

LABORATORIUM PROGRAMOWANIA W CHMURACH KOMPUTEROWYCH

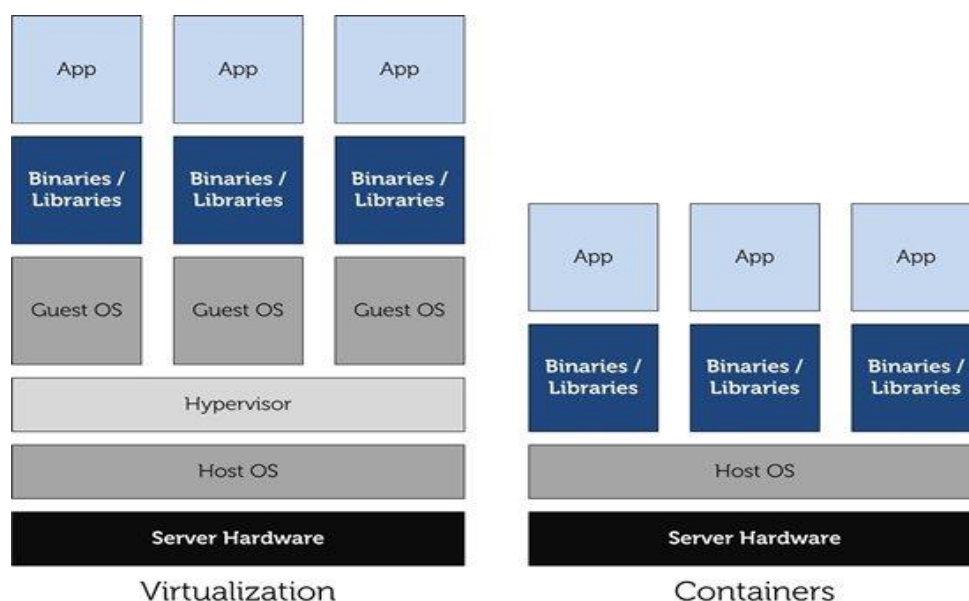
LABORATORIUM NR 1.

Ćwiczenie laboratoryjne podzielone jest na dwie części. Pierwsza ma na celu zapoznanie podstawami operacji na kontenerach LXC w systemie Ubuntu. Druga zawiera wprowadzenie do konfiguracji trybów sieciowych kontenerów LXC.

CZESC I

Wprowadzenie do lekkiej wirtualizacji. Kontenery LXC.

Kontenery LXC (ang. Linux containers) są lekką formą wirtualizacji. Od standardowych hipervisorów takich jak KVM, VirtualBox różnią się tym, że nie emulują sprzętu fizycznego i współdzielą ten sam system operacyjny. Różnicę tą ilustruje rysunek poniżej.



W chwili obecnej dostępne są dwie implementacje kontenerów LXC. Pierwszy bazuje na jądrze Linux-a a drugi opiera się o implementacje biblioteki libvirt. Pierwsze z tych rozwiązań jest zdecydowanie popularniejsze (szczególnie jeżeli pod uwagę muszą być brane kwestie bezpieczeństwa systemów wirtualnych) Poniższe ćwiczenia wykorzystują kontenery LXC oparte o jądro Linux. LXC będą wykorzystywane w kilku kolejnych ćwiczeniach laboratoryjnych. Informacje ogólne na temat kontenerów można odnaleźć pod adresem <https://linuxcontainers.org/>. Pomoc przy budowie systemów wirtualnych opartych o LXC w dystrybucji Ubuntu można odnaleźć w <https://help.ubuntu.com/lts/serverguide/lxc.html#lxc-network>

A. System kontenerów LXC nie jest domyślnie zainstalowany w systemie Ubuntu. Dlatego też przed rozpoczęciem pracy należy zainstalować niezbędne pakiety programowe poprzez wydanie polecenia:

```
sudo apt-get install lxc
```

Po instalacji, dobrym zwyczajem jest sprawdzenie poprawności konfiguracji zainstalowanych komponentów. W przypadku LXC służy do tego polecenie jak niżej.

```

student@student-VirtualBox: ~
student@student-VirtualBox:~$ lxc-checkconfig
Kernel configuration not found at /proc/config.gz; searching...
Kernel configuration found at /boot/config-4.4.0-31-generic
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled

--- Control groups ---
Cgroup: enabled
Cgroup clone_children flag: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: enabled
Cgroup cpuset: enabled

--- Misc ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
Bridges: enabled
Advanced netfilter: enabled
CONFIG_NF_NAT_IPV4: enabled
CONFIG_NF_NAT_IPV6: enabled
CONFIG_IP_NF_TARGET_MASQUERADE: enabled
CONFIG_IP6_NF_TARGET_MASQUERADE: enabled
CONFIG_NETFILTER_XT_TARGET_CHECKSUM: enabled
FUSE (for use with lxcfs): enabled

--- Checkpoint/Restore ---
checkpoint restore: enabled
CONFIG_FHANDLE: enabled
CONFIG_EVENTFD: enabled
CONFIG_EPOLL: enabled
CONFIG_UNIX_DIAG: enabled
CONFIG_INET_DIAG: enabled
CONFIG_PACKET_DIAG: enabled
CONFIG_NETLINK_DIAG: enabled
File capabilities: enabled

Note : Before booting a new kernel, you can check its configuration
usage : CONFIG=/path/to/config /usr/bin/lxc-checkconfig

```

Instalacja LXC wprowadza też zmiany w interfejsach sieciowych w systemie macierzystym. Dostępny jest nowy most o nazwie lxcbr0. Sposób jego użycia zostanie omówiony w dalszej części laboratorium. Wraz z tym interfejsem, instalacja pakietu LXC zmodyfikowała ustawienia zapory sieciowej iptables. Zmiana polega na dodaniu reguły translacji adresów PAT (ang. masquerade). Elementy te przedstawia rysunek poniżej.

```

student@cloudlab: ~
student@cloudlab:~$ ifconfig lxcbr0
lxcbr0      Link encap:Ethernet  HWaddr 9a:58:66:0c:71:7d
            inet addr:10.0.3.1  Bcast:10.0.3.255  Mask:255.255.255.0
            inet6 addr: fe80::9858:66ff:fe0c:717d/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:2293 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 B)  TX bytes:829791 (829.7 KB)

student@cloudlab:~$ sudo iptables -t nat -nL
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE all  --  10.0.3.0/24            !10.0.3.0/24
student@cloudlab:~$

```

Uwaga: Na chwilę obecną, pomimo, że instalator LXC powinien zainstalować pakiet iptables (jeśli stwierdzi jego brak) to dobrze jest iptables zainstalować wcześniej a w przypadku jego używania, sprawdzić czy wszystkie reguły są poprawne po instalacji. Użytkownicy innych zapór sieciowych muszą samodzielnie zmodyfikować ich ustawienia.

B. Poprawnie zainstalowany system LXC pozwala na uruchomienie pierwszego kontenera. Tworzenie i zarządzanie kontenerami możliwe jest z poziomu konsoli (narzędzia trybu tekstowego). Podstawowa składnia polecenia dla kontenerów LXC to:

```
lxc-DZIAŁANIE -t SZABLON - NAZWA
```

gdzie:

- DZIAŁANIE – jest częścią nazwy związaną z określoną funkcjonalnością.
- SZABLON – jest związany z dystrybucją Linux-a, którą ma być podstawą systemu kontenera
- NAZWA – nazwa identyfikująca kontener w systemie macierzystym.

Przykładowo:

- instalacja kontenera z systemem Ubuntu:

```
sudo lxc-create -t ubuntu -n mycloudlab
```

- instalacja kontenera z minimalną wersją używanej przez system macierzysty dystrybucji Linux-a:

```
sudo lxc-create -n minimal_linux
```

- uruchomienie zainstalowanego kontenera:

```
sudo lxc-start -n mycloudlab
```

- uruchomienie zainstalowanego kontenera w trybie daemona

```
sudo lxc-start -n mycloudlab -d
```

- zalogowanie się do kontenera (otworzenie konsoli z kontenerem):

```
sudo lxc-console -n mycloudlab
```

- zatrzymanie kontenera:

```
sudo lxc-stop -n mycloudlab
```

- usunięcie danych kontenera i samego kontenera:

```
sudo lxc-destroy -n mycloudlab
```

Krok 1:

Należy zainstalować kontener na bazie szablonu Ubuntu i nazwać go mycloudlab. W tym celu należy wydać polecenie:

```
sudo lxc-create -t ubuntu -n mycloudlab
```

Na koniec instalacji pojawi się informacja o danych logowania do systemu operacyjnego kontenera.

```
student@cloudlab: ~  
##  
# The default user is 'ubuntu' with password 'ubuntu'!  
# Use the 'sudo' command to run tasks as root in the container.  
##  
student@cloudlab:~$
```

Uwaga: Podczas tworzenia kontenera w oparciu o dany szablon, pobierane są pakiety danej dystrybucji co, zależnie od jakości posiadanego łącza sieciowego, trwa stosunkowo długi czas (typowo kilka minut). Każda kolejna instalacja kontenera wykorzystująca ten szablon jest zdecydowanie krótsza, ze względu na fakt, że pakiety zostały już pobrane a dane konfiguracyjne dystrybucji są zapisane w cache.

Krok 2:

Zainstalowany kontener mycloudlab można teraz uruchomić za pomocą polecenia:

```
sudo lxc-start -n mycloudlab
```

Rozpocznie się proces ładowania składników systemów i usług jak w przypadku standardowego systemu operacyjnego. Przy standardowych ustawieniach jest on niewidoczny dla użytkownika ponieważ kontener jest uruchamiany w trybie deamona (szczegółowe informacje o opcjach polecenia lxc-start można znaleźć w podręczniku systemowym). Na szczęście, uruchomiony kontener może być monitorowany i zarządzany z poziomu maszyny gospodarza. Podstawowe informacje o konfiguracji stworzonego kontenera można uzyskać poleceniami: `lxc-info` oraz `lxc-ls` z odpowiednikami opcjami formatującymi treść odpowiedzi.

```
student@student-VirtualBox:~$ sudo lxc-ls --fancy  
NAME      STATE   AUTOSTART GROUPS IPV4      IPV6  
mycloudlab RUNNING 0        -        10.0.3.199 -  
student@student-VirtualBox:~$ sudo lxc-info --name mycloudlab  
Name:      mycloudlab  
State:     RUNNING  
PID:       23052  
IP:        10.0.3.199  
CPU use:   0.41 seconds  
BlkIO use: 11.68 MiB  
Memory use: 25.13 MiB  
KMem use:  0 bytes  
Link:      vethB8BNLL  
TX bytes:  1.76 KiB  
RX bytes:  4.99 KiB  
Total bytes: 6.75 KiB
```

Uwaga: Kontener LXC stworzony i uruchomiony w poprzednich krokach, jest tzw. kontenerem w trybie uprzywilejowanym (utworzonym z poziomu uprawnień root-a). W SYSTEMIE Ubuntu a także niektórych innych dystrybucjach jest możliwe utworzenie kontenera w trybie użytkownika (ang. unprivileged container). Szczegółowe informacje na ten temat można odnaleźć w dokumentacji LXC dla systemu Ubuntu, np. pod adresem: <https://help.ubuntu.com/lts/serverguide/lxc.html>

Krok 3:

Aby zalogować się do uruchomionego kontenera LXC należy użyć polecenia

```
sudo lxc-console -n mycloudlab
```

Najpierw otrzymamy informacje o sposobie wyjścia z konsoli kontenera (trzeba o niej pamiętać) oraz zostaniemy poproszeni o wprowadzenie nazwy i hasła użytkownika. Dane te były wyświetlone na końcu tworzenia kontenera (patrz: rysunek na końcu kroku 1) i w naszym przypadku są to: user: ubuntu, passwd:

ubuntu. Cały proces przedstawia rysunek poniżej.

```
student@student-VirtualBox:~$ sudo lxc-console -n mycloudlab
Connected to tty 1
Type <Ctrl+a q> to exit the console, <Ctrl+a Ctrl+a> to enter Ctrl+a itself
Ubuntu 16.04.1 LTS mycloudlab pts/0
mycloudlab login: ubuntu
Password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Po zalogowaniu się, standardowo jesteśmy w katalogu użytkownika. Do jego dyspozycji jest system operacyjny Ubuntu o standardowym drzewie katalogów. Ilustruje to poniższy rysunek.

```
ubuntu@mycloudlab:~$ ls -al
total 24
drwxr-xr-x 3 ubuntu ubuntu 4096 Oct  9 22:35 .
drwxr-xr-x 3 root   root   4096 Oct  9 22:16 ..
-rw-r--r-- 1 ubuntu ubuntu 220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Aug 31  2015 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Oct  9 22:35 .cache
-rw-r--r-- 1 ubuntu ubuntu 655 Jun 24 15:44 .profile
ubuntu@mycloudlab:~$ ls /
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
ubuntu@mycloudlab:~$
```

Wyjście z konsoli i powrót do systemu gospodarza uzyskuje się po naciśnięciu Ctrl+a a potem klawisza q.

Po opuszczeniu konsoli należy zatrzymać kontener za pomocą polecenia lxc-stop. Proszę sprawdzić składnię tego polecenia w podręczniku systemowym oraz zwrócić uwagę na inne dostępne opcje (reboot, kill itd.).

1P1: Rysunek poniżej przedstawia dostępne polecenia związane z kontenerami LXC. Korzystając z podręczników systemowych (man) oraz dokumentacji LXC proszę podać do czego służą polecenia:

- lxc-autostart,
- lxc-freeze,
- lxc-unfreeze,
- lxc-wait.

```
student@student-VirtualBox: ~
student@student-VirtualBox:~$ lxc-
lxc-attach          lxc-console          lxc-info             lxc-top
lxc-autostart       lxc-copy             lxc-ls               lxc-unfreeze
lxc-cgroup          lxc-create           lxc-monitor          lxc-unshare
lxc-checkconfig     lxc-destroy          lxc-snapshot         lxc-usernsexec
lxc-checkpoint      lxc-device            lxc-start            lxc-wait
lxc-clone           lxc-execute          lxc-start-ephemeral
lxc-config          lxc-freeze           lxc-stop
```


1P2: Jednym z bardzo użytecznych poleceń w środowisku LXC są polecenia: *lxc-attach*, *lxc-execute*, *lxc-monitor*. Wykorzystując stworzony kontener mycloudlab, proszę zaproponować przykładowe użycie tego polecenia w codziennej pracy z kontenerem. Odpowiedź proszę krótko opisać i zilustrować odpowiednimi zrzutami ekranowymi.

CZESC II

Kontenery LXC mogą być skonfigurowane do pracy w różnych trybach sieciowych. Dokumentacja LXC, na chwilę obecną, wymienia następujące tryby:

- empty
- veth
- macvlan
- vlan
- phys

Tryb empty jest trybem najmniej ciekawym z punktu widzenia tworzenia systemów zwirtualizowanych. W tym trybie, kontener dysponuje tylko interfejsem loopback. Wobec tego nie ma możliwości połączenia z siecią zewnętrzną i tym samym wykorzystanie tego trybu ogranicza się do testowania systemów i aplikacji izolowanych do jednego systemu operacyjnego.

Z kolei tryb vlan jest stosowany w infrastrukturach sieciowych korzystających z techniki tagowanych sieci VLAN (opartych o standard IEEE802.1q) Tryb ten zdecydowanie wykracza poza typowe zastosowania kontenerów i nie będzie omawiany w trakcie tego laboratorium.

Uwaga: W przypadku niektórych systemów operacyjnych, np. od wersji Ubuntu Trusty możliwe jest skonfigurowanie jeszcze jednego trybu sieciowego, tzw. trybu none.

A

Domyślnym trybem sieciowym dla kontenerów LXC jest tryb veth. W trybie tym pracował pierwszy uruchomiony kontener o nazwie mycloudlab. Proszę raz jeszcze go uruchomić (jeśli do tej pory nie był uruchomiony). W systemie gospodarza, poza wspomnianym (pkt. A części I) mostem lxcbr0, pojawił się interfejs wirtualny skojarzony z interfejsem systemu operacyjnego kontenera. Sprawdzając dostępne interfejsy w systemie macierzystym powinny być aktywne m.in. dwa interfejsy, analogicznie jak na rysunku poniżej (nazwa interfejsu wirtualnego może się różnić pomiędzy systemami).

```
student@cloudlab: ~  
lxcbr0    Link encap:Ethernet  HWaddr fe:1c:fa:fe:56:72  
          inet addr:10.0.3.1  Bcast:10.0.3.255  Mask:255.255.255.0  
          inet6 addr: fe80::9858:66ff:fe0c:717d/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:54 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:2760 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:4680 (4.6 KB)  TX bytes:993005 (993.0 KB)  
  
vethXDIYA4 Link encap:Ethernet  HWaddr fe:1c:fa:fe:56:72  
          inet6 addr: fe80::fc1c:faff:fefe:5672/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:22 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:2284 (2.2 KB)  TX bytes:6083 (6.0 KB)
```

Ponadto można w prosty sposób sprawdzić, że interfejs wirtualny jest przyłączony do mostu lxcbr0. W tym celu można użyć polecenia:

```
student@cloudlab: ~  
student@cloudlab:~$ brctl show lxcbr0  
bridge name      bridge id        STP enabled      interfaces  
lxcbr0           8000.fe1cfafe5672 no                vethXDIYA4  
student@cloudlab:~$
```

Dodatkowo ustawienia sieciowe w tym trybie (jak i w każdym innym) można wyświetlić posługując się poznanym już poleceniem `lxc-ls --fancy`.

Pliki, dane i konfiguracje kontenera przechowywane są domyślnie w systemie plików systemu gospodarza, w katalogu `/var/lib/lxc/NAZWA` gdzie NAZWA oznacza nazwę danego kontenera. Aby sprawdzić ich istnienie, należy na chwile przejąć uprawnienia root-a. Przykładowo, dla kontenera mycloudlab:

```
root@student-VirtualBox: /var/lib/lxc  
student@student-VirtualBox:~$ sudo su -  
root@student-VirtualBox:~# cd /var/lib/lxc  
root@student-VirtualBox:/var/lib/lxc# ls -a  
..  mycloudlab  
root@student-VirtualBox:/var/lib/lxc# ls -al mycloudlab/  
total 16  
drwxrwx--- 3 root root 4096 paź 10 00:16 .  
drwx----- 3 root root 4096 paź 10 00:09 ..  
-rw-r--r-- 1 root root 706 paź 10 00:16 config  
drwxr-xr-x 21 root root 4096 paź 10 01:27 rootfs  
root@student-VirtualBox:/var/lib/lxc#
```

Z punktu widzenia ustawień trybu sieciowego, istotne są zapisy w pliku konfiguracji kontenera o nazwie `config`. Po wyjściu z ustawień root-a (polecenie `exit`) można sprawdzić aktualne ustawienia. Ponownie, dla kontenera mycloudlab, zapisy odnoszące się do ustawień sieciowych są następujące:

```
student@cloudlab: /  
student@cloudlab:/$ sudo grep network /var/lib/lxc/mycloudlab/config  
lxc.network.type = veth  
lxc.network.flags = up  
lxc.network.link = lxcbr0  
lxc.network.hwaddr = 00:16:3e:13:68:c9  
student@cloudlab:/$
```

Potwierdza się, że kontener mycloudlab został domyślnie skonfigurowany na w trybie veth.

1P3: Proszę utworzyć kolejny kontener na bazie szablonu ubuntu i nazwać go *nettest*. Po jego uruchomieniu należy sprawdzić czy stan i ustawienia obu działających kontenerów są takie jak na rysunku poniżej. Proszę sprawdzić stan połączeń na moście lxcbr0, czy możliwe jest „pingowanie” kontenerów z systemu macierzystego, czy możliwe jest „pingowanie” systemu gospodarza z kontenerów. Dodatkowo proszę przetestować czy kontenery „widzą” się nawzajem i czy mogą łączyć się z adresami zewnętrznymi (w Internecie). W sprawozdaniu proszę odpowiedzieć na powyższe pytania a odpowiedzi zilustrować odpowiednimi wynikami działania użytych poleceń.

Uwaga: W trybie veth most lxcbr0 dla systemu kontenera pełni rolę bramy sieciowej oraz serwera DNS. Adresy i parametry sieciowe dla poszczególnych kontenerów są przyznawane poprzez zaimplementowany w systemie LXC serwer DHCP. Możliwe jest sztywne ustawienie tych adresów, Szczegóły podane są np. pod adresem: <https://help.ubuntu.com/lts/serverguide/lxc.html> (podrozdział „Networking”)

B. Kolejnym trybem sieciowym, przydatnym w codziennej pracy z kontenerami jest tryb macvlan. Tryb ten posiada zdefiniowane w dokumentacji trzy podwersje:

- private
- vepa
- bridge

Pierwszy z nich, tryb macvlan-private izoluje od siebie poszczególne kontenery (uniemożliwia komunikację pomiędzy nimi) wobec czego jest mało przydatny w środowiskach wirtualnych. Drugi z kolei, tryb macvlan-vepa wymaga konfiguracji zewnętrznego (względem domyślnego mostu linux-owego) wyposażonego w bardzo rzadko stosowaną funkcję reflective relay. Oba wymienione wyżej podwersje trybu macvlan mają niewielką przydatność z punktu widzenia głównego tematu laboratorium.

Tryb macvlan-bridge jest natomiast bardzo szeroko wykorzystywany w systemach zwirtualizowanych, opartych o kontenery LXC.

W celu demonstracji cech tego trybu sieciowego, utworzone i skonfigurowane zostaną dwa kontenery. Kontenery te przyłączone zostaną do ręcznie utworzonego mostu linux-owego.

Krok 1:

na początku należy utworzyć i uaktywnić w systemie gospodarza dodatkowy most np. o nazwie lxcbr1. W tym celu należy wydać następujące polecenia:

```
student@cloudlab: /
student@cloudlab:/$ sudo brctl addbr lxcbr1
student@cloudlab:/$ sudo ip link set lxcbr1 up
student@cloudlab:/$ sudo brctl show
bridge name      bridge id                STP enabled        interfaces
lxcbr0           8000.fe304ca8b373        no                 vethFKVFVA
                  vethIY2926
lxcbr1           8000.00000000000000      no
```

Krok 2:

Kolejnym krokiem jest utworzenie dwóch kontenerów o nazwach odpowiednio: 1macvlan oraz 2macvlan. Oba kontenery mają korzystać z szablonu ubuntu. Przed uruchomieniem nowych kontenerów należy skonfigurować je do pracy w trybie. W tym celu należy zmodyfikować zawartość pliku konfiguracyjnego każdego z kontenerów (tj. 1macvlan oraz 2macvlan) za pomocą wybranego edytora (np. nano) tak by była zgodna z treścią przedstawioną na rysunku poniżej (adresy MAC mogą pozostać różne od tych z rysunku).

```
student@cloudlab: /
student@cloudlab:/$ sudo grep network /var/lib/lxc/1macvlan/config
lxc.network.type = macvlan
lxc.network.macvlan.mode = bridge
lxc.network.flags = up
lxc.network.link = lxcbr1
lxc.network.ipv4 = 10.10.0.3/16
lxc.network.hwaddr = 00:16:3e:e1:de:7e
student@cloudlab:/$ sudo grep network /var/lib/lxc/2macvlan/config
lxc.network.type = macvlan
lxc.network.macvlan.mode = bridge
lxc.network.flags = up
lxc.network.link = lxcbr1
lxc.network.ipv4 = 10.10.0.4/16
lxc.network.hwaddr = 00:16:3e:7b:49:bc
student@cloudlab:/$
```


Krok 3:

po poprawnym skonfigurowaniu tryby sieciowego macvlan-bridge w poprzednim kroku, można przystąpić do uruchomienia kontenerów 1macvlan oraz 2macvlan, obu w trybie daemon. Wyświetlając stan kontenerów skonfigurowanych w systemie macierzystym powinno otrzymać się rezultat analogiczny do przedstawionego na rysunku poniżej.

```
student@student-VirtualBox:~$ sudo lxc-start -n 1macvlan
student@student-VirtualBox:~$ sudo lxc-start -n 2macvlan
student@student-VirtualBox:~$ sudo lxc-ls --fancy
```

NAME	STATE	AUTOSTART	GROUPS	IPV4	IPV6
1macvlan	RUNNING	0	-	10.10.0.3	-
2macvlan	RUNNING	0	-	10.10.0.4	-
mycloudlab	RUNNING	0	-	10.0.3.199	-
nettest	RUNNING	0	-	10.0.3.179	-

IP4: Proszę zalogować się na kontenerach macvlan1 oraz macvlan2. Następnie proszę sprawdzić, które interfejsy są podłączone do mostu lxcbr1. Wreszcie, należy dokonać testów łączności utworzonych kontenerów: pomiędzy sobą, z systemem macierzystym oraz siecią zewnętrzną (Internetem). W sprawozdaniu proszę odpowiedzieć na powyższe pytania a odpowiedzi zilustrować odpowiednimi wynikami działania użytych poleceń.

C. Ostatnim trybem, który będzie testowany w tym zadaniu jest tryb phys. Idea działania tego trybu jest stosunkowo prosta i polega na przeniesieniu sieciowego interfejsu fizycznego (karty sieciowej na maszynie gospodarza) do kontenera.

Uwaga: Skonfigurowanie kontenera w trybie sieciowym phys oznacza, że w systemie macierzystym nie będzie dostępny żaden interfejs sieciowy i tym samym nie będzie on mógł komunikować się z siecią zewnętrzną.

Konfiguracja tego trybu, podobnie jak poprzednich trybów sieciowych, polega na edycji pliku konfiguracyjnego danego kontenera LXC. Wobec tego należy na początku utworzyć nowy kontener na bazie szablonu ubuntu. Kontener ten należy nazwać *physnet*. Następnie należy poddać edycji plik konfiguracyjny tego kontenera tak by zawierał informacje zgodne z rysunkiem poniżej. !!!!! Nazwę interfejsu maszyny gospodarza (*lxc.network.link*) należy zmienić na nazwę zgodną z używanym systemem !!!!!

```
student@student-VirtualBox: ~
student@student-VirtualBox:~$ sudo grep network /var/lib/lxc/physnet/config
lxc.network.type = phys
lxc.network.link = enp0s3
lxc.network.flags = up
lxc.network.hwaddr = 00:16:3e:57:34:94
#lxc.network.ipv4 = 192.168.1.100/24
#lxc.network.ipv4.gateway = 192.168.1.1
student@student-VirtualBox:~$
```

Uwaga: Konfiguracja interfejsu sieciowego w kontenerze pracującym w trybie phys odbywa się na tych samych zasadach co uprzednia konfiguracja karty w systemie macierzystym. Jeżeli wykorzystywany był serwer DHCP to konieczne jest zwrócenie uwagi na adres MAC. W większości przypadków powinien on być zgodny z adresem MAC karty fizycznej (w takiej sytuacji należy przed konfiguracją odnaleźć i zapisać adres MAC interfejsu fizycznego). Jeżeli nie jest wykorzystywany DHCP to należy parametry sieciowe ustawić ręcznie (należy usunąć komentarz w dwóch ostatnich linia pliku konfiguracyjnego przedstawionego na rysunku powyżej. Adres kontenera i bramy sieciowej musu odpowiadać planowi adresowemu jaki jest zdefiniowany dla danej sieci LAN.

Po wykonaniu poprawnej konfiguracji i uruchomieniu kontenera physnet, polecenie `lxc-ls --fancy` powinno dać rezultat jak niżej (z wyjątkiem adresu IP dla kontenera physnet, który może być różny z powodów przytoczonych w ostatniej z uwag). Proszę również zauważyć, że system gospodarza zapewne poinformował o niedostępności sieci.

```
student@student-VirtualBox:~$ sudo lxc-start -n physnet
student@student-VirtualBox:~$ sudo lxc-ls --fancy
NAME      STATE   AUTOSTART GROUPS IPV4          IPV6
1macvlan  RUNNING 0        -      10.10.0.3     -
2macvlan  RUNNING 0        -      10.10.0.4     -
mycloudlab RUNNING 0        -      10.0.3.199    -
nettest   RUNNING 0        -      10.0.3.179    -
physnet    RUNNING 0        -      192.168.1.100 -
student@student-VirtualBox:~$
```

IP5: Proszę zalogować się na kontenerze physnet. Na systemie macierzystym proszę sprawdzić czy dostępny jest jakikolwiek interfejs fizyczny. Wreszcie, należy dokonać testów łączności utworzonego kontenera: z innymi kontenerami oraz z siecią zewnętrzną (Internetem). W sprawozdaniu proszę odpowiedzieć na powyższe pytania a odpowiedzi zilustrować odpowiednimi wynikami działania użytych poleceń.

Po zakończeniu testów proszę zatrzymać kontener physnet tak by w systemie macierzystym ponownie dostępny był interfejs wirtualny.

D: Kontenery LXC mogą posiadać więcej niż jeden interfejs sieciowy. Każdy z nich może pracować w innym trybie sieciowym.

IP6: Proszę utworzyć kontener z systemem ubuntu o nazwie mytest. Następnie należy skonfigurować go zgodnie z przedstawionym niżej plikiem konfiguracyjnym i uruchomić.

```
student@cloudlab: ~
student@cloudlab:~$ sudo grep network /var/lib/lxc/mytest/config
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = lxcbr0
lxc.network.hwaddr = 00:16:3e:bd:2f:4a
lxc.network.type = macvlan
lxc.network.macvlan.mode = bridge
lxc.network.flags = up
lxc.network.link = lxcbr1
lxc.network.ipv4 = 10.10.0.10/16
student@cloudlab:~$
```

Poprawnie wykonana konfiguracja powinna doprowadzić do sytuacji, w której na systemie macierzystym dostępne będą następujące kontenery:

```
student@student-VirtualBox: ~
student@student-VirtualBox:~$ sudo lxc-start -n mytest
student@student-VirtualBox:~$ sudo lxc-ls --fancy
NAME      STATE   AUTOSTART GROUPS IPV4          IPV6
1macvlan  RUNNING 0        -      10.10.0.3     -
2macvlan  RUNNING 0        -      10.10.0.4     -
mycloudlab RUNNING 0        -      10.0.3.199    -
mytest     RUNNING 0        -      10.0.3.170, 10.10.0.10 -
nettest   RUNNING 0        -      10.0.3.179    -
physnet    STOPPED 0        -      -             -
student@student-VirtualBox:~$
```

W sprawozdaniu proszę narysować schemat sieci prezentujący połączenia kontenera mytest z pozostałymi kontenerami i z interfejsem sieciowym systemu macierzystego. Proszę też podać z którymi kontenerami może on się komunikować i przez który swój interfejs.

Na zakończenie ćwiczenia **PROSZĘ KONIECZNIE** zatrzymać wszystkie utworzone kontenery LXC. Następnie proszę usunąć je z systemu macierzystego za pomocą polecenia:

```
sudo lxc-destroy -n NAZWA
```

gdzie w miejsce NAZWA proszę wstawić nazwę kolejnego usuwanego kontenera.

Plik ze sprawozdanie prosze wgrać do systemu moodle. Preferowany format to pdf. Plik proszę nazwać zgodnie ze schematem:

CW1_dzień_godz_Nazwisko.pdf

gdzie:

zamiast **dzień** prosze wstawić dzień Państwa zajęć, np. *wt*, *czw* lub *pt*

zamiast **godz** prosze wstawić godzinę rozpoczęcia Państwa zajęć, np.. *8*, *14* , *16* lub *18*

zamiast **Nazwisko** prosze wstawić swoje nazwisko
