Configuration conteneur xfce

Configuration de l'hote

```
echo "/dev/dm-0 /backup  btrfs defaults 0 0" >> /etc/fstab
echo "/dev/dm-1 /share/home/ btrfs defaults,subvol=@home 0 0" >> /etc/fstab
echo "/dev/dm-3 /share/data/ btrfs defaults 0 0" >> /etc/fstab
```

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sdb2 during installation
UUID=4637cc09-4076-479f-9ffd-38fade38e1b1 /               ext4
errors=remount-ro 0       1
/dev/mapper/vg01-sp01 none            swap    sw              0       0
/dev/dm-0 /backup  btrfs defaults 0 0
/dev/dm-1 /share/home/ btrfs defaults,subvol=@home 0 0
/dev/dm-3 /share/data/ btrfs defaults 0 0
```

```
sed -i "s/main/main\ non-free/g" /etc/apt/sources.list.d/sources.list
apt update
apt install firmware-realtek
apt install tklpatch
apt install btrfs-tools
apt install console-data
apt install alsa-utils
aplay -L
ls -l /dev/snd

total 0
drwxr-xr-x 2 root root       80 Dec 17 15:41 by-path
crw-rw---- 1 root audio 116,  7 Dec 17 15:41 controlC0
crw-rw---- 1 root audio 116,  2 Dec 17 15:41 controlC1
crw-rw---- 1 root audio 116,  9 Dec 17 15:41 hwC0D0
crw-rw---- 1 root audio 116,  6 Dec 17 15:41 hwC1D0
crw-rw---- 1 root audio 116,  8 Dec 17 15:41 pcmC0D3p
crw-rw---- 1 root audio 116,  4 Dec 17 15:41 pcmC1D0c
crw-rw---- 1 root audio 116,  3 Dec 17 15:41 pcmC1D0p
crw-rw---- 1 root audio 116,  5 Dec 17 15:41 pcmC1D2c
crw-rw---- 1 root audio 116,  1 Dec 17 15:41 seq
crw-rw---- 1 root audio 116, 33 Dec 17 15:41 timer
```

Creation de la VM

```
btrfs subvolume create /backup/lxc
```

/etc/lxc/lxc.conf

```
# override lxc's hardwired defaults
lxc.default_config=/etc/lxc/default.conf
lxc.lxcpath=/backup/lxc
lxc.bdev.lvm.vg=turnkey
lxc.bdev.lvm.thin_pool=turnkey
lxc.bdev.zfs.root=lxc
```

```
lxc-create -t download -n xfce -- -d ubuntu -r bionic -a amd64
lxc-execute -n xfce -- sudo adduser jacques.nougat --force-badname
lxc-execute -n xfce -- passwd root

lxc-stop -n xfce
lxc-start -n xfce
lxc-console -n xfce
```

Configuration de la VM

Configuration du fichier de config

/backup/lxc/xfce/config

```
# Template used to create this container: /usr/share/lxc/templates/lxc-
download
# Parameters passed to the template: -d ubuntu -r bionic -a amd64
# Template script checksum (SHA-1): 740c51206e35463362b735e68b867876048a8baf
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)

# use nat bridge by default

# Distribution configuration
lxc.include = /usr/share/lxc/config/ubuntu.common.conf
lxc.arch = linux64

# Container specific configuration
```

```
lxc.include = /etc/lxc/natbridge.conf
lxc.rootfs = /backup/lxc/xfce/rootfs
lxc.rootfs.backend = dir
lxc.utsname = xfce
lxc.cgroup.devices.allow = c 116:* rwm
lxc.cgroup.devices.allow = c 226:0 rwm
lxc.cgroup.devices.allow = c 226:128 rwm
lxc.cgroup.devices.allow = c 4:7 rwm
lxc.cgroup.devices.allow = c 29:0 rwm
lxc.cgroup.devices.allow = c 13:* rwm
lxc.mount.entry = /dev/dri/card0 dev/dri/card0 none
bind,optional,create=file
lxc.mount.entry = /dev/tty7 dev/tty7 none bind,optional,create=file
lxc.mount.entry = /dev/fb0 dev/fb0 none bind,optional,create=file
lxc.mount.entry = /dev/input dev/input none bind,optional,create=dir
lxc.mount.entry = /dev/snd dev/snd none bind,optional,create=dir
lxc.mount.entry = /share/home/jacques.nougat home/jacques.nougat none
bind,rw 0 0
lxc.mount.entry = /share/data data none bind,rw,create=dir 0  0
```

Configuration du proxy

Pour connaître l'adresse IP de la VM utiliser la commande suivante:
`lxc-info -n xfce -iH`
`192.168.121.187`

Ajout suppression de règles iptables nat

Syntaxe	`iptables-nat action s_port d_addr:d_port`
Arguments	* **action**: action to perform (add\del\info) * **s_port**: source port on host * **d_addr:d_port**: destination ip address and port
Exemples	`iptables-nat add 2222 192.168.121.150:22` `iptables-nat del 2222 192.168.121.150:22`

```
iptables-nat add 5900 192.168.121.87:5900
iptables-nat add 5901 192.168.121.87:5901
iptables-nat add 3389 192.168.121.87:3389
iptables-nat add 80 192.168.121.87:80
iptables-nat add 443 192.168.121.87:443
iptables-nat add 2222 192.168.121.87:22
```

```
iptables -S

# Network configuration
-P INPUT DROP
-P FORWARD DROP
```

```
-P OUTPUT ACCEPT
-N f2b-sshd
-A INPUT -p tcp -m multiport --dports 22 -j f2b-sshd
-A INPUT -m state --state INVALID -j DROP
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s 10.13.251.0/24 -i br0 -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -s 192.168.121.0/24 -i natbr0 -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -i natbr0 -p udp -m udp --sport 68 --dport 67 -j ACCEPT
-A FORWARD -d 192.168.121.87/32 -p tcp -m state --state NEW -m tcp --dport
22 -j ACCEPT
-A FORWARD -d 192.168.121.87/32 -p tcp -m state --state NEW -m tcp --dport
443 -j ACCEPT
-A FORWARD -d 192.168.121.87/32 -p tcp -m state --state NEW -m tcp --dport
80 -j ACCEPT
-A FORWARD -d 192.168.121.87/32 -p tcp -m state --state NEW -m tcp --dport
3389 -j ACCEPT
-A FORWARD -d 192.168.121.87/32 -p tcp -m state --state NEW -m tcp --dport
5901 -j ACCEPT
-A FORWARD -d 192.168.121.87/32 -p tcp -m state --state NEW -m tcp --dport
5900 -j ACCEPT
-A FORWARD -d 192.168.121.0/24 -p icmp -m icmp --icmp-type 8 -j DROP
-A FORWARD -d 192.168.121.0/24 -p tcp -m conntrack --ctstate NEW -j DROP
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 10.13.251.0/24 -i br0 -m conntrack --ctstate NEW -j ACCEPT
-A FORWARD -s 192.168.121.0/24 -i natbr0 -m conntrack --ctstate NEW -j
ACCEPT
-A f2b-sshd -j RETURN
```

Définition des users

```
lxc-snapshot -n xfce -c /backup/xfce-config.txt -o /backup/xfce-snapshot.txt
apt install xfce4
export https_proxy="http://proxy.infra.dgfip:3128"
export httpt_proxy="http://proxy.infra.dgfip:3128"
apt install nginx
adduser jacques.nougat sudo
addgroup admin
adduser jacques.nougat admin
```

Configuration du clavier et de la souris

```
vi /usr/share/X11/xorg.conf.d/10-evdev.conf
#
# Catch-all evdev loader for udev-based systems
# We don't simply match on any device since that also adds accelerometers
# and other devices that we don't really want to use. The list below
# matches everything but joysticks.
```

```
Section "InputClass"
        Identifier "evdev pointer catchall"
        MatchIsPointer "on"
        MatchDevicePath "/dev/input/event*"
        Driver "evdev"
EndSection

Section "InputClass"
        Identifier "evdev keyboard catchall"
        MatchIsKeyboard "on"
        MatchDevicePath "/dev/input/event*"
        Driver "evdev"
EndSection

Section "InputClass"
        Identifier "evdev touchpad catchall"
        MatchIsTouchpad "on"
        MatchDevicePath "/dev/input/event*"
        Driver "evdev"
EndSection

Section "InputClass"
        Identifier "evdev tablet catchall"
        MatchIsTablet "on"
        MatchDevicePath "/dev/input/event*"
        Driver "evdev"
EndSection

Section "InputClass"
        Identifier "evdev touchscreen catchall"
        MatchIsTouchscreen "on"
        MatchDevicePath "/dev/input/event*"
        Driver "evdev"
EndSection
```

Configuration du serveur X

```
vi /etc/X11/Xwrapper.config
apt install thunar-archive-plugin -y
apt install xterm
```

Configuration USB

```
udevadm info --query all --path /sys/block/sdc/ --attribute-walk |grep model
    ATTRS{model}=="USB DISK 2.0    "
SUBSYSTEM="scsi", ATTRS{model}=="USB DISK 2.0    ", SYMLINK+="usb%n"
apt install usbutils
lsusb
```

```
Bus 004 Device 002: ID 8087:8000 Intel Corp.
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 002: ID 8087:8008 Intel Corp.
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 030: ID 046d:c077 Logitech, Inc. M105 Optical Mouse
Bus 001 Device 002: ID 05dc:b055 Lexar Media, Inc.
Bus 001 Device 089: ID 6557:4200
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 002: ID 8087:8000 Intel Corp.
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 002: ID 8087:8008 Intel Corp.
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 030: ID 046d:c077 Logitech, Inc. M105 Optical Mouse
Bus 001 Device 002: ID 05dc:b055 Lexar Media, Inc.
Bus 001 Device 031: ID 05dc:a81d Lexar Media, Inc. LJDTT16G [JumpDrive 16GB]
Bus 001 Device 089: ID 6557:4200
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
ls -al /dev/bus/usb/001/002
crw-rw-r-- 1 root root 189, 1 Feb 15 07:53 /dev/bus/usb/001/002
ls -al /dev/bus/usb/001/031
crw-rw-r-- 1 root root 189, 30 Feb 16 08:03 /dev/bus/usb/001/031
```

```
# Template used to create this container: /usr/share/lxc/templates/lxc-
download
# Parameters passed to the template: -d ubuntu -r bionic -a amd64
# Template script checksum (SHA-1): 740c51206e35463362b735e68b867876048a8baf
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)

# use nat bridge by default

# Distribution configuration
lxc.include = /usr/share/lxc/config/ubuntu.common.conf
lxc.arch = linux64

# Container specific configuration
lxc.include = /etc/lxc/natbridge.conf
lxc.rootfs = /backup/lxc/xfce/rootfs
lxc.rootfs.backend = dir
lxc.utsname = xfce
lxc.cgroup.devices.allow = c 116:* rwm
lxc.cgroup.devices.allow = c 189:* rwm
lxc.cgroup.devices.allow = c 226:0 rwm
lxc.cgroup.devices.allow = c 226:128 rwm
lxc.cgroup.devices.allow = c 4:7 rwm
```

```
lxc.cgroup.devices.allow = c 29:0 rwm
lxc.cgroup.devices.allow = c 13:* rwm
lxc.mount.entry = /dev/dri/card0 dev/dri/card0 none
bind,optional,create=file
lxc.mount.entry = /dev/tty7 dev/tty7 none bind,optional,create=file
lxc.mount.entry = /dev/fb0 dev/fb0 none bind,optional,create=file
lxc.mount.entry = /dev/input dev/input none bind,optional,create=dir
lxc.mount.entry = /dev/snd dev/snd none bind,optional,create=dir
lxc.mount.entry = /share/home/jacques.nougat home/jacques.nougat none
bind,rw 0 0
lxc.mount.entry = /share/data data none bind,rw,create=dir 0  0
lxc.mount.entry = /media media none bind,rw,create=dir 0  0
lxc.mount.entry = /dev/bus/usb/001 dev/bus/usb/001 none
bind,optional,create=dir
```

```
cat <<'EOF' >/usr/local/bin/usb-mount.sh
#!/bin/bash

ACTION=$1
DEVBASE=$2
DEVICE="/dev/${DEVBASE}"

# See if this drive is already mounted
MOUNT_POINT=$(/bin/mount | /bin/grep ${DEVICE} | /usr/bin/awk '{ print $3
}')

do_mount()
{
    if [[ -n ${MOUNT_POINT} ]]; then
        # Already mounted, exit
        exit 1
    fi
    # Get info for this drive: $ID_FS_LABEL, $ID_FS_UUID, and $ID_FS_TYPE
    eval $(/sbin/blkid -o udev ${DEVICE})

    # Figure out a mount point to use
    LABEL=${ID_FS_LABEL}
    if [[ -z "${LABEL}" ]]; then
        LABEL=${DEVBASE}
    elif /bin/grep -q " /media/${LABEL} " /etc/mtab; then
        # Already in use, make a unique one
        LABEL+="-${DEVBASE}"
    fi
    MOUNT_POINT="/media/${LABEL}"

    /bin/mkdir -p ${MOUNT_POINT}

    # Global mount options
    OPTS="rw,relatime"
```

```
    # File system type specific mount options
    if [[ ${ID_FS_TYPE} == "vfat" ]]; then
        OPTS+=",users,gid=100,umask=000,shortname=mixed,utf8=1,flush"
    fi

    if ! /bin/mount -o ${OPTS} ${DEVICE} ${MOUNT_POINT}; then
        # Error during mount process: cleanup mountpoint
        /bin/rmdir ${MOUNT_POINT}
        exit 1
    fi
    # Bonus track: send desktop notification to user
    sudo -u andrea DISPLAY=:0
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus notify-send "Device
${DEVICE} mounted at ${MOUNT_POINT}"
}

do_unmount()
{
    if [[ -n ${MOUNT_POINT} ]]; then
        /bin/umount -l ${DEVICE}
    fi

    # Delete all empty dirs in /media that aren't being used as mount
points.
    for f in /media/* ; do
        if [[ -n $(/usr/bin/find "$f" -maxdepth 0 -type d -empty) ]]; then
            if ! /bin/grep -q " $f " /etc/mtab; then
                /bin/rmdir "$f"
            fi
        fi
    done
}
case "${ACTION}" in
    add)
        do_mount
        ;;
    remove)
        do_unmount
        ;;
esac

EOF
```

```
cat <<'EOF' >/etc/systemd/system/usb-mount@.service
[Unit]
Description=Mount USB Drive on %i

[Service]
Type=oneshot
RemainAfterExit=true
```

```
ExecStart=/usr/local/bin/usb-mount.sh add %i
ExecStop=/usr/local/bin/usb-mount.sh remove %i

EOF
```

```
cat <<'EOF' >/etc/udev/rules.d/99-local.rules
KERNEL=="sd[a-z][0-9]", SUBSYSTEMS=="usb", ACTION=="add",
RUN+="/bin/systemctl start usb-mount@%k.service"
KERNEL=="sd[a-z][0-9]", SUBSYSTEMS=="usb", ACTION=="remove",
RUN+="/bin/systemctl stop usb-mount@%k.service"

EOF
```

```
chmod +x /usr/local/bin/usb-mount.sh
udevadm control --reload-rules
systemctl daemon-reload
```