LABORATORIUM PROGRAMOWANIA W CHMURACH OBLICZENIOWYCH

LABORATORIUM NR 3.

UWAGA: Tryb sieciowy maszyn wirtualnych z systemem Ubuntu 16.04 należy ustawić na tryb mostowania (ang. bridge) i w tym trybie uruchomić w środowisku VirtualBox.

Ćwiczenie laboratoryjne, w części pierwszej zawiera wprowadzenie do zaawansowane zarządzania środowiskiem LXD. W części drugiej zawarte jest wprowadzenie do pracy LXD w konfiguracji zawierającej wiele hostów.

CZESC I

Zaawansowane wykorzystanie środowiska LXD.

A. Podobnie jak w przypadku pojedynczych kontenerów LXC, w środowisku LXD jest możliwe definiowanie poszczególnych trybów sieciowych. Można konfigurować ustawienia interfejsów sieciowych dla pojedynczego kontenera (poprzez zmianę jego konfiguracji) lub dla zestawu kontenerów (poprzez konfigurację profilu, do którego te kontenery są przypisane). Zagadnienia zmian w konfiguracji kontenera i profilu są omawiane w kolejnym podpunkcie (podpunkt B).

Uwaga: Jeżeli potrzebne są zmiany w ustawieniach sieciowych całego środowiska LXD to konieczne jest wykonanie części zadań ze skryptu init (zagadnienie przedstawione w ostatniej części poprzedniego zadania laboratoryjnego).

Domyślnie środowisko LXD wykorzystuje dedykowany most *lxdbr0*. Wszystkie nowo stworzone kontenery LXC są przyłączane do tego mostu.

3P1. 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 dokumentacje (użyte polecenia, wyniki ich działań, komentarz czego dowodzi dany test) następujących zadań cząstkowych (zagadnień):

- ustawienia firewall-a na maszynie gospodarza w odniesieniu do pulu adresowej, przypisanej kontenerom w konfiguracji środowiska LXD.

- czy interfejsy kontenerów są przyłączone do mostu lxdbr0 ?

- czy można „pingować” (należy włączyć opcje rejestracji trasy) kontener *test2* z kontenera *test1* i na odwrót ?

Podpowiedź: w celu dokonania testu ping nie trzeba logować się w kontenerze. Można (należy) wykorzystać możliwość wykonywania zadań „z zewnątrz”, przykładowo jak ilustruje to rysunek poniżej.

```
student@student-VirtualBox:~$ sudo lxc list -c n4
+-------+---------------------+
| NAME  |        IPV4         |
+-------+---------------------+
| test1 | 10.0.100.196 (eth0) |
+-------+---------------------+
| test2 | 10.0.100.54 (eth0)  |
+-------+---------------------+
student@student-VirtualBox:~$ sudo lxc exec test2 -- ping -R 10.0.100.196
PING 10.0.100.196 (10.0.100.196) 56(124) bytes of data.
64 bytes from 10.0.100.196: icmp_seq=1 ttl=64 time=0.085 ms
RR:     10.0.100.54
        10.0.100.196
        10.0.100.196
        10.0.100.54

64 bytes from 10.0.100.196: icmp_seq=2 ttl=64 time=0.048 ms     (same route)
^C
--- 10.0.100.196 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.048/0.066/0.085/0.020 ms
```

- czy można „pingować" kontenery z maszyny gospodarza (ubuntu na VirtualBox) i z OpenSuse ?

- czy można „pingować" z danego kontenera na system gospodarza (Ubuntu) i do OpenSuse.

- 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.

Zadanie proszę zakończyć krótkim podsumowaniem, jak domyślnie skonfigurowane są połączenia sieciowe kontenerów LXC w środowisku LXD.

Bardzo częstym przypadkiem jest gdy chcemy aby maszyna gospodarza (i inne podłączone zasoby) znajdowały się w tej samej podsieci co kontenery. W takim przypadku konieczna jest zmiana konfiguracji środowiska LXD i samodzielne utworzenie mostu, który będzie wykorzystywany przez środowisko LXD. Szczegółową procedura jest opisana pod adresem: https://www.simpleprecision.com/ubuntu-16-04-lxd-networking-simple-bridge/

B. Projektanci środowiska LXD duży nacisk położyli na jego elastyczność i konfigurowalność w stosunku do standardowych kontenerów LXC. Na tym laboratorium skoncentrujemy się na dwóch konkretnych rozwiązaniach tj. na operowaniu obrazami oraz na konfiguracji kontenerów i grup kontenerów.

LXD (podobnie jak LXC) bazuje na obrazach tj. każdy kontener bazuje na określonym obrazie. Istnieją trzy podstawowe zasoby, z których można pobierać obrazy (a następnie konfigurować i modyfikować):
- zewnętrzne serwery z obrazami,
- lokalny (również na zdalnym LXD) serwer obrazów,
- import obrazu z pliku.
Opis oraz omówienie dostępnych narzędzi dla ostatniego z wymienionych rozwiązań można znaleźć pod adresem: https://insights.ubuntu.com/2016/04/01/lxd-2-0-image-management-512/

Zewnętrzne serwery są podzielone na kolejne trzy grupy, odpowiednio:

1. ubuntu: (zawiera stabilne/oficjalne wydania systemu Ubuntu)
2. ubuntu-daily: (zawiera wersje rozwojowe systemu ubuntu)

3. images: (zawiera obrazy innych dystrybucji niż ubuntu)

W poprzednim laboratorium podane były przykłady wykorzystania repozytoriów 1 oraz 3. Analogicznie uzyskuje się dostęp do obrazów z repozytorium 2 (w razie wątpliwości proszę zapoznać się z pomocą dla polecenia: `lxc image list`). Szczegóły o repozytoriach (po domyślnej konfiguracji środowiska LXD) powinny wyglądać jak na zrzucie ekranowym poniżej (polecenie: `lxc remote list`).

```
student@student-VirtualBox:~$ sudo lxc remote list
[sudo] password for student:
+-----------------+-----------------------------------------+---------------+--------+--------+
|      NAME       |                  URL                    |   PROTOCOL    | PUBLIC | STATIC |
+-----------------+-----------------------------------------+---------------+--------+--------+
| images          | https://images.linuxcontainers.org      | simplestreams | YES    | NO     |
+-----------------+-----------------------------------------+---------------+--------+--------+
| local (default) | unix://                                 | lxd           | NO     | YES    |
+-----------------+-----------------------------------------+---------------+--------+--------+
| ubuntu          | https://cloud-images.ubuntu.com/releases | simplestreams | YES    | YES    |
+-----------------+-----------------------------------------+---------------+--------+--------+
| ubuntu-daily    | https://cloud-images.ubuntu.com/daily   | simplestreams | YES    | YES    |
+-----------------+-----------------------------------------+---------------+--------+--------+
```

Wśród wyświetlonych repozytoriów jest repozytorium o nazwie local. Jest ono bardzo użyteczne w codziennej pracy z kontenerami LXC w środowisku LXD. Wykonane dotąd ćwiczenia potwierdziły, że obraz pobierany jest ze zdalnego repozytorium tylko za pierwszym razem. Tworzenie kolejnego kontenera na bazie tego obrazu korzysta już z zasobów cache systemu (korzysta z pobranych danych). Istnieje możliwość zapisania obrazów bezpośrednio do repozytorium local i potem wykorzystywania ich tak jakby to było repozytorium zewnętrzne.

W systemie Ubuntu (po wykonaniu punktu A) powinny być działające dwa kontenery, *test1* oraz *test2*. Oba powstały na bazie obrazu Ubuntu 16.04 amd64. Wobec tego, bez konieczności pobierania danych z sieci, można szybko przenieść z cache obraz do repozytorium lokalnego.

UWAGA: Oczywiście jeśli dany obraz nie jest dostępny w cache albo nie jest aktualny, to dane zostaną pobrane z repozytorium zdalnego i dodane do repozytorium lokalnego.

Należy wykorzystać polecenia (nazwa systemu oczywiście może być zmieniona zależnie od potrzeb):

```
sudo lxc image copy ubuntu:16.04 local:
```

W większości przypadków najwygodniej jest posługiwać się własną nazwą obrazu (aby np. odróżnić lokalny obraz od obrazów dostępnych na zdalnych repozytoriach). Nowe kontenery można wtedy tworzyć powołując się na przypisany alias. Należy wydać polecenie (nazwa aliasu przykładowa):

```
sudo lxc image copy ubuntu:16.04 local: --alias base-ubuntu
```

Przykładowy wynik działania tego ostatniego polecenia oraz potwierdzenie dodania obrazu do lokalnego repozytorium jest przedstawione poniżej:

```
student@student-VirtualBox:~$ sudo lxc image copy ubuntu:16.04 local: --alias base_ubuntu
Image copied successfully!
student@student-VirtualBox:~$ sudo lxc image list local:
+-------------+-------------+--------+----------------------------------------+--------+----------+
|    ALIAS    | FINGERPRINT | PUBLIC |               DESCRIPTION              |  ARCH  |   SIZE   |
+-------------+-------------+--------+----------------------------------------+--------+----------+
| base_ubuntu | 315bedd32580 | no     | ubuntu 16.04 LTS amd64 (release) (20161020) | x86_64 | 144.29MB |
+-------------+-------------+--------+----------------------------------------+--------+----------+
```

Należy zapoznać się z podręcznikiem systemowym zestawu poleceń lxc image ponieważ zawiera on opis wielu narzędzi i opcji do manipulowania obrazami, co zresztą podkreśla początek tego podręcznika (poniżej).

```
student@student-VirtualBox:~$ sudo lxc image -h
Usage: Manipulate container images.

In LXD containers are created from images. Those images were themselves
either generated from an existing container or downloaded from an image
server.

When using remote images, LXD will automatically cache images for you
and remove them upon expiration.

The image unique identifier is the hash (sha-256) of its representation
as a compressed tarball (or for split images, the concatenation of the
metadata and rootfs tarballs).

Images can be referenced by their full hash, shortest unique partial
hash or alias name (if one is set).
```

3P2. Po zapoznaniu się ze wskazanym wyżej podręcznikiem należy:

- skopiować do repozytorium lokalnego obraz ubuntu 16.04 i zapisać go pod nazwą (aliasem) *first_ubuntu*,
- potwierdzić, że obraz o zadanym aliasie został dodany do repozytorium lokalnego,
- skopiować do repozytorium lakolnego obraz ubuntu 16.04 i zapisać go z aliasem identycznym jak alias używany w repozytorium zdalnym,
- wylistować wszystkie aliasy obrazów dostępnych w repozytorium lokalnym,
- 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,
- wyświetlić informację o ostatnim z dodanych obrazów, które potwierdzą, że ma on ustawioną flagę auto-update.
- usunąć wszystkie obrazy z repozytorium lokalnego z wyjątkiem obrazu o aliasie *first_ubuntu*
- utworzyć kontener o nazwie test3 na podstawie obrazu a aliasie *first_ubuntu* znajdującym się w repozytorium lokalnym. Potwierdź, że kontener *test3* jest uruchomiony.
- usuń kontenet test3. Sprawdź czy obraz, na podstawie którego powstał (*first_ubuntu*) jest wciąż obecny w lokalnym repozytorium.

Wszystkie polecenia oraz wyniki ich działań umieścić z opisem w sprawozdaniu.

Kolejnym elementem zapewniającym elastyczność środowiska LXD jest możliwość konfiguracji poszczególnych kontenerów, jak i grup kontenerów (poprzez profile).

Każdy kontener utworzony w LXD ma swoją konfigurację, którą można edytować. Podstawowe polecenia służące do operowania na konfiguracji pojedynczego kontenera przedstawione są poniżej:

```
lxc config show <nazwa kontenera>
```

albo w wersji rozszerzonej (zawierającej dane profilu)

```
lxc config show --expanded <nazwa kontenera>
```

Przykładowo, dla (zakładając, że wciąż jest uruchomiony) kontenera o nazwie *test1* wynik działania polecenia rozszerzonego jest przedstawiony poniżej.

```
student@student-VirtualBox:~$ sudo lxc config show --expanded test1
name: test1
profiles:
- default
config:
  volatile.base_image: 315bedd32580c3fb79fd2003746245b9fe6a8863fc9dd990c3a2dc90f4930039
  volatile.eth0.hwaddr: 00:16:3e:83:7f:87
  volatile.last_state.idmap: '[{"Isuid":true,"Isgid":false,"Hostid":165536,"Nsid":0,"Ma
prange":65536},{"Isuid":false,"Isgid":true,"Hostid":165536,"Nsid":0,"Maprange":65536}]'
devices:
  eth0:
    name: eth0
    nictype: bridged
    parent: lxdbr0
    type: nic
  root:
    path: /
    type: disk
ephemeral: false
```

Każdą konfiguracja może zostać poddana edycji

```
lxc config edit <nazwa kontenera>
```

Istnieje też możliwość bezpośredniej zmiany danego parametru konfiguracji (bez konieczności otwierania całości konfiguracji w edytorze co wykonuje polecenie przedstawione wyżej). Służy do tego polecenie o składni:

```
lxc config set <nazwa kontenera> <klucz> <wartość>
```

Z kolei polecenie służące do zmiany/zdefiniowania parametrów konfiguracji sieciowej kontenera ma składnie:

```
lxc network set <sieć> <klucz> <wartość>
```

Jeszcze inna składnia obowiązuje w przypadku dodawania urządzeń wejścia-wyjścia.

```
lxc config device set <nazwa kontenera> <typ urządzenia> <wartość klucza>
```

Przykładowo, aby dodać urządzenie znakowe kvm do kontenera o nazwie testowty_kontener w ścieżce /dev/kvm :

```
lxc config device add testowy_kontener kvm unix-char path=/dev/kvm
```

Uwaga: urządzenia wejścia i wyjścia (w tym interfejsy sieciowe) mogą być dodawane do działającego kontenera.

W wersji LXD 2.0 (obecna wersja) wspierane są następujące rodzaje urządzeń:

- **disc** – może to być fizyczny dysk lub partycja mountowana w kontenerze jak i zasób zewnętrzny (np. iscsi, nfs)

- **nic** – interfejs sieciowy, który może być skonfigurowany w trybach sieciowych poznanych na poprzednim laboratorium,
- **unix-block** – np. /dev/sda,
- **unix-char** – np. /dev/kvm
- **none** – specjalne urządzenie służące np. "ukrywania" urządzeń, których konfiguracja mogłaby zostać nadpisana przez ustawienia profilu.

Pełna lista kluczy i przypisanych im wartości jest dostępna pod adresem: https://github.com/lxc/lxd/blob/master/doc/configuration.md

Jeżeli ustawienia/zmiana ustawień ma dotyczyć grupy kontenerów to można wykorzystać w tym celu tzw. profile. Posługiwanie się profilami jest analogiczne do zarządzania konfiguracją poszczególnych kontenerów. Podstawowe polecenia przedstawia rysunek poniżej:

```
student@student-VirtualBox:~$ sudo lxc profile list
default
docker
student@student-VirtualBox:~$ sudo lxc profile show default
name: default
config: {}
description: Default LXD profile
devices:
  eth0:
    name: eth0
    nictype: bridged
    parent: lxdbr0
    type: nic
```

Uwaga: wszystkie ćwiczenia na poprzednich I obecnym laboratorium wykonywane były na kontenerach opartych o profil *default*. Wersja LXD 2.0 wprowadziła nowy profil *docker*. Zostanie on wykorzystany w kolejnym laboratorium.

Pozostałe polecenia są odpowiednikiem tych, które służyły manipulowaniu konfiguracją pojedynczego kontenera, np. edycja konfiguracji profilu:

```
lxc profile edit <nazwa profilu>
```

Analogicznie rzecz się ma z poleceniami:

```
lxc profile set ....

lxc profile device add ....
```

3P3. Po zapoznaniu się opisem kluczy i ich wartości (link powyżej) zmodyfikuj konfigurację kontenera *test1* tak aby:
- wykorzystywał tylko 2 CPU (dwa rdzenie procesora),
- wykorzystywał 1GB pamięci RAM,
- 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 umieść polecenie i jego wynik, które wyświetli zmodyfikowaną konfigurację kontenera i potwierdzi dokonane zmiany.

3P4. Które ustawienia są nadrzędne, kontenera czy profilu. Uzasadnij odpowiedź wybranym, prostym przykładem.

PYTANIE DODATKOWE (dla osób posiadających system Linux bez VirtualBox):
3D1A Podaj czy a jeśli tak, to w jaki sposób można zmienić konfigurację kontener LXC tak by miał on dostęp do zawartości pena USB. (system plików na pen-ie dowolny)

PYTANIE DODATKOWE (dla osób korzystających z systemu Linux za pośrednictwem VirtualBox):
3D1B Typowy okres ważności obrazu w repozytorium to 10 dni. Opierając się o poznane polecenia oraz dokumentacje LXD, podaj jak zmienić ten czas na np. 30 dni.

C. Kontenery LXC w środowisku LXD pozwalają na bardzo wygodną pracę z zasobami zawartymi w kontenerze. Poniżej podstawowe, przydatne polecenia:

Aby mieć dostęp do shella kontenera (założono, że bash jest obecny w kontenerze):

```
lxc exec <nazwa kontenera> bash
```

Powrót do system gospodarza następuje przez komendę: *exit*

Uwaga: Wszystkie polecenia wykonywane w shellu kontenera są z poziomu użytkownika root z minimalnymi ustawieniami i ścieżką domową ustawioną na /root. Dodatkowe zmienne środowiska mogą być ustawione z linii komend (tymczasowo) lub poprzez zmianę konfiguracji kontenera (profilu do którego należy)/ Ustawienia są wtedy permanentne a wykorzystuje się wtedy „environment <key>”

Bardziej złożone polecenia wymagają użycia seperatora jak w przykładzie poniżej:

```
lxc exec <container> -- ls -lh /
```

Ponieważ LXD pozwala na bezpośredni dostęp do system plików kontenera to możliwa jest prosta wymiana plików. Służą do tego polecenia `lxc file pull` oraz `lxc file push`

Składnia polecenia do pobrania pliku z kontenera:

```
lxc file pull <nazwa kontenera>/<ścieżka> <cel>
```

Przykładowo, aby pobrać /etc/host z kontenera *test1* i zapisać w bieżącym katalogu pod nazwą host:

```
sudo lxc file pull test1/etc/hosts hosts
```

Można też ten plik bezpośrednio wyświetlić na konsoli, tak jak pokazuje to przykład poniżej:

```
student@student-VirtualBox:~$ sudo lxc file pull test1/etc/hosts -
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Analogicznie, składnia polecenia do przesłania pliku do danego kontenera:

```
lxc file push <źródło> <nazwa kontenera>/<ścieżka>
```

Możliwe jest też bezpośrednie edytowanie plików w kontenerze:

```
lxc file edit <container>/<path>
```

3P5. Zakładając, że wciąż działające są dwa kontenery z punktu A tej części ćwiczenia tj. *test1* oraz *test 2* (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.

CZESC II

Konfiguracja środowiska LXD do pracy na wielu hostach.

Środowisko LXD w wersji 2.0 zostało opracowane z myślą o możliwości pracy na wielu hostach. W tym celu zdefiniowane zostały dwie podstawowe metody komunikacji pomiędzy demonami LXD jak również LXD a światem zewnętrznym:
- LXD API – bazujący na REST API
- Simplestreams – protokół komunikacji opracowany głównie dla potrzeb dystrybucji obrazów z repozytoriów zewnętrznych. (proszę wrócić do zrzutu ekranowego z początku instrukcji – część I B str.3 gdzie listing dostępnych repozytorium wskazywał na wykorzystanie tego protokołu).

LXD pozwala nie tylko na zdalną pracę na kontenerach, ale również na swobodne kopiowanie i przenoszenie (również w trakcie działania) kontenerów pomiędzy hostami (maszynami fizycznymi). Aby w praktyce przećwiczyć te cechy funkcjonalne środowiska LXD proszę najpierw zapoznać się z informacjami i przykładami zawartymi pod adresem:
https://www.stgraber.org/2016/04/12/lxd-2-0-remote-hosts-and-container-migration-612/

!!
Wykonanie poniższego zadania na zajęciach i prezentacja rezultatów prowadzącemu będzie dodatkowo oceniona (doceniona).
!!

3P6. 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 podac 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 odpowienie adresy). Jeśli brak jest połączenia, proszę zaproponować rozwiązanie.

W przypadku, gdy zadanie nie zostanie wykonane na zajęciach, w sprawozdaniu punkty 1 – 6 proszę opisać, zilustrować wynikami działań odpowiednich poleceń oraz krótko skomentować.

ZADANIE DODATKOWE (termin oddania – 2 tygodnie po zajęciach)

Podobnie jak w przypadku LXC, środowisko LXD nie dostarcza domyślnego środowiska graficznego. Uwaga developerów skoncentrowana została na LXD API. Zainteresowani dokładniejszym jego opisem powinni zapoznać się z informacjami na stronach:

https://www.stgraber.org/2016/04/18/lxd-api-direct-interaction/
https://github.com/lxc/lxd/blob/master/doc/rest-api.md

Tym niemniej powstały próby opracowania WebGUI dla LXD. Popularnymi przykładami są:

https://github.com/aarnaud/lxd-webui
https://github.com/dobin/lxd-webgui

Pod względem funkcjonalnym, większe zainteresowanie budzi to drugie rozwiązanie (https://github.com/dobin/lxd-webgui)

Proszę szczegółowo opisać poszczególne etapy instalacji i konfiguracji tego rozwiązania w swoim środowisku LXD oraz przedyskutować dostrzeżone jego zalety i wady.

Na zakończenie ćwiczenia PROSZĘ KONIECZNIE zatrzymań wszystkie utworzone kontenery LXC.

Plik ze sprawozdanie prosze wgrać do systemu moodle. Preferowany format to pdf. Plik proszę nazwać zgodnie ze schematem, jak przy poprzednich sprawozdaniach:
CW3_dzień_godz_Nazwisko.pdf

Plik z zadaniem dodatkowym również proszę wgrać do odpowiedniego katalogu na moodle oraz proszę go nazwać:
Z_DOD1_dzień_godz_Nzwisko.pdf
