

Laboratorium 3 Chmury obliczeniowe

Mateusz Markowski gr.7.4/7

Zadanie 1

0) Proszę utworzyć i uruchomić dwa kontenery LXC na bazie dystrybucji Ubuntu, odpowiednio o nazwach *test1* oraz *test2*. Proszę sprawdzić i opisać domyślną konfigurację sieciową kontenerów LXC w środowisku LXD. (analogiczna analiza była przeprowadzana dla pojedynczych kontenerów LXC, w trakcie laboratorium nr 1). Wykonanie zadania powinno obejmować dokumentację (użyte polecenia, wyniki ich działań, komentarz czego dowodzi dany test) następujących zadań cząstkowych (zagadnień):

1. ustawienia firewall-a na maszynie gospodarza w odniesieniu do puli adresowej, przypisanej kontenerom w konfiguracji środowiska LXD.
2. czy interfejsy kontenerów są przyłączone do mostu *lxdbr0* ?
3. czy można „pingować” (należy włączyć opcje rejestracji trasy) kontener *test2* z kontenera *test1* i na odwrót ?
4. czy można „pingować” kontenery z maszyny gospodarza (ubuntu na VirtualBox) i z OpenSuse ?
5. czy można „pingować” z danego kontenera na system gospodarza (Ubuntu) i do OpenSuse.
6. kontenery *test1* oraz *test2* otrzymały adresy z puli DHCP zdefiniowanej podczas konfiguracji środowiska LXD. Czy można wyłączyć (zrezygnować z DHCP) i skonfigurować adresy kontenerów ręcznie (oczywiście wykorzystując adresy z tej samej podsieci co most *lxdbr0*) ? Jeśli tak to proszę zilustrować procedurę postępowania.

0)

```
student@student-VirtualBox:~$ sudo lxc launch ubuntu:16.04 test1
Creating test1
Starting test1
student@student-VirtualBox:~$ sudo lxc launch ubuntu:16.04 test2
Creating test2
Starting test2
student@student-VirtualBox:~$
```

```
student@student-VirtualBox:~$ sudo lxc list
```

NAME	STATE	IPV4	IPV6
	TYPE	SNAPSHOTS	
test1	RUNNING	10.243.137.109 (eth0)	fd8a:dcb1:9c8d:8189:216:3eff:feeb:b623 (eth0)
	PERSISTENT	0	
test2	RUNNING	10.243.137.74 (eth0)	fd8a:dcb1:9c8d:8189:216:3eff:fe01:cb07 (eth0)
	PERSISTENT	0	

Powyższe dwa zrzuty ekranowe prezentują utworzenie oraz uruchomienie kontenerów w środowisku lxd. W celu poprawnego sprawdzenia ustawień kontenerów należy poleceniem `cd` przejść do folderu `/etc/default` oraz polecenie `nano lxd-bridge` wyświetlić dane o adresie ip dla kontenerów w środowisku lxd. Widać że kontenery są w takiej samej podsieci dlatego utworzenie kontenerów powiodło się

```
# IPv4
## IPv4 address (e.g. 10.0.8.1)
LXD_IPV4_ADDR="10.243.137.1"

## IPv4 netmask (e.g. 255.255.255.0)
LXD_IPV4_NETMASK="255.255.255.0"

## IPv4 network (e.g. 10.0.8.0/24)
LXD_IPV4_NETWORK="10.243.137.1/24"

## IPv4 DHCP range (e.g. 10.0.8.2,10.0.8.254)
LXD_IPV4_DHCP_RANGE="10.243.137.2,10.243.137.254"

## IPv4 DHCP number of hosts (e.g. 250)
LXD_IPV4_DHCP_MAX="252"

## NAT IPv4 traffic
LXD_IPV4_NAT="true"
```

1)

```

student@student-VirtualBox:~$ sudo iptables -L -n
[sudo] password for student:
Chain INPUT (policy ACCEPT)
target     prot opt source                destination            tcp dpt:53 /* mana
ACCEPT     tcp  --  0.0.0.0/0              0.0.0.0/0              tcp dpt:53 /* mana
ged by lxd-bridge */
ACCEPT     udp  --  0.0.0.0/0              0.0.0.0/0              udp dpt:53 /* mana
ged by lxd-bridge */
ACCEPT     udp  --  0.0.0.0/0              0.0.0.0/0              udp dpt:67 /* mana
ged by lxd-bridge */
ACCEPT     tcp  --  0.0.0.0/0              0.0.0.0/0              tcp dpt:53
ACCEPT     udp  --  0.0.0.0/0              0.0.0.0/0              udp dpt:53
ACCEPT     tcp  --  0.0.0.0/0              0.0.0.0/0              tcp dpt:67
ACCEPT     udp  --  0.0.0.0/0              0.0.0.0/0              udp dpt:67

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination            /* managed by lxd-
bridge */
ACCEPT     all  --  0.0.0.0/0              0.0.0.0/0              /* managed by lxd-
bridge */
ACCEPT     all  --  0.0.0.0/0              0.0.0.0/0              /* managed by lxd-
bridge */
ACCEPT     all  --  0.0.0.0/0              0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0              0.0.0.0/0

student@student-VirtualBox:~$ sudo ufw status verbose
Status: inactive
student@student-VirtualBox:~$

```

Poleceniem `sudo ufw status verbose` wyświetlamy aktualny status firewall'a jak widać jest on nieaktywnym. Poleceniem `iptables -L -n` sprawdzamy czy firewall dodał jakieś adresy do blokowania. Jak widać firewall pozwala na komunikację między konterami a maszyną gospodarza.

2) Tak interfejsy kontenerów zostały podłączone do mostu `lxdbr0`, co pokazuje poniższy zrzut ekranowy. W celu sprawdzenia czy kontenery zostały podłączone do mostu `lxdbr0` wydać należy polecenie `brctl show lxdbr0`

```

student@student-VirtualBox:~$ brctl show lxdbr0
bridge name      bridge id        STP enabled      interfaces
lxdbr0           8000.fe4f72672026 no                veth7HX1M3
                  vethX00DYA
student@student-VirtualBox:~$

```

3)

```

student@student-VirtualBox:~$ sudo lxc exec test2 -- ping -R 10.243.137.109
PING 10.243.137.109 (10.243.137.109) 56(124) bytes of data.
64 bytes from 10.243.137.109: icmp_seq=1 ttl=64 time=0.049 ms
RR: 10.243.137.74
    10.243.137.109
    10.243.137.109
    10.243.137.74
64 bytes from 10.243.137.109: icmp_seq=2 ttl=64 time=0.097 ms (same route)
64 bytes from 10.243.137.109: icmp_seq=3 ttl=64 time=0.098 ms (same route)
64 bytes from 10.243.137.109: icmp_seq=4 ttl=64 time=0.077 ms (same route)
64 bytes from 10.243.137.109: icmp_seq=5 ttl=64 time=0.038 ms (same route)
^C
--- 10.243.137.109 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4074ms
rtt min/avg/max/mdev = 0.038/0.071/0.098/0.026 ms
student@student-VirtualBox:~$

```



```

student@student-VirtualBox:~$ sudo lxc exec test1 -- ping -R 10.243.137.74
PING 10.243.137.74 (10.243.137.74) 56(124) bytes of data.
64 bytes from 10.243.137.74: icmp_seq=1 ttl=64 time=0.028 ms
RR:      10.243.137.109
        10.243.137.74
        10.243.137.74
        10.243.137.109

64 bytes from 10.243.137.74: icmp_seq=2 ttl=64 time=0.100 ms      (same route)
64 bytes from 10.243.137.74: icmp_seq=3 ttl=64 time=0.101 ms      (same route)
64 bytes from 10.243.137.74: icmp_seq=4 ttl=64 time=0.111 ms      (same route)
64 bytes from 10.243.137.74: icmp_seq=5 ttl=64 time=0.060 ms      (same route)
64 bytes from 10.243.137.74: icmp_seq=6 ttl=64 time=0.039 ms      (same route)
^C
--- 10.243.137.74 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5110ms
rtt min/avg/max/mdev = 0.028/0.073/0.111/0.032 ms

```

Jak prezentują powyższe zrzuty ekranowe kontenery się wzajemnie pingują.

4)

```

C:\Users\Mateusz>ping 10.243.137.109

Pinging 10.243.137.109 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.243.137.109:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Mateusz>ping 10.243.137.74

Pinging 10.243.137.74 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.243.137.74:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

Pingowanie z systemu Windows do kontenerów test1 oraz test2 nie powiodło się co prezentuje powyższy zrzut ekranowy.

```

student@student-VirtualBox:~$ ping 10.243.137.109
PING 10.243.137.109 (10.243.137.109) 56(84) bytes of data.
64 bytes from 10.243.137.109: icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from 10.243.137.109: icmp_seq=2 ttl=64 time=0.059 ms
64 bytes from 10.243.137.109: icmp_seq=3 ttl=64 time=0.052 ms
64 bytes from 10.243.137.109: icmp_seq=4 ttl=64 time=0.075 ms
^C
--- 10.243.137.109 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3063ms
rtt min/avg/max/mdev = 0.052/0.059/0.075/0.012 ms
student@student-VirtualBox:~$

```

```

student@student-VirtualBox:~$ ping 10.243.137.74
PING 10.243.137.74 (10.243.137.74) 56(84) bytes of data.
64 bytes from 10.243.137.74: icmp_seq=1 ttl=64 time=0.039 ms
64 bytes from 10.243.137.74: icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from 10.243.137.74: icmp_seq=3 ttl=64 time=0.062 ms
64 bytes from 10.243.137.74: icmp_seq=4 ttl=64 time=0.039 ms
^C
--- 10.243.137.74 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.039/0.048/0.062/0.012 ms

```

Pingowanie z maszyny wirtualnej do kontenerów test1 oraz test2 zakończyło się sukcesem co prezentują dwa powyższe zrzuty ekranowe.

5)

```

student@student-VirtualBox:~$ sudo lxc exec test1 -- ping -R 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(124) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.038 ms
RR:      10.243.137.109
         10.0.2.15
         10.0.2.15
         10.243.137.109
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.075 ms      (same route)
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.101 ms      (same route)
64 bytes from 10.0.2.15: icmp_seq=4 ttl=64 time=0.044 ms      (same route)
^C
--- 10.0.2.15 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3049ms
rtt min/avg/max/mdev = 0.038/0.064/0.101/0.026 ms
student@student-VirtualBox:~$ sudo lxc exec test2 -- ping -R 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(124) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.039 ms
RR:      10.243.137.74
         10.0.2.15
         10.0.2.15
         10.243.137.74
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.114 ms      (
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.078 ms      (
^C
--- 10.0.2.15 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
rtt min/avg/max/mdev = 0.039/0.077/0.114/0.030 ms
student@student-VirtualBox:~$ sudo lxc exec test2 -- ping -R 192.168.8.104
PING 192.168.8.104 (192.168.8.104) 56(124) bytes of data.
64 bytes from 192.168.8.104: icmp_seq=1 ttl=126 time=1.03 ms
64 bytes from 192.168.8.104: icmp_seq=2 ttl=126 time=1.90 ms
^C
--- 192.168.8.104 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.030/1.467/1.904/0.437 ms

```



```
student@student-VirtualBox:~$ sudo lxc exec test1 -- ping -R 192.168.8.104
PING 192.168.8.104 (192.168.8.104) 56(124) bytes of data.
64 bytes from 192.168.8.104: icmp_seq=1 ttl=126 time=0.776 ms
64 bytes from 192.168.8.104: icmp_seq=2 ttl=126 time=0.854 ms
^C
--- 192.168.8.104 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1026ms
rtt min/avg/max/mdev = 0.776/0.815/0.854/0.039 ms
```

Pingowanie z kontenerów test1 oraz test2 do maszyny wirtualnej oraz do systemu macierzystego Windows zakończyło się powodzeniem co prezentują powyższe zrzuty ekranowe.

6) Jest możliwe statyczne przypisanie adresu IP dla kontenerów wystarczy utworzyć plik dns.conf w folderze /etc/default poleceniem `sudo nano /etc/default/dns.conf`. W tym pliku dodać dwie linijki, które zaprezentowano na poniższym zrzucie ekranowym.

```
dhcp-host=test1,10.243.137.105
dhcp-host=test2,10.243.137.104
```

Dla kontenera test1 oraz test2 przypisujemy hosta-dhcpa. Następnie zapisujemy zmiany w pliku oraz poleceniem `nano /etc/default/lxd-bridge`. Edytujemy plik konfiguracyjny, który wygenerowany został w momencie zainicjalizowania kontenera lxd. W linii LXD_CONFILe podajemy ścieżkę do pliku konfiguracyjnego dns.conf.

```
# WARNING: This file is generated by a debconf template!
# It is recommended to update it by using "dpkg-reconfigure -p medium lxd"

# Whether to setup a new bridge or use an existing one
USE_LXD_BRIDGE="true"

# Bridge name
# This is still used even if USE_LXD_BRIDGE is set to false
# set to an empty value to fully disable
LXD_BRIDGE="lxdbr0"

# Update the "default" LXD profile
UPDATE_PROFILE="true"

# Path to an extra dnsmasq configuration file
LXD_CONFILe="/etc/default/dns.conf"

# DNS domain for the bridge
LXD_DOMAIN="lxd"
```

```
student@student-VirtualBox:~$ sudo lxc list -c n4
+-----+-----+
| NAME   | IPV4               |
+-----+-----+
| test1  | 10.243.137.105 (eth0) |
+-----+-----+
| test2  | 10.243.137.104 (eth0) |
+-----+-----+
```

Domyślnie kontenery lxc są skonfigurowane w trybie prywatnym tzn, nie są widoczne z zewnątrz. Jednak da się je udostępnić na porcie 443 w tym celu należy zmodyfikować tablicę firewalle.

Zadanie 2

Po zapoznaniu się ze wskazanym wyżej podręcznikiem należy:

1) skopiować do repozytorium lokalnego obraz ubuntu 16.04 i zapisać go pod nazwą (aliasem)

first_ubuntu,

2) potwierdzić, że obraz o zadanym aliasie został dodany do repozytorium lokalnego,

3) skopiować do repozytorium lokalnego obraz ubuntu 16.04 i zapisać go z aliasem identycznym

jak alias używany w repozytorium zdalnym,

4) wylistować wszystkie aliasy obrazów dostępnych w repozytorium lokalnym,

5) dodać do repozytorium lokalnego kolejny obraz na bazie ubuntu 16.04 ale w taki sposób by

był on synchronizowany z repozytorium zewnętrznym. Proszę nadać temu obrazowi alias o

nazwie *auto_ubuntu*. Następnie potwierdzić dodanie tego obrazu do repozytorium lokalnego,

6) wyświetlić informację o ostatnim z dodanych obrazów, które potwierdzą, że ma on ustawioną flagę auto-update.

7) usunąć wszystkie obrazy z repozytorium lokalnego z wyjątkiem obrazu o aliasie *first_ubuntu*

8) utworzyć kontener o nazwie *test3* na podstawie obrazu a aliasie *first_ubuntu* znajdującym się

w repozytorium lokalnym. Potwierdź, że kontener *test3* jest uruchomiony.

9) usuń kontener *test3*. Sprawdź czy obraz, na podstawie którego powstał (*first_ubuntu*) jest wciąż obecny w lokalnym repozytorium.

1) Wynik udanego skopiowania prezentuje poniższy zrzut ekranowy

```
student@student-VirtualBox:~$ sudo lxc image copy ubuntu:16.04 local: --alias first_ubuntu
Image copied successfully!
student@student-VirtualBox:~$
```

2)

W celu potwierdzenia dodania obrazu do lokalnego repozytorium należy wydać polecenie `sudo lxc image list local: .` Wynik polecenia przedstawia poniższy zrzut ekranowy.

```
student@student-VirtualBox:~$ sudo lxc image list local:
+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | | DESCRIPTION |
| ARCH | SIZE | UPLOAD DATE | |
+-----+-----+-----+-----+-----+
| first_ubuntu | 0e237317de16 | no | ubuntu 16.04 LTS i386 (release) (20211001) | i686 | 170.27MB | Nov 2, 2021 at 6:03pm (UTC) |
+-----+-----+-----+-----+-----+
```

3)

W celu skopiowania obrazu pod tym samym aliasem w repozytorium należy wydać polecenie `sudo lxc image copy ubuntu:16.04 local:`. Automatycznie przeniesie to obraz do repozytorium nadając alias taki sam jak nazwa obrazu. Nie powinno się tak robić ponieważ w niektórych przypadkach może to utrudniać rozróżnianie obrazów lokalnych od zdalnych.

```
student@student-VirtualBox:~$ sudo lxc image copy ubuntu:16.04 local:
Image copied successfully!
student@student-VirtualBox:~$
```

4)

W celu wylistowania wszystkich obrazów dostępnych w repozytorium lokalnym należy posłużyć się poleceniem `sudo lxc image alias list`.

```
student@student-VirtualBox:~$ sudo lxc image alias list
+-----+-----+-----+
| ALIAS   | FINGERPRINT | DESCRIPTION |
+-----+-----+-----+
| alias   | 0e237317de16 |             |
+-----+-----+-----+
| first_ubuntu | 0e237317de16 |             |
+-----+-----+-----+
| ubuntu:16.04 | 0e237317de16 |             |
+-----+-----+-----+
```

5)

W celu stworzenia kopii obrazu `ubuntu:16.04` synchronizowanego z obrazem zewnętrznym należy wydać polecenie `sudo lxc image copy` dodatkowo dodając flagę `--auto-update`. Flaga ta umożliwi synchronizację ze zdalnym repozytorium. Informacja o tej flagie znajduje się po wpisaniu polecenia `sudo lxc image -h`. Poniższe zrzuty ekranowe przedstawiają fakt pomyślnego wykonania tej operacji.

```
error: flag provided but not defined: --alias
student@student-VirtualBox:~$ sudo lxc image copy ubuntu:16.04 local: --alias au
to_ubuntu --auto-update
Image copied successfully!
```

```
Image copied successfully.
student@student-VirtualBox:~$ sudo lxc image alias list
+-----+-----+-----+
| ALIAS   | FINGERPRINT | DESCRIPTION |
+-----+-----+-----+
| alias   | 0e237317de16 |             |
+-----+-----+-----+
| auto_ubuntu | 0e237317de16 |             |
+-----+-----+-----+
| first_ubuntu | 0e237317de16 |             |
+-----+-----+-----+
| ubuntu:16.04 | 0e237317de16 |             |
+-----+-----+-----+
```


6)

W celu wyświetlenia informacji o ustawionej fladze w obrazie auto_ubuntu należy wydać polecenie `sudo lxc image info auto_ubuntu`. Następnie odszukać linijki `auto update` w niej znajduje się informacja `enabled` jeśli flaga jest włączona oraz `disabled` jeśli jest wyłączona. Poniższe zrzuty ekranowe przedstawiają wykonanie polecenia.

```
student@student-VirtualBox:~$ sudo lxc image info auto_ubuntu
Fingerprint: 0e237317de16cecd297f6592bbfd0ac31851d13a0fb5f0660025cc7dd24f7ce
Size: 170.27MB
Architecture: i686
Public: no
Timestamps:
  Created: 2021/10/01 00:00 UTC
  Uploaded: 2021/11/02 18:03 UTC
  Expires: 2021/04/21 00:00 UTC
  Last used: 2021/11/02 18:12 UTC
Properties:
  architecture: i386
  description: ubuntu 16.04 LTS i386 (release) (20211001)
  label: release
  os: ubuntu
  release: xenial
  serial: 20211001
  version: 16.04
Aliases:
  - ubuntu:16.04
  - alias
  - first_ubuntu
  - auto_ubuntu
Cached: yes
Auto update: enabled
Source:
  Server: https://cloud-images.ubuntu.com/releases
  Protocol: simplestreams
  Alias: 16.04
```

7)

W celu usunięcia wszystkich aliasów należy wydać polecenie `sudo lxc image delete alias nazwa_aliasu`. Wykonanie tego polecenia przedstawia poniższy zrzut ekranowy.

```
student@student-VirtualBox:~$ sudo lxc image alias list
+-----+-----+-----+
| ALIAS   | FINGERPRINT | DESCRIPTION |
+-----+-----+-----+
| first_ubuntu | 0e237317de16 |
```

8)

W celu stworzenia kontenera na bazie obrazu first_ubuntu należy wydać polecenia `sudo lxc launch first_ubuntu test3`. Następnie poleceniem `sudo lxc list` wyświetlić informacje o kontenerach znajdujących się w systemie macierzystym. Przedstawiają to poniższe zrzuty ekranowe

```

student@student-VirtualBox:~$ sudo lxc launch first_ubuntu test3
Creating test3
Starting test3
student@student-VirtualBox:~$ sudo lxc list
+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 |
| | | TYPE | SNAPSHOTS | |
+-----+-----+-----+-----+
| test1 | RUNNING | 10.243.137.105 (eth0) | fd8a:dcb1:9c8d:8189:216:3eff:fe01:cb23 (eth0) | PERSISTENT | 0 |
+-----+-----+-----+-----+
| test2 | RUNNING | 10.243.137.104 (eth0) | fd8a:dcb1:9c8d:8189:216:3eff:fe01:cb b7 (eth0) | PERSISTENT | 0 |
+-----+-----+-----+-----+
| test3 | RUNNING | 10.243.137.194 (eth0) | fd8a:dcb1:9c8d:8189:216:3eff:fe26:51 e5 (eth0) | PERSISTENT | 0 |
+-----+-----+-----+-----+

```

9)

W celu usunięcia kontenera test3 najpierw za pomocą polecenia `sudo lxc stop test3` należy zatrzymać kontener. Następnie poleceniem `sudo lxc delete test3` usunąć kontener. Poleceniem `sudo lxc list` sprawdzamy czy kontener został usunięty. Natomiast poleceniem `sudo lxc image alias list` sprawdzamy wszystkie lokalne obrazy. Jaki prezentują to poniższe zrzuty ekranowe usunięcie kontenera nie powoduje usunięcia obrazu `first_ubuntu`.

```

student@student-VirtualBox:~$ sudo lxc delete test3
error: The container is currently running, stop it first or pass --force.
student@student-VirtualBox:~$ sudo lxc stop test3
student@student-VirtualBox:~$ sudo lxc delete test3
student@student-VirtualBox:~$ sudo lxc list
+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 |
| | | TYPE | SNAPSHOTS | |
+-----+-----+-----+-----+
| test1 | RUNNING | 10.243.137.105 (eth0) | fd8a:dcb1:9c8d:8189:216:3eff:fe01:cb23 (eth0) | PERSISTENT | 0 |
+-----+-----+-----+-----+
| test2 | RUNNING | 10.243.137.104 (eth0) | fd8a:dcb1:9c8d:8189:216:3eff:fe01:cb b7 (eth0) | PERSISTENT | 0 |
+-----+-----+-----+-----+

```



```

student@student-VirtualBox:~$ sudo lxc image alias list
+-----+-----+-----+
| ALIAS | FINGERPRINT | DESCRIPTION |
+-----+-----+-----+
| first_ubuntu | 0e237317de16 | |
+-----+-----+-----+

```

Zadanie 3

Po zapoznaniu się opisem kluczy i ich wartości (link powyżej) zmodyfikuj konfigurację kontenera `test1` tak aby:

- 1) wykorzystywał tylko 2 CPU (dwa rdzenie procesora),
- 2) wykorzystywał 1GB pamięci RAM,
- 3) miał ograniczenie na prędkość ruchu sieciowego (tak wchodzącego jak i wychodzącego) do 100Mb/s

Wszystkie polecenia oraz wyniki ich działań umieścić z opisem w sprawozdaniu. Dodatkowo umieścić polecenie i jego wynik, które wyświetli zmodyfikowaną konfigurację kontenera i potwierdzi dokonane zmiany.

1)

W celu wykorzystania tylko dwóch CPU należy wykonać polecenie zaprezentowane na poniższym zrzucie ekranowym. Za pomocą polecenia `sudo lxc config show --expanded test1` jesteśmy w stanie wyświetlić informacje o kontenerze jak przedstawia to drugi zrzut ekranowy

```
student@student-VirtualBox:~$ sudo lxc config set test1 limits.cpu 2
student@student-VirtualBox:~$
```

```
student@student-VirtualBox:~$ sudo lxc config show --expanded test1
architecture: i686
config:
  environment.http_proxy: ""
  limits.cpu: "2"
  user.network_mode: ""
  volatile.base_image: 0e237317de16cecd297f6592bbfd0ac31851d13a0fb5f0660025ccd24f7ce
```

- 2) W celu ustawienia ograniczonej pamięci dla kontenera należy użyć polecenia `sudo lxc config set test1 limits.memory 1GB`. Następnie informacje o kontenerze wyświetlić tak samo jak w pierwszym podpunkcie.

```
student@student-VirtualBox:~$ sudo lxc config set test1 limits.memory 1GB
student@student-VirtualBox:~$ sudo lxc config show --expanded test1
architecture: i686
config:
  environment.http_proxy: ""
  limits.cpu: "2"
  limits.memory: 1GB
  user.network_mode: ""
  volatile.base_image: 0e237317de16cecd297f6592bbfd0ac31851d13a0fb5f0660025ccd24f7ce
  volatile.eth0.hwaddr: 00:16:3e:cb:b6:23
  volatile.idmap.base: "0"
  volatile.idmap.next: '[{"Isuid":true,"Isgid":false,"Hostid":165536,"Nsuid":0,"Maprange":65536},{"Isuid":false,"Isgid":true,"Hostid":165536,"Nsuid":0,"Maprange":65536}]'
```

3)

W celu ustawienia ruchu sieciowego przychodzącego jak i wychodzącego należy ustawić profil wykorzystując polecenie `sudo lxc profile device set default eth0 limits.egress 100Mbit`.


```

student@student-VirtualBox:~$ sudo lxc profile device set default eth0 limits.eg
ress 100Mbit
student@student-VirtualBox:~$ sudo lxc config show --expanded test1
architecture: i686
config:
  environment.http_proxy: ""
  limits.cpu: "2"
  limits.memory: 1GB
  user.network_mode: ""
  volatile.base_image: 0e237317de16cecd297f6592bbfd0ac31851d13a0fb5f0660025cc7d
d24f7ce
  volatile.eth0.hwaddr: 00:16:3e:cb:b6:23
  volatile.idmap.base: "0"
  volatile.idmap.next: '[{"Isuid":true,"Isgid":false,"Hostid":165536,"Nsid":0,"M
aprange":65536},{"Isuid":false,"Isgid":true,"Hostid":165536,"Nsid":0,"Maprange":
65536}]'
  volatile.last_state.idmap: '[{"Isuid":true,"Isgid":false,"Hostid":165536,"Nsid
":0,"Maprange":65536},{"Isuid":false,"Isgid":true,"Hostid":165536,"Nsid":0,"Mapr
ange":65536}]'
  volatile.last_state.power: RUNNING
devices:
  eth0:
    limits.egress: 100Mbit

```

Zadanie 4

Które ustawienia są nadrzędne, kontenera czy profilu. Uzasadnij odpowiedź wybranym, prostym przykładem.

Nadrzędnymi ustawieniami są ustawienia profilu. Przykładowo podczas ustawiania limitu ruchu sieciowego dla kontenera pierwszego taki sam limit ustawił się dla kontenera test2.

Zadanie 5

Zakładając, że wciąż działające są dwa kontenery z punktu A tej części ćwiczenia tj. *test1* oraz *test2* (jeśli nie to proszę je utworzyć/uruchomić), proszę odpowiedzieć na pytanie: Czy możliwe jest utworzenie na kontenerze *test1* dowolnego, pustego pliku tekstowego i przesłanie go na kontener *test2* w wybrane miejsce? Odpowiedź uzasadnij i zilustruj odpowiedniki zrzutami ekranowymi.

Bezpośrednie przesłanie pliku z kontenera test1 na kontener test2 nie powiodło się ilustrują to dwa pierwsze zrzuty ekranowe. Jednak mechanizm ten można obejść. Najpierw stworzyć plik w kontenerze test1 pobrać go na maszynę wirtualną z wykorzystaniem polecenia `sudo lxc file pull`. Następnie z maszyny gospodarza za pomocą polecenia `sudo lxc file push` przesłać go do kontenera test2. Przedstawiają to zrzuty ekranowe 3 oraz 4.

```

student@student-VirtualBox:~$ sudo lxc exec test1 -- touch pliktest
student@student-VirtualBox:~$ sudo lxc exec test1 --ls -l
error: flag provided but not defined: --ls
student@student-VirtualBox:~$ sudo lxc exec test1 -- ls -l
total 0
-rw-r--r-- 1 root root 0 Nov  3 19:52 pliktest
student@student-VirtualBox:~$ sudo lxc exec test1 -- pwd
/root
student@student-VirtualBox:~$ sudo lxc exec test1 -- ls
student@student-VirtualBox:~$ sudo lxc file push /root/pliktest test2/root/plik
error: open /root/pliktest: no such file or directory
student@student-VirtualBox:~$

```

```

student@student-VirtualBox:~$ sudo lxc file pull test1/root/pliktest plik
student@student-VirtualBox:~$ ls -l
total 44
drwxr-xr-x 2 student student 4096 paź 29 16:07 Desktop
drwxr-xr-x 2 student student 4096 paź 29 16:07 Documents
drwxr-xr-x 2 student student 4096 paź 29 16:07 Downloads
-rw-r--r-- 1 student student 8980 paź 29 10:31 examples.desktop
drwxr-xr-x 2 student student 4096 paź 29 16:07 Music
drwxr-xr-x 2 student student 4096 paź 29 16:07 Pictures
-rw-r--r-- 1 root root 0 lis  3 21:09 plik
drwxr-xr-x 2 student student 4096 paź 29 16:07 Public
drwxr-xr-x 2 student student 4096 paź 29 16:07 Templates
drwxr-xr-x 2 student student 4096 paź 29 16:07 Videos
student@student-VirtualBox:~$

```

```

student@student-VirtualBox:~$ sudo lxc file pull test1/root/pliktest plik
student@student-VirtualBox:~$ ls -l
total 44
drwxr-xr-x 2 student student 4096 paź 29 16:07 Desktop
drwxr-xr-x 2 student student 4096 paź 29 16:07 Documents
drwxr-xr-x 2 student student 4096 paź 29 16:07 Downloads
-rw-r--r-- 1 student student 8980 paź 29 10:31 examples.desktop
drwxr-xr-x 2 student student 4096 paź 29 16:07 Music
drwxr-xr-x 2 student student 4096 paź 29 16:07 Pictures
-rw-r--r-- 1 root root 0 lis  3 21:09 plik
drwxr-xr-x 2 student student 4096 paź 29 16:07 Public
drwxr-xr-x 2 student student 4096 paź 29 16:07 Templates
drwxr-xr-x 2 student student 4096 paź 29 16:07 Videos
student@student-VirtualBox:~$ sudo lxc file push plik test2/root/pliktest
student@student-VirtualBox:~$ sudo lxc exec test2 -- ls
pliktest
student@student-VirtualBox:~$

```

Zadanie 6

Należy usunąć wszystkie działające kontenery. Potem należy wyłączyć maszynę wirtualną w Virtualbox. Następnie należy w wykonać kopie maszyny wirtualnej PwChO_D i skonfigurować ją do pracy w trybie mostowania (bridge). Uruchomić obie maszyny. Zmienić nazwy maszyn wirtualnych na odpowiednio: *first* oraz *second*. Na maszynie *second* proszę skonfigurować środowisko LXD. Należy zastanowić się jakie adresy IPv4 należy podać przy konfiguracji i czy ich dobór ma znaczenie dla dalszych działań.

1. Na maszynie wirtualnej *first* należy utworzyć kontener LXC o nazwie *master*. Na maszynie

wirtualnej *second* utworzyć kontener o nazwie *slave*. W obu przypadkach należy użyć obrazu: ubuntu 16.04.

2. Należy skonfigurować możliwość współpracy pomiędzy demonami LXD na obu maszynach

wirtualnych, zgodnie z przykładem z linku powyżej.

3. Należy upewnić się czy zdalny kontener *slave* jest „widoczny” z maszyny wirtualnej *first*. A następnie, będąc na maszynie wirtualnej *first* sprawdzić możliwość korzystania z shella na zdalnym kontenerze *slave* .

4. Jeśli powyższe działania zakończyły się niepowodzeniem – proszę podać przyczynę i propozycję rozwiązania problemu. Jeśli zakończyły się sukcesem to proszę sprawdzić czy obecna konfiguracja jest „symetryczna” tj. czy można pracować na zdalnym kontenerze

master z maszyny wirtualnej *second*.

5. Proszę sprawdzić czy wykonane konfiguracje oznaczają, że kontenery „widzą się” na poziomie połączeń sieciowych (proszę wykonać test ping na odpowiednie adresy). Jeśli brak jest połączenia, proszę zaproponować rozwiązanie.

```
student@student-VirtualBox:~$ sudo lxc stop test1
student@student-VirtualBox:~$ sudo lxc stop test2
student@student-VirtualBox:~$ sudo lxc delete test1
student@student-VirtualBox:~$ sudo lxc delete test2
student@student-VirtualBox:~$ sudo lxc list
+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
student@student-VirtualBox:~$
```



1)

Stworzenie kontenerów master oraz slave przedstawiają pierwsze dwa poniższe zrzuty ekranowe. W celu sprawdzenia czy kontenery zostały uruchomione oraz utworzone wykorzystano polecenie `sudo lxc list` na obu maszynach. Wyniki tych poleceń prezentują drugie dwa poniższe zrzuty ekranowe.

```
student@student-VirtualBox:~$ sudo lxc launch ubuntu:16.04 master
[sudo] password for student:
Creating master
Starting master
student@student-VirtualBox:~$
```



```
student@student-VirtualBox:~$ sudo lxc launch ubuntu:16.04 slave
Creating slave
Starting slave
student@student-VirtualBox:~$
```

```
student@student-VirtualBox:~$ sudo lxc list
+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 |
| | TYPE | SNAPSHOTS | |
+-----+-----+-----+-----+
| master | RUNNING | 10.243.137.36 (eth0) | fd8a:dcb1:9c8d:8189:216:3eff:fe67:eeac (eth0) |
| | PERSISTENT | 0 |
+-----+-----+-----+-----+
student@student-VirtualBox:~$
```

```
student@student-VirtualBox:~$ sudo lxc list
+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 |
| | TYPE | SNAPSHOTS | |
+-----+-----+-----+-----+
| slave | RUNNING | 10.243.137.38 (eth0) | fd8a:dcb1:9c8d:8189:216:3eff:fe8c:b1c9 (eth0) |
| | PERSISTENT | 0 |
+-----+-----+-----+-----+
student@student-VirtualBox:~$
```

2)

```
student@student-VirtualBox:~$ sudo lxc config set core.trust_password admin
student@student-VirtualBox:~$ sudo lxc config set core.https_address [::]:8443
student@student-VirtualBox:~$
```

```
student@student-VirtualBox:~$ sudo lxc config set core.https_address [::]:8443
student@student-VirtualBox:~$
```

```
student@student-VirtualBox:~$ sudo lxc remote add slave 192.168.8.112
Certificate fingerprint: c3066a5c30566891ad8ae7142ba39209426322721fffd2df6e26d46c4fb2ada4
ok (y/n)? y
Admin password for slave:
Client certificate stored at server: slave
student@student-VirtualBox:~$
```

3)

W celu sprawdzenia czy w liście zdalnych obrazów znajduje się obraz slave należy wydać polecenie `sudo lxc remote list`. Jak prezentuje to pierwszy zrzut ekranowy udało się skonfigurować jako serwer zdalnych obraz slave z maszyny wirtualnej

ubuntu second. W celu wejścia do powłoki shelowej zdalnie z maszyny wirtualnej master należy poleceniem `sudo lxc list slave` sprawdzić jakie są dostępne działające kontenery na maszynie slave. Następnie wydając polecenie `sudo lxc exec slave:slave bash` jesteśmy w stanie wejść do powłoki shell kontenera slave. Prezentuje to drugi zrzut ekranowy.

```
student@student-VirtualBox:~$ sudo lxc remote list
+-----+-----+-----+-----+-----+
| NAME | URL | PROTOCOL | P |
+-----+-----+-----+-----+
| images | https://images.linuxcontainers.org | simplestreams | Y |
+-----+-----+-----+-----+
| local (default) | unix:// | lxd | N |
+-----+-----+-----+-----+
| slave | https://192.168.8.112:8443 | lxd | N |
+-----+-----+-----+-----+
| ubuntu | https://cloud-images.ubuntu.com/releases | simplestreams | Y |
+-----+-----+-----+-----+

student@student-VirtualBox:~$ sudo lxc list slave:
+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 |
| TYPE | SNAPSHOTS |
+-----+-----+-----+-----+
| slave | RUNNING | 10.243.137.38 (eth0) | fd8a:dcb1:9c8d:8189:216:3eff:fe8c:b1c9 (eth0) |
| PERSISTENT | 0 |
+-----+-----+-----+-----+

student@student-VirtualBox:~$ sudo lxc exec slave:slave bash
root@slave:~#
```

4)

Konfiguracja nie jest symetryczna i nie da się z maszyny wirtualnej second uruchomić kontenera master na maszynie first. Dzieje się tak dlatego, że na maszynie slave nie został skonfigurowany żaden zdalny obraz. W celu takiej komunikacji należało by skonfigurować serwer zdalnych z maszyny first na maszynie second.

```

student@student-VirtualBox:~$ sudo lxc list remote:
[sudo] password for student:
error: The remote "remote" doesn't exist
student@student-VirtualBox:~$ sudo lxc list remote
+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
student@student-VirtualBox:~$ sudo lxc exec master:master bash
error: The remote "master" doesn't exist
student@student-VirtualBox:~$ █

```

5)

Obecnie dwa kontenery nie mogą się pingować, ponieważ znajdują się w innej sieci. W celu rozwiązania tego problemu można by stworzyć bridge do którego można było by podłączyć oba kontenery i w ten sposób za pomocą dhcp łączyły by się ze sobą. Można też statycznie przypisać adresy ip dla kontenerów tak by znajdowały się w jednej sieci.

```

student@student-VirtualBox:~$ sudo lxc list
+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| slave | RUNNING | 10.243.137.38 (eth0) | fd8a:dcb1:9c8d:8189:216:3eff:fe8c:b1c9 (eth0) | PERSISTENT | 0 |
+-----+-----+-----+-----+-----+-----+
student@student-VirtualBox:~$ ping 10.243.137.36
PING 10.243.137.36 (10.243.137.36) 56(84) bytes of data.
From 10.243.137.1 icmp_seq=1 Destination Host Unreachable
From 10.243.137.1 icmp_seq=2 Destination Host Unreachable
From 10.243.137.1 icmp_seq=3 Destination Host Unreachable
From 10.243.137.1 icmp_seq=4 Destination Host Unreachable
From 10.243.137.1 icmp_seq=5 Destination Host Unreachable
From 10.243.137.1 icmp_seq=6 Destination Host Unreachable
^C
--- 10.243.137.36 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet loss, time 7152ms
pipe 4
student@student-VirtualBox:~$ █

```

```

student@student-VirtualBox:~$ ping 10.243.137.38
PING 10.243.137.38 (10.243.137.38) 56(84) bytes of data.
From 10.243.137.1 icmp_seq=1 Destination Host Unreachable
From 10.243.137.1 icmp_seq=2 Destination Host Unreachable
From 10.243.137.1 icmp_seq=3 Destination Host Unreachable
^C
--- 10.243.137.38 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 4089ms
pipe 4
student@student-VirtualBox:~$

```