

目 录

1	介绍相关的元素和属性	2
1.1	一般性的元数据	2
1.2	控制网络连接的元素	2
1.3	设置 vlan 标签	4
1.4	静态路由	5
1.5	IP 地址	6
2	网络配置举例	7
2.1	基于 NAT 的网络	7
2.2	基于 route 的网络	8
2.3	配置被隔离的网络	8
2.4	使用宿主机的网桥	8

1 介绍相关的元素和属性

对于网络的配置，根元素是“network”，如下所示：

```
<network>
...
</network>
```

1.1 一般性的元数据

首先看下面的例子：

```
<network ipv6='yes' trustGuestRxFilters='no'>
  <name>default</name>
  <uuid>3e3fce45-4f53-4fa7-bb32-11f34168b82b</uuid>
  <metadata>
    <app1:foo xmlns:app1="http://app1.org/app1/">...</app1:foo>
    <app2:bar xmlns:app2="http://app2.org/app2/">...</app2:bar>
  </metadata>
</network>
```

下面是对例子中各元素和属性的介绍：

- network，是网络配置的根元素。
 - ipv6，是 network 的属性。当 ipv6=“yes”时，虚拟机有 IPv6 的网关地址。
 - trustGuestRxFilters，当 trustGuestRxFilters=“yes”时，宿主机将检测并信任虚拟机中接口 mac 地址的变化，并且接受过滤器。
- name，虚拟网络的名称，只能是英文。
- uuid，虚拟网络的 unique identifier。

1.2 控制网络连接的元素

首先看下面的例子：

```
<network>
  <bridge name="virbr0" stp="on" delay="5" macTableManager="libvirt"/>
  <mtu size="9000"/>
  <domain name="example.com" localOnly="no"/>
  <forward mode="nat" dev="eth0"/>
</network>
```

下面是对例子中各元素和属性的介绍：

- bridge 元素
 - name, 建立虚拟网络的网桥的名称, 建议以 “virbr” 开头。
 - stp, 用于决定 Spanning Tree Protocol 是否开启。
 - delay, 用于设置网桥 forward 的延迟秒数。
 - macTableManager, 当 macTableManager= “kernel” 时, 内核将自动管理网络的 mac 地址表。当 macTableManager= “libvirt” 时, libvirt 将管理网络的 mac 地址表, 相对而言性能更好。
- mtu 元素, 用于决定 Maximum Transmission Unit, 默认值为 1500。
- domain 元素
 - name, DHCP 服务器的 DNS domain 的名称。只在 nat 和 route 模式中使用。
 - localOnly, 当 localOnly= “yes” 时, 虚拟机中 DNS 请求只会被它自己的 DNS 服务器解析, 而不会被宿主机的 DNS 服务器解析。
- forward 元素, 如果没有这个元素, 说明这个虚拟机是独立于其他网络的。
 - mode, 决定了 forward 的方式, mode= “nat” 时, 虚拟机对外部网络的连接都会经过宿主机, 宿主机会将虚拟机的 IP 地址变为自己的 IP 地址。在这种模式下, 即使宿主机只能拥有一个 IP 地址, 也可以让多个虚拟机连接 Internet。可以通过 <nat> 这个子元素来设置虚拟机的网络, 通过 <address> 设置虚拟机的 IP 地址, 通过 <port> 来设置虚拟机的端口, 如下所示:

```
<forward mode='nat'>
  <nat>
    <address start='1.2.3.4' end='1.2.3.10' />
    <port start='500' end='1000' />
  </nat>
</forward>
```

当 mode= “route” 时, 虚拟机将通过宿主机的 IP routing stack 连接到网络。如果 dev 有赋值, 比如 dev= “eth0”, 那么虚拟机的网络请求将仅转发到 eth0 网卡上。

当 mode= “open” 时, libvirt 将不会加入任何防火墙规则, 所以也不能设置 dev 属性。

当 mode= “bridge” 时, 虚拟机直接通过网桥连接到外部网络, libvirt 不会管理这个网桥接口。

当 mode= “private” 时, 每个物理网络接口只能被一个虚拟机使用, 不能多个虚拟机同时使用。

当 mode= “passthrough” 时, 虚拟机要使用的物理接口将从 <forward> 元素的 <interface> 子元素中列出的物理接口中选择。每个物理网络接口一次只能由

单个客户机使用，并且只会分配未使用的网络接口。如果当一个域的接口连接时没有可用的物理接口，将导致此次连接失败并同时记录错误。如下所示：

```
<forward mode='passthrough'>
  <interface dev='eth10' />
  <interface dev='eth11' />
  <interface dev='eth12' />
  <interface dev='eth13' />
  <interface dev='eth14' />
</forward>
```

此外，还可以通过 `<pf>` 子元素调用与多个虚拟接口相关联的相应物理接口来指定和单个物理功能相关联的所有虚拟接口。如下所示：

```
<forward mode='passthrough'>
  <pf dev='eth0' />
</forward>
```

当 `mode="hostdev"` 时，该网络模式利用了网络设备的 PCI passthrough 方式。在首先可选地为设备的 mac 地址和 vlan 标签赋值后，从接口池中选择网络设备并将其直接分配给虚拟机。其中接口池由 `<address>` 元素指定，它的属性包括 `<type>`、`<domain>`、`<bus>`、`<slot>` 和 `<function>`。如下所示：

```
<forward mode='hostdev' managed='yes'>
  <driver name='vfio' />
  <address type='pci' domain='0' bus='4' slot='0' function='1' />
  <address type='pci' domain='0' bus='4' slot='0' function='2' />
  <address type='pci' domain='0' bus='4' slot='0' function='3' />
  <address type='pci' domain='0' bus='4' slot='0' function='4' />
</forward>
```

需要注意的是，属性 `<type>` 的值一定要是 “pci”。

可选的，这个接口池也可以由 `<pf>` 子元素定义，用于调用与多个虚拟接口相关联的相应物理接口。如下所示：

```
<forward mode='hostdev' managed='yes'>
  <pf dev='eth0' />
</forward>
```

1.3 设置 vlan 标签

首先看下面的例子：

```
<network>
  <name>ovs-net</name>
  <forward mode='bridge' />
  <bridge name='ovsbr0' />
  <virtualport type='openvswitch' />
```

```

        <parameters interfaceid='09b11c53-8b5c-4eeb-8f00-d84eaa0aaa4f' />
    </virtualport>
    <vlan trunk='yes'>
        <tag id='42' nativeMode='untagged' />
        <tag id='47' />
    </vlan>
    <portgroup name='dontpanic'>
        <vlan>
            <tag id='42' />
        </vlan>
    </portgroup>
</network>

```

从上例可以看出，一个 <vlan> 元素中可以设置多个标签。

1.4 静态路由

静态路由用于向宿主机提供路由信息，该路由信息不能直接到达宿主机，但是可以到达虚拟机，然后再从虚拟机到达宿主机。

首先看一个例子：

```

<network ipv6='yes'>
    <name>nogw</name>
    <uuid>7a3b7497-1ec7-8aef-6d5c-38dff9109e93</uuid>
    <bridge name='virbr2' stp='on' delay='0' />
    <mac address='00:16:3E:5D:C7:9E' />
</network>

```

上述例子中虚拟网络接口没有 IPv4 和 IPv6 的地址。这样的网络可以用于向宿主机提供网络，而这个网络只能通过虚拟机访问。一个既可以访问 guest-only 网络又可以访问与宿主机相连的网络的虚拟机可以作为一个网关。向“主机可见”的网络添加静态路由来提供路由信息，可以使得 IP 包能够从宿主机发送到隐藏网络上的客户机。

看下面一个例子：

```

<ip address="192.168.122.1" netmask="255.255.255.0">
    <dhcp>
        <range start="192.168.122.128" end="192.168.122.254" />
    </dhcp>
</ip>
<route address="192.168.222.0" prefix="24" gateway="192.168.122.2" />
<ip family="ipv6" address="2001:db8:ca2:2::1" prefix="64" />
<route family="ipv6" address="2001:db8:ca2:3::" prefix="64" gateway="2001:db8:ca2:2::2" />
<route family="ipv6" address="2001:db9:4:1::" prefix="64" gateway="2001:db8:ca2:2::3" metric='2' />

```

上面是定义网络的一个片段，显示了静态路由规范和网络地址的 IPv4 和 IPv6 定义。第三个静态路由规范中的“metric”属性用于定义静态路由规范的优先级，数值越低优先级越高，默认值为 1。

1.5 IP 地址

对于 IP 地址的配置，仅适于被孤立的网络或者模式为 “route” 或 “nat” 的网络。首先看一个例子：

```
<mac address='00:16:3E:5D:C7:9E' />
<domain name="example.com" />
<dns>
  <txt name="example" value="example value" />
  <forwarder addr="8.8.8.8" />
  <forwarder domain='example.com' addr="8.8.4.4" />
  <forwarder domain='www.example.com' />
  <srv service='name' protocol='tcp' domain='test-domain-name' target='.' port='
    1024' priority='10' weight='10' />
  <host ip='192.168.122.2'>
    <hostname>myhost</hostname>
    <hostname>myhostalias</hostname>
  </host>
</dns>
<ip address="192.168.122.1" netmask="255.255.255.0" localPtr="yes">
  <dhcp>
    <range start="192.168.122.100" end="192.168.122.254" />
    <host mac="00:16:3e:77:e2:ed" name="foo.example.com" ip="192.168.122.10" />
    <host mac="00:16:3e:3e:a9:1a" name="bar.example.com" ip="192.168.122.11" />
  </dhcp>
</ip>
<ip family="ipv6" address="2001:db8:ca2:2:1" prefix="64" localPtr="yes" />
<route family="ipv6" address="2001:db9:ca1:1::" prefix="64" gateway="2001
  :db8:ca2:2::2" />
```

下面是对例子中各元素和属性的介绍：

- mac 元素，其中 address 属性用于定义网卡的 mac 地址。如果不设置的话 libvirt 会自动产生随机的 mac 地址。
- dns 元素，用于设置虚拟网络的 DNS 服务器。它的属性 enable= “no” 时，虚拟网络将不会有 DNS 服务器。它的属性 forwardPlainNames= “no” 时，不符合规格的域名将不会被宿主机解析，只会在虚拟机中解析。
- forwarder，dns 的子元素，可以有 0 或多个。它有 domain 和 addr 两个属性，用于定义 DNS 服务器。
- txt，dns 的子元素，可以有 0 或多个。它有 name 和 value 两个属性，name 是被查询的名字，value 是被解析后返回的值。
- host，dns 的子元素，属性 ip 用于定义 IP 地址，子元素 hostname 用于定义 IP 地址对应的名称。
- srv，dns 的子元素，其中 name 和 protocol 必须赋值，而 target、port、priority、weight 和 domain 可选。

- ip, 用于定义 address、netmask。localPtr=“yes”时, DNS 服务器被配置为不从由 address 和 netmask/prefix 属性配置的网络转发任何针对 IP 地址的反向 DNS 请求。
- tftp, ip 的子元素, 它的 root 属性必须被赋值, 用于指定 TFTP 服务的根目录。tftp 元素不支持 IPv6 地址。
- dhcp, ip 的子元素, 用于设置虚拟网络的 DHCP 服务。range 子元素用于设置 IP 池的范围, host 子元素可以有 0 到多个。

2 网络配置举例

2.1 基于 NAT 的网络

配置如下所示:

```
<network>
  <name>default</name>
  <bridge name="virbr0"/>
  <forwarder 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:ca2:2::1" prefix="64"/>
</network>
```

加上对 IPv6 的配置, 如下所示:

```
<network>
  <name>default6</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:ca2:2::1" prefix="64">
    <dhcp>
      <range start="2001:db8:ca2:2:1::10" end="2001:db8:ca2:2:1::ff"/>
    </dhcp>
  </ip>
</network>
```

2.2 基于 route 的网络

基于 route 的网络将流量从虚拟网络路由到 LAN。该网络要求 IP 地址范围在主机网络上的路由器的路由表中预先配置。以下例子还指定虚拟机流量仅由 eth1 主机网络设备出去：

```
<network>
  <name>local</name>
  <bridge name="virbr1"/>
  <forward mode="route" dev="eth1"/>
  <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:ca2:2::1" prefix="64"/>
</network>
```

2.3 配置被隔离的网络

这种网络使得虚拟机不能访问 LAN 上的其他机器，但是可以相互访问，也可以访问宿主机。如下所示：

```
<network>
  <name>private</name>
  <bridge name="virbr2"/>
  <ip address="192.168.152.1" netmask="255.255.255.0">
    <dhcp>
      <range start="192.168.152.2" end="192.168.152.254"/>
    </dhcp>
  </ip>
  <ip family="ipv6" address="2001:db8:ca2:3::1" prefix="64"/>
</network>
```

2.4 使用宿主机的网桥

宿主机中存在网桥“br0”，就可以配置虚拟机网络来使用“br0”。配置如下所示：

```
<network>
  <name>host-bridge</name>
  <forward mode="bridge"/>
  <bridge name="br0"/>
</network>
```