Task Problem 1 :

1 - Create bridge network with subnet 192.168.0.0/24.

2 - Run 2 containers and attach containers to this network.

3 - Create another bridge network with subnet 10.5.0.0/24.

4 - Run any container and attach it to the new network.

5 - Make sure that the containers at different network can't ping each other

Solutions :

1 – check the currently network

2 – I will create 2 network [n1 – n2]

A – n1 with subnet 192.168.0.0/24 and driver bridge type

B – n2 with subnet 10.5.0.0/24 and driver bridge type

3 – check the currently network again to make sure that I create two network successfully

4 – create 3 containers names [first_con – second_con – third_con] using nginx image

A – connect first_con and second_con to network n1

B – connect third_con to network n2

5 – inspect network n1 and inspect network n2 to make sure that the containers connected to them

6 – I will use first_con container as a tester ping for the other two conatiners and install ping on it

7 –I will ping from first_con to second_con >> both on same network n1

8 – finally , I will ping from fist_con to third_con >> both on different network n1,n2

The Results :

Solution Step 1

```
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
1d077082d916   bridge    bridge    local
9bc7c7df79aa   host      host      local
51e0f5bcd2d4   none      null      local
```

Solution Step 2

```
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker network create --driver=bridge --subnet=192.168.0.0/24 n1
cf88b66914429814e7de7d8c45a5e854cda83e44670f73607ab2831cde80e1fd
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker network create --driver=bridge --subnet=10.5.0.0/24 n2
5bc21a7374b5ba80f49b635b93f2d6340c0fbf7ae6789a8a9e03dd5941b2b6ce
```

Solution Step 3

```
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
1d077082d916   bridge    bridge    local
9bc7c7df79aa   host      host      local
cf88b6691442   n1        bridge    local
5bc21a7374b5   n2        bridge    local
51e0f5bcd2d4   none      null      local
```

Solution Step 4

```
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker run -d --network n1 --name first_con nginx
c6366ad0e66915f6556d49890c6962789102bdb4c7571b6be45ac32c23432f09
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker run -d --network n1 --name second_con nginx
23fde53f013adf1548c5ae819247b12643d5d1b61958e8ce9c592d1e024b588d
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker run -d --network n2 --name third_con nginx
bcf833bb80e8c1993c113fdf370baf0130700c506ee06988ceb646a167cfed2e
```

Solution Step 5 for n1

```
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker network inspect n1
[
    {
        "Name": "n1",
        "Id": "cf88b66914429814e7de7d8c45a5e854cda83e44670f73607ab2831cde80e1fd",
        "Created": "2023-05-30T12:24:51.340163833+03:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "192.168.0.0/24"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "23fde53f013adf1548c5ae819247b12643d5d1b61958e8ce9c592d1e024b588d": {
                "Name": "second_con",
                "EndpointID": "51485ba45528b67c02f7415099287655bcbbfb66744659207a8d47dcb1277efb",
                "MacAddress": "02:42:c0:a8:00:03",
                "IPv4Address": "192.168.0.3/24",
                "IPv6Address": ""
            },
            "c6366ad0e66915f6556d49890c6962789102bdb4c7571b6be45ac32c23432f09": {
                "Name": "first_con",
                "EndpointID": "896f14ca7f5426e1ac95a02d8e34d877ec491996fc3d916784df06671122ed66",
                "MacAddress": "02:42:c0:a8:00:02",
                "IPv4Address": "192.168.0.2/24",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

Solution Step 5 for n2

```
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker network inspect n2
[
    {
        "Name": "n2",
        "Id": "5bc21a7374b5ba80f49b635b93f2d6340c0fbf7ae6789a8a9e03dd5941b2b6ce",
        "Created": "2023-05-30T12:25:03.825209138+03:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "10.5.0.0/24"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "bcf833bb80e8c1993c113fdf370baf0130700c506ee06988ceb646a167cfed2e": {
                "Name": "third_con",
                "EndpointID": "e2cac2d342d40aa8bbd4e6590b89247bbb5315be754484117866bc5a94f31c77",
                "MacAddress": "02:42:0a:05:00:02",
                "IPv4Address": "10.5.0.2/24",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

Solution Step 6

```
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker exec -it first_con bash
root@c6366ad0e669:/# sudo apt update && sudo apt upgrade
bash: sudo: command not found
root@c6366ad0e669:/# apt update && apt upgrade
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:2 http://deb.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8183 kB]
Get:5 http://deb.debian.org/debian-security bullseye-security/main amd64 Packages [245 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [14.8 kB]
Fetched 8650 kB in 6s (1478 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@c6366ad0e669:/# apt install iputils-ping
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcap2 libcap2-bin libpam-cap
The following NEW packages will be installed:
  iputils-ping libcap2 libcap2-bin libpam-cap
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 121 kB of archives.
After this operation, 348 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://deb.debian.org/debian bullseye/main amd64 libcap2 amd64 1:2.44-1 [23.6 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 libcap2-bin amd64 1:2.44-1 [32.6 kB]
```

Solution Step 7

```
debconf: falling back to frontend: Teletype
Setting up iputils-ping (3:20210202-1) ...
Processing triggers for libc-bin (2.31-13+deb11u6) ...
root@c6366ad0e669:/# exit
exit
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker exec -it first_con ping second_con
PING second_con (192.168.0.3) 56(84) bytes of data.
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=1 ttl=64 time=0.080 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=2 ttl=64 time=0.066 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=3 ttl=64 time=0.062 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=4 ttl=64 time=0.059 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=5 ttl=64 time=0.063 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=6 ttl=64 time=0.062 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=7 ttl=64 time=0.058 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=8 ttl=64 time=0.060 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=9 ttl=64 time=0.060 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=10 ttl=64 time=0.079 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=11 ttl=64 time=0.059 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=12 ttl=64 time=0.065 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=13 ttl=64 time=0.058 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=14 ttl=64 time=0.055 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=15 ttl=64 time=0.054 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=16 ttl=64 time=0.061 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=17 ttl=64 time=0.061 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=18 ttl=64 time=0.086 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=19 ttl=64 time=0.060 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=20 ttl=64 time=0.054 ms
64 bytes from second_con.n1 (192.168.0.3): icmp_seq=21 ttl=64 time=0.088 ms
^C
--- second_con ping statistics ---
21 packets transmitted, 21 received, 0% packet loss, time 20354ms
rtt min/avg/max/mdev = 0.054/0.064/0.088/0.009 ms
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ ^C
```

Solution Step 8

```
amr@amr-VirtualBox:~/Docker_Tasks/Last_Task$ docker exec -it first_con ping third_con
ping: third_con: Temporary failure in name resolution
```