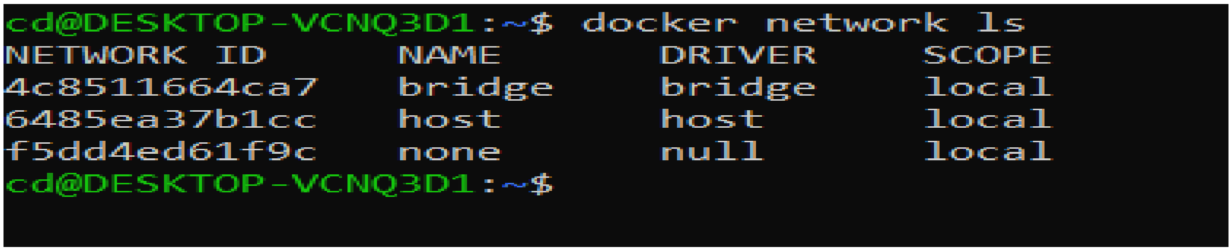
Networking

Listing networks

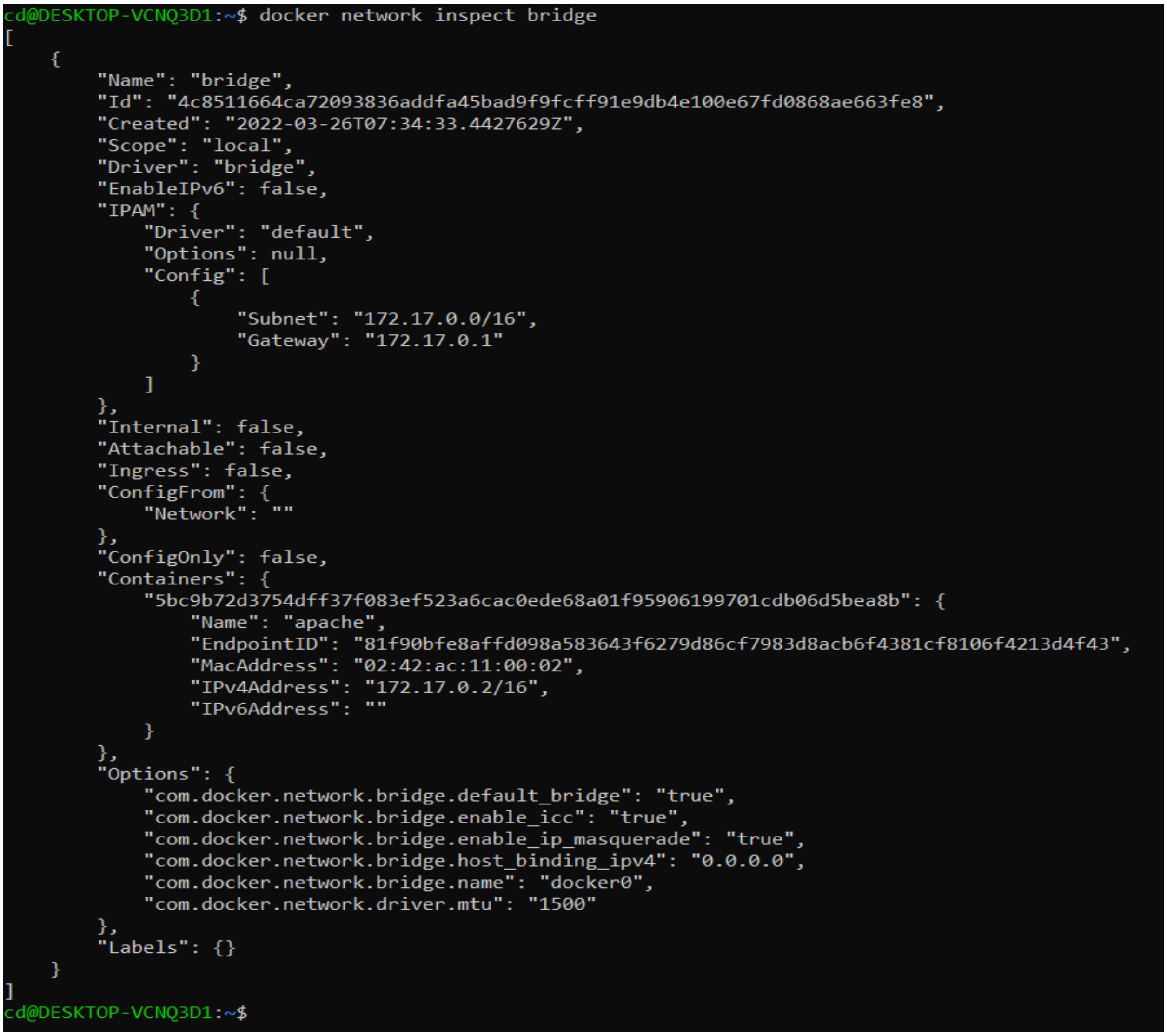
$ docker network ls



```
cd@DESKTOP-VCNQ3D1:~$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
4c8511664ca7   bridge    bridge    local
6485ea37b1cc   host      host      local
f5dd4ed61f9c   none      null      local
cd@DESKTOP-VCNQ3D1:~$
```

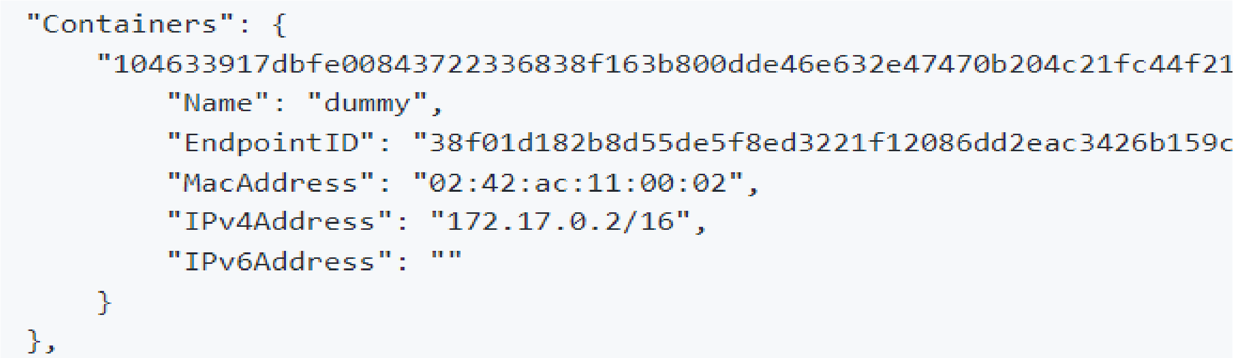
The default bridge network

$ docker network inspect bridge



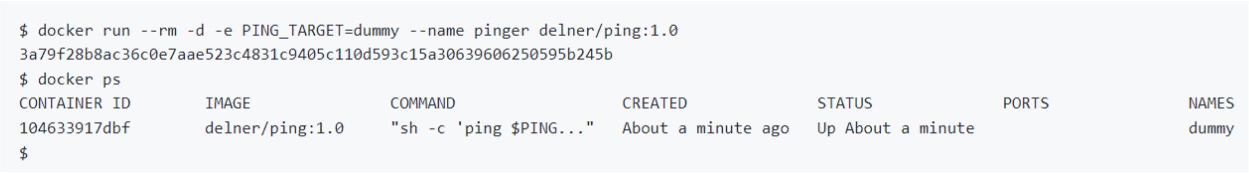
```
cd@DESKTOP-VCNQ3D1:~$ docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "4c8511664ca72093836addfa45bad9f9fcff91e9db4e100e67fd0868ae663fe8",
        "Created": "2022-03-26T07:34:33.4427629Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "5bc9b72d3754dff37f083ef523a6cac0ede68a01f95906199701cdb06d5bea8b": {
                "Name": "apache",
                "EndpointID": "81f90bfe8affd098a583643f6279d86cf7983d8acb6f4381cf8106f4213d4f43",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "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": {}
    }
]
cd@DESKTOP-VCNQ3D1:~$
```

$ docker run –rm –d –name dummy delner/ping:1.0



```
"Containers": {
    "104633917dbfe00843722336838f163b800dde46e632e47470b204c21fc44f21
        "Name": "dummy",
        "EndpointID": "38f01d182b8d55de5f8ed3221f12086dd2eac3426b159c
        "MacAddress": "02:42:ac:11:00:02",
        "IPv4Address": "172.17.0.2/16",
        "IPv6Address": ""
    }
},
```

$ docker run –rm –d –e PING_TARGET=172.17.0.2 –name pinger delner/ping:1.0

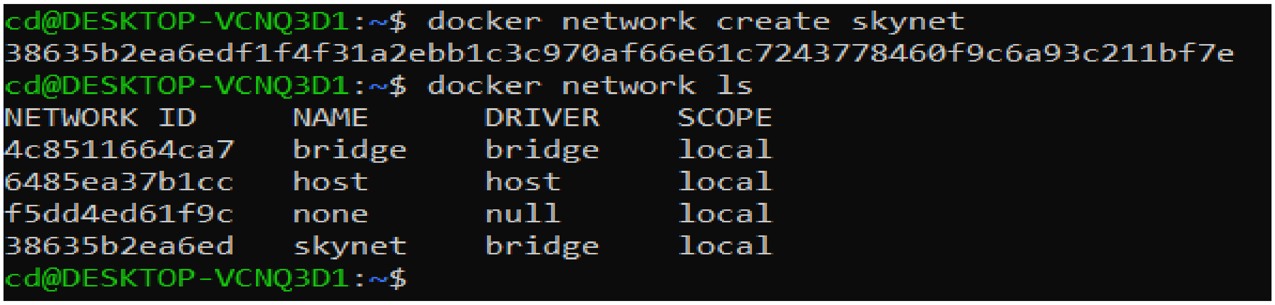


```
$ docker run --rm -d -e PING_TARGET=dummy --name pinger delner/ping:1.0
3a79f28b8ac36c0e7aae523c4831c9405c110d593c15a30639606250595b245b
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS              PORTS               NAMES
104633917dbf        delner/ping:1.0     "sh -c 'ping $PING..."   About a minute ago   Up About a minute                       dummy
$
```

Managing custom networks

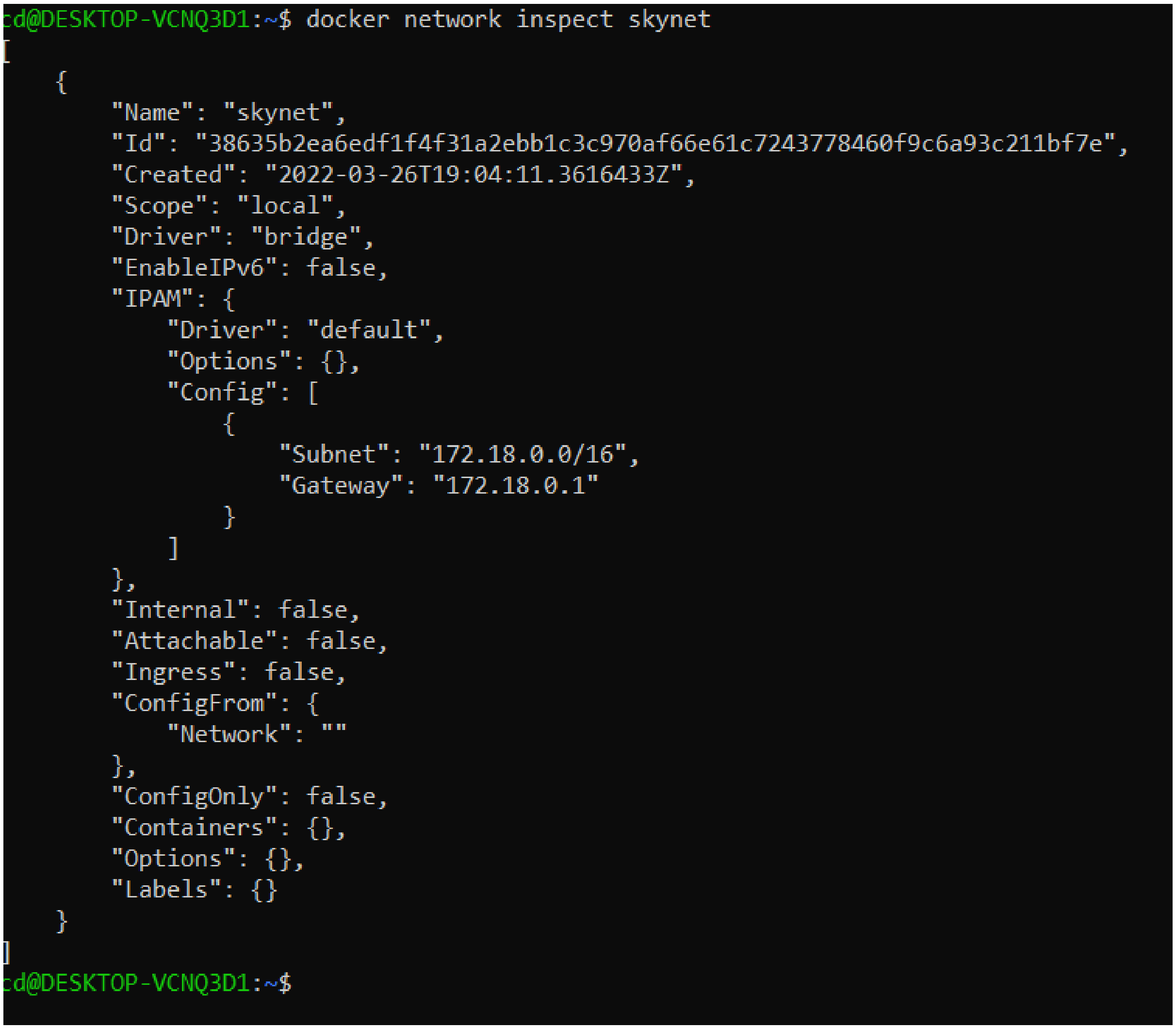
$ docker network create skynet

$ docker network ls



```
cd@DESKTOP-VCNQ3D1:~$ docker network create skynet
38635b2ea6edf1f4f31a2ebb1c3c970af66e61c7243778460f9c6a93c211bf7e
cd@DESKTOP-VCNQ3D1:~$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
4c8511664ca7   bridge    bridge    local
6485ea37b1cc   host      host      local
f5dd4ed61f9c   none      null      local
38635b2ea6ed   skynet    bridge    local
cd@DESKTOP-VCNQ3D1:~$
```

$ docker network inspect skynet

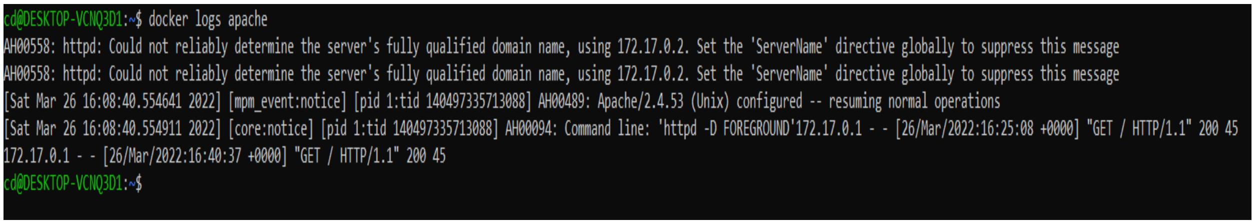


```
cd@DESKTOP-VCNQ3D1:~$ docker network inspect skynet
[
    {
        "Name": "skynet",
        "Id": "38635b2ea6edf1f4f31a2ebb1c3c970af66e61c7243778460f9c6a93c211bf7e",
        "Created": "2022-03-26T19:04:11.3616433Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {},
        "Labels": {}
    }
]
cd@DESKTOP-VCNQ3D1:~$
```

Adding containers to a network

$ docker run --rm -d --network skynet --name dummy delner/ping:1.0

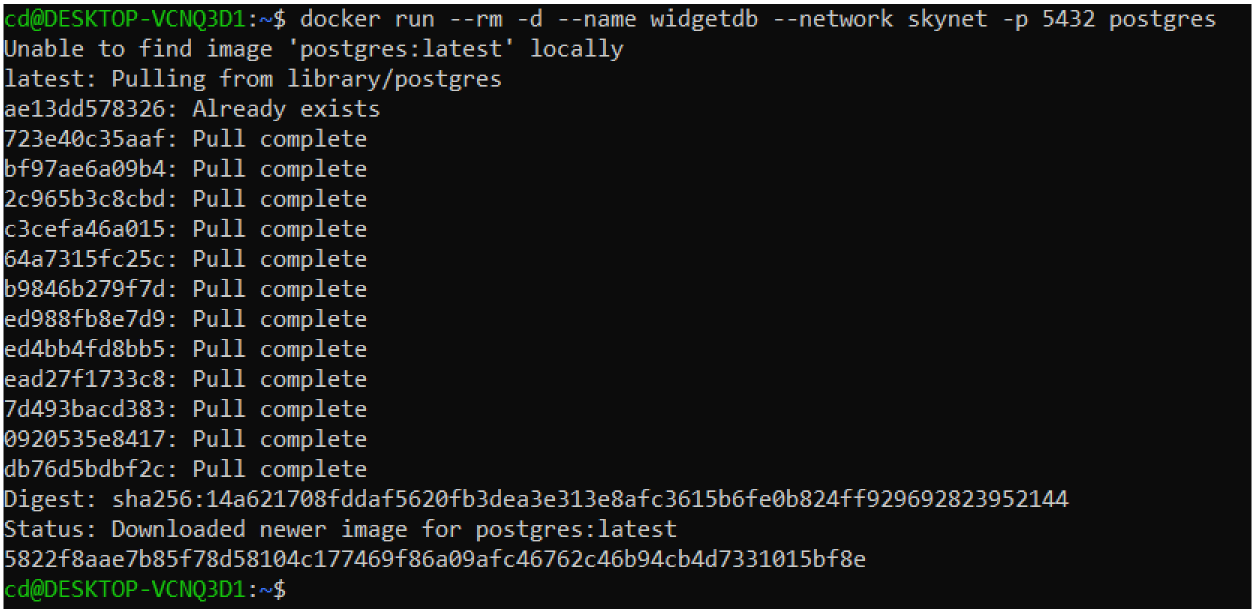
$ docker run –rm –d –network skynet –e PING_TARGET=dummy –name pinger delner/ping:1.0



```
cd@DESKTOP-VCNQ3D1:~$ docker logs apache
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[Sat Mar 26 16:08:40.554641 2022] [mpm_event:notice] [pid 1:tid 140497335713088] AH00489: Apache/2.4.53 (Unix) configured -- resuming normal operations
[Sat Mar 26 16:08:40.554911 2022] [core:notice] [pid 1:tid 140497335713088] AH00094: Command line: 'httpd -D FOREGROUND'172.17.0.1 - - [26/Mar/2022:16:25:08 +0000] "GET / HTTP/1.1" 200 45
172.17.0.1 - - [26/Mar/2022:16:40:37 +0000] "GET / HTTP/1.1" 200 45
cd@DESKTOP-VCNQ3D1:~$
```

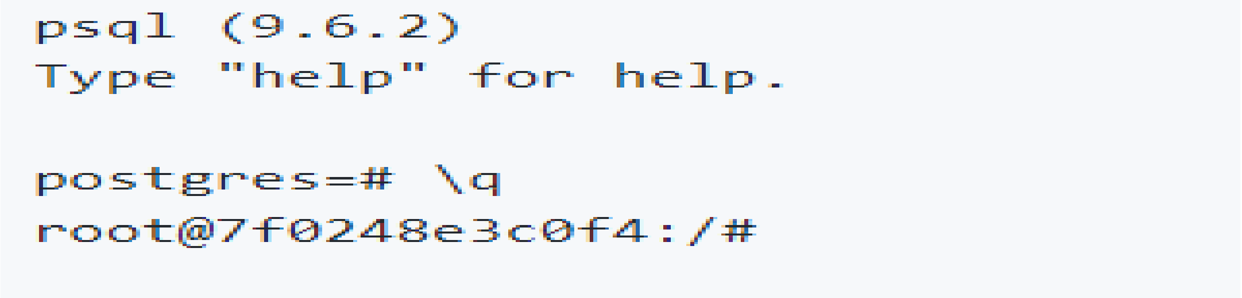
Connecting between containers in a network

$ docker run --rm -d --name widgetdb --network skynet -p 5432 postgres



```
cd@DESKTOP-VCNQ3D1:~$ docker run --rm -d --name widgetdb --network skynet -p 5432 postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
ae13dd578326: Already exists
723e40c35aaf: Pull complete
bf97ae6a09b4: Pull complete
2c965b3c8cbd: Pull complete
c3cefa46a015: Pull complete
64a7315fc25c: Pull complete
b9846b279f7d: Pull complete
ed988fb8e7d9: Pull complete
ed4bb4fd8bb5: Pull complete
ead27f1733c8: Pull complete
7d493bacd383: Pull complete
0920535e8417: Pull complete
db76d5bdbf2c: Pull complete
Digest: sha256:14a621708fddaf5620fb3dea3e313e8afc3615b6fe0b824ff929692823952144
Status: Downloaded newer image for postgres:latest
5822f8aae7b85f78d58104c177469f86a09afc46762c46b94cb4d7331015bf8e
cd@DESKTOP-VCNQ3D1:~$
```

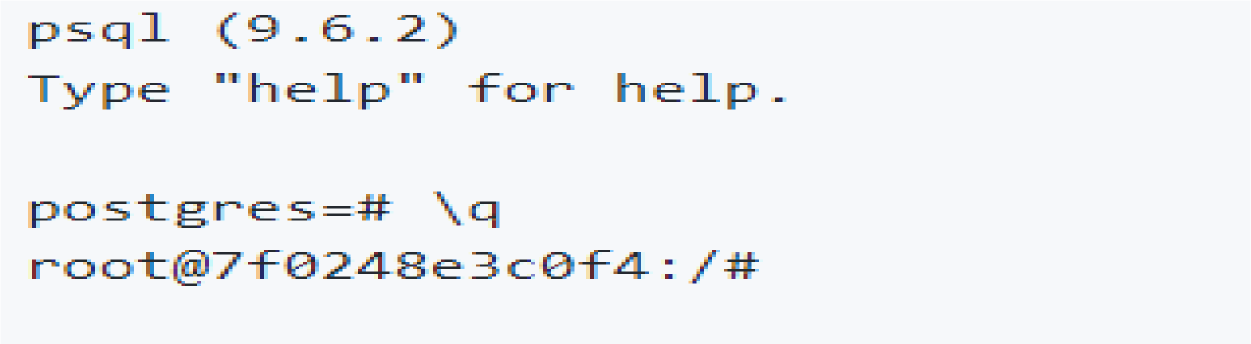
root@7f0248e3c0f4:/# psql -U postgres



```
psql (9.6.2)
Type "help" for help.

postgres=# \q
root@7f0248e3c0f4:/#
```

root@7f0248e3c0f4:/# psql -U postgres -h gadgetdb

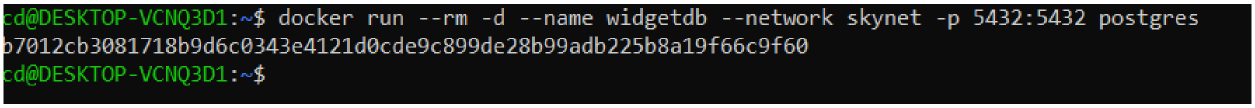


```
psql (9.6.2)
Type "help" for help.

postgres=# \q
root@7f0248e3c0f4:/#
```

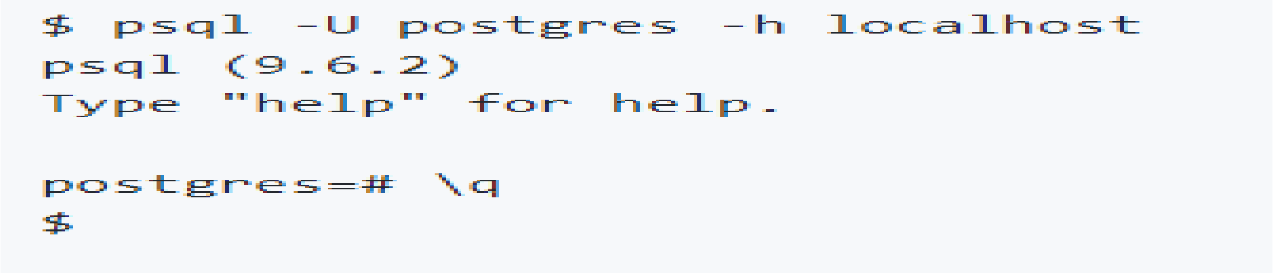
Binding ports to the host

$ docker run --rm -d --name widgetdb --network skynet -p 5432:5432 postgres



```
cd@DESKTOP-VCNQ3D1:~$ docker run --rm -d --name widgetdb --network skynet -p 5432:5432 postgres
b7012cb3081718b9d6c0343e4121d0cde9c899de28b99adb225b8a19f66c9f60
cd@DESKTOP-VCNQ3D1:~$
```

$ psql -U postgres -h localhost



```
$ psql -U postgres -h localhost
psql (9.6.2)
Type "help" for help.

postgres=# \q
$
```

$ Docker stop widgetdb