

Routed network

A routed network is usually only used when a [*Bridged network*](#) is unavailable, either due to hosting provider restrictions or because the libvirt server is connected wirelessly to the LAN. Virtual machines (VMs) have their own IP addresses, but do not bind directly to them. Instead, packets destined for those addresses are [*statically routed*](#) to the libvirt server and forwarded to VMs (without using NAT).

Limitations

Unfortunately, libvirt's built-in routed network automatically inserts iptables rules whether you

want them or not, in an order that is difficult to control. If you would rather be in full control and prevent libvirt from interfering, create a [*Custom routed network*](#) instead.

On a dedicated server, a routed network is only possible when there are enough IP addresses to allocate one per VM. This is not a problem for IPv6, as hosting providers usually provide many free IPv6 addresses. However, extra IPv4 addresses are rarely free. If you only have one public IPv4 address (and need to serve clients over IPv4), either buy more IPv4 addresses or create a [*NAT-based network*](#).

Initial steps

In this example:

- The server has an Ethernet device called `eth0`.
- VMs bind to a virtual bridge called `virbr1`, which is created by libvirt.
- The hosting provider has allocated two address blocks (see [CIDR notation](#)):
 - one public IPv4 address block (`203.0.113.80/28`)
 - one public IPv6 address block (`2001:db8::/56`)
- The server binds statically to `203.0.113.86` and `2001:db8::1`.

To begin, choose which IP addresses to make available to VMs. In this example, a subnet of six IPv4 addresses (`203.0.113.88/29`) will be sliced from the allocated address block and made available to VMs. In an Intranet or home lab, you might choose a whole `/24` subnet (eg, `192.168.50.0/24`).

For IPv6, always use a `/64` subnet prefix to avoid [breaking various IPv6 features](#). In this example, `2001:db8::/56` can be sliced into 256 different `/64` subnets, one of which (`2001:db8:aa::/64`) will be made available to VMs. (Most hosting providers allocate a `/64` block for every server, and should allocate a free `/56` on request.)

Configure static routes

The LAN router is unaware that the subnets chosen above are located on the libvirt server, so network packets can only reach VMs if you configure static routes on the LAN router. In this example, two static routes are required:

- `ip -4 route add 203.0.113.88/29 via 203.0.113.86`

- `ip -6 route add 2001:db8:aa::/64 via 2001:db8::1`

For a dedicated server, ask your hosting provider to configure these routes for you. For an Intranet or home lab, consult the documentation for your router.

! Note

If the router in your home lab is consumer-grade, it might have buggy support for static routes or it might not allow static routes at all. If possible, install alternative firmware such as OpenWRT or Merlin. (A painful last resort is to add static routes to every client on the LAN.)

Configure the virtual network

Create `/tmp/mynetwork1.xml` with the following contents. If only a single protocol version is required, omit either the IPv4 or IPv6 `<ip>` block. (Optionally, use [DHCP host entries](#) to always assign the same IP address to a particular VM.)

```
<network>
  <name>mynetwork1</name>
  <bridge name="virbr1" />
  <forward mode="route"/>
  <ip address="203.0.113.88" netmask="255.255.255.248">
    <dhcp>
      <range start="203.0.113.89" end="203.0.113.94"/>
    </dhcp>
  </ip>
  <ip family="ipv6" address="2001:db8:aa::1" prefix="64"/>
</network>
```

IPv6 addresses are allocated via stateless address autoconfiguration (SLAAC). Alternatively, you may want to allocate IPv6 addresses via DHCPv6 instead, in which case add a `<dhcp>` element with a range of addresses to offer to VMs:

```
<network>
  <name>mynetwork1</name>
  <bridge name="virbr1" />
  <forward mode="route"/>
  <ip address="203.0.113.88" netmask="255.255.255.248">
    <dhcp>
      <range start="203.0.113.89" end="203.0.113.94"/>
    </dhcp>
  </ip>
  <ip family="ipv6" address="2001:db8:aa::1" prefix="64">
    <dhcp>
      <range start="2001:db8:aa::1000" end="2001:db8:aa::1fff"/>
    </dhcp>
  </ip>
</network>
```

Use `/tmp/mynetwork1.xml` to define a new network.

```
# virsh net-define /tmp/mynetwork1.xml
# virsh net-autostart mynetwork1
```

```
# virsh net-start mynetwork1
```

Multiple virtual networks

You can create as many routed networks as required. Simply choose a different name for the network (eg, `mynetwork2`), a different name for the virtual bridge (eg, `virbr2`), and a different range of IP addresses. Also see [Multiple networks](#).

Configure virtual machines

New VM

```
# virt-install --network network=mynetwork1 ...
```

Optionally, pass `--network` more than once to create additional virtual Ethernet interfaces for the VM.


```
# virt-install --network network=mynetwork1 --network network=mynetwork2 ...
```

Existing VM

Open the XML configuration for the VM in a text editor.

```
# virsh edit name-of-vm
```

There should already be an `<interface>` section that configures a virtual Ethernet interface for the VM. Note down the MAC address.

```
<interface type="network">  
  <source network="default"/>  
  <mac address="52:54:00:4f:47:f2"/>  
</interface>
```

To reconfigure the virtual Ethernet interface, replace the `<interface>` section with the following contents. Use the MAC address noted above, otherwise the MAC address of the VM will change.

```
<interface type="network">  
  <source network="mynetwork1"/>  
  <mac address="52:54:00:4f:47:f2"/>  
</interface>
```

To add an additional virtual Ethernet interface, append a new `<interface>` section. libvirt generates a random MAC for the new interface if `<mac>` is omitted.

```
<interface type="network">  
  <source network="mynetwork1"/>  
</interface>
```

Reboot the VM to apply the changes. You may need to amend the VM's network initialization scripts to account for the network interface changes.

[Comments](#) 

[← Previous](#)

[Next →](#)

Version 1.0.1 — Last updated on 2015-12-16.

© Copyright 2015, [Jamie Nguyen](#). Created with Sphinx using a custom-built theme.