**Setting up Linux Bridge Network among Namespaces**

This guide outlines the steps to create three namespaces named **blue-ns**, **gray-ns** and **lime-ns**. Then establish a linux bridge network among them using ***veth*** interfaces. The goal is to enable communication among the namespaces and allow them to ping each other.

**Prerequisites**

* Linux operating system
* Root or sudo access
* Packages

sudo apt update

sudo apt upgrade -y

sudo apt install iproute2 -y

sudo apt install net-tools

sudo apt install curl telnet tcpdump iputils-ping -y

**Step 1: Create a Linux bridge:**

sudo ip link add dev v-net type bridge

Lets run sudo ip links show to check and the ***expected output*** might look like

12: v-net: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000

link/ether d2:22:49:28:a4:c8 brd ff:ff:ff:ff:ff:ff

**But here v-net is now in down state. So lets turn into up.**

sudo ip link set dev v-net up

Here, the "UP" flag indicates that the interface is enabled and operational, while the "DOWN" state indicates that the interface is currently inactive or not functioning as there is no any physical connectivity of the interface right now.

**Step 2: Assign an IP address to the bridge interface 'v-net':**

sudo ip address add 10.0.0.1/24 dev v-net

**Step 3: Create three network namespaces:**

sudo ip netns add blue-ns

sudo ip netns add gray-ns

sudo ip netns add lime-ns

Run sudo ip netns list to check the list of namespaces.

**Step 4: Create virtual Ethernet pairs:**

sudo ip link add veth-blue-ns type veth peer name veth-blue-br

sudo ip link add veth-gray-ns type veth peer name veth-gray-br

sudo ip link add veth-lime-ns type veth peer name veth-lime-br

**Step 5: Move each end of veth cable to a different namespace:**

sudo ip link set dev veth-blue-ns netns blue-ns

sudo ip link set dev veth-gray-ns netns gray-ns

sudo ip link set dev veth-lime-ns netns lime-ns

**Step 6: Add the other end of the virtual interfaces to the bridge:**

sudo ip link set dev veth-blue-br master v-net

sudo ip link set dev veth-gray-br master v-net

sudo ip link set dev veth-lime-br master v-net

Again run sudo ip link show command to verify.

**Step 7: Set the bridge interfaces up:**

sudo ip link set dev veth-blue-br up

sudo ip link set dev veth-gray-br up

sudo ip link set dev veth-lime-br up

**Step 8: Set the namespace interfaces up:**

sudo ip netns exec blue-ns ip link set dev veth-blue-ns up

sudo ip netns exec gray-ns ip link set dev veth-gray-ns up

sudo ip netns exec lime-ns ip link set dev veth-lime-ns up

Now, run sudo ip link show command again and it should show all interfaces are currently in UP state.

**Step 9: Assign IP addresses to the virtual interfaces within each namespace and set the default routes:**

**In the blue-ns namespace:**

sudo ip netns exec blue-ns ip address add 10.0.0.11/24 dev veth-blue-ns

sudo ip netns exec blue-ns ip route add default via 10.0.0.1

**In the gray-ns namespace:**

sudo ip netns exec gray-ns ip address add 10.0.0.21/24 dev veth-gray-ns

sudo ip netns exec gray-ns ip route add default via 10.0.0.1

**In the lime-ns namespace:**

sudo ip netns exec lime-ns ip address add 10.0.0.31/24 dev veth-lime-ns

sudo ip netns exec lime-ns ip route add default via 10.0.0.1

**Step 10: Firewall rules:**

sudo iptables --append FORWARD --in-interface v-net --jump ACCEPT

sudo iptables --append FORWARD --out-interface v-net --jump ACCEPT

These rules enabled traffic to travel across the v-net virtual bridge.These are useful to allow all traffic to pass through the v-net 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**

sudo ip netns exec lime-ns ping -c 2 10.0.0.11

***Expected Output***

PING 10.0.0.11 (10.0.0.11) 56(84) bytes of data.

64 bytes from 10.0.0.11: icmp\_seq=1 ttl=64 time=0.064 ms

64 bytes from 10.0.0.11: icmp\_seq=2 ttl=64 time=0.169 ms

--- 10.0.0.11 ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1006ms

rtt min/avg/max/mdev = 0.064/0.116/0.169/0.052 ms

**Another one:**

sudo ip netns exec gray-ns ping -c 2 10.0.0.11

***Expected Output***

PING 10.0.0.11 (10.0.0.11) 56(84) bytes of data.

64 bytes from 10.0.0.11: icmp\_seq=1 ttl=64 time=0.051 ms

64 bytes from 10.0.0.11: icmp\_seq=2 ttl=64 time=0.112 ms

--- 10.0.0.11 ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1024ms

rtt min/avg/max/mdev = 0.051/0.081/0.112/0.030 ms

**Another one:**

sudo ip netns exec blue-ns ping -c 2 10.0.0.21

***Expected Output***

PING 10.0.0.21 (10.0.0.21) 56(84) bytes of data.

64 bytes from 10.0.0.21: icmp\_seq=1 ttl=64 time=0.056 ms

64 bytes from 10.0.0.21: icmp\_seq=2 ttl=64 time=0.109 ms

--- 10.0.0.21 ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1018ms

rtt min/avg/max/mdev = 0.056/0.082/0.109/0.026 ms

**Another one:**

sudo ip netns exec blue-ns ping -c 2 10.0.0.31

***Expected Output***

PING 10.0.0.31 (10.0.0.31) 56(84) bytes of data.

64 bytes from 10.0.0.31: icmp\_seq=1 ttl=64 time=0.073 ms

64 bytes from 10.0.0.31: icmp\_seq=2 ttl=64 time=0.127 ms

--- 10.0.0.31 ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1031ms

rtt min/avg/max/mdev = 0.073/0.100/0.127/0.027 ms

**Clean Up (optional)**

sudo ip netns del <namespace>

sudo ip link delete <bridge network name> type bridge

If you want to remove the namespaces and bridge network device run these commands to clean up the setup.