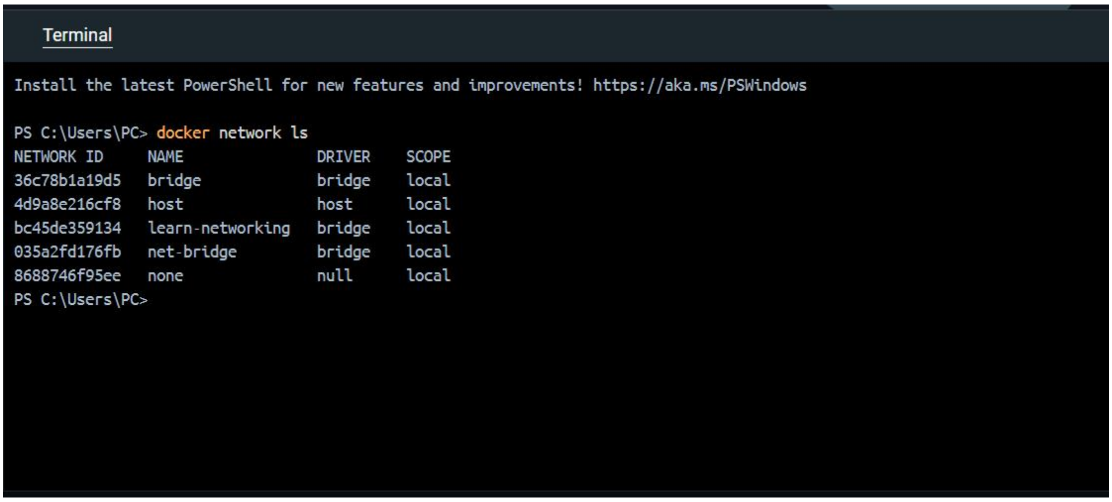
Name : Vanshika singh
Batch : 02
Sap'id : 500107391

LAB 6 :: List All Networks Command:

docker network ls

Usage: This command lists all Docker networks available on the system



```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\PC> docker network ls
NETWORK ID     NAME               DRIVER    SCOPE
36c78b1a19d5   bridge             bridge    local
4d9a8e216cf8   host               host      local
bc45de359134   learn-networking   bridge    local
035a2fd176fb   net-bridge         bridge    local
8688746f95ee   none               null      local
PS C:\Users\PC>
```

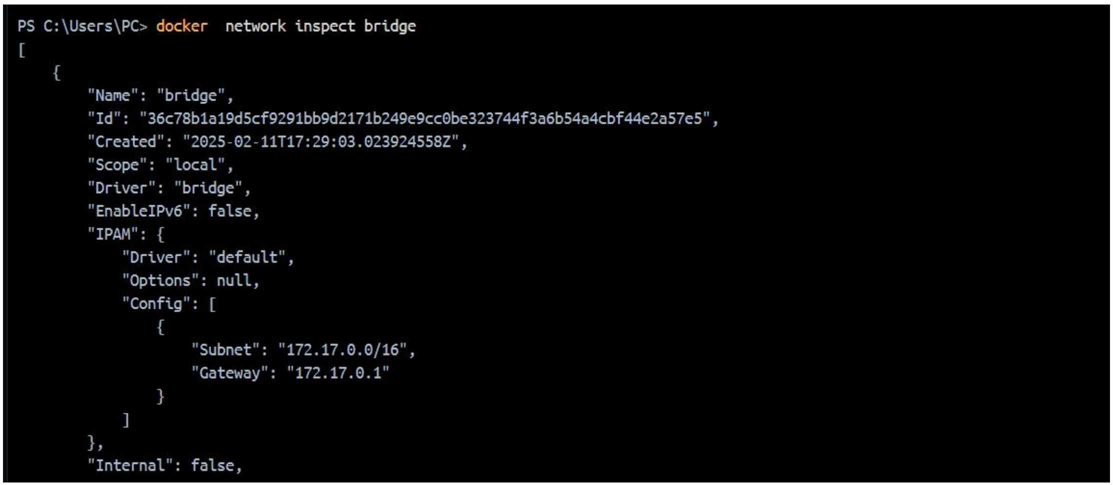
By default,
Docker comes with three networks:

- bridge: Default network for containers.
- host: Shares the host's network.
- none: Disables networking

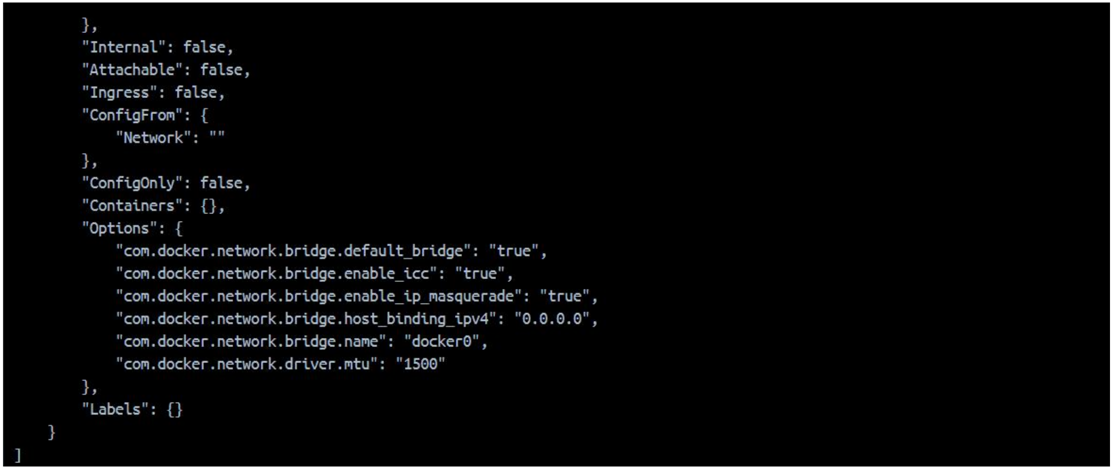
2. Inspect a Network

Command: docker network inspect

Usage: Provides detailed information about a network, including connected containers and their IP addresses.



```
PS C:\Users\PC> docker  network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "36c78b1a19d5cf9291bb9d2171b249e9cc0be323744f3a6b54a4cbf44e2a57e5",
        "Created": "2025-02-11T17:29:03.023924558Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
```



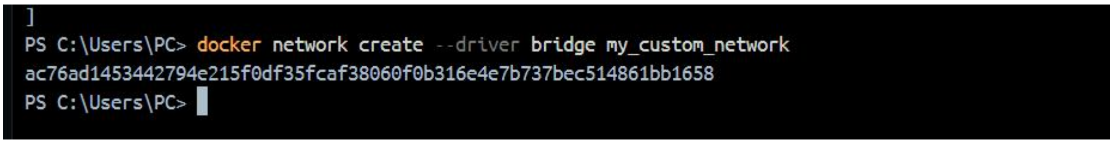
```
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
            "com.docker.network.bridge.name": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
        "Labels": {}
    }
]
```

3. Create a New Network Command:

docker network create --driver

Usage: Creates a new network. The most common drivers are:

- bridge (default)
- host
- Overlay

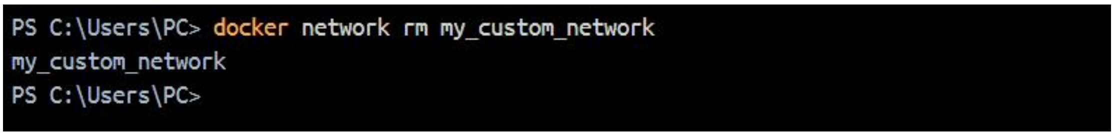


```
]
PS C:\Users\PC> docker network create --driver bridge my_custom_network
ac76ad1453442794e215f0df35fcaf38060f0b316e4e7b737bec514861bb1658
PS C:\Users\PC> 
```

4. Remove a Network

Command: docker network rm

Usage: Deletes a Docker network

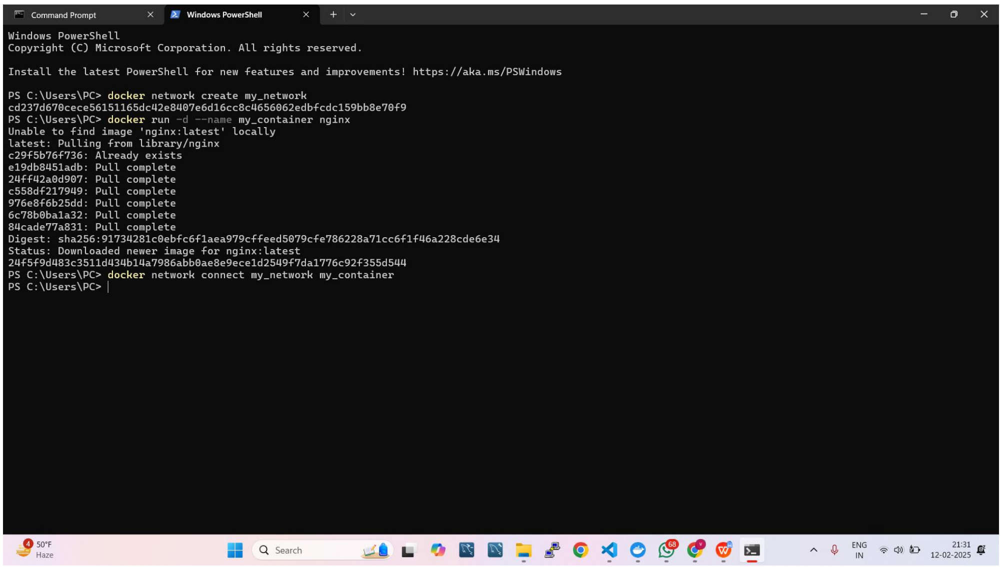


```
PS C:\Users\PC> docker network rm my_custom_network
my_custom_network
PS C:\Users\PC>
```

5. Connect a Container to a Network

Command: docker network connect

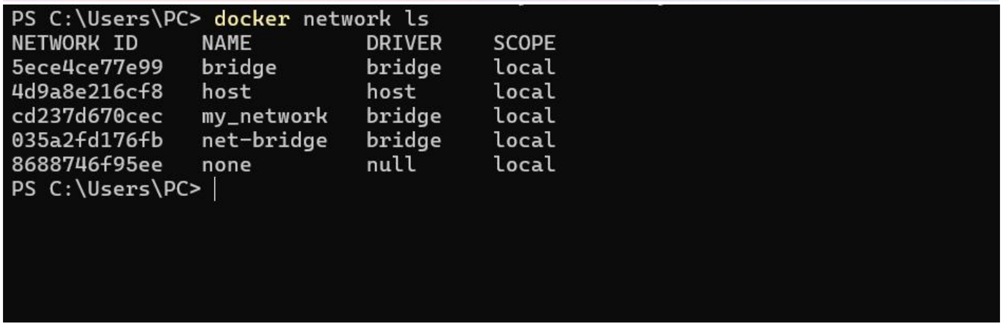
Usage: Adds an existing container to a specific network.



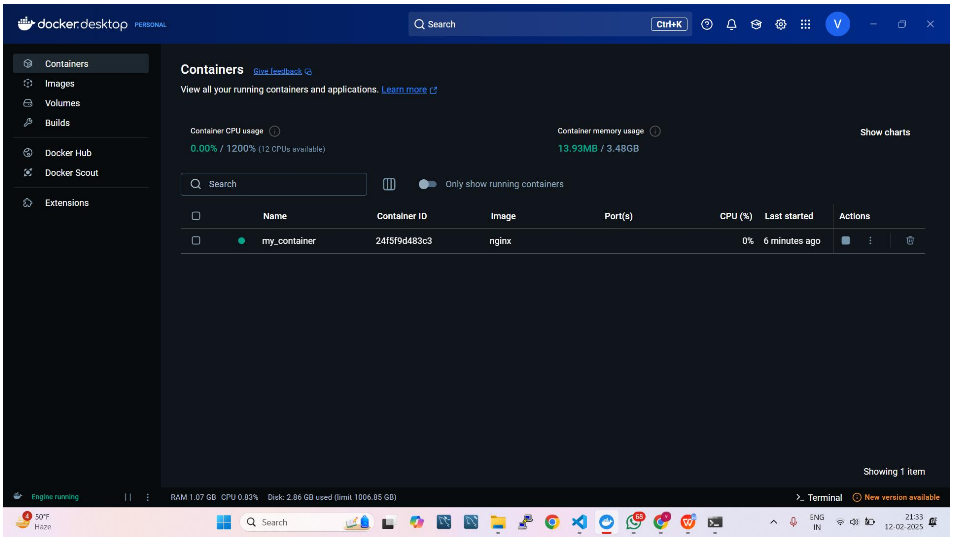
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\PC> docker network create my_network
cd237d670cece56151165dc42e8407e6d16cc8c4656062edbfcdc159bb8e70f9
PS C:\Users\PC> docker run -d --name my_container nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
c29f5b76f736: Already exists
e19db8451adb: Pull complete
24ff42a0d907: Pull complete
c558df217949: Pull complete
976e8f6b25dd: Pull complete
6c78b0ba1a32: Pull complete
84cade77a831: Pull complete
Digest: sha256:91734281c0ebfc6f1aea979cffeed5079cfe786228a71cc6f1f46a228cde6e34
Status: Downloaded newer image for nginx:latest
24f5f9d483c3511d434b14a7986abb0ae8e9ece1d2549f7da1776c92f355d544
PS C:\Users\PC> docker network connect my_network my_container
PS C:\Users\PC>
```



```
PS C:\Users\PC> docker network ls
NETWORK ID     NAME         DRIVER    SCOPE
5ece4ce77e99   bridge       bridge    local
4d9a8e216cf8   host         host      local
cd237d670cec   my_network   bridge    local
035a2fd176fb   net-bridge   bridge    local
8688746f95ee   none         null      local
PS C:\Users\PC>
```



Containers Give feedback

View all your running containers and applications. Learn more

Container CPU usage 0.00% / 1200% (12 CPUs available)

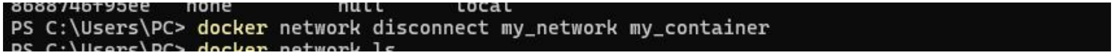
Container memory usage 13.93MB / 3.48GB

□		Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
□	●	my_container	24f5f9d483c3	nginx		0%	6 minutes ago	

Showing 1 item

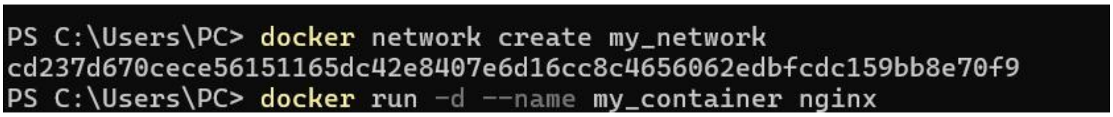
RAM 1.07 GB CPU 0.83% Disk: 2.86 GB used (limit 1006.85 GB)

6. Disconnect a Container from a Network
Command: docker network disconnect
Usage: Removes a container from a network.



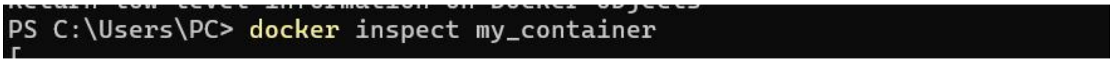
```
8688746f95ee   none     null     local
PS C:\Users\PC> docker network disconnect my_network my_container
PS C:\Users\PC> docker network ls
```

7.Run a Container with a Specific Network
Command: docker run --network= -d
Usage: Starts a container with a specified network.

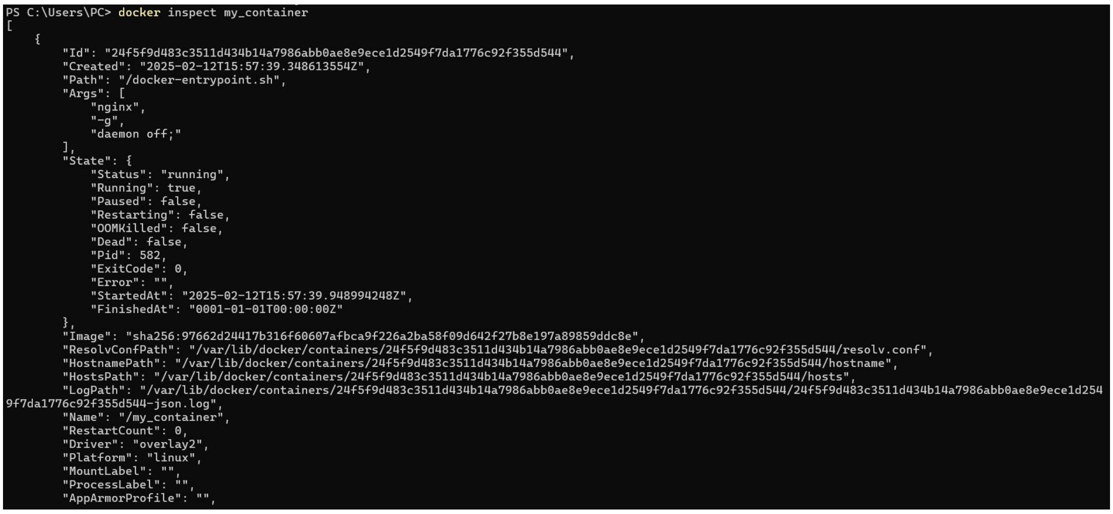


```
PS C:\Users\PC> docker network create my_network
cd237d670cece56151165dc42e8407e6d16cc8c4656062edbfcdc159bb8e70f9
PS C:\Users\PC> docker run -d --name my_container nginx
```

8. Check a Container’s Network Configuration
Command: docker inspect
Usage: Displays the network details of a container, including IP addresses.

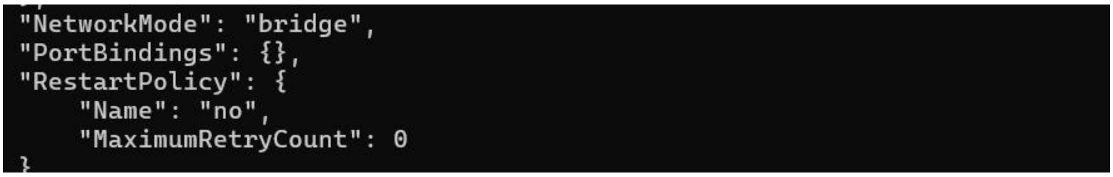


```
Return low-level information on Docker objects
PS C:\Users\PC> docker inspect my_container
[
```



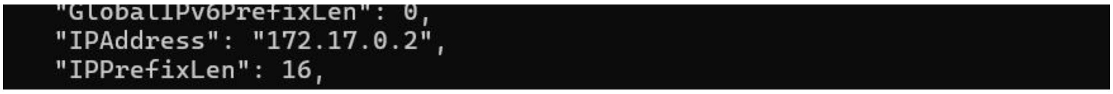
```
PS C:\Users\PC> docker inspect my_container
[
    {
        "Id": "24f5f9d483c3511d434b14a7986abb0ae8e9ece1d2549f7da1776c92f355d544",
        "Created": "2025-02-12T15:57:39.348613554Z",
        "Path": "/docker-entrypoint.sh",
        "Args": [
            "nginx",
            "-g",
            "daemon off;"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 582,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2025-02-12T15:57:39.948994248Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:97662d24417b316f60607afbca9f226a2ba58f09d642f27b8e197a89859ddc8e",
        "ResolvConfPath": "/var/lib/docker/containers/24f5f9d483c3511d434b14a7986abb0ae8e9ece1d2549f7da1776c92f355d544/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/24f5f9d483c3511d434b14a7986abb0ae8e9ece1d2549f7da1776c92f355d544/hostname",
        "HostsPath": "/var/lib/docker/containers/24f5f9d483c3511d434b14a7986abb0ae8e9ece1d2549f7da1776c92f355d544/hosts",
        "LogPath": "/var/lib/docker/containers/24f5f9d483c3511d434b14a7986abb0ae8e9ece1d2549f7da1776c92f355d544/24f5f9d483c3511d434b14a7986abb0ae8e9ece1d254
9f7da1776c92f355d544-json.log",
        "Name": "/my_container",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "",
```

NetworkMode::



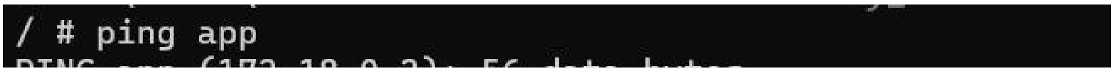
```
"NetworkMode": "bridge",
"PortBindings": {},
"RestartPolicy": {
    "Name": "no",
    "MaximumRetryCount": 0
}
```

IPAddress::

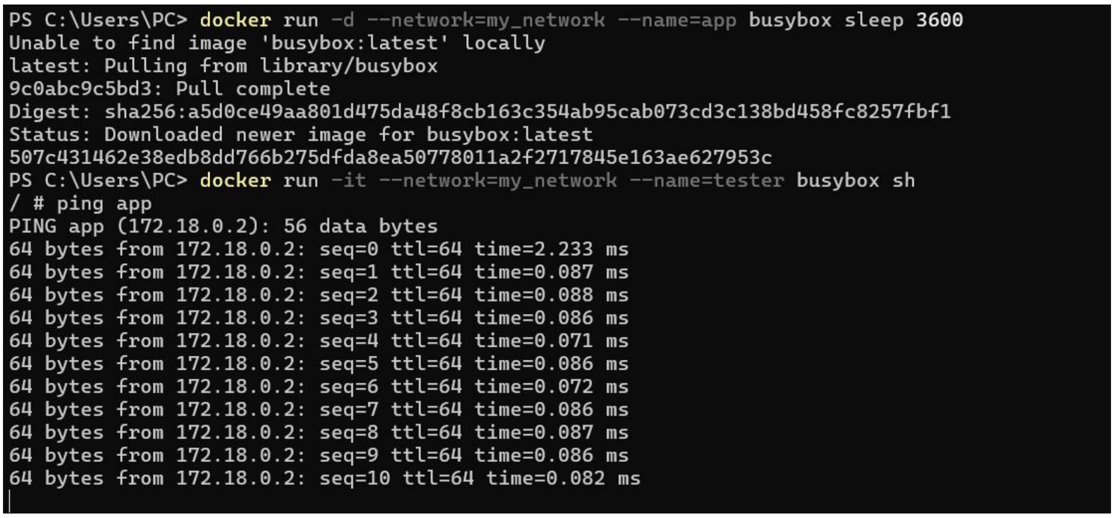


```
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "172.17.0.2",
  "IPPrefixLen": 16,
```

9. Use Docker DNS for Container Communication
Command: ping
Usage: When containers are in the same Docker network, they can communicate using container names instead of IPs.

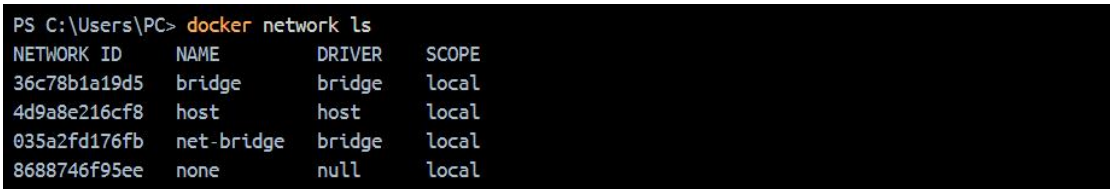


```
/ # ping app
```



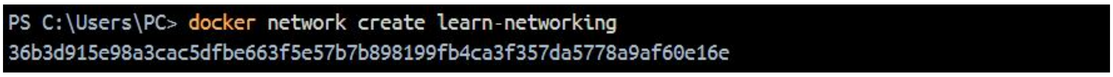
```
PS C:\Users\PC> docker run -d --network=my_network --name=app busybox sleep 3600
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
9c0abc9c5bd3: Pull complete
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Downloaded newer image for busybox:latest
507c431462e38edb8dd766b275dfda8ea50778011a2f2717845e163ae627953c
PS C:\Users\PC> docker run -it --network=my_network --name=tester busybox sh
/ # ping app
PING app (172.18.0.2): 56 data bytes
64 bytes from 172.18.0.2: seq=0 ttl=64 time=2.233 ms
64 bytes from 172.18.0.2: seq=1 ttl=64 time=0.087 ms
64 bytes from 172.18.0.2: seq=2 ttl=64 time=0.088 ms
64 bytes from 172.18.0.2: seq=3 ttl=64 time=0.086 ms
64 bytes from 172.18.0.2: seq=4 ttl=64 time=0.071 ms
64 bytes from 172.18.0.2: seq=5 ttl=64 time=0.086 ms
64 bytes from 172.18.0.2: seq=6 ttl=64 time=0.072 ms
64 bytes from 172.18.0.2: seq=7 ttl=64 time=0.086 ms
64 bytes from 172.18.0.2: seq=8 ttl=64 time=0.087 ms
64 bytes from 172.18.0.2: seq=9 ttl=64 time=0.086 ms
64 bytes from 172.18.0.2: seq=10 ttl=64 time=0.082 ms
```

Use Case :



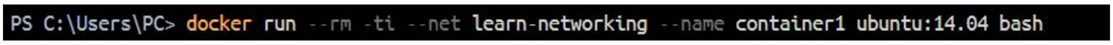
```
PS C:\Users\PC> docker network ls
NETWORK ID     NAME         DRIVER    SCOPE
36c78b1a19d5   bridge       bridge    local
4d9a8e216cf8   host         host      local
035a2fd176fb   net-bridge   bridge    local
8688746f95ee   none         null      local
```

Step 1: Create a Docker Network:



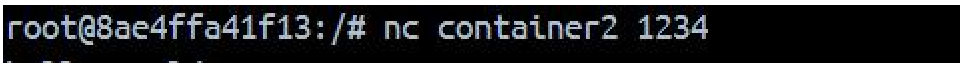
```
PS C:\Users\PC> docker network create learn-networking
36b3d915e98a3cac5dfbe663f5e57b7b898199fb4ca3f357da5778a9af60e16e
```

Step 2: Start the First Ubuntu Container (Server):



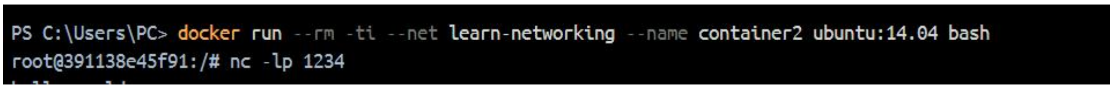
```
PS C:\Users\PC> docker run --rm -ti --net learn-networking --name container1 ubuntu:14.04 bash
```

Start a Netcat listener on port 1234:

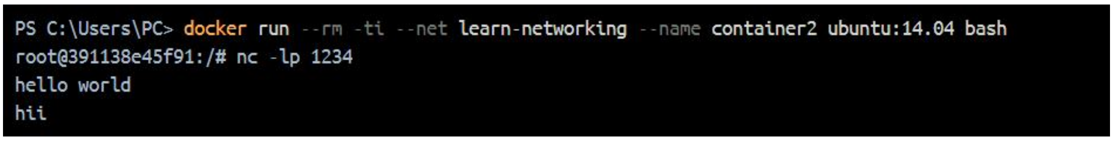


```
root@8ae4ffa41f13:/# nc container2 1234
```

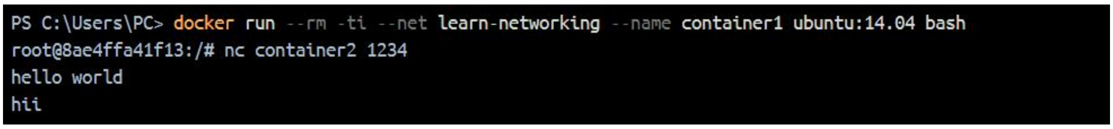
Step 3: Start the Second Ubuntu Container (Client)



```
PS C:\Users\PC> docker run --rm -ti --net learn-networking --name container2 ubuntu:14.04 bash
root@391138e45f91:/# nc -lp 1234
```

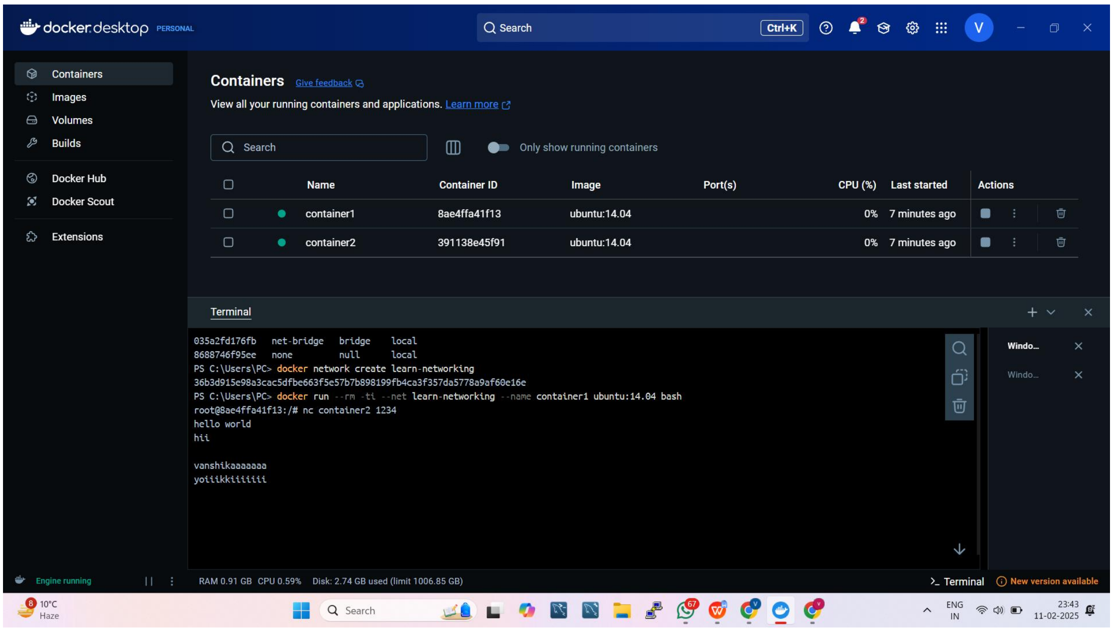


```
PS C:\Users\PC> docker run --rm -ti --net learn-networking --name container2 ubuntu:14.04 bash
root@391138e45f91:/# nc -lp 1234
hello world
hii
```



```
PS C:\Users\PC> docker run --rm -ti --net learn-networking --name container1 ubuntu:14.04 bash
root@8ae4ffa41f13:/# nc container2 1234
hello world
hii
```

Step 4: Verify the Message on the Server::



Containers

View all your running containers and applications. Learn more

Only show running containers

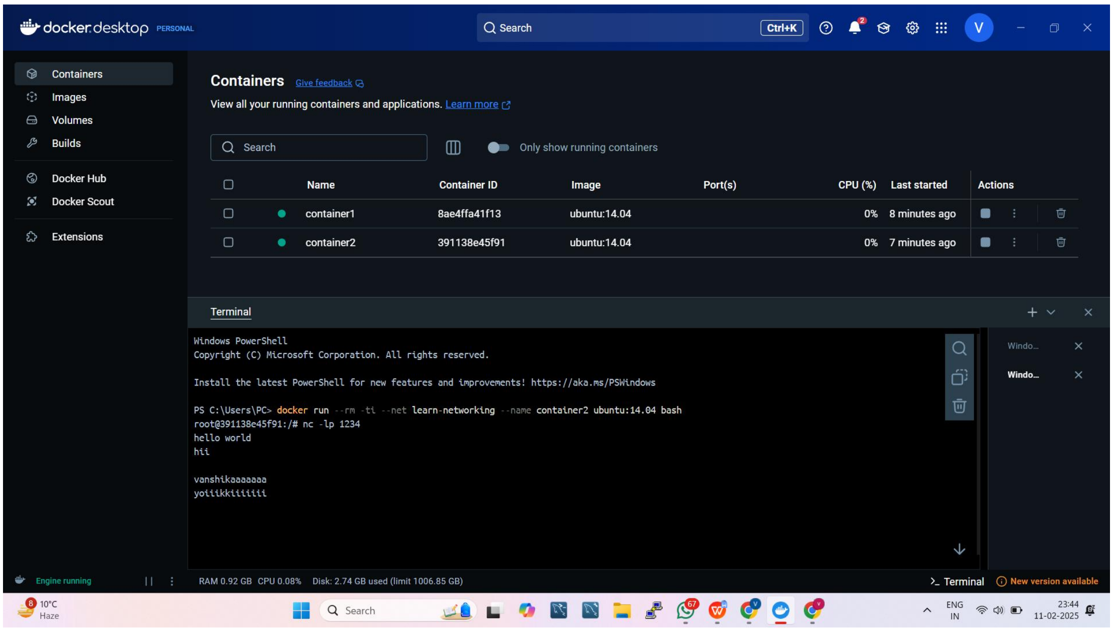
	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
□	● container1	8ae4ffa41f13	ubuntu:14.04		0%	7 minutes ago	
□	● container2	391138e45f91	ubuntu:14.04		0%	7 minutes ago	

Terminal

```
035a2fd176fb   net-bridge   bridge    local
8688746f95ee   none         null      local
PS C:\Users\PC> docker network create learn-networking
36b3d915e98a3cac5dfbe663f5e57b7b898199fb4ca3f357da5778a9af60e16e
PS C:\Users\PC> docker run --rm -ti --net learn-networking --name container1 ubuntu:14.04 bash
root@8ae4ffa41f13:/# nc container2 1234
hello world
hii

vanshikaaaaaaa
yoiiikkiiiiiii
```

RAM 0.91 GB CPU 0.59% Disk: 2.74 GB used (limit 1006.85 GB)



Containers

View all your running containers and applications. Learn more

Only show running containers

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
□	● container1	8ae4ffa41f13	ubuntu:14.04		0%	8 minutes ago	
□	● container2	391138e45f91	ubuntu:14.04		0%	7 minutes ago	

Terminal

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\PC> docker run --rm -ti --net learn-networking --name container2 ubuntu:14.04 bash
root@391138e45f91:/# nc -lp 1234
hello world
hii

vanshikaaaaaaa
yoiiikkiiiiiii
```

RAM 0.92 GB CPU 0.08% Disk: 2.74 GB used (limit 1006.85 GB)