

Bridged network

Routed network

▸ **NAT-based network**

Limitations

Configure the default network

Forward incoming connections

Configure IPv6

Configure virtual machines

Custom routed network

Custom NAT-based network

Multiple networks

Appendix

NAT-based network

A [NAT](#)-based network is ideal if virtual machines (VMs) only need outbound IPv4 network access. The libvirt server acts as a router, and VM traffic appears to originate from the IPv4 address of the server.

Limitations

The default virtual network is NAT-based (with a [fragile hook system](#) to forward incoming connections). Unfortunately, it automatically inserts iptables rules whether you want them or not — in an order that is difficult to control — unless you disable the default network completely.

If you would rather be in full control and prevent libvirt from interfering, create a [Custom NAT-based network](#) instead.

Configure the default network

The default NAT-based network should already be available after installing libvirt.

```
# virsh net-info default
Name:                default
UUID:                f63c210f-67e7-4aed-8d21-066d9f
Active:              no
Persistent:          yes
Autostart:           no
Bridge:              virbr0

# virsh net-autostart default
# virsh net-start default
```

If the default network is missing, create `/tmp/default.xml` with the following contents. (Optionally, use [DHCP host entries](#) to always assign the same IP address to a particular VM.)

```
<network>
  <name>default</name>
  <bridge name="virbr0"/>
  <forward mode="nat"/>
  <ip address="192.168.122.1" netmask="255.255.255.0">
    <dhcp>
      <range start="192.168.122.2" end="192.168.122.254"/>
    </dhcp>
  </ip>
</network>
```

Use `/tmp/default.xml` to create the default network.

```
# virsh net-define /tmp/default.xml
# virsh net-start default
# virsh net-autostart default
```

Multiple virtual networks

You can create as many NAT-based networks as required. Simply choose a different name for the network (eg, `default2`), a different name for the virtual bridge (eg, `virbr2`), and a different range of IP addresses. Also see [Multiple networks](#).

Forward incoming connections

❗ Note

This step is optional. It is only necessary if one or more VMs are running services (eg, web applications)

that need to be available over the network.

If one of the VMs has a web application listening on ports `80`/`443`, connections to those ports on the server can be forwarded to the VM using a [fragile hook system](#).

The main limitation is that a specific port on the server can only be forwarded to a single VM. This is problematic if many VMs are fighting over ports `80`/`443`. One option is to forward connections to ports `80`/`443` on the server to a VM running a reverse proxy (eg, NGINX or HAProxy), which can then proxy those connections to other VMs. Alternatively, run a reverse proxy on the libvirt server itself.

If you want more control over your firewall than the hook system can provide, create a [Custom NAT-based network](#).

Configure IPv6

! Note

This step is optional.

NAT makes little sense for IPv6, so create a [Routed network](#) instead. A routed network can be run alongside a NAT-based network if both are needed.

If preferred, you can avoid creating a separate routed network and allow VMs on a NAT-based network to bind to IPv6 addresses. However, no NAT is performed for IPv6 traffic so you still have to configure static routes (see [Routed network](#)). Open the XML configuration for the default network in a text editor.

```
# virsh net-edit default
```

Replace the configuration with the following content. Addresses are allocated via stateless address autoconfiguration (SLAAC).

```
<network>
  <name>default</name>
  <bridge name="virbr0"/>
  <forward mode="nat"/>
  <ip address="192.168.122.1" netmask="255.255.255.0">
    <dhcp>
      <range start="192.168.122.2" end="192.168.122.254"/>
    </dhcp>
  </ip>
  <ip family="ipv6" address="2001:db8::1" prefixlen="64"/>
</network>
```

Alternatively, you may want to allocate addresses via DHCPv6 instead of SLAAC. Add a `<dhcp>` element with a range of addresses to offer to VMs. (Optionally, use *DHCP host entries* to always assign the same IP address to a particular VM.)

```
<network>
  <name>default</name>
  <bridge name="virbr0"/>
  <forward mode="nat"/>
  <ip address="192.168.122.1" netmask="255.255.255.0">
    <dhcp>
      <range start="192.168.122.2" end="192.168.122.254"/>
    </dhcp>
  </ip>
  <ip family="ipv6" address="2001:db8::1" prefixlen="64">
    <dhcp>
      <range start="2001:db8::1000" end="2001:db8::10ff"/>
    </dhcp>
  </ip>
</network>
```

Configure virtual machines

New VM

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

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

```
# virt-install --network network=default --net
```

Existing VM

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

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

To configure a virtual Ethernet interface for the VM, add an `<interface>` section. If required, you can add multiple `<interface>` sections.

```
<interface type="network">  
  <source network="default"/>  
</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.

[< Previous](#)[Next >](#)