’interface de l’hôte

Exploitation des politiques réseau du réseau de l’hôte

- Adressage
- DHCP
- …

Configuration LXC

/var/lib/lxc/ct1/config

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br
lxc.network.veth.pair = br-ct1

lxc.rootfs = /var/lib/lxc/ct1/rootfs

# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf

# Container specific configuration
lxc.mount = /var/lib/lxc/ct1/fstab
lxc.utsname = ct1
lxc.arch = amd64
lxc.autodev = 1
lxc.kmsg = 0
```

Script up

1. Arrêt du dhcp (selon configuration)
2. Désactivation de l'interface eth0
3. Suppression de la configuration de eth0
4. Activation de eth0
5. Création du bridge
6. Activation du bridge
7. Ajout de eth0 dans le bridge
8. Activation du dhcp pour le bridge
    1. Configuration réseau du bridge
9. Lancement du conteneur
    1. Création du TAP associé
    2. Intégration du TAP dans le bridge

```
#!/bin/bash
piddhcp=$(pgrep -f dhcp)
kill -9 $piddhcp
ip link set dev eth0 down
ip addr flush eth0
ip link set dev eth0 up
ip link add name br type bridge
ip link set dev br up
ip link set dev eth0 master br
dhclient br
lxc-start -n ct1 -d
```

Script down

1. Arrêt du dhcp (selon configuration)
2. Arrêt du conteneur
3. Extraction de eth0 du bridge
4. Destruction du bridge
5. Activation du DHCP sur eth0s

```
#!/bin/bash
piddhcp=$(pgrep -f dhcp)
kill -9 $piddhcp
lxc-stop –n ct1
ip link set dev eth0 nomaster
ip link delete dev br
dhclient eth0
```

LXC - Bridge NAT

Principe

Mise en place d’un réseau privé pour l’invité

Mise en place de règle de transfert de paquet entre le réseau privé et l’interface de sortie

Configuration LXC

/var/lib/lxc/ct1/config

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br
lxc.network.veth.pair = br-ct1
lxc.network.ipv4 = 192.168.10.101/24

lxc.rootfs = /var/lib/lxc/ct1/rootfs

# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf

# Container specific configuration
lxc.mount = /var/lib/lxc/ct1/fstab
lxc.utsname = ct1
lxc.arch = amd64
lxc.autodev = 1
lxc.kmsg = 0
```

Script up

1. Création du bridge
2. Affectation d'une adresse au bridge
3. Activation du bridge
4. Mise en place de la règle iptables de NAT
5. Lancement du conteneur
    1. Création du TAP associé
    2. Intégration du TAP dans le bridge

```
#!/bin/bash


ip link add name br type bridge

ip addr add 192.168.10.1/24 dev br

ip link set dev br up

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

lxc-start -n ct2 -d
```

Script down

1. Arrêt du conteneur
2. Destruction de la règle Iptables
3. Destruction du bridge

```
#!/bin/bash

lxc-stop -n ct2
iptables –F
ip link delete dev br
```

VLAN

Vlan

- connexion à un vlan d'un hôte
- spécification du numéro de vlan dans le fichier config
- permet l'isolation réseau des conteneurs

MacVlan

`macvlan`

- connexion à l'extérieur par l'interface de l'hôte
- communication inter invités impossible par défaut
- communication invité / hôte impossible par défaut
- isolation totale d'un invité

`macvlan` en mode bridge

- connexion à l'extérieur par l'interface de l'hôte
- communication inter invités possible par défaut
- communication invité / hôte impossible par défaut
- isolation totale d'un groupe d'invités

Qemu et le réseau

Modes réseau

Fonctionnement par émulation de carte réseau

Emulation de divers matériels

Quatre modes connus

- user mode
- redirection
- tap
- VDE

La gestion du réseau

Commande

- `net`
- `netdev`

Modes

- `user` mode utilisateur
- `nic` création d'une interface réseau
- `tap` lien vers une interface tap
- `ifname` nom de l'interface
- `script` script à exécuter
- `id` identifiant du réseau

Le mode réseau user

SLIRP

Mode par défaut

Ne nécessite pas de droits administrateur

Emulation par défaut d'une carte Intel e1000 PCI

Placé en NAT sur l'hôte

Autorise l'accès sortant

Interdit l'accès entrant

Ne supporte que TCP et UDP mais pas ICMP

Propriétés par défaut :

- serveur DHCP en 10.0.2.2
- adresses distribuées à partir de 10.0.2.15
- DNS virtuel en 10.0.2.3
- serveur samba virtuel en 10.0.2.4 (accès à l'hôte)

Commande

```
qemu-system-x86_64 -k fr -m 512 -hda myDeb.img &
```

```
qemu-system-x86_64  -k fr -m 512 -hda myDeb.img -net nic -net user &
```

```
qemu-system-x86_64 -k fr -m 512 -hda myDeb.img -netdev user,id=network0
                   -device e1000,netdev=network0 &
```

Mode réseau redirection de ports

Extension du mode user

Redirection d'un port de l'hôte sur l'invité

Utilisé pour le partage de ressources ou accès SSH

Pas de règles iptables à spécifier

`redir` directive de redirection

`tcp:port_hôte::port_invité`

```
qemu-system-x86_64 -k fr -m 512 -redir tcp:5555::80 -hda myDeb.img &
```

Le mode TAP

Utilisation

- accès aux tap offert par l'hôte
- utilisation dans le cadre de pont réseau
- utilisable en mode
  - réseau ponté : hôte et invité sur le même réseau
  - NAT : invité dans un sous réseau de l'hôte

Exploitation possible de script de configuration réseau

- possibilité de script depuis un compte utilisateur
  - /etc/qemu-ifup
  - /etc/qemu-ifdown

```
qemu-system-x86_64 -k fr -m 512 -net nic -netdev tap,ifname=tap0,script=no  -hda myDeb.img &
```

Déclinaison du mode TAP

Mode TAP utilisé pour les différents mode *bridge*

- Mode NAT définit par défaut (SLIRP)

Définition des liaisons et des connexions en fonction de la connectivité du bridge

Intégration du *lien TAP,* dans les bridges définis

Nécessite de mettre en place un système d'adressage

Open Vswitch

Généralités

Equipement réseau virtuel

- Switch
- Routeur

Permet de connecter des machines virtuelles

Utilisable comme un équipement réel

- Cascadable
- Connaissance des VLAN

Interconnectable sur plusieurs machines

Installation

Installation des packages

- openvswitch-common
- openvswitch-switch

De nombreux autres packages

Extension des capacités

Commandes de base

Création d'un switch (bridge)

- ovs-vsctl add-br brName

Destruction d'un switch (bridge)

- ovs-vsctl del-br brName

Ajout d'un port dans un switch

- ovs-vsctl add-port brName portName

Suppression d'un port dans un switch

- ovs-vsctl del-port portName

Récupération de l'état

- ovs-vsctl show

```
root@deb-ovs-103:~# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:b4:f6:a5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feb4:f6a5/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:6d:bb:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.59.103/24 brd 192.168.59.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe6d:bb68/64 scope link
       valid_lft forever preferred_lft forever
4: ovs-system: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether ca:fd:ae:85:7d:e4 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::c8fd:aeff:fe85:7de4/64 scope link
       valid_lft forever preferred_lft forever
root@deb-ovs-103:~# ovs-vsctl show
dd42b6cf-c861-41ba-be4a-ccefb9dd2bcd
    ovs_version: "2.3.0"
```

Intégration des invités

Ajout de l'interface TAP définie

Qemu / KVM

- Création du TAP
- Intégration dans le bridge
- Ajout dans la ligne de commande

LXC

- Définition du TAP dans le fichier de configuration & intégration dans le bridge
- Mise en place d'un script automatique

Réseau privé

Configuration conteneur

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.veth.pair = br-deb1
lxc.network.ipv4 = 192.168.1.102/24

lxc.rootfs = /var/lib/lxc/deb1/rootfs

# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf

# Container specific configuration
lxc.mount = /var/lib/lxc/deb1/fstab
lxc.utsname = deb1
lxc.arch = amd64
lxc.autodev = 1
lxc.kmsg = 0
```

Commandes

```
ovs-vsctl add-br br0
ip addr add 192.168.1.1/24 dev br0
ip link set dev br0 up
ip link set dev ovs-system up
lxc-start -n deb0 -d
ovs-vsctl add-port br0 br-deb0
lxc-start -n deb1 -d
ovs-vsctl add-port br0 br-deb1
```

```
lxc-stop -n deb1
lxc-stop -n deb0
ovs-vsctl del-br br0
```

Full bridge

Configuration conteneur

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.veth.pair = br-deb1

lxc.rootfs = /var/lib/lxc/deb1/rootfs

# Common configuration
lxc.include = /usr/share/lxc/config/debian.common.conf

# Container specific configuration
lxc.mount = /var/lib/lxc/deb1/fstab
lxc.utsname = deb1
lxc.arch = amd64
lxc.autodev = 1
lxc.kmsg = 0
```

Commandes

```
piddhcp=$(pgrep -f dhcp)
kill -9 $piddhcp
ip link set dev eth0 down
ip addr flush eth0
ip link set dev eth0 up

ovs-vsctl add-br br0
ip link set dev br0 up
ip link set dev ovs-system up
ovs-vsctl add-port br0 eth0
dhclient br0
lxc-start -n deb0 -d
ovs-vsctl add-port br0 br-deb0
```

```
piddhcp=$(pgrep -f dhcp)
kill -9 $piddhcp
lxc-stop -n deb0
ovs-vsctl del-port eth0
ovs-vsctl del-br br0
dhclient eth0
```