**Connecting a container network namespace to root network namespace**

Let's create a custom network namespace ns0 and a bridge br0. In Linux networking, a bridge is a virtual network device that connects multiple network interfaces, allowing them to function as a single logical network.

sudo ip netns add ns0

sudo ip link add br0 type bridge

**Configure a bridge interface**

A new device, the br0 bridge interface, has been created, but it's now in a DOWN state. Let's assign ip address and turn it into UP state.

sudo ip link set br0 up

sudo ip addr add 192.168.0.1/16 dev br0

**Configure virtual ethernet cable**

It's time to set up a virtual Ethernet cable. One cable hand will be configured as a nic card in the ns0 namespace, while the other hand will be configured in the br0 interface.

sudo ip link add veth0 type veth peer name ceth0

sudo ip link set ceth0 netns ns0

sudo ip link set veth0 master br0

Both end of this cable is now in DOWN state. Let's turn into UP state

sudo ip netns exec ns0 ip link set ceth0 up

sudo ip link set veth0 up

**Configure ns0 namespace**

We need to assign an ip address to ceth0 and turn loopback interface into UP state.

sudo ip link set lo up

sudo ip addr add 192.168.0.2/16 dev ceth0

**Namespace ns0 to root ns Communication**

Let's check the Ip address assigned to primary ethernet interface of host machine.

ip addr show

**Now, ping to this ip address**

sudo ip netns exec ns0 bash

ping 10.0.0.25 [ act according to your eth0 ]

It says network in unreachable. So, something is not okay. Let's check the route table.

route

The output may look like.

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.0.0 0.0.0.0 255.255.0.0 U 0 0 0 ceth0

This routing table entry indicates that any destination IP address within the 192.168.0.0/16 network should be reached directly through the ceth0 interface, without the need for a specific gateway.

So we need to add a Default Gateway in the route table.

ip netns exec ns0 bash

ip route add default via 192.168.0.1

**Now we are good to go! Let's ping again.**

sudo ip netns exec ns0 bash

ping 10.0.0.25 -c 5

**Ping 8.8.8.8 from ns0 [ egress traffic ]**

We have come so far. Now we can ping our root ns or primary ethernet interface from custom namesapce. But can we ping outside the world? Let's ping to 8.8.8.8.

sudo ip netns exec ns0 bash

ping 8.8.8.8 -c 5

Okay, it does not seem that the network is unreachable. It's something else. Maybe somewhere, packets have stuck. Let's dig it out. We can do it using the Linux utility tcpdump. We will check two interfaces. One is br0, and the other is ens3 (in the case of my device).

Here we can see that packets are coming to those interfaces but are still stuck in ns0. But why?

**Root Cause**

See! The source IP address, 192.168.0.2, is attempting to connect to Google DNS 8.8.8.8 using its private IP address. So it's very basic that the outside world can't reach that private IP because they have no idea about that particular private IP address. That's why packets are able to reach Google DNS, but we are not getting any replies from 8.8.8.8.

**Solution**

We must somehow change the private IP address to a public address in order to sort out this issue with the help of NAT (Network Translation Address). So, a SNAT (source NAT) rule must be added to the IP table in order to modify the POSTROUTING chain.

sudo iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -j MASQUERADE

**Firewall rules**

sudo iptables -t nat -L -n -v

In case if still it not works then we may need to add some additional firewall rules.

sudo iptables --append FORWARD --in-interface br0 --jump ACCEPT

sudo iptables --append FORWARD --out-interface br0 --jump ACCEPT

These rules enabled traffic to travel across the br0 virtual bridge.These are useful to allow all traffic to pass through the br0 interface without any restrictions. However, keep in mind that using such rules without any filtering can expose your system to potential security risks. But for now we re good to ping!

**Test Connectivity**

ping 8.8.8.8 -c 5