

[] Network Namespace

: CentOS 7.6.1810
: root

Network Namespace

가 .
Network Space() , , IP

Host .

Default Network Namespace Check

Local Host



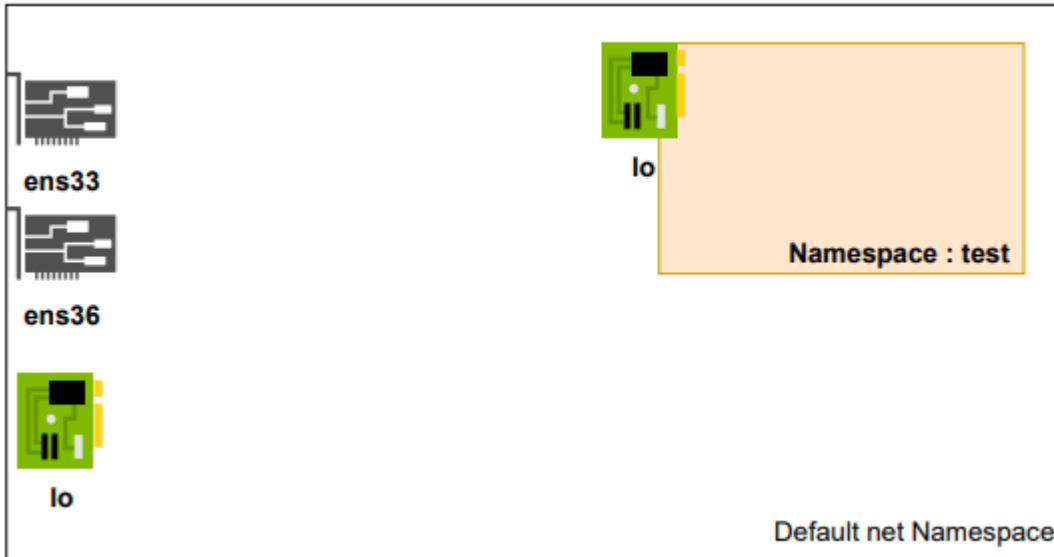
```
# Host          Network Namespace
$ lsns -t net -o pid,uid,user,command
PID  UID  USER  COMMAND
  1    0  root  /sbin/init maybe-ubiquity
```

Host PID 1 (Init)

가 nic(: eth0) lo 가

Create Network Namespace

Local Host



lo 가

```
# test 가 Namespace
```

```
$ ip netns add test
```

```
$ ip netns  
test
```

```
# Check
```

PID 가

lsns

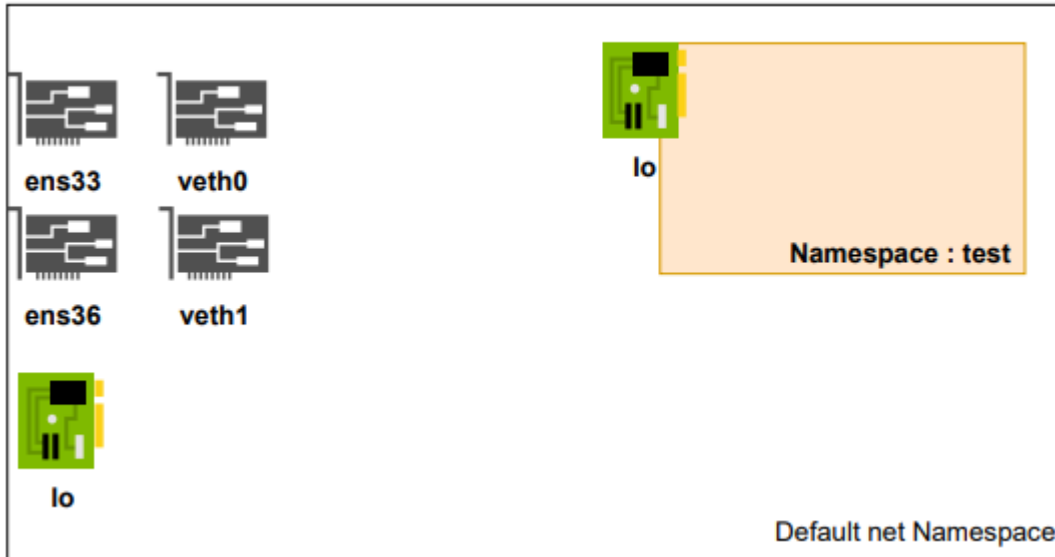
```
$ lsns -t net
```

```
PID USER TYPE COMMAND
```

```
1 root net /usr/lib/systemd/systemd --switched-root --  
system --deserialize 22
```

Namespace Network 1 - 가

Local Host

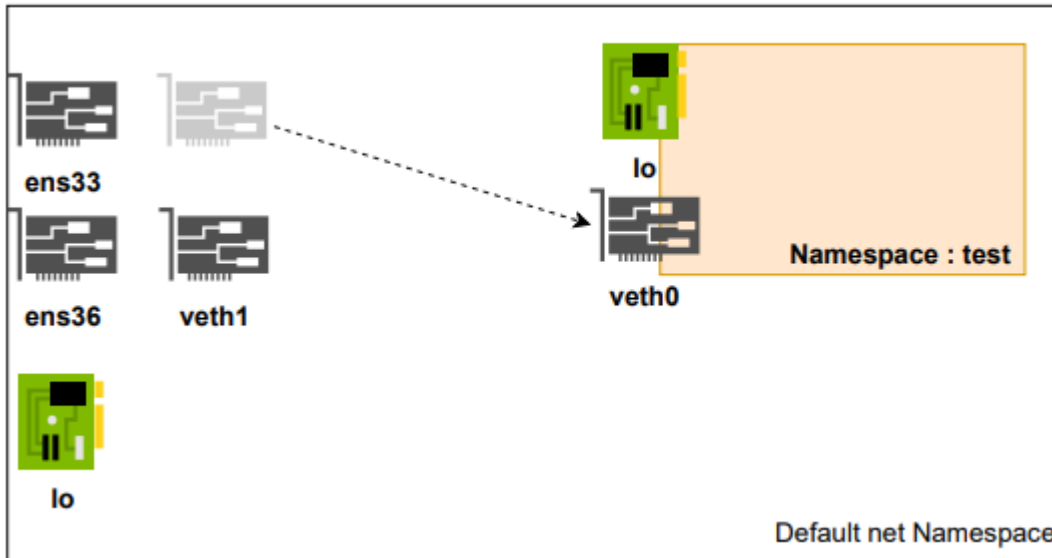


```

가   Network Namespace   가
      veth
veth   HOST <---->
.
# HOST   가   가   . veth type peer pair
$ ip link add veth0 type veth peer name veth1
.
# HOST   veth0/veth1   2   가   가   .
$ ip -br -c addr
lo           UNKNOWN           127.0.0.1/8
ens33       UP                 211.239.150.48/23
ens36       UP                 192.168.0.2/24
veth1@veth0 DOWN
veth0@veth1 DOWN

Namespace   Network   2 -   가
  
```

Local Host



가 , test

```
# veth0 test Namespace Set
$ ip link set veth0 netns test
```

```
# HOST veth0 test namespace
```

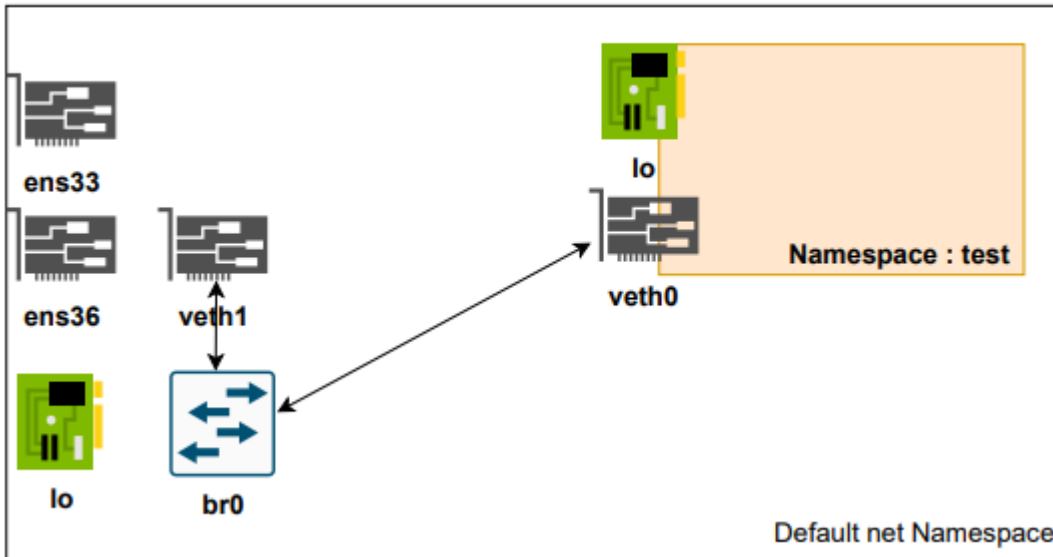
```
$ ip -br -c addr
lo UNKNOWN 127.0.0.1/8
ens33 UP 211.239.150.48/23
ens36 UP 192.168.0.2/24
veth1@if5 DOWN
```

```
test namespace 가 veth0
```

```
# test namespace netns exec
$ ip netns exec test ip -br addr
lo DOWN
veth0@if4 DOWN
```

Namespace Network 3 - bridge

Local Host



```
HOST test namespace veth0 veth1 가
DOWN 가 IP
HOST (bridge)
가
```

Check

```
$ ( 가 ) yum install -y bridge-utils-1.5-9.el7.x86_64
```

```
$ brctl show
```

```
bridge name bridge id STP enabled
interfaces
```

```
가 . br0 HOST
```

Bridge Create && Check

```
$ ip link add br0 type bridge
```

```
$ brctl show
```

```
bridge name bridge id STP enabled
interfaces
br0 8000.000000000000 no
```

가

```
$ ip -br -c addr
```

```
lo UNKNOWN 127.0.0.1/8
```

```
ens33          UP          211.239.150.48/23
ens36          UP          192.168.0.2/24
veth1@if5     DOWN
br0           DOWN
```

```
br0 vethx
```

```
# HOST          veth1  Host  br0
$ ip link set veth1 master br0
```

```
# check bridge veth1 가
$ brctl show
bridge name      bridge id          STP enabled
interfaces
br0              8000.46df623e69e4 no                veth1
```

```
가 , , IP
```

```
ifconfig          net-util          ip
```

```
# netns exec          test Namespace  veth0          IP
UP
```

```
$ ip netns exec test ip addr add 10.10.10.2/24 dev veth0
$ ip netns exec test ip link set veth0 up
```

```
# host veth1          bridge          up
$ ip link set br0 up
$ ip link set veth1 up
```

```
# UP check          가          UP
```

```
$ ip -br -c addr
lo          UNKNOWN          127.0.0.1/8
ens33      UP          211.239.150.48/23
ens36      UP          192.168.0.2/24
veth1@if5  UP
br0        UP
```

```
# test namespace          UP
```

```
$ ip netns exec test ip link
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
5: veth0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
noqueue state UP mode DEFAULT group default qlen 1000
    link/ether f2:1c:09:d4:47:fc brd ff:ff:ff:ff:ff:ff link-
netnsid 0
```

```
# lo          가 DOWN          가
. UP        UNKNOWN
```

```
$ ip netns exec test ip link set dev lo up
```

```
$ ip netns exec test ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN group default qlen 1000
```

```
# Check
```

```
$ ip netns exec test ping 127.0.0.1
```

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
```

```
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.063 ms
```

```
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.058 ms
```

```
# Check 2
```

```
Host
```

```
Gateway
```

```
IP
```

```
Routing
```

```
가
```

```
,
```

```
ip
```

```
#
```

```
IP
```

```
$ ip addr add 10.10.10.200/24 dev br0
```

```
# test          veth0          Ping
```

```
$ ping 10.10.10.2
```

```
ping 10.10.10.2 -c 2
```

```
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
```

```
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=0.073 ms
```

```
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.071 ms
```

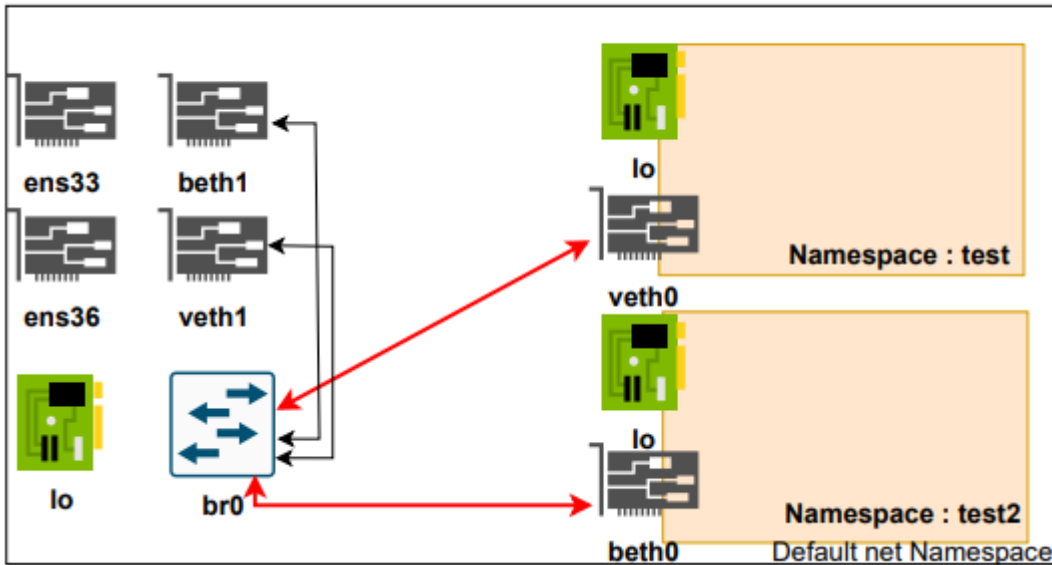
```
--- 10.10.10.2 ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 999ms
```

rtt min/avg/max/mdev = 0.071/0.072/0.073/0.001 ms

Namespace Network 4 -

Local Host



```
test namespace 가
.
가 , 가
# test2 namespace 가 beth0/beth1
# IP test <---> test2 Ping
.
ip netns add test2
ip link add beth0 type veth peer name beth1
ip link set beth0 netns test2
ip link set beth1 master br0
ip netns exec test2 ip addr add 10.10.10.3/24 dev beth0
ip netns exec test2 ip link set beth0 up
ip netns exec test2 ip link set dev lo up
ip link set beth1 up

# test2 namespace
$ ip netns
test2 (id: 1)
test (id: 0)
```



```
$ ip -br -c addr
```

```
lo                UNKNOWN          127.0.0.1/8
ens33             UP                211.239.150.48/23
ens36             UP                192.168.0.2/24
veth1@if5        UP
br0               UP
beth1@if8        UP
```

```
$ brctl show
```

```
bridge name      bridge id          STP enabled
interfaces
br0              8000.2e0e64ccb0e5 no                beth1
                                                         veth1
```

```
# test namespace veth0(10.10.10.2) Ping
```

```
ip netns exec test2 ping 10.10.10.2 -c 2
```

```
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
```

```
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=0.112 ms
```

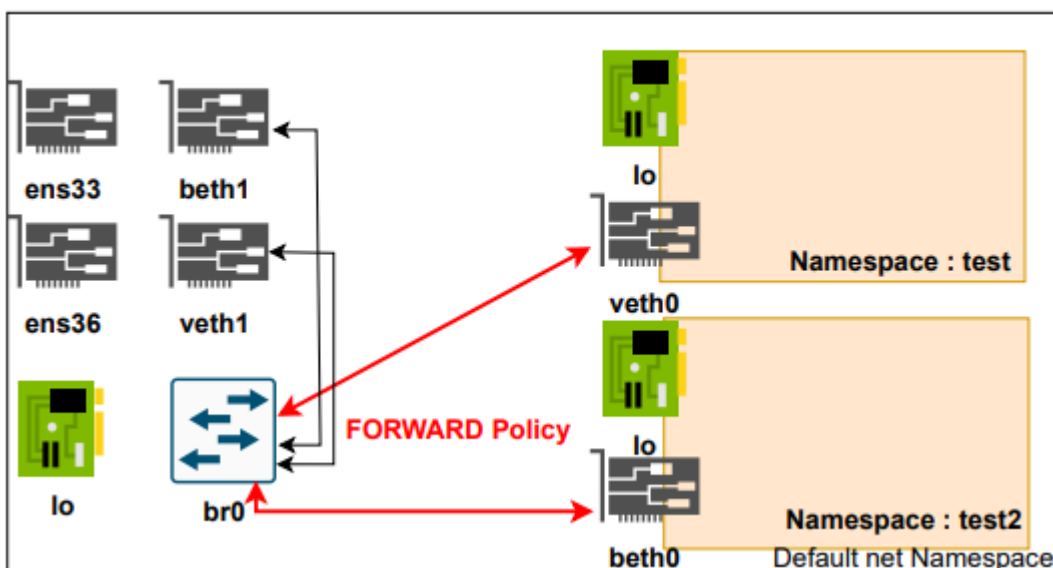
```
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=0.076 ms
```

```
--- 10.10.10.2 ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
```

```
rtt min/avg/max/mdev = 0.076/0.094/0.112/0.018 ms
```

Local Host



```
#  
HOST
```

```
?  
lo ( )
```

```
, Host
```

```
    NAT
ip4          FORWARD      HOST

#          HOST iptables FORWARD      ACCEPT
$ iptables -nL | grep -i forward
Chain FORWARD (policy DROP)

#
$ iptables --policy FORWARD ACCEPT
$ iptables -nL | grep -i forward
Chain FORWARD (policy ACCEPT)

$ service iptables save (          OS          )

#          ip4v.forward
echo 1 > /proc/sys/net/ipv4/ip_forward
sysctl --system

# check
#
```