Lesson 07 Demo 06

Connecting Docker with SSHs for Networking

Objective: To set up Docker networking and SSH integration for seamless communication and connectivity in a containerized environment

Tools required: Docker configuration

Prerequisites: Ubuntu configuration and Docker

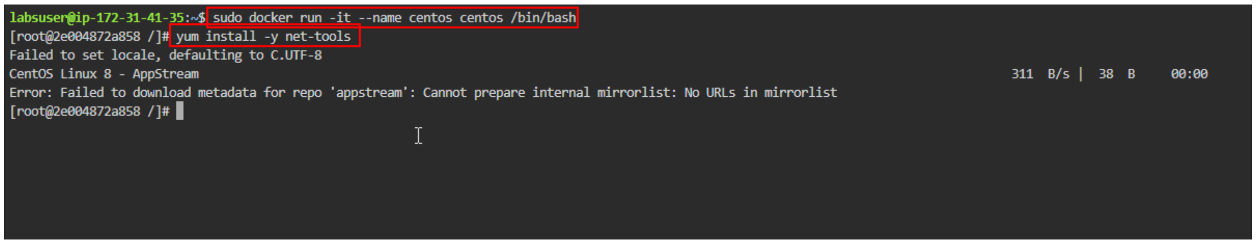
Steps to be followed:

1. Create a container and commit it
2. Create a bridge network and find its IP address
3. Connect the network from another SSH server

Step 1: Create a container and commit it

1.1 Create a CentOS Docker container and install net-tools using the following commands:

sudo docker run -it --name centos centos /bin/bash
yum install -y net-tools

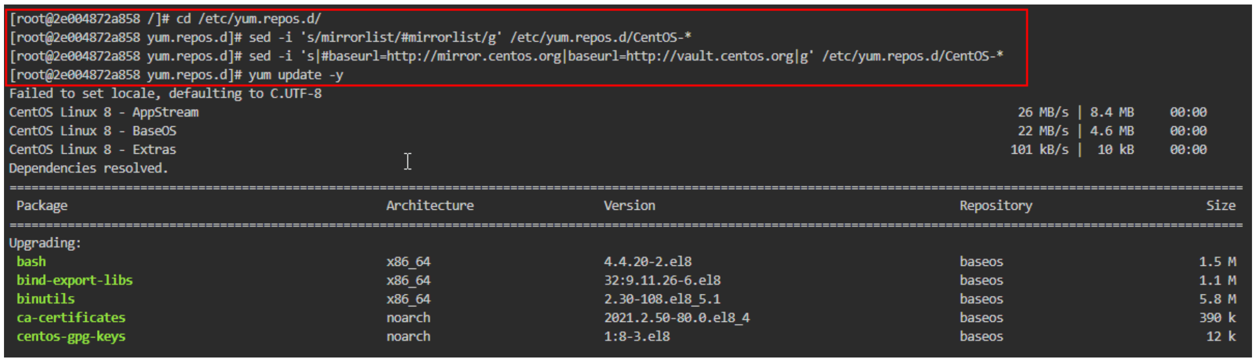


```
labsuser@ip-172-31-41-35:~$ sudo docker run -it --name centos centos /bin/bash
[root@2e004872a858 /]# yum install -y net-tools
Failed to set locale, defaulting to C.UTF-8
CentOS Linux 8 - AppStream                                                              311  B/s |  38  B     00:00
Error: Failed to download metadata for repo 'appstream': Cannot prepare internal mirrorlist: No URLs in mirrorlist
[root@2e004872a858 /]# 
```

Note: In the above step, you will face an error because CentOS Linux 8 reached its end of life (EOL) on December 31st, 2021, implying no further official development support. To update CentOS Linux, switch mirrors to vault.centos.org or consider upgrading to CentOS stream.

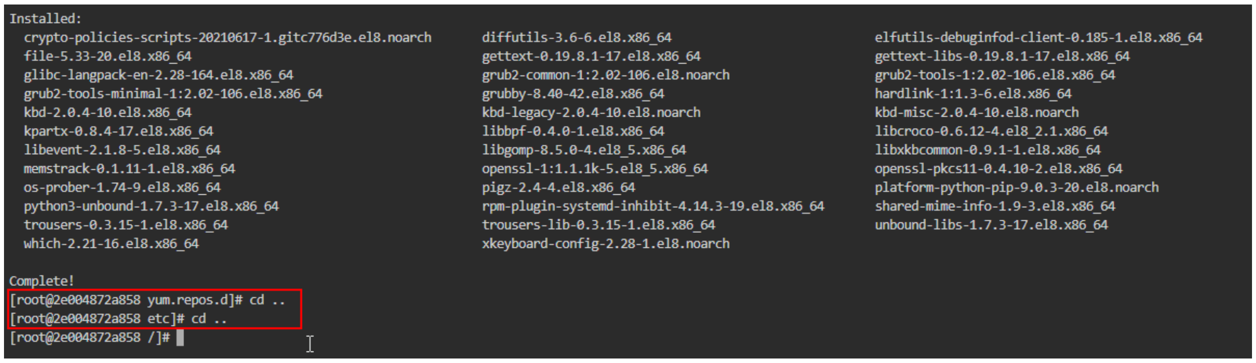
1.2 Execute the following commands to fix the error in the previous step:

cd /etc/yum.repos.d/
sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*
sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-*
yum update -y



```
[root@2e004872a858 /]# cd /etc/yum.repos.d/
[root@2e004872a858 yum.repos.d]# sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*
[root@2e004872a858 yum.repos.d]# sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-*
[root@2e004872a858 yum.repos.d]# yum update -y
Failed to set locale, defaulting to C.UTF-8
CentOS Linux 8 - AppStream                                                                   26 MB/s | 8.4 MB     00:00
CentOS Linux 8 - BaseOS                                                                      22 MB/s | 4.6 MB     00:00
CentOS Linux 8 - Extras                                                                     101 kB/s |  10 kB     00:00
Dependencies resolved.
================================================================================================================================
 Package                            Architecture          Version                              Repository                  Size
================================================================================================================================
Upgrading:
 bash                               x86_64                4.4.20-2.el8                         baseos                     1.5 M
 bind-export-libs                   x86_64                32:9.11.26-6.el8                     baseos                     1.1 M
 binutils                           x86_64                2.30-108.el8_5.1                     baseos                     5.8 M
 ca-certificates                    noarch                2021.2.50-80.0.el8_4                 baseos                     390 k
 centos-gpg-keys                    noarch                1:8-3.el8                            baseos                      12 k
```

1.3 Navigate to the root directory using the **cd** command as shown in the screenshot below:

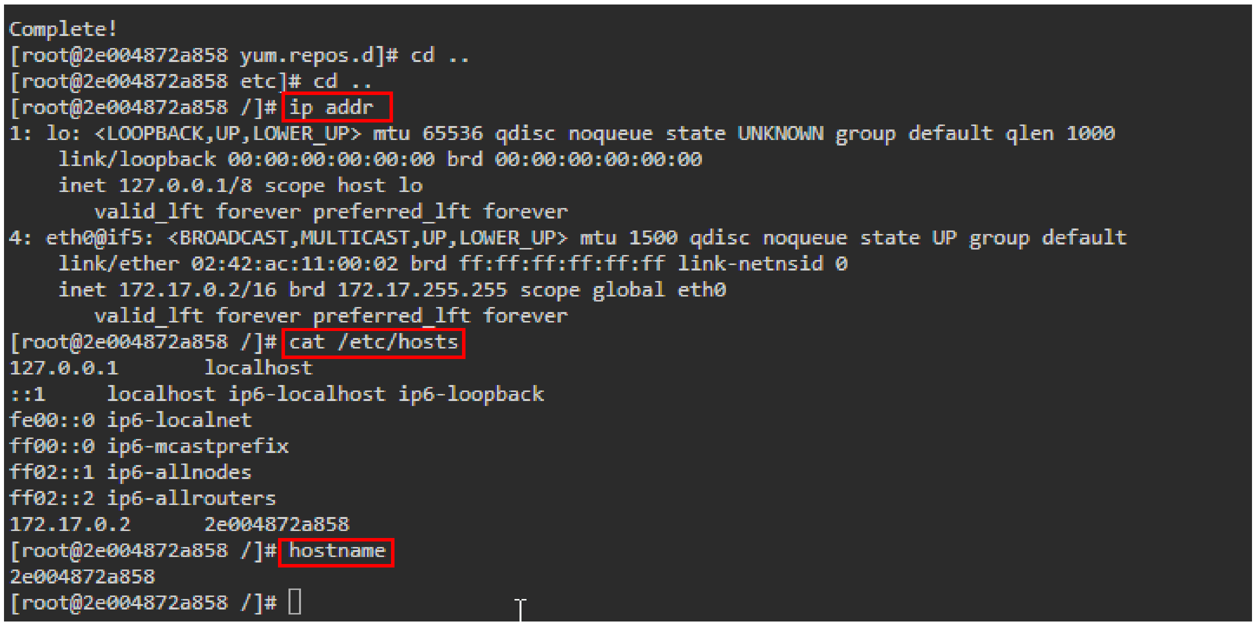


```
Installed:
  crypto-policies-scripts-20210617-1.gitc776d3e.el8.noarch      diffutils-3.6-6.el8.x86_64                          elfutils-debuginfod-client-0.185-1.el8.x86_64
  file-5.33-20.el8.x86_64                                       gettext-0.19.8.1-17.el8.x86_64                      gettext-libs-0.19.8.1-17.el8.x86_64
  glibc-langpack-en-2.28-164.el8.x86_64                         grub2-common-1:2.02-106.el8.noarch                  grub2-tools-1:2.02-106.el8.x86_64
  grub2-tools-minimal-1:2.02-106.el8.x86_64                     grubby-8.40-42.el8.x86_64                           hardlink-1:1.3-6.el8.x86_64
  kbd-2.0.4-10.el8.x86_64                                       kbd-legacy-2.0.4-10.el8.noarch                      kbd-misc-2.0.4-10.el8.noarch
  kpartx-0.8.4-17.el8.x86_64                                    libbpf-0.4.0-1.el8.x86_64                           libcroco-0.6.12-4.el8_2.1.x86_64
  libevent-2.1.8-5.el8.x86_64                                   libgomp-8.5.0-4.el8_5.x86_64                        libxkbcommon-0.9.1-1.el8.x86_64
  memstrack-0.1.11-1.el8.x86_64                                 openssl-1:1.1.1k-5.el8_5.x86_64                     openssl-pkcs11-0.4.10-2.el8.x86_64
  os-prober-1.74-9.el8.x86_64                                   pigz-2.4-4.el8.x86_64                               platform-python-pip-9.0.3-20.el8.noarch
  python3-unbound-1.7.3-17.el8.x86_64                           rpm-plugin-systemd-inhibit-4.14.3-19.el8.x86_64     shared-mime-info-1.9-3.el8.x86_64
  trousers-0.3.15-1.el8.x86_64                                  trousers-lib-0.3.15-1.el8.x86_64                    unbound-libs-1.7.3-17.el8.x86_64
  which-2.21-16.el8.x86_64                                      xkeyboard-config-2.28-1.el8.noarch

Complete!
[root@2e004872a858 yum.repos.d]# cd ..
[root@2e004872a858 etc]# cd ..
[root@2e004872a858 /]#
```

1.4 Check the IP address and hostname

ip addr
cat /etc/hosts
hostname

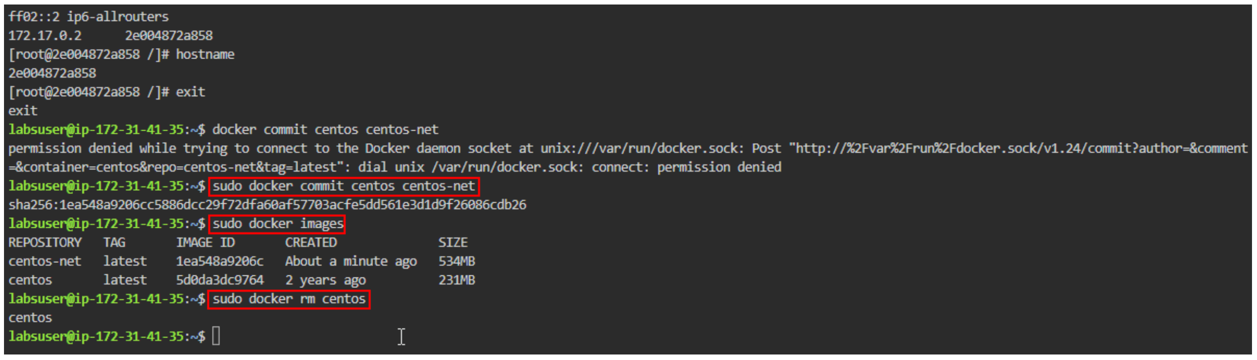


```
Complete!
[root@2e004872a858 yum.repos.d]# cd ..
[root@2e004872a858 etc]# cd ..
[root@2e004872a858 /]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
4: eth0@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
[root@2e004872a858 /]# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      2e004872a858
[root@2e004872a858 /]# hostname
2e004872a858
[root@2e004872a858 /]# 
```

Note: Type **exit** and press the **enter** key

1.5 Commit the container to an image using the following commands as shown in the screenshot below:

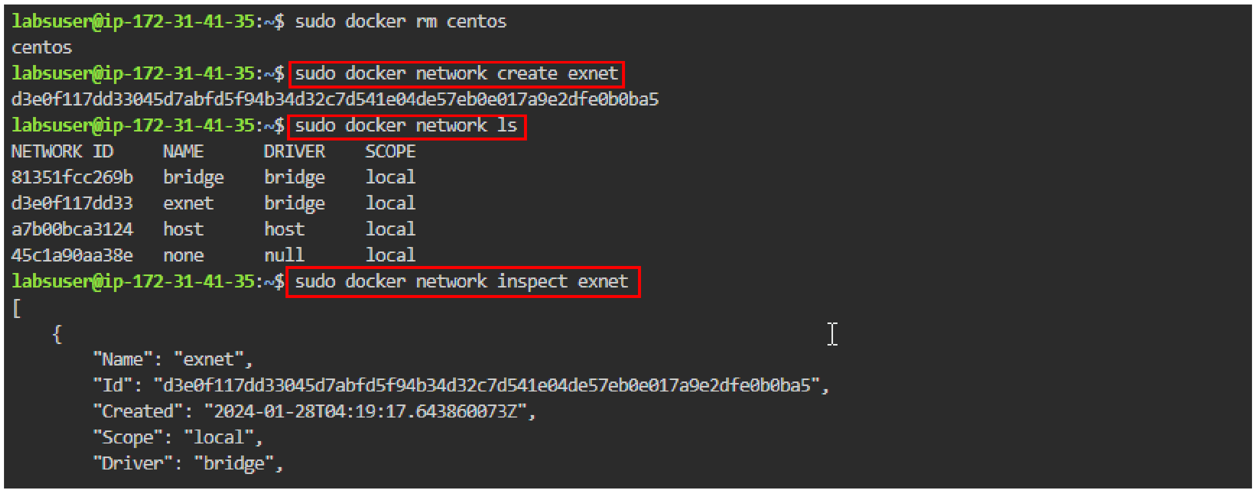
sudo docker commit centos centos-net
sudo docker images
sudo docker rm centos



```
ff02::2 ip6-allrouters
172.17.0.2      2e004872a858
[root@2e004872a858 /]# hostname
2e004872a858
[root@2e004872a858 /]# exit
exit
labsuser@ip-172-31-41-35:~$ docker commit centos centos-net
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/commit?author=&comment
=&container=centos&repo=centos-net&tag=latest": dial unix /var/run/docker.sock: connect: permission denied
labsuser@ip-172-31-41-35:~$ sudo docker commit centos centos-net
sha256:1ea548a9206cc5886dcc29f72dfa60af57703acfe5dd561e3d1d9f26086cdb26
labsuser@ip-172-31-41-35:~$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED              SIZE
centos-net   latest    1ea548a9206c   About a minute ago   534MB
centos       latest    5d0da3dc9764   2 years ago          231MB
labsuser@ip-172-31-41-35:~$ sudo docker rm centos
centos
labsuser@ip-172-31-41-35:~$ 
```

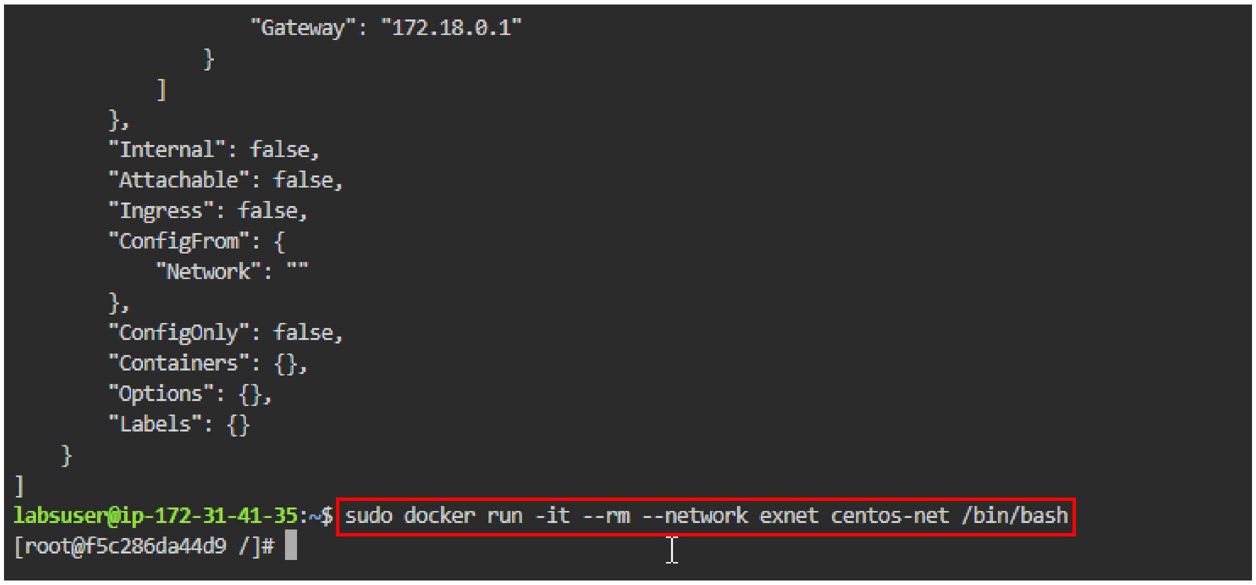
Step 2: Create a bridge network and find its IP address

2.1 Execute the following commands to create, list, and inspect a network named **exnet**:
sudo docker network create exnet
sudo docker network ls
sudo docker network inspect exnet



```
labsuser@ip-172-31-41-35:~$ sudo docker rm centos
centos
labsuser@ip-172-31-41-35:~$ sudo docker network create exnet
d3e0f117dd33045d7abfd5f94b34d32c7d541e04de57eb0e017a9e2dfe0b0ba5
labsuser@ip-172-31-41-35:~$ sudo docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
81351fcc269b   bridge    bridge    local
d3e0f117dd33   exnet     bridge    local
a7b00bca3124   host      host      local
45c1a90aa38e   none      null      local
labsuser@ip-172-31-41-35:~$ sudo docker network inspect exnet
[
    {
        "Name": "exnet",
        "Id": "d3e0f117dd33045d7abfd5f94b34d32c7d541e04de57eb0e017a9e2dfe0b0ba5",
        "Created": "2024-01-28T04:19:17.643860073Z",
        "Scope": "local",
        "Driver": "bridge",
```

2.2 Execute the following command to run the CentOS container using the new network:
sudo docker run -it --rm --network exnet centos-net /bin/bash



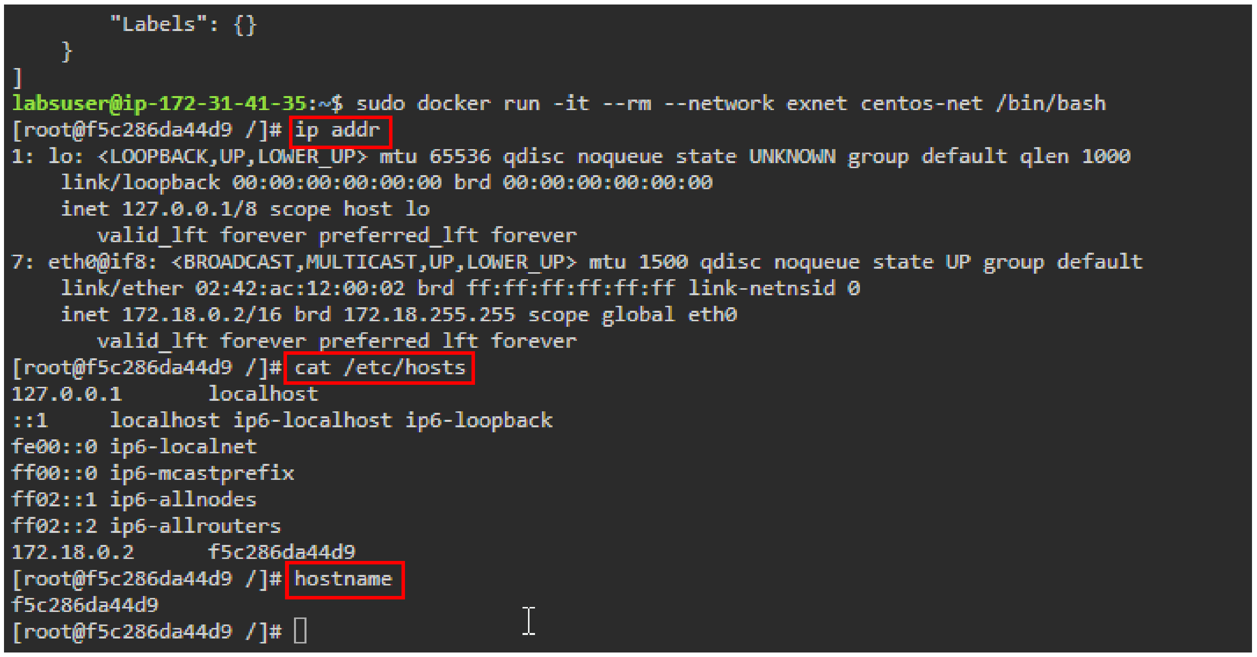
```
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {},
        "Labels": {}
    }
]
labsuser@ip-172-31-41-35:~$ sudo docker run -it --rm --network exnet centos-net /bin/bash
[root@f5c286da44d9 /]#
```

2.3 Check the IP address and hostname

ip addr

cat /etc/hosts

hostname

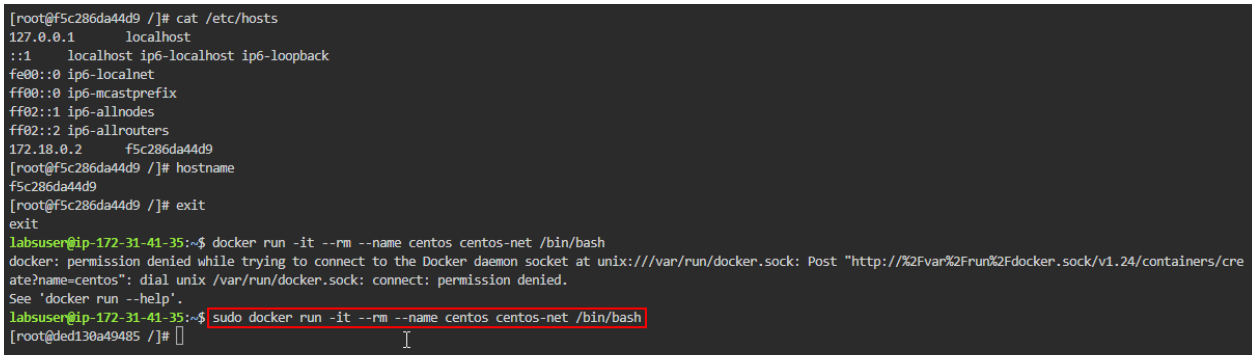


```
        "Labels": {}
    }
]
labsuser@ip-172-31-41-35:~$ sudo docker run -it --rm --network exnet centos-net /bin/bash
[root@f5c286da44d9 /]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
7: eth0@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:12:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.18.0.2/16 brd 172.18.255.255 scope global eth0
       valid_lft forever preferred_lft forever
[root@f5c286da44d9 /]# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.18.0.2      f5c286da44d9
[root@f5c286da44d9 /]# hostname
f5c286da44d9
[root@f5c286da44d9 /]#
```

Note: Type **exit** and press the **enter** key

2.4 Execute the following command to start a new container using the default network:

sudo docker run -it --rm --name centos centos-net /bin/bash



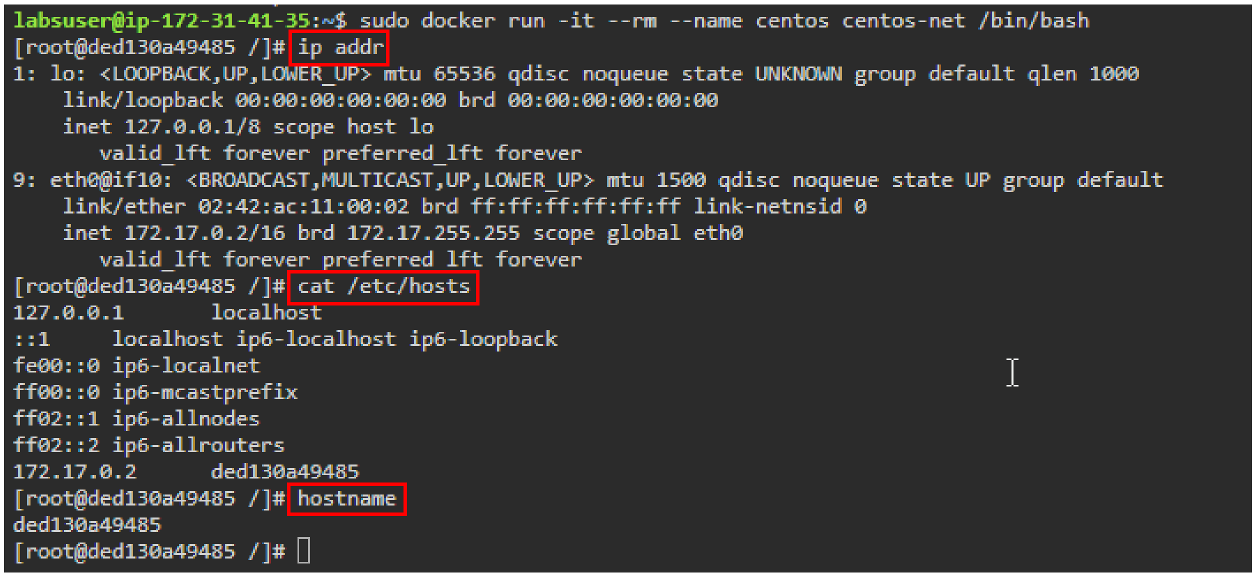
```
[root@f5c286da44d9 /]# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.18.0.2      f5c286da44d9
[root@f5c286da44d9 /]# hostname
f5c286da44d9
[root@f5c286da44d9 /]# exit
exit
labsuser@ip-172-31-41-35:~$ docker run -it --rm --name centos centos-net /bin/bash
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/cre
ate?name=centos": dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
labsuser@ip-172-31-41-35:~$ sudo docker run -it --rm --name centos centos-net /bin/bash
[root@ded130a49485 /]#
```

2.5 Check the IP address and hostname

ip addr

cat /etc/hosts

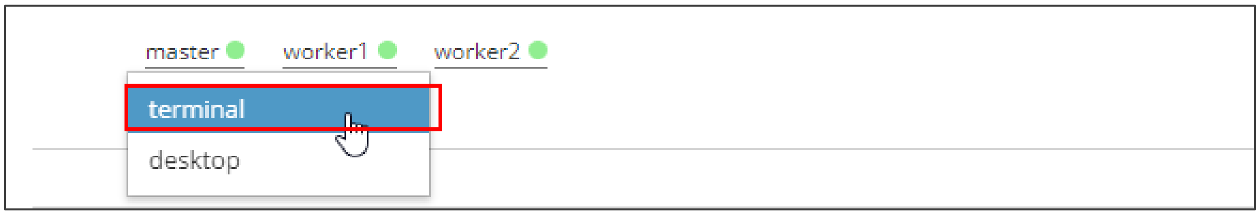
hostname



```
labsuser@ip-172-31-41-35:~$ sudo docker run -it --rm --name centos centos-net /bin/bash
[root@ded130a49485 /]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
9: eth0@if10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
[root@ded130a49485 /]# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      ded130a49485
[root@ded130a49485 /]# hostname
ded130a49485
[root@ded130a49485 /]# 
```

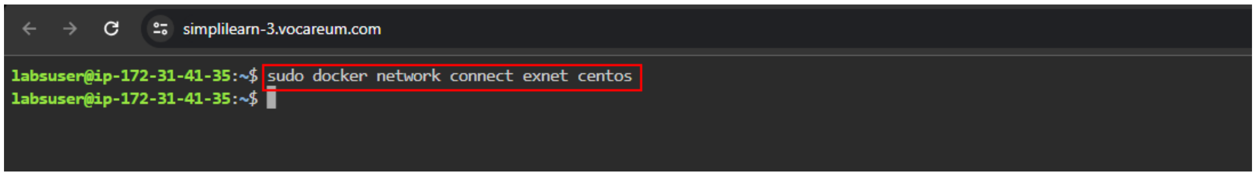
Step 3: Connect the network from another SSH server

3.1 Click on the **master** button and then select the **terminal** option



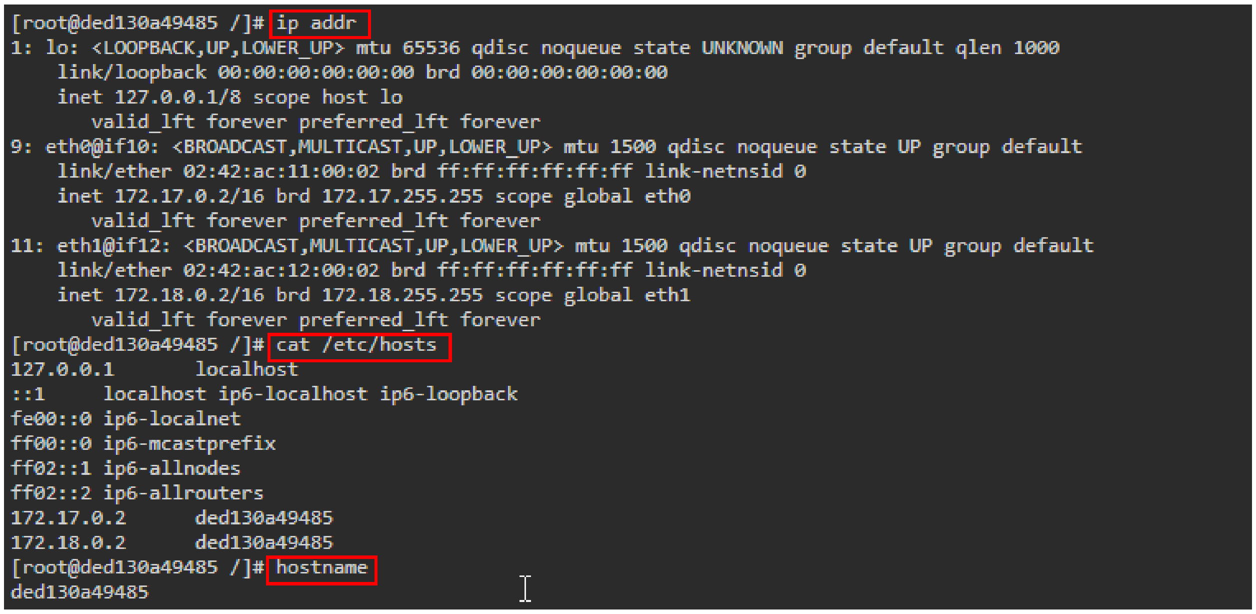
3.2 Execute the following command in the second SSH terminal to connect the network to the container:

sudo docker network connect exnet centos



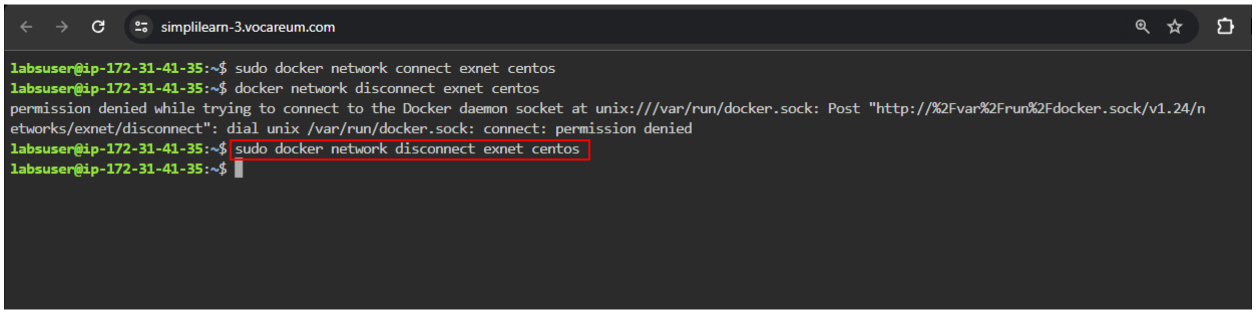
```
labsuser@ip-172-31-41-35:~$ sudo docker network connect exnet centos
labsuser@ip-172-31-41-35:~$ 
```

3.3 Navigate to the running container and verify the IP address as shown in the screenshot below:
ip addr
cat /etc/hosts
hostname



```
[root@ded130a49485 /]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
9: eth0@if10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
11: eth1@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:12:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.18.0.2/16 brd 172.18.255.255 scope global eth1
       valid_lft forever preferred_lft forever
[root@ded130a49485 /]# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      ded130a49485
172.18.0.2      ded130a49485
[root@ded130a49485 /]# hostname
ded130a49485
```

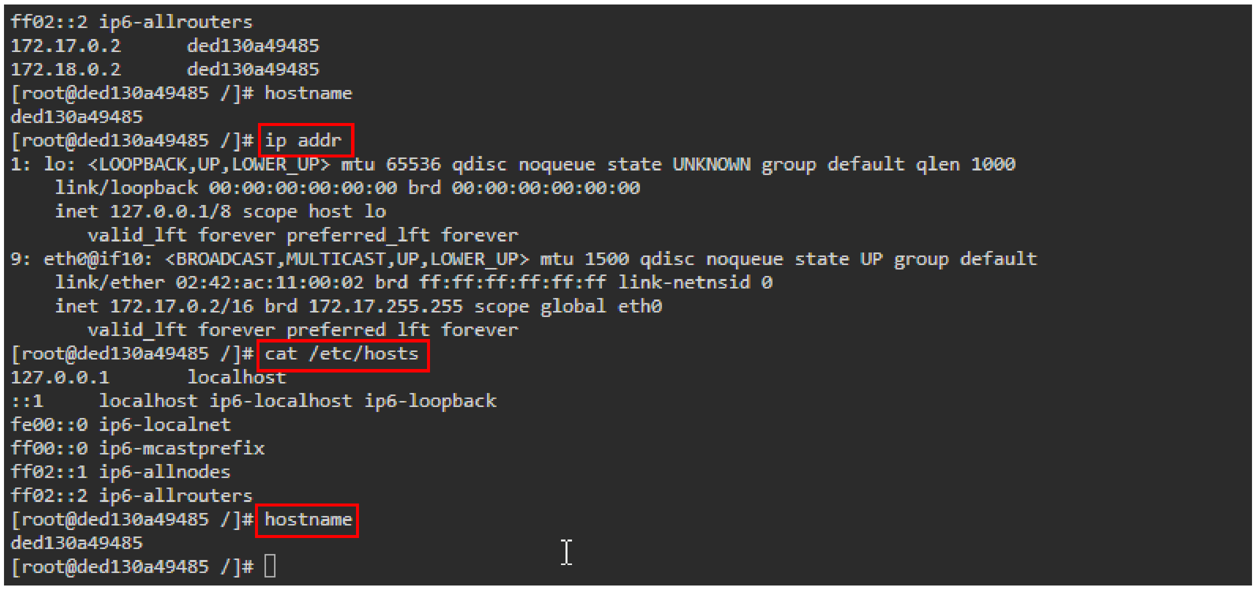
3.4 Execute the following command in the second SSH terminal to disconnect the network to the container:
docker network disconnect exnet centos



simplilearn-3.vocareum.com

```
labsuser@ip-172-31-41-35:~$ sudo docker network connect exnet centos
labsuser@ip-172-31-41-35:~$ docker network disconnect exnet centos
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/n
etworks/exnet/disconnect": dial unix /var/run/docker.sock: connect: permission denied
labsuser@ip-172-31-41-35:~$ sudo docker network disconnect exnet centos
labsuser@ip-172-31-41-35:~$
```

3.5 Navigate to the running container and verify the IP address as shown in the screenshot below:
ip addr
cat /etc/hosts
hostname



```
ff02::2 ip6-allrouters
172.17.0.2      ded130a49485
172.18.0.2      ded130a49485
[root@ded130a49485 /]# hostname
ded130a49485
[root@ded130a49485 /]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
9: eth0@if10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
[root@ded130a49485 /]# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
[root@ded130a49485 /]# hostname
ded130a49485
[root@ded130a49485 /]# 
```

Note: Type **exit** and press the **enter** key

By following these steps, you have successfully set up Docker networking and SSH integration to facilitate seamless communication and connectivity within your containerized environment.