

Setting Up a Basic Linux Network
Namespace with Two Hosts, Two Routers,
Tunnel Interfaces, and Static Routes

Introduction

This is a demonstration of how to create four namespaces on two Linux hosts: two namespaces act as hosts and two act as routers. It includes assigning IP addresses, setting up a GRE tunnel, configuring static routes, and performing a ping test between them.

**You can follow my GitHub repository, [BuildNetWorkToCloud](https://github.com/zenithsoul/BuildNetWorkToCloud),
to see examples of automated network creation
using open-source tools**



<https://github.com/zenithsoul>

Requirement

1) -----

2 Linux Hosts

2) -----

Ubuntu / Debian: `iproute2`

CentOS / RHEL / Fedora: `iproute`

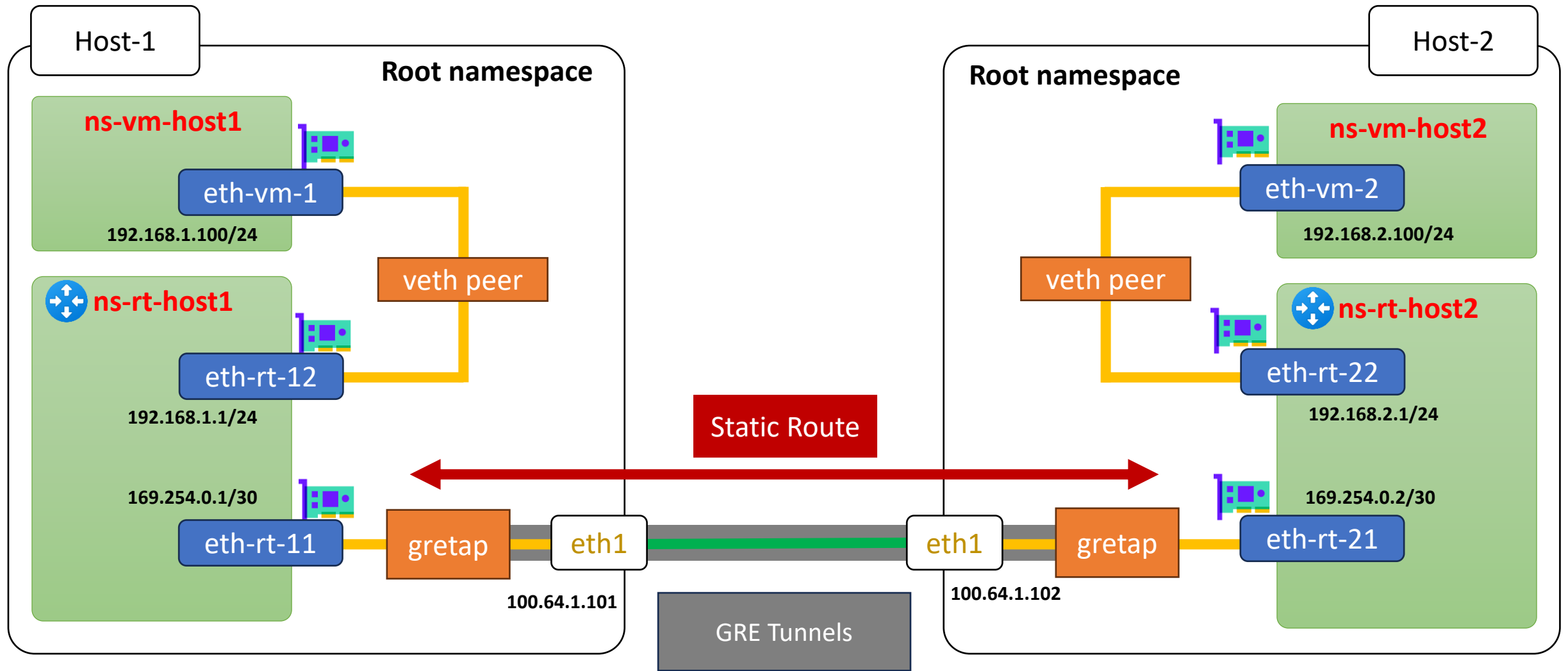
3) -----

Run with root account

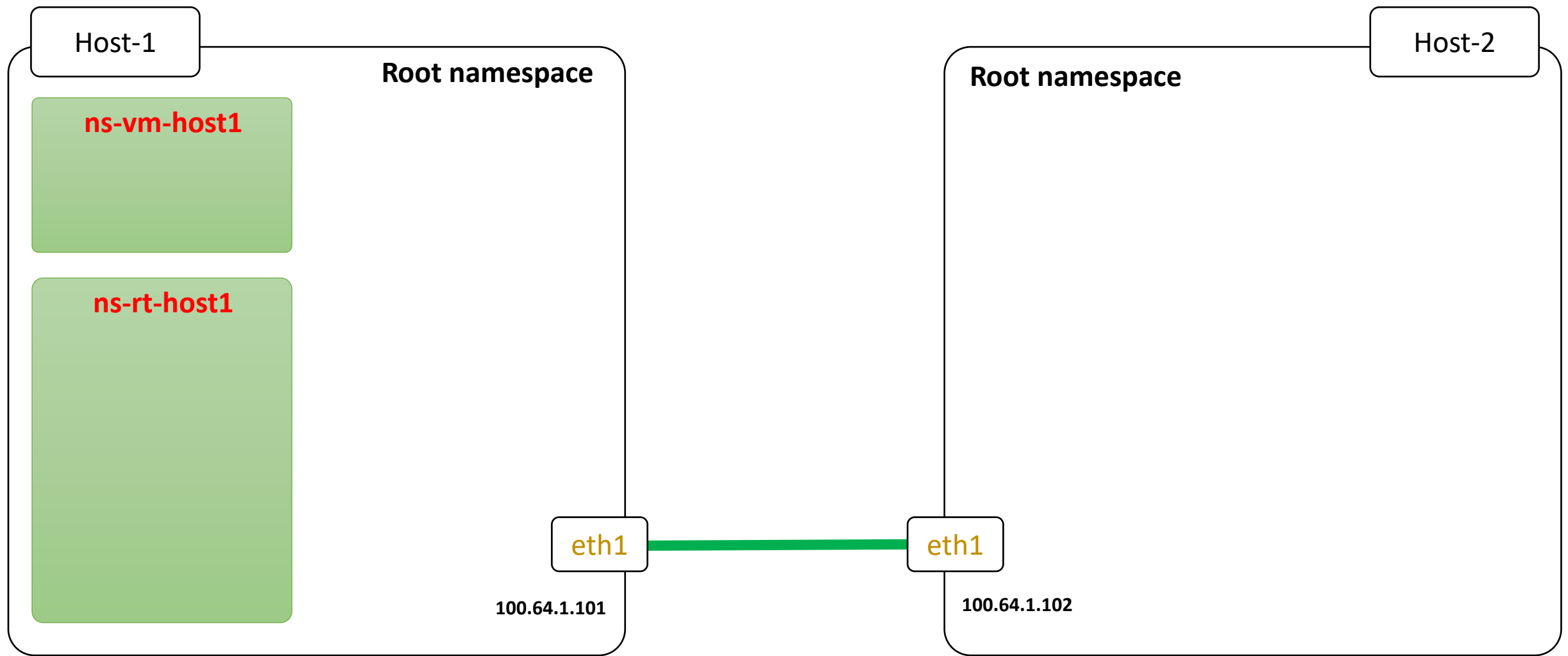
4) -----

Set the IP addresses according to my diagram. If you want to make changes, don't forget to update the IPs in the Linux command line.

If your network interface names are different, please adjust them to fit your setup.



High Level Design - Diagram



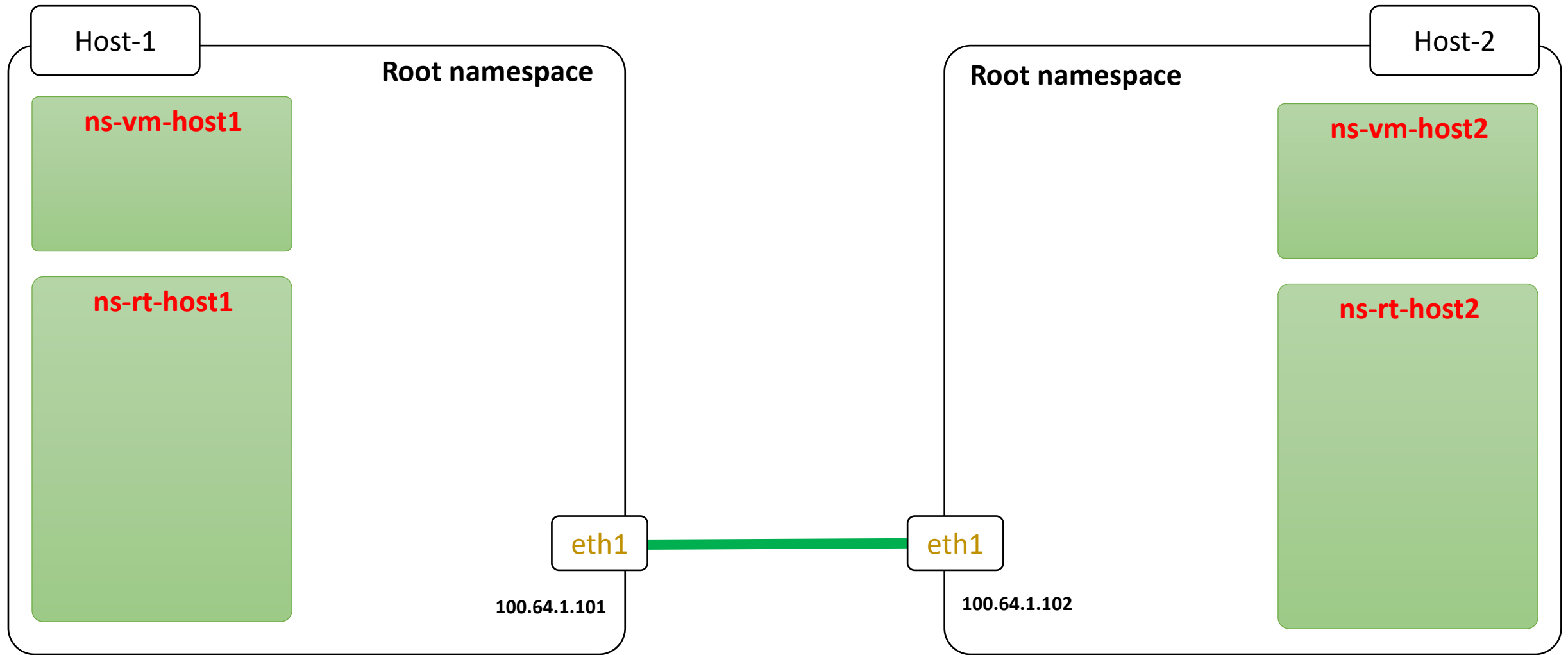
Create namespaces on **Host-1** and enable IP forwarding in **ns-rt-host1**

```
Host-1 # ip netns add ns-vm-host1
```

```
Host-1 # ip netns add ns-rt-host1
```

```
# enable IP forwarding
```

```
Host-1 # ip netns exec ns-rt-host1 sysctl -w net.ipv4.ip_forward=1
```



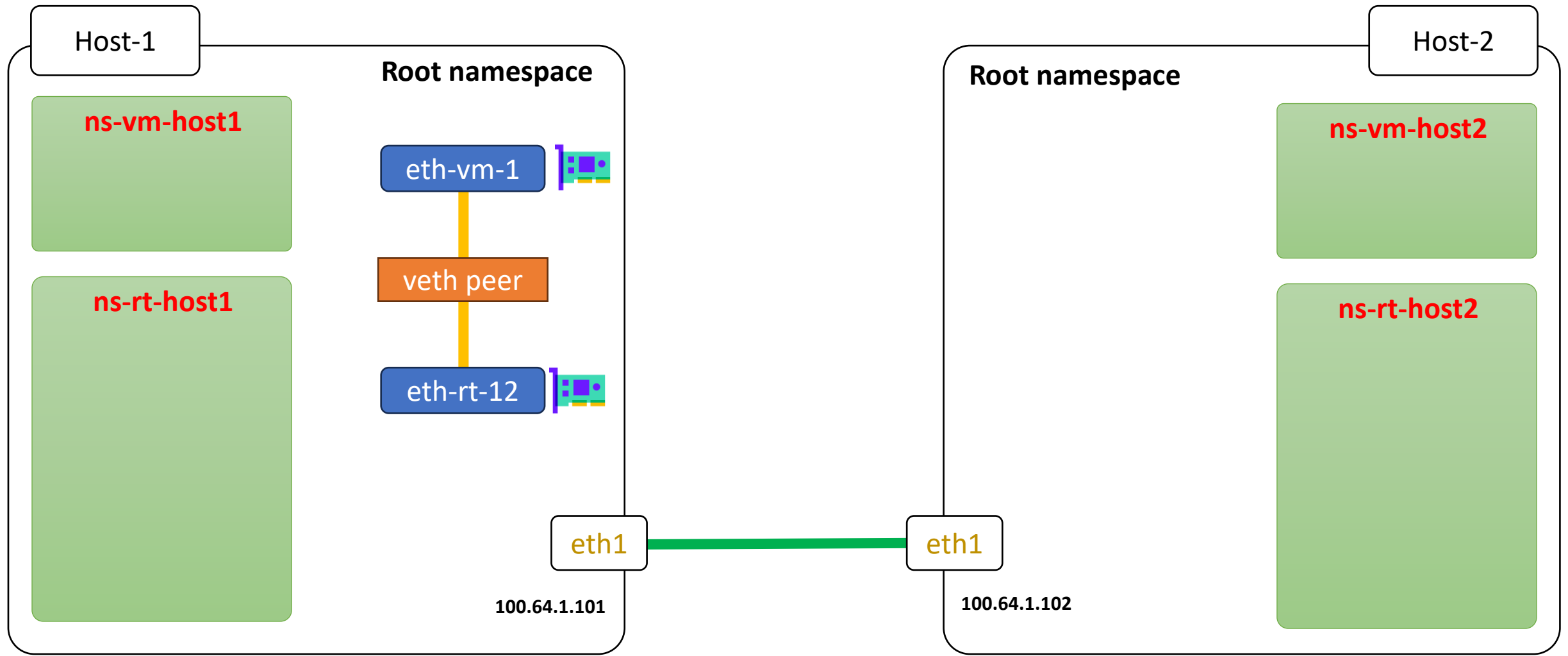
Create namespaces on **Host-2** and enable IP forwarding in **ns-rt-host2**

```
Host-2 # ip netns add ns-vm-host2
```

```
Host-2 # ip netns add ns-rt-host2
```

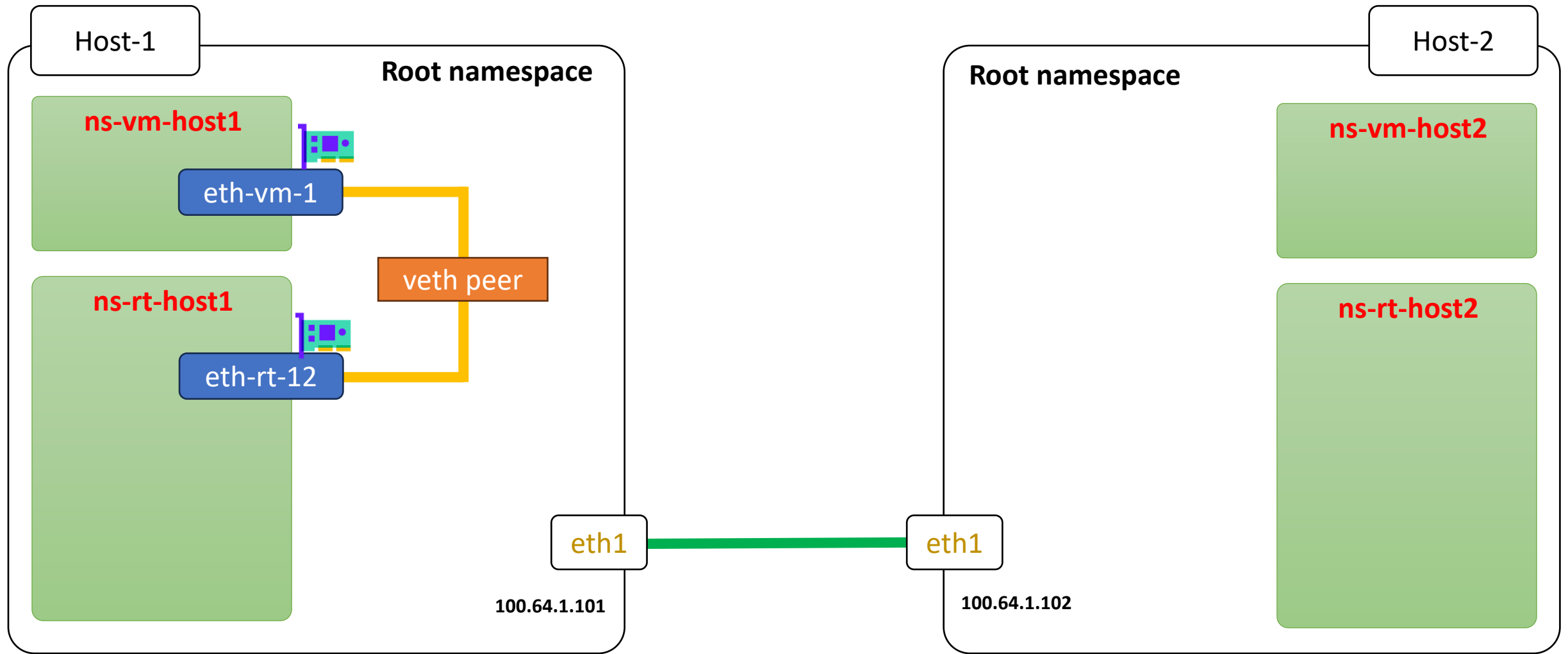
```
# enable IP forwarding
```

```
Host-2 # ip netns exec ns-rt-host2 sysctl -w net.ipv4.ip_forward=1
```



Create veth pair on **Host-1**

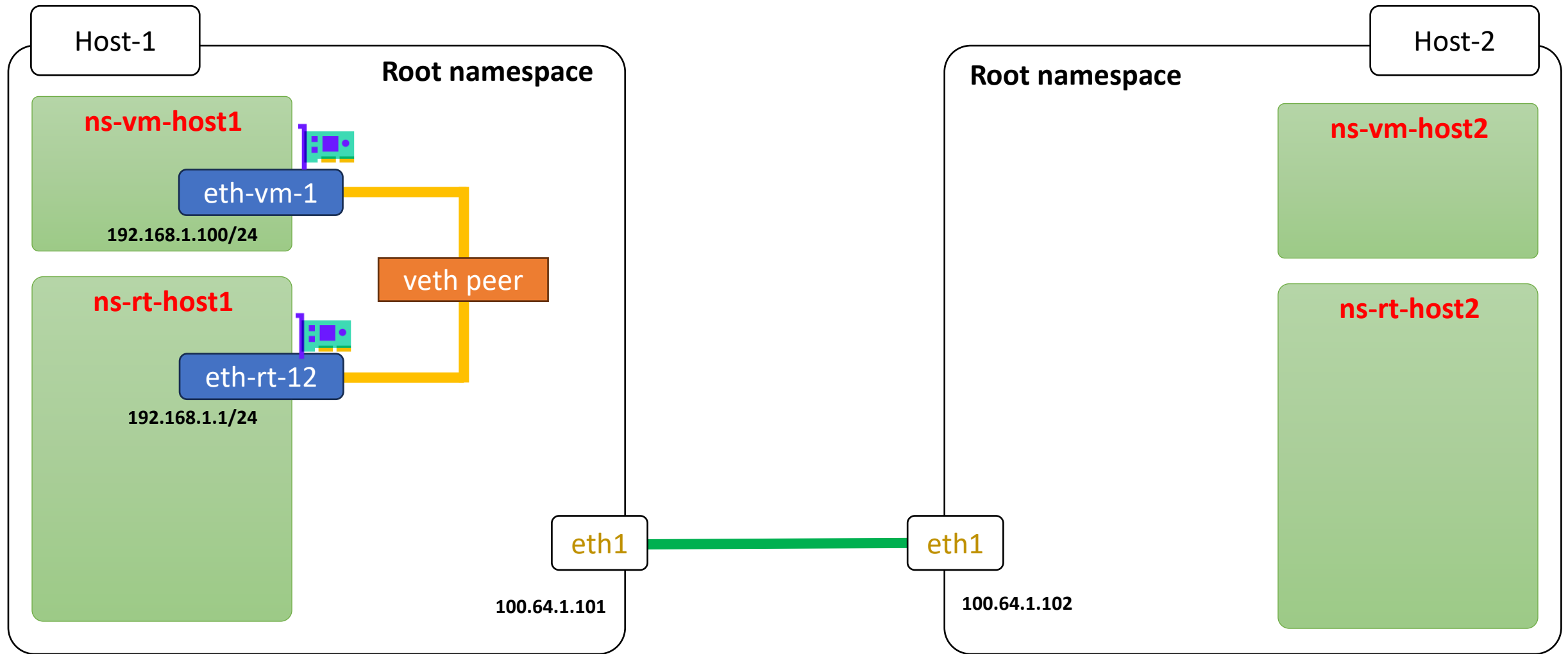
```
Host-1 # ip link add eth-vm-1 type veth peer name eth-rt-12
```



Move All interfaces into namespace

Host-1 # ip link set **eth-vm-1** netns **ns-vm-host1**

Host-1 # ip link set **eth-rt-12** netns **ns-rt-host1**



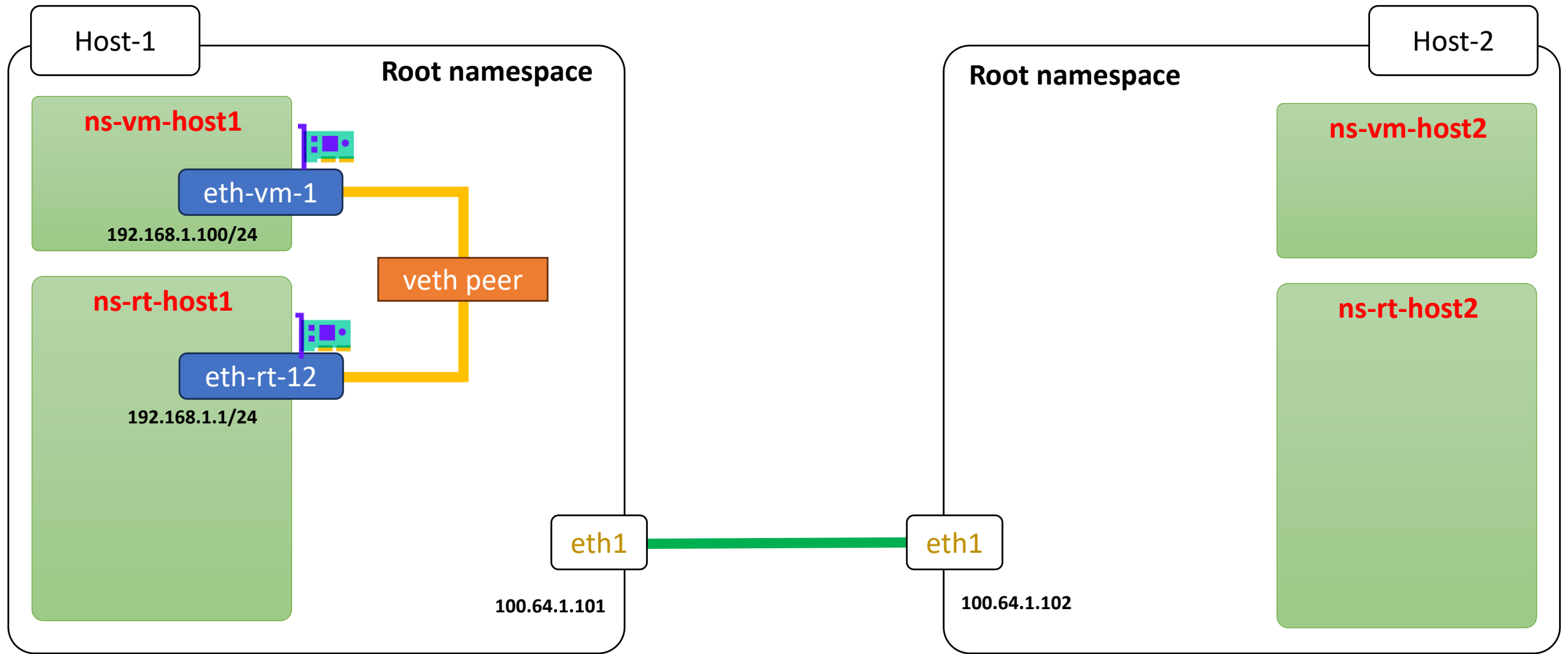
Assign IP inside the namespace **ns-vm-host1** and **ns-rt-host1**

Host-1 # ip netns exec **ns-vm-host1** ip addr add 192.168.1.100/24 dev `eth-vm-1`

Host-1 # ip netns exec **ns-vm-host1** ip link set `eth-vm-1` up

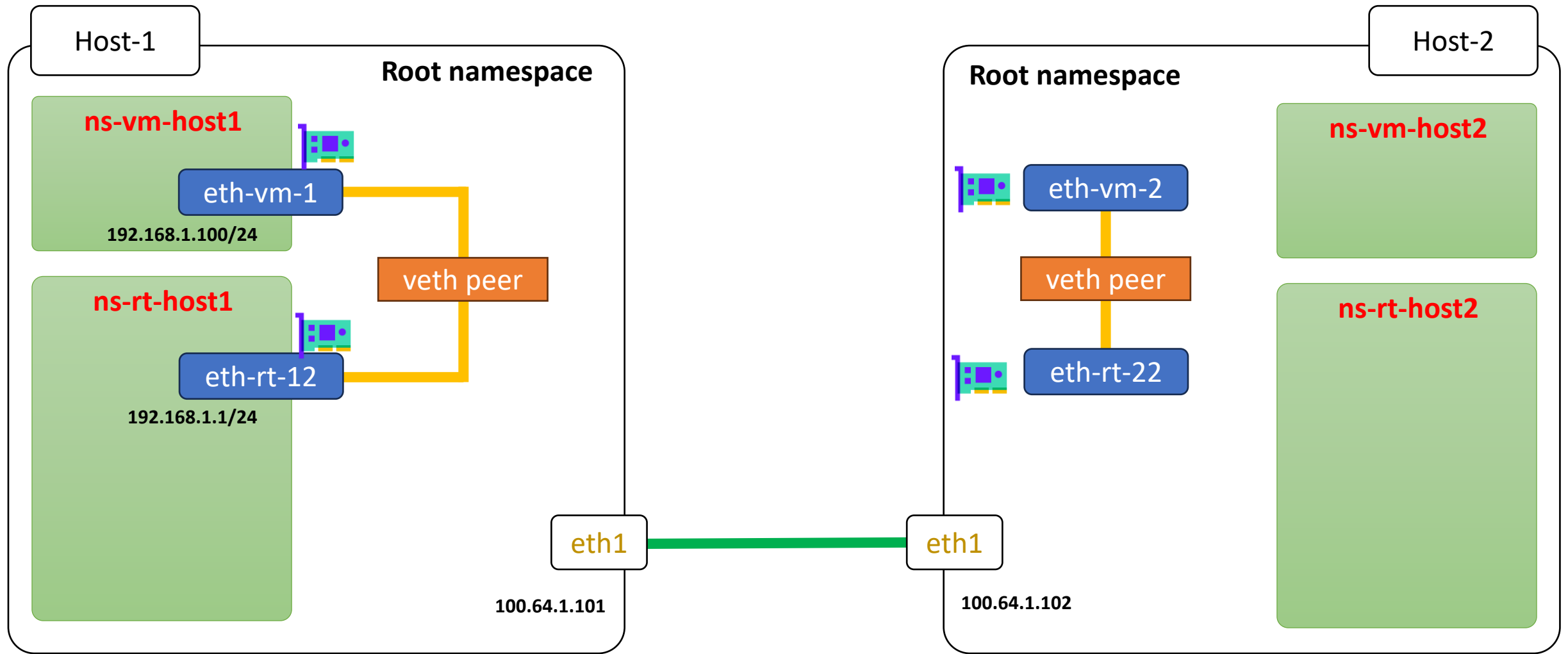
Host-1 # ip netns exec **ns-rt-host1** ip addr add 192.168.1.1/24 dev `eth-rt-12`

Host-1 # ip netns exec **ns-rt-host1** ip link set `eth-rt-12` up



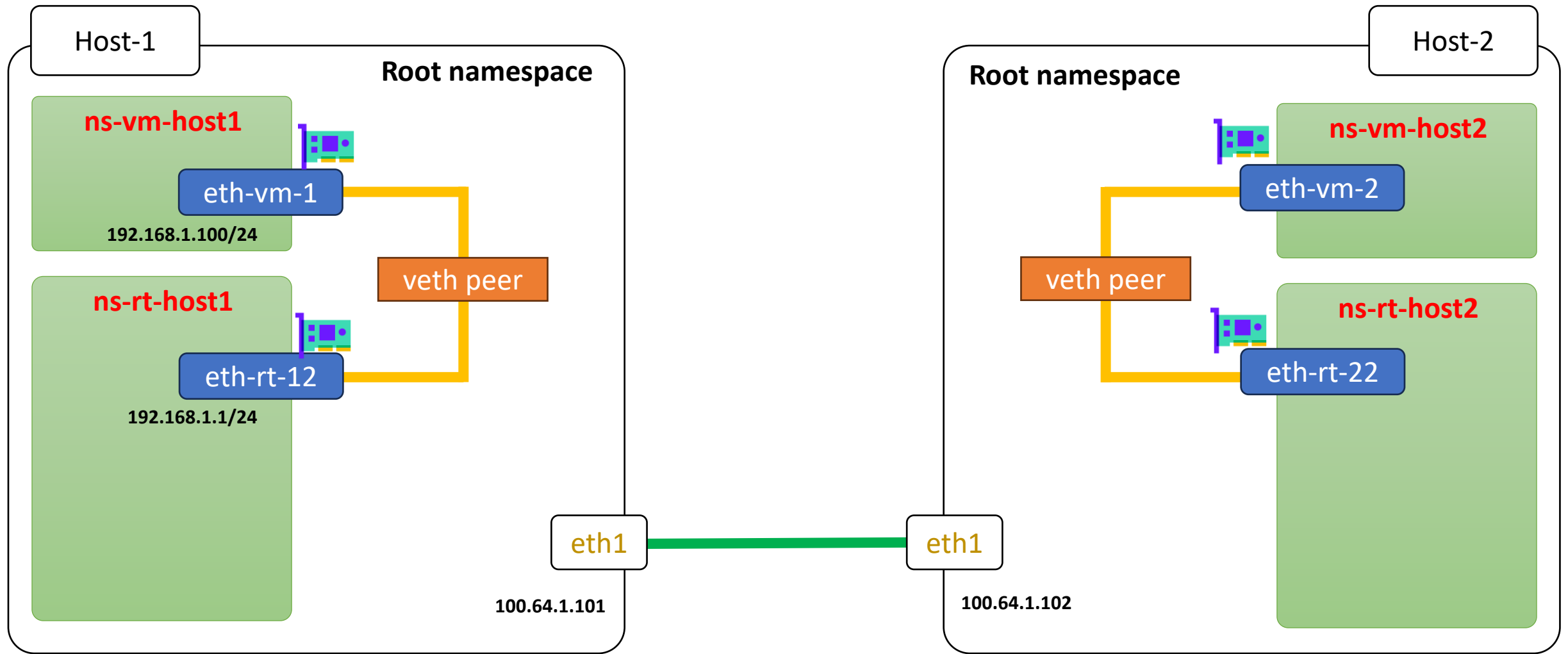
Check ping between **ns-vm-host1** and **ns-rt-host1**

```
Host-1 # ip netns exec ns-vm-host1 ping 192.168.1.1 -c 4
```



Create veth pair on Host-2

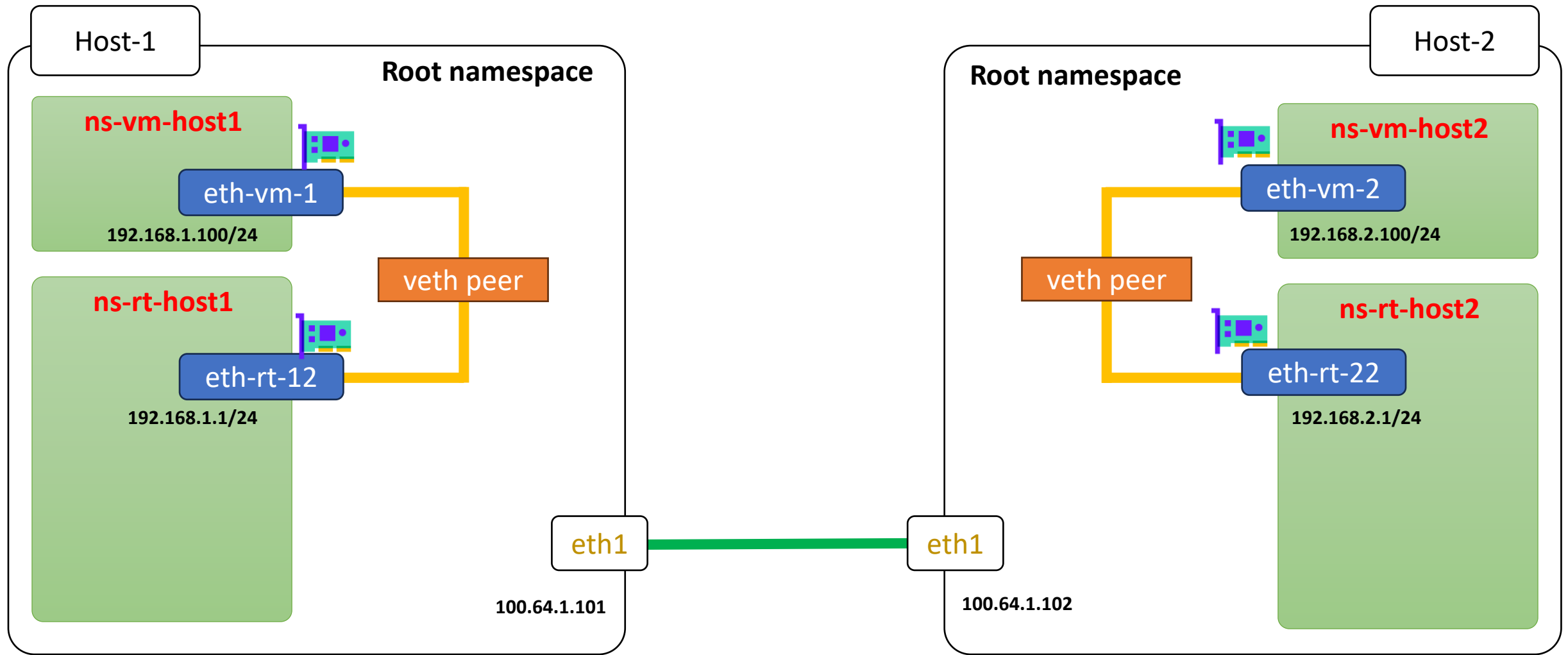
Host-2 # ip link add eth-vm-2 type veth peer name eth-rt-22



Move All interfaces into namespace

Host-2 # ip link set **eth-vm-2** netns **ns-vm-host1**

Host-2 # ip link set **eth-rt-22** netns **ns-rt-host1**



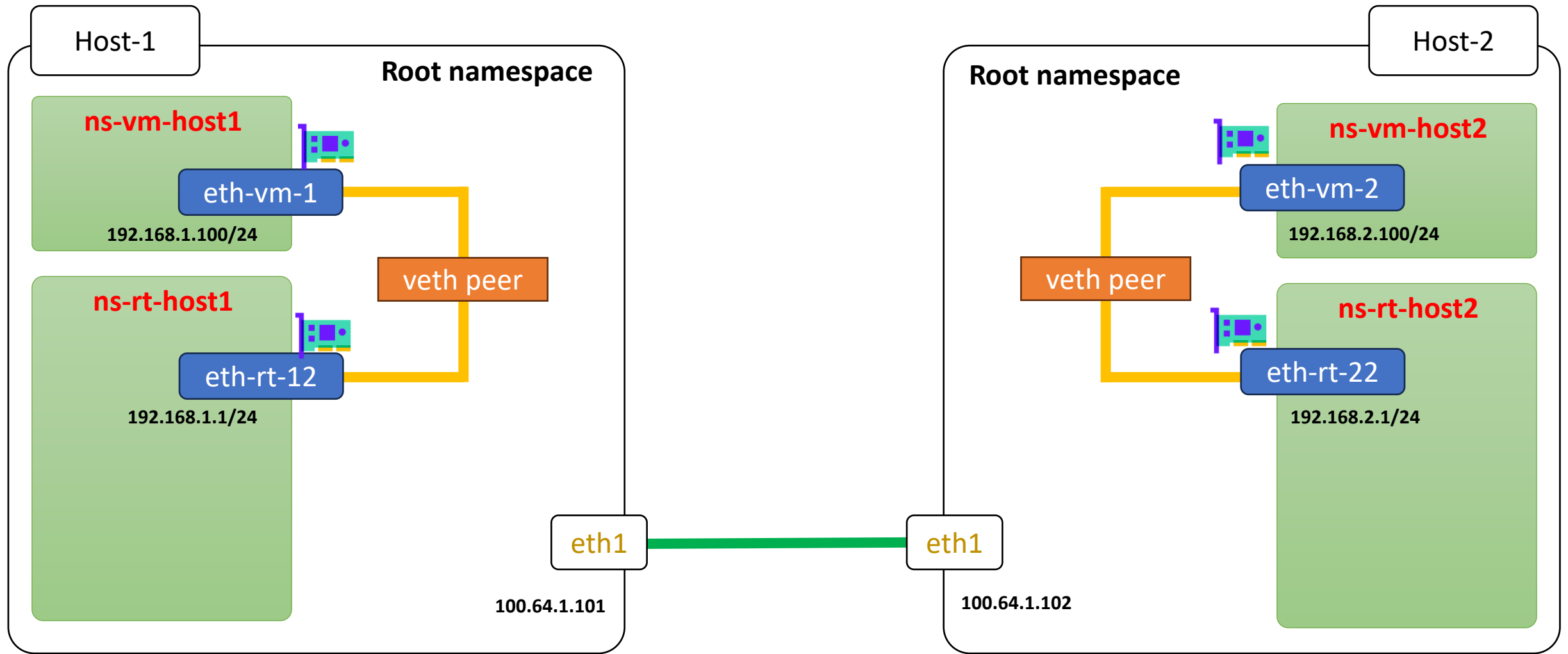
Assign IP inside the namespace **ns-vm-host2** and **ns-rt-host2**

Host-2 # ip netns exec **ns-vm-host1** ip addr add 192.168.2.100/24 dev **eth-vm-2**

Host-2 # ip netns exec **ns-vm-host1** ip link set **eth-vm-2** up

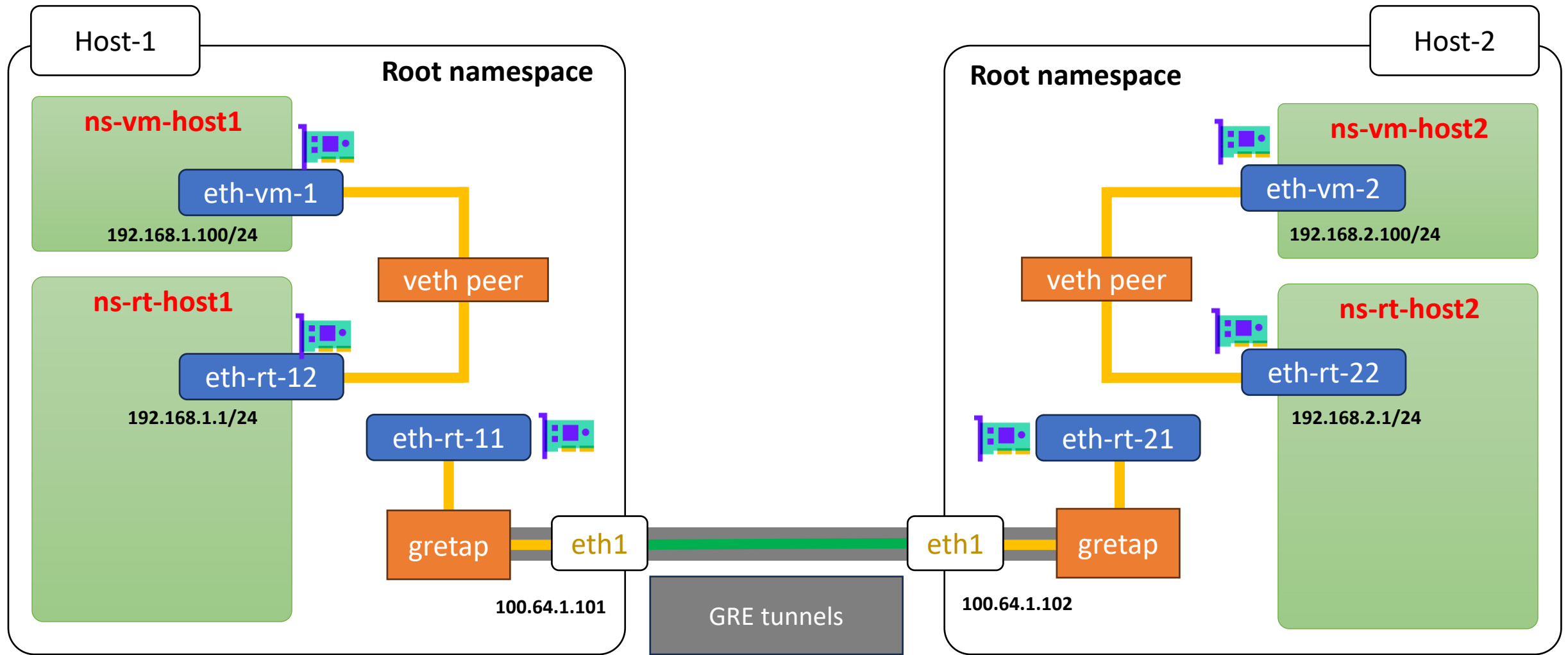
Host-2 # ip netns exec **ns-rt-host1** ip addr add 192.168.2.1/24 dev **eth-rt-22**

Host-2 # ip netns exec **ns-rt-host1** ip link set **eth-rt-22** up



Check ping between **ns-vm-host2** and **ns-rt-host2**

```
Host-2 # ip netns exec ns-vm-host2 ping 192.168.2.1 -c 4
```

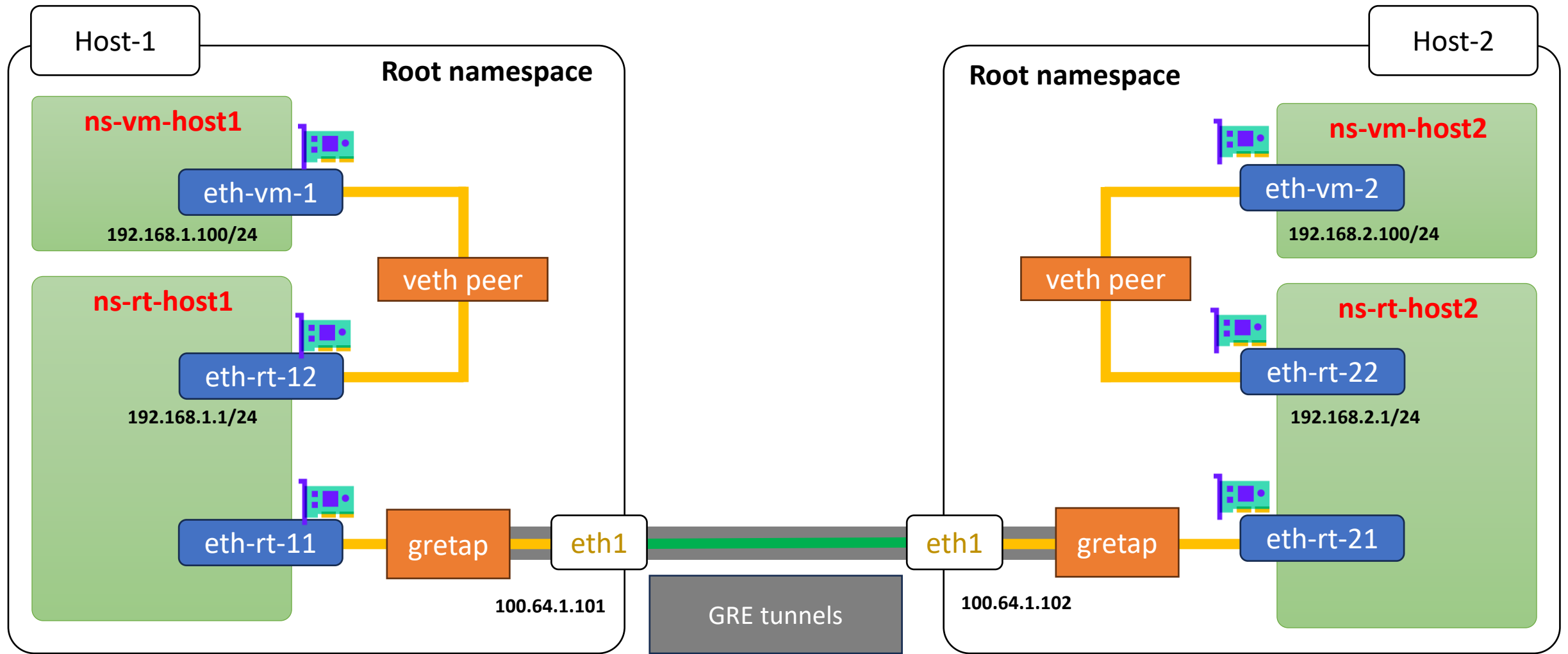


Create a GRETAP on Host-1

```
Host-1 # ip link add eth-rt-11 type gretap \
remote 100.64.1.102 \
local 100.64.1.101 dev eth1
```

Create a GRETAP on Host-2

```
Host-2 # ip link add eth-rt-21 type gretap \
remote 100.64.1.101 \
local 100.64.1.102 dev eth1
```

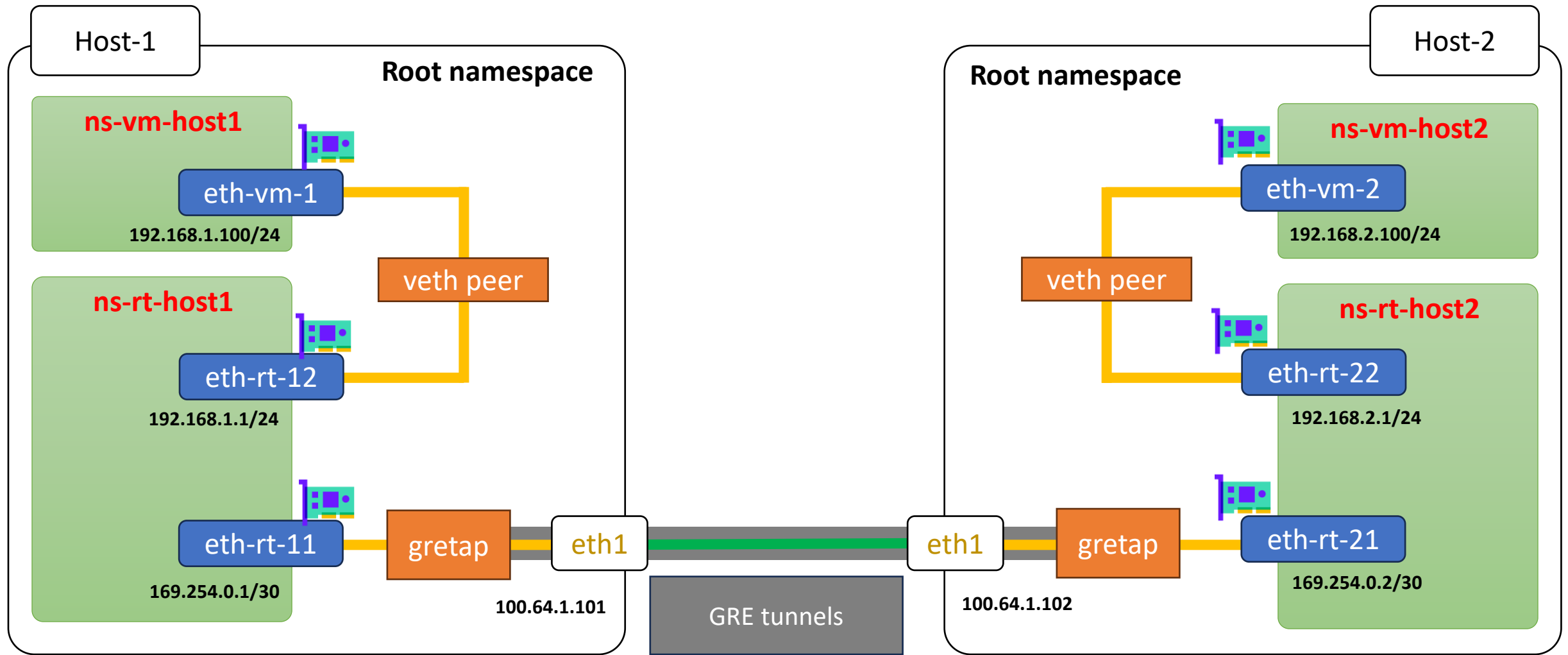


Assign [eth-rt-11] interface to **ns-rt-host1**

Host-1 # ip link set **eth-rt-11** netns **ns-rt-host1**

Assign [eth-rt-21] interface to **ns-rt-host2**

Host-2 # ip link set **eth-rt-21** netns **ns-rt-host2**



Assign IP to [eth-rt-11] interface

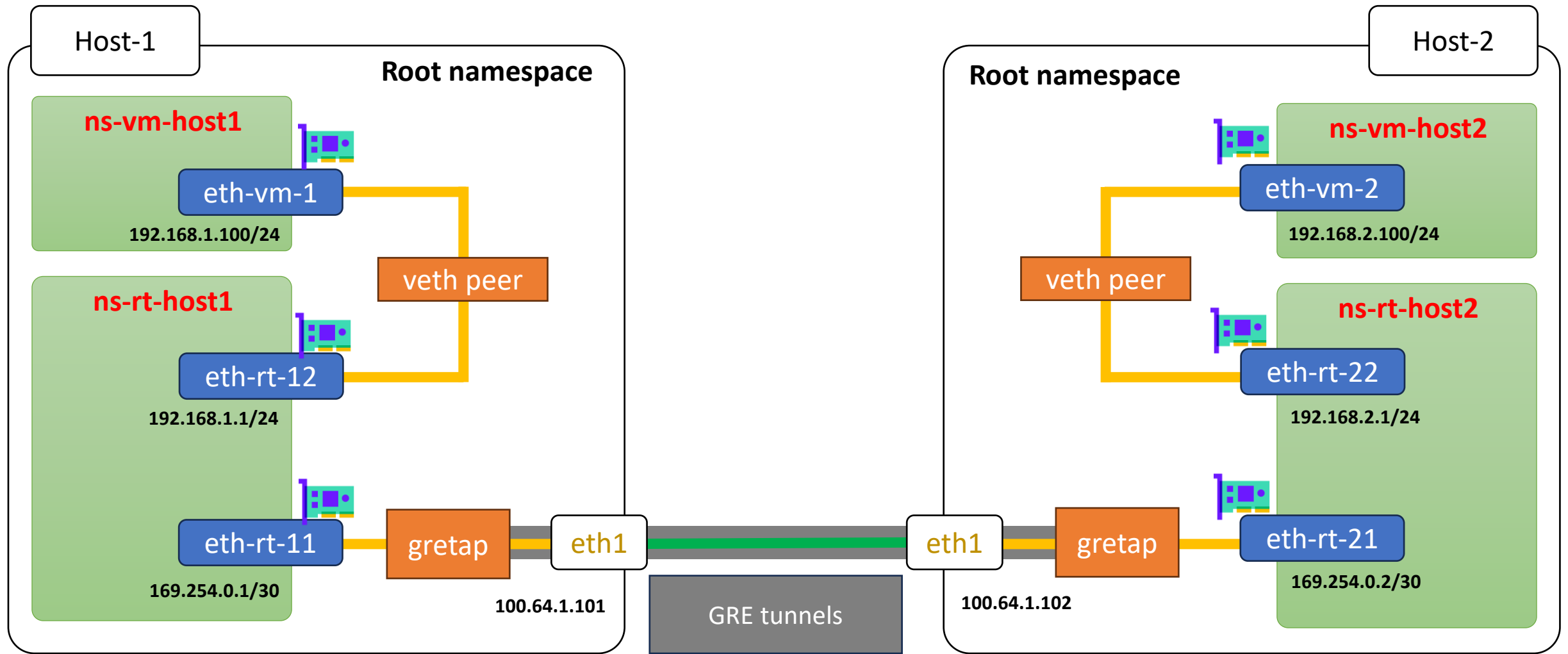
```
Host-1 # ip netns exec ns-rt-host1 \
ip addr add 169.254.0.1/30 dev eth-rt-11
```

```
Host-1 # ip netns exec ns-rt-host1 ip link set eth-rt-11 up
```

Assign IP to [eth-rt-21] interface

```
Host-2 # ip netns exec ns-rt-host2 \
ip addr add 169.254.0.1/30 dev eth-rt-21
```

```
Host-2 # ip netns exec ns-rt-host2 ip link set eth-rt-21 up
```

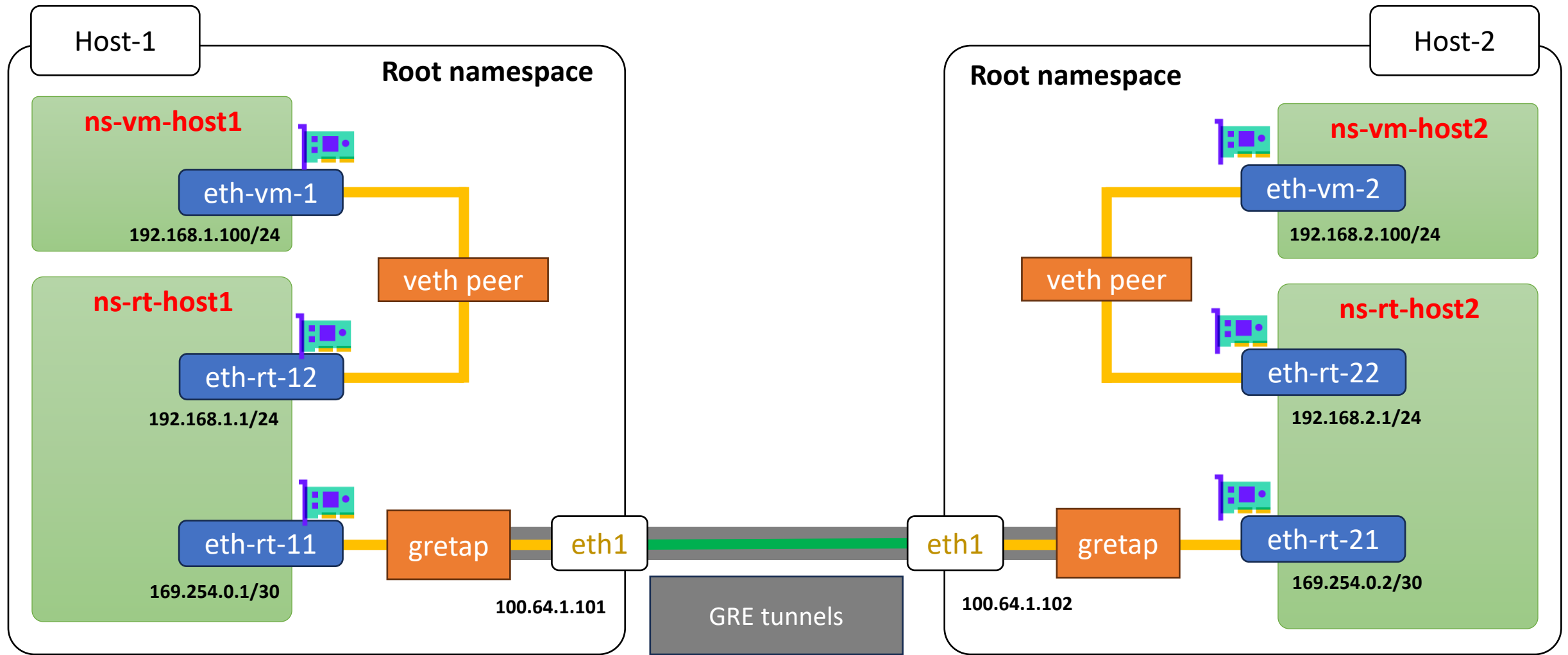


Test ping from **ns-rt-host1** to **ns-rt-host2**

Host-1 # ip netns exec **ns-rt-host1** ping 169.254.0.2 -c 4

Test ping from **ns-rt-host2** to **ns-rt-host1**

Host-2 # ip netns exec **ns-rt-host1** ping 169.254.0.1 -c 4



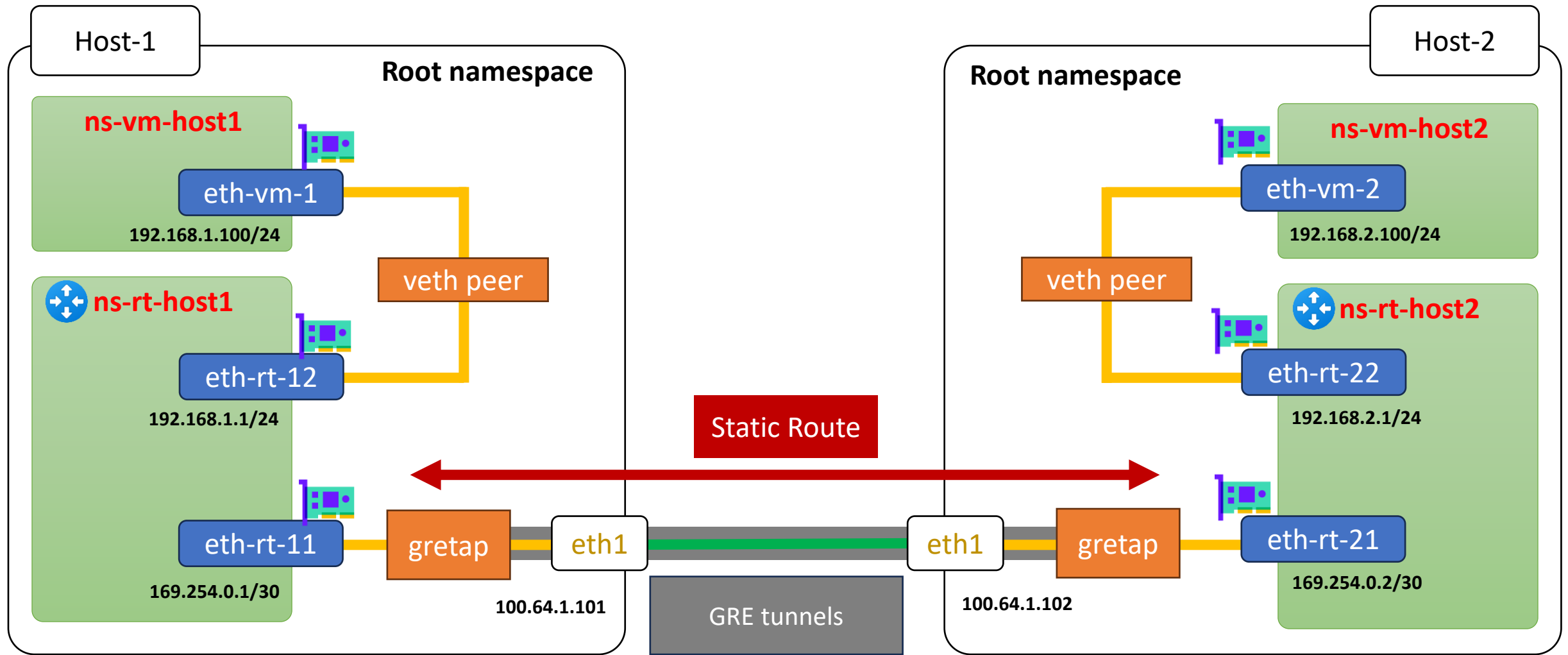
Test ping from **ns-vm-host1** to **ns-vm-host2**

Host-1 # ip netns exec **ns-vm-host1** ping 192.168.2.100 -c 4

Test ping from **ns-vm-host2** to **ns-vm-host1**

Host-2 # ip netns exec **ns-vm-host2** ping 192.168.1.100 -c 4

Both hosts can't reach to each other

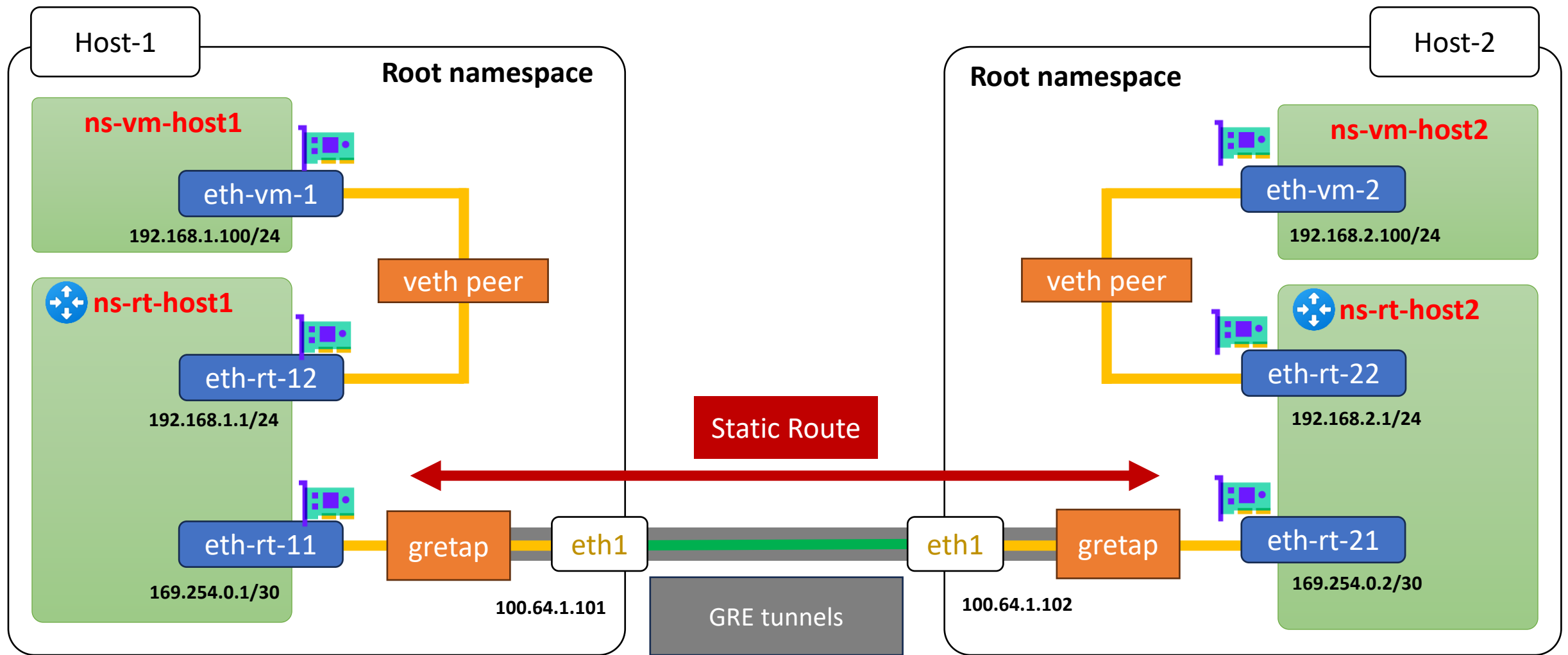


Add static route to network **192.168.2.0/24**
in namespace **ns-rt-host1**

Host-1 # ip netns exec **ns-rt-host1** ip route add \
192.168.2.0/24 via **169.254.0.2** dev **eth-rt-11**

Add static route to network **192.168.1.0/24**
in namespace **ns-rt-host2**

Host-2 # ip netns exec **ns-rt-host2** ip route add \
192.168.1.0/24 via **169.254.0.1** dev **eth-rt-21**

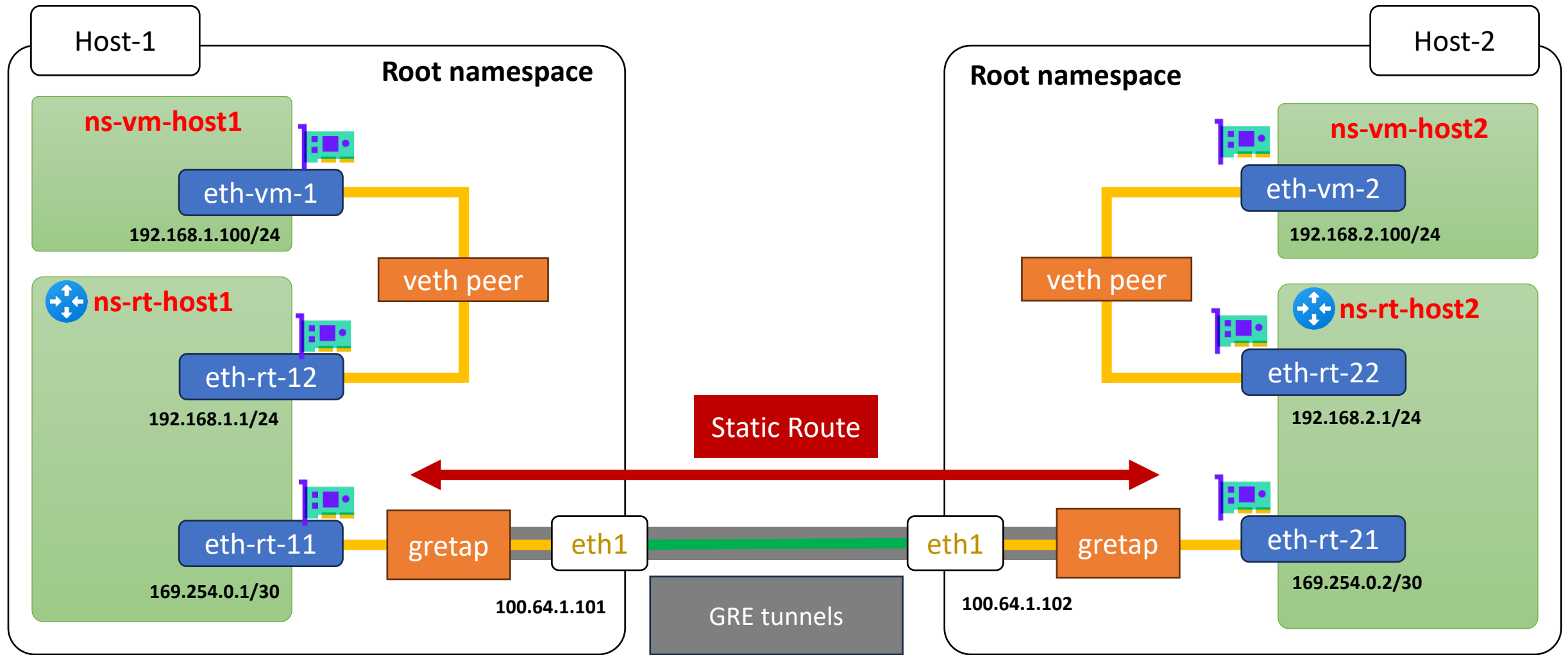


Add default route in namespace **ns-vm-host1** to **ns-rt-host1**

```
Host-1 # ip netns exec ns-vm-host1 ip route add default \
via 192.168.1.1 dev eth-vm-1
```

Add default route in namespace **ns-vm-host2** to **ns-rt-host2**

```
Host-2 # ip netns exec ns-vm-host2 ip route add default \
via 192.168.2.1 dev eth-vm-2
```



Test ping from **ns-vm-host1** to **ns-vm-host2**

Host-1 # ip netns exec **ns-vm-host1** ping 192.168.2.100 -c 4

Test ping from **ns-vm-host2** to **ns-vm-host1**

Host-2 # ip netns exec **ns-vm-host2** ping 192.168.1.100 -c 4