

## LABORATORIUM 4. TRYBY SIECIOWE KONTENERA DOCKER

### Zadanie 4.1. Konfiguracja sieci typu user-defined bridge

Należy zbudować następującą strukturę połączeń pomiędzy kontenerami i siecią zewnętrzną, opisaną w postaci skryptu bash z odpowiednimi komentarzami.

- Kontenery „T2” oraz „D2” nasłuchują na wewnętrznych portach 80 (portach swoich wirtualnych interfejsów sieciowych). Porty mają być mapowane na podane porty na interfejsie hosta macierzystego (odpowiednio 80 i 8000). Proszę zwracać uwagę, że mapowanie jest realizowane indywidualnie dla każdego Endpointa (interfejsu wirtualnego kontenera) przyłączonego do danej sieci.
- Kontener na bazie obrazu Ubuntu (np. latest) „late” ma zostać przyłączony do dwóch sieci pracujących pod nadzorem User-defined Network Driver. Powinno być to wykonane jako ostatnie działanie. Aby uniknąć domyślnego przyłączenia do mostu docker0, można utworzyć ten kontener (`$ docker create`) a następnie przyłączyć do wymaganych sieci (`$ docker network connect`). Na końcu można go uruchomić.
- Kontenery „D1” oraz „S1” mają być przyłączone do sieci wykorzystującej Userdefined Network Driver. Należy uzyskać możliwość niefiltrowanej (bez ograniczeń) i dwukierunkowej komunikacji z interfejsem hosta macierzystego. Odpowiada to sytuacji, jakby wymienione kontenery i host macierzysty przyłączone były do przełącznika sieciowego (można też takie połączenie porównać do trybu sieciowego bridge w Virtualbox). Należy przyjrzeć się ustawieniom iptables dla tegotrybu i w opracowywanym skrypcie umieścić odpowiednie reguły (ewentualnie usunąć wybrane reguły) tak by możliwa była taka komunikacja. Pomocny link: <https://docs.docker.com/network/iptables/>
- Kontenery „D2” oraz „S1” mają mieć bezpośredni dostęp (via most) do interfejsu hosta macierzystego. W jaki sposób można ustalić i skonfigurować pulę adresową dla podsieci na moście „bridge2”?
- Czy kontenery „D2” i „S1” mogą wykorzystać mechanizm dynamicznego przypisywania adresów (DHCP) jeśli w tym segmencie sieci dostępny byłby serwer DHCP. Czy wymaga to dodatkowej konfiguracji któregoś z elementów zbudowanej infrastruktury? Jeśli tak, to proszę podać co i jak należało skonfigurować.

W sprawozdaniu, oprócz wymienionego wyżej skryptu, należy umieścić końcową konfigurację wszystkich sieci, tablicę routingu na hoście macierzystym oraz kontenerach „T2” oraz „D2”, jak również istotne fragmenty konfiguracji iptables. W ocenie końcowej brane będzie pod uwagę wskazanie miejsc w w/w listingach, które potwierdzają poprawność wykonania zadania.

```
Otwórz ▾ J+ jl_2022_03_24_lab4_zad1.sh

1#!/bin/bash
2
3#zatrzymaj kontenery
4sudo docker container stop T1 T2 D1 D2 S1 late
5
6#usun kontenery
7sudo docker container rm T1 T2 D1 D2 S1 late
8
9#usun sieci
10sudo docker network rm bridge1
11sudo docker network rm bridge2
12
13#utworz sieci
14sudo docker network create --subnet 10.0.10.0/24 bridge1
15sudo docker network create --subnet 192.168.1.0/24 bridge2
16
17#uruchom kontenery
18sudo docker run -itd --name T1 alpine sh
19sudo docker run -itd --name T2 --net bridge1 -p 10.0.10.1:8080:8080 nginx sh
20sudo docker run -itd --name D1 --net bridge1 --ip 10.0.10.254 alpine sh
21sudo docker run -itd --name D2 httpd
22
23sudo docker network connect bridge2 D2
24sudo docker network connect bridge1 D2
25
26sudo docker run -itd --name late ubuntu
27
28sudo docker network connect bridge2 late
29sudo docker network connect bridge1 late
30
31sudo docker run -itd --name S1 --net bridge2 ubuntu|
```

```
student@student-VirtualBox:~$ ./jl_2022_03_24_lab4_zad1.sh
T1
T2
D1
D2
S1
late
T1
T2
D1
D2
S1
late
bridge1
bridge2
04c4f1a67d77df097ceabff64e16e890a1f4ba1f82d4959559486fd51814125
e066a6248062ad3edb666eeb5b9ae4cf55d9982bcf3cadc53aa990b71ea7029a
17a688e959a209bc0617e5f3ec3489944aa69c1646ffb97162ce99ed41aa26ad
d1eb665db6a7ea7fc35683079dcf59e4faab74e0d521691ae1c39379a77e2722
46eba50e66158df3be64cb7b716f80b7eef9c5213fefafa2304839644164bb3693
ea891ea487a400e7fbb063208b490b468bc20c9db7c56c2681f35954eb967198
2d56432df1fd15e59a8dcraf06f77e6b6af3a9d9ece172115724c4e80fdf671eb
1a50282183fc295bb9e0023be691f9501329ef1633d1124dcdb811b4eaa5adf6
```

```
student@student-VirtualBox:~$ sudo docker network inspect bridge1
[{"Name": "bridge1",
 "Id": "04c4f1a67d77df097ceaabff64e16e890a1f4ba1f82d4959559486fd51814125",
 "Created": "2022-03-30T17:45:58.025358639+02:00",
 "Scope": "local",
 "Driver": "bridge",
 "EnableIPv6": false,
 "IPAM": {
     "Driver": "default",
     "Options": {},
     "Config": [
         {
             "Subnet": "10.0.10.0/24"
         }
     ]
 },
 "Internal": false,
 "Attachable": false,
 "Ingress": false,
 "ConfigFrom": {
     "Network": ""
 },
 "ConfigOnly": false,
 "Containers": {
     "2d56432df1fd15e59a8dcf06f77e6b6af3a9d9ece172115724c4e80fdf671eb": {
         "Name": "late",
         "EndpointID": "5e20a24e462a598aa7642094aeb4d0967fc6cc50f9c03a7e9a2aea6999499183",
         "MacAddress": "02:42:0a:00:0a:04",
         "IPv4Address": "10.0.10.4/24",
         "IPv6Address": ""
     },
     "46eba50e66158df3be64cb7b716f80b7eef9c5213fefafa2304839644164bb3693": {
         "Name": "D1",
         "EndpointID": "9904e2ef8ee88b7fa4bbb9de5769e0544111fe65a183bea064c16b08ecbc1f9",
         "MacAddress": "02:42:0a:00:0a:fe",
         "IPv4Address": "10.0.10.254/24",
         "IPv6Address": ""
     },
     "d1eb665db6a7ea7fc35683079dcf59e4faab74e0d521691ae1c39379a77e2722": {
         "Name": "T2",
         "EndpointID": "3c898932d09ed2b460f308ee5203e92cb7f2eaeb4edf137c28e28ca85abb7d38",
         "MacAddress": "02:42:0a:00:0a:02",
         "IPv4Address": "10.0.10.2/24",
         "IPv6Address": ""
     }
 }
```

```
student@student-VirtualBox:~$ sudo docker network inspect bridge2
[{"Name": "bridge2",
 "Id": "e066a6248062ad3edb666eeb5b9ae4cf55d9982bcf3cad53aa990b71ea7029a",
 "Created": "2022-03-30T17:45:58.3877635+02:00",
 "Scope": "local",
 "Driver": "bridge",
 "EnableIPv6": false,
 "IPAM": {
   "Driver": "default",
   "Options": {},
   "Config": [
     {
       "Subnet": "192.168.1.0/24"
     }
   ]
 },
 "Internal": false,
 "Attachable": false,
 "Ingress": false,
 "ConfigFrom": {
   "Network": ""
 },
 "ConfigOnly": false,
 "Containers": {
   "1a50282183fc295bb9e0023be691f9501329ef1633d1124dcdb811b4eaa5adf6": {
     "Name": "S1",
     "EndpointID": "d3e58775ad8f1e61afdc23a20f00d96c23fbda51baac57a9dd55987328bfe3bd",
     "MacAddress": "02:42:c0:a8:01:04",
     "IPv4Address": "192.168.1.4/24",
     "IPv6Address": ""
   },
   "2d56432df1fd15e59a8dcf06f77e6b6af3a9d9ece172115724c4e80fdf671eb": {
     "Name": "late",
     "EndpointID": "cf947961843aeb0057b96094542a5c3aff2cb98d6a4759d0946d99713147c8c7",
     "MacAddress": "02:42:c0:a8:01:03",
     "IPv4Address": "192.168.1.3/24",
     "IPv6Address": ""
   },
   "ea891ea487a400e7fbb063208b490b468bc20c9db7c56c2681f35954eb967198": {
     "Name": "D2",
     "EndpointID": "408b022128335191b43799ae881197612dc73ab8152de27581e1182482cdd025",
     "MacAddress": "02:42:c0:a8:01:02",
     "IPv4Address": "192.168.1.2/24",
     "IPv6Address": ""
   }
 },
 "Options": []
}
```

**W jaki sposób można ustalić i skonfigurować pulę adresową dla podsieci na moście „bridge2”?**

Podczas tworzenia sieci bridge2 trzeba użyć opcji “--ip-range”

**Czy kontenery „D2” i „S1” mogą wykorzystać mechanizm dynamicznego przypisywania adresów (DHCP) jeśli w tym segmencie sieci dostępny byłby serwer DHCP. Czy wymaga to dodatkowej konfiguracji któregoś z elementów zbudowanej infrastruktury? Jeśli tak, to proszę podać co i jak należałoby skonfigurować.**

Należy jako sterownika sieci bridge2 użyć DHCP IPAM (IP Adress Management).

## Zadanie 4.2. Użycie aliasów w komunikacji sieciowej kontenerów

Zmodyfikuj skrypt z zadania 4.1 tak by uwzględniał on użycie aliasów (aliasy są podane na rysunku za pomocą nazw w kolorze czerwonym). Skrypt w nowej postaci powinien zawierać niezbędne komentarze.

- Czy można używać aliasów do komunikacji pomiędzy kontenerami przyłączonymi do dwóch różnych sieci ale pracujących w trybie mostu definiowanego przez użytkownika (np. pomiędzy host1 a host2)? Odpowiedź uzasadnij powołując się na sposób implementacji aliasów o zasięgu sieci.
- W przypadku pomyślnej konfiguracji w zadaniu 4.1 sieci wykorzystującej most bridge2 sprawdź czy możliwe jest korzystanie ze zdefiniowanych aliasów na hoście macierzystym przy połączeniach do kontenera S1 lub D2? Uzasadnij otrzymany rezultat.

```
Otwórz ▾ J+ jl_2022_03_24_lab4_zad2.sh

1 #!/bin/bash
2
3 #zatrzymaj kontenery
4 sudo docker container stop T1 T2 D1 D2 S1 late
5
6 #usun kontenery
7 sudo docker container rm T1 T2 D1 D2 S1 late
8
9 #usun sieci
10 sudo docker network rm bridge1
11 sudo docker network rm bridge2
12
13 #utworz sieci
14 sudo docker network create --subnet 10.0.10.0/24 bridge1
15 sudo docker network create --subnet 192.168.1.0/24 bridge2
16
17 #uruchom kontenery
18 sudo docker run -itd --name T1 alpine sh
19 sudo docker run -itd --name T2 --net bridge1 -p 10.0.10.1:80:8080 nginx sh
20 sudo docker run -itd --name D1 --net bridge1 --net-alias host1 --ip 10.0.10.254 alpine sh
21 sudo docker run -itd --name D2 httpd
22
23 sudo docker network connect --alias apa2 bridge2 D2
24 sudo docker network connect --alias apa1 bridge1 D2
25
26 sudo docker run -itd --name late ubuntu
27
28 sudo docker network connect bridge2 late
29 sudo docker network connect bridge1 late
30
31 sudo docker run -itd --name S1 --net bridge2 --net-alias host2 ubuntu
```

```
student@student-VirtualBox:~$ ./jl_2022_03_24_lab4_zad2.sh
T1
T2
D1
D2
S1
late
T1
T2
D1
D2
S1
late
bridge1
bridge2
295eddb9776d85d966c9d9ca20e0fa1019257d6931576becb8ac7845555353f4
ca9675dd7ac47aea49c87cbb0fb029040e048a083aa9e2c0e22108989ff71cea
933c5c5ff66900a6770f8e9032d58f249b1b318f4eb38071417e8671f2939682
fd2aa25245a5745d6f36f8ec819e81acb5aa7eaec0aa5657411dc3d81ca227a1
3473407677d8b1ce6288ffa2d7b9685ad9a792c2ba300903526505684e0fb9b9
8b2a5f8e92ef3c49c11ca1962b86ce9b458c281337170b8b976e69c672f55601
c5e1433b8511c0784e9fd16c6b3e375ddefc7d36ff1fd9036f9cf0c32473f83
75c3fe5a987a7522421374eb21d2c688041d4af9f75e4e6af2f6e18c36302406
```

```
student@student-VirtualBox:~$ sudo docker ps
CONTAINER ID        IMAGE       COMMAND           CREATED          STATUS          PORTS          NAMES
75c3fe5a987a        ubuntu      "bash"            36 seconds ago   Up 35 seconds
c5e1433b8511        ubuntu      "bash"            37 seconds ago   Up 36 seconds
8b2a5f8e92ef        httpd      "httpd-foreground" 38 seconds ago   Up 37 seconds   80/tcp
3473407677d8        alpine     "sh"              38 seconds ago   Up 38 seconds
fd2aa25245a5        nginx      "/docker-entrypoint...." 39 seconds ago   Up 38 seconds   80/tcp, 10.0.10.1:80->8080/tcp T2
933c5c5ff669        alpine     "sh"              39 seconds ago   Up 39 seconds   T1
```

```
student@student-VirtualBox:~$ sudo docker network inspect bridge1
[
    {
        "Name": "bridge1",
        "Id": "295eddb9776d85d966c9d9ca20e0fa1019257d6931576becb8ac7845555353f4",
        "Created": "2022-03-30T18:23:38.491628668+02:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "10.0.10.0/24"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "3473407677d8b1ce6288ffa2d7b9685ad9a792c2ba300903526505684e0fbbb9": {
                "Name": "D1",
                "EndpointID": "f0cfdd464d3d6e4671b9216d5475ac0edc6c852e55b3ab718d3f44d92992e43b",
                "MacAddress": "02:42:0a:00:0a:fe",
                "IPv4Address": "10.0.10.254/24",
                "IPv6Address": ""
            },
            "8b2a5f8e92ef3c49c11ca1962b86ce9b458c281337170b8b976e69c672f55601": {
                "Name": "D2",
                "EndpointID": "3b05c1a9d3c0225815975812dc2b84463a374758483d4a297a94ea66a7dba852",
                "MacAddress": "02:42:0a:00:0a:03",
                "IPv4Address": "10.0.10.3/24",
                "IPv6Address": ""
            },
            "c5e1433b8511c0784e9fd16c6b3e375ddefc7d36ff1fd9036f9cf0c32473f83": {
                "Name": "late",
                "EndpointID": "d291248e7df19d005ea7510126f1de33ac6d81f6a603a3129a853e149edb23f7",
                "MacAddress": "02:42:0a:00:0a:04",
                "IPv4Address": "10.0.10.4/24",
                "IPv6Address": ""
            },
            "fd2aa25245a5745d6f36f8ec819e81acb5aa7eaec0aa5657411dc3d81ca227a1": {
                "Name": "T2",
            }
        }
    }
]
```

```
student@student-VirtualBox:~$ sudo docker network inspect bridge2
[{"Name": "bridge2",
 "Id": "ca9675dd7ac47aea49c87ccb0fb029040e048a083aa9e2c0e22108989ff71cea",
 "Created": "2022-03-30T18:23:38.762480424+02:00",
 "Scope": "local",
 "Driver": "bridge",
 "EnableIPv6": false,
 "IPAM": {
   "Driver": "default",
   "Options": {},
   "Config": [
     {
       "Subnet": "192.168.1.0/24"
     }
   ]
 },
 "Internal": false,
 "Attachable": false,
 "Ingress": false,
 "ConfigFrom": {
   "Network": ""
 },
 "ConfigOnly": false,
 "Containers": {
   "75c3fe5a987a7522421374eb21d2c688041d4af9f75e4e6af2f6e18c36302406": {
     "Name": "S1",
     "EndpointID": "ec99b40c0a33228c88386bc3c7618709ae1786a078aa2bc2b3e0394d3110cf4b",
     "MacAddress": "02:42:c0:a8:01:04",
     "IPv4Address": "192.168.1.4/24",
     "IPv6Address": ""
   },
   "8b2a5f8e92ef3c49c11ca1962b86ce9b458c281337170b8b976e69c672f55601": {
     "Name": "D2",
     "EndpointID": "05937254199b2311bb888af492bd9ddda2a92f73b29d6c4d77ef0d74ecbcd3fc",
     "MacAddress": "02:42:c0:a8:01:02",
     "IPv4Address": "192.168.1.2/24",
     "IPv6Address": ""
   },
   "c5e1433b8511c0784e9fd16c6b3e375ddefc7d36ff1fd9036f9cf0c032473f83": {
     "Name": "late",
     "EndpointID": "06f4d69f58d968f7dd9a585db708ec39a3d64dca7293e8a57bc1226e0787aea0",
     "MacAddress": "02:42:c0:a8:01:03",
     "IPv4Address": "192.168.1.3/24",
     "IPv6Address": ""
   }
 }
}
```

**Czy można używać aliasów do komunikacji pomiędzy kontenerami przyłączonymi do dwóch różnych sieci ale pracujących w trybie mostu definiowanego przez użytkownika (np. pomiędzy host1 a host2)? Odpowiedź uzasadnij powołując się na sposób implementacji aliasów o zasięgu sieci.**

Nie można używać aliasów do komunikacji pomiędzy kontenerami przyłączonymi do dwóch różnych sieci ale pracujących w trybie mostu definiowanego przez użytkownika.

Aliases, w przeciwieństwie do nazw kontenerów, definiowane są w obrębie danej sieci. Jeśli jeden kontener chciałby skomunikować się z drugim kontenerem o pewnym aliasie za pomocą tego aliasu, musiałby wysłać polecenie do swojej sieci, a ta sieć nie zna tego aliasu lub używa go dla innego kontenera.