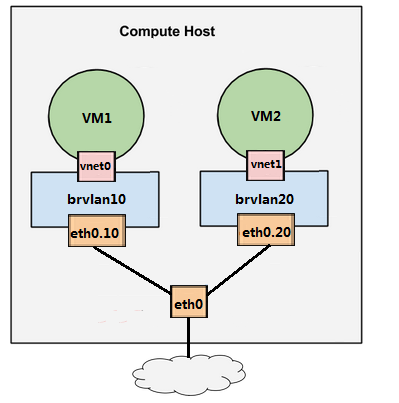
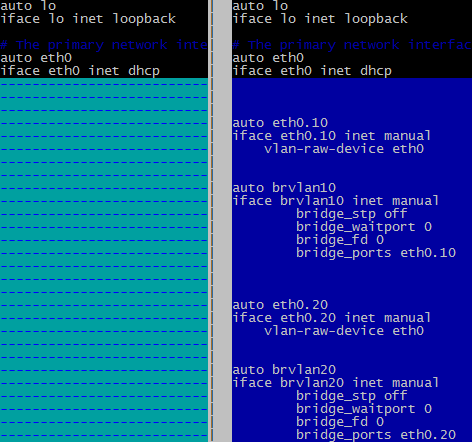


本节我们来看如何在实验环境中实施和配置如下 VLAN 网络  


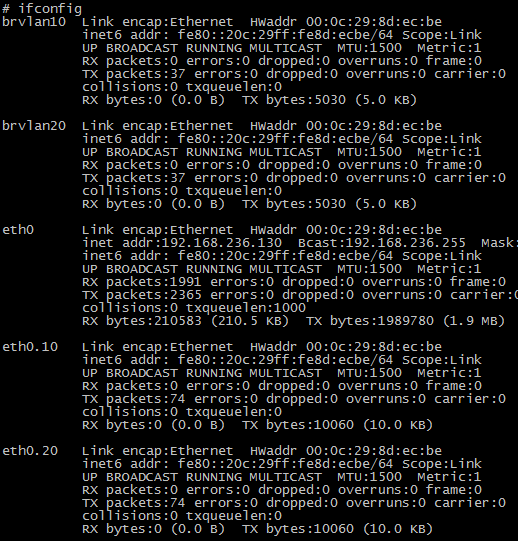
**配置 VLAN**

编辑 /etc/network/interfaces，配置 eth0.10、brvlan10、eth0.20 和 brvlan20。

下面用 vmdiff 展示了对 /etc/network/interfaces 的修改

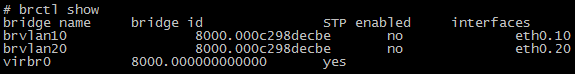


重启宿主机，ifconfig 各个网络接口

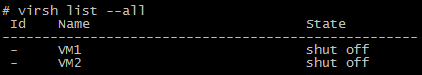


用 brctl show 查看当前 Linux Bridge 的配置。

eth0.10 和 eth0.20 分别挂在 brvlan10 和 brvlan20上 了

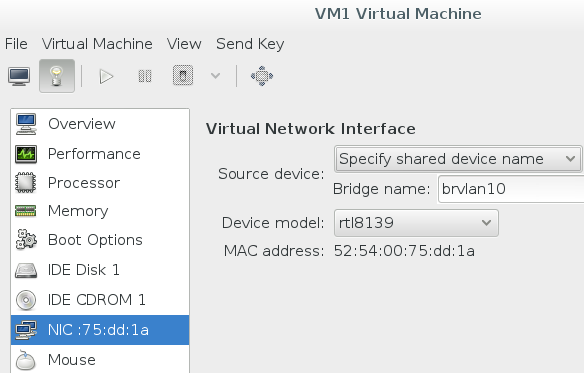


在宿主机中已经提前创建好了虚机 VM1 和 VM2，现在都处于关机状态



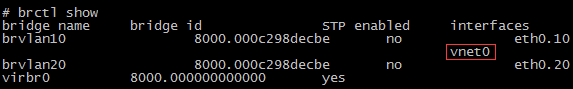
**配置 VM1**

在 virt-manager 中将 VM1 的虚拟网卡挂到 brvlan10 上。

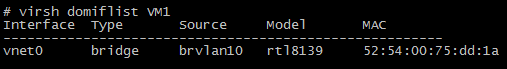


启动 VM1  
http://7xo6kd.com1.z0.glb.clouddn.com/upload-ueditor-image-20160327-1459082009515093103.jpg

查看 Bridge，发现 brvlan10 已经连接了一个 vnet0 设备

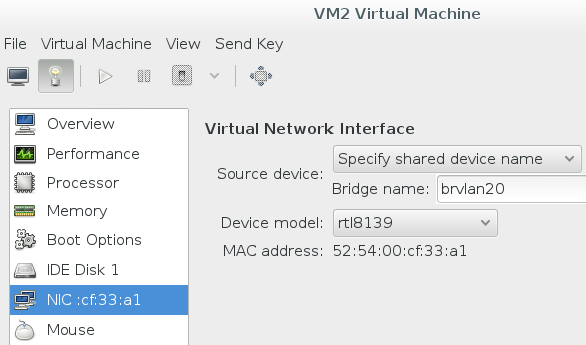


通过 virsh 确认这就是 VM1 的虚拟网卡。



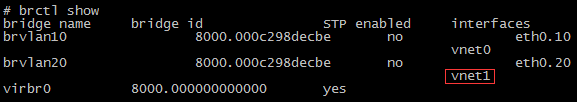
**配置VM2**

类似的，将 VM2 的网卡挂在 brvlan20 上

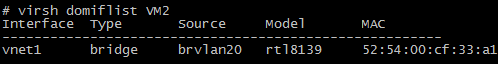


启动 VM2  
http://7xo6kd.com1.z0.glb.clouddn.com/upload-ueditor-image-20160327-1459082012102081391.jpg

查看 Bridge，发现 brvlan20 已经连接了一个 vnet1 设备



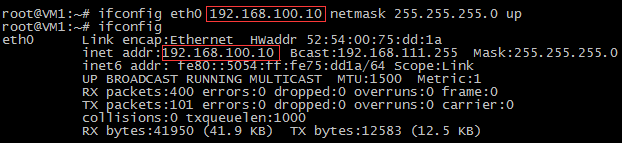
通过 virsh 确认这就是 VM2 的虚拟网卡。



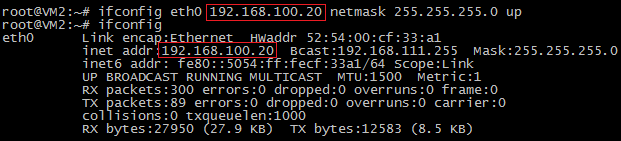
**验证 VLAN 的隔离性**

为了验证 VLAN10 和 VLAN20 之间的隔离，我们为 VM1 和 VM2 配置同一网段的 IP。

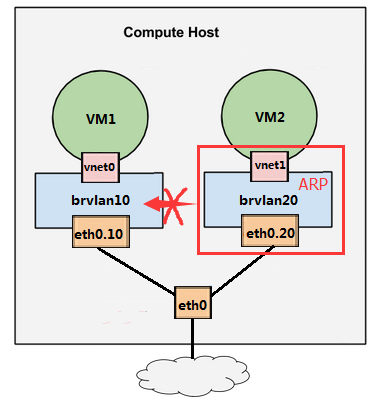
配置 VM1 的 IP



配置 VM2 的 IP



**Ping 测试结果： VM1 与 VM2 是不通的**



原因如下：

1. VM2 向 VM1 发 Ping 包之前，需要知道 VM1 的 IP 192.168.100.10 所对应的 MAC 地址。VM2 会在网络上广播 ARP 包，其作用就是问 “谁知道 192.168.100.10 的 MAC 地址是多少？”

2. ARP 是二层协议，VLAN 的隔离作用使得 ARP 只能在 VLAN20 范围内广播，只有 brvlan20 和 eth0.20 能收到，VLAN10 里的设备是收不到的。VM1 无法应答 VM2 发出的ARP包。

3. VM2 拿不到 VM1 vnet0 的 MAC 地址，也就 Ping 不到 VM1。

**Linux Bridge + VLAN = 虚拟交换机**

现在对 KVM 的网络虚拟化做个总结。

1. 物理交换机存在多个 VLAN，每个 VLAN 拥有多个端口。
2. 同一 VLAN 端口之间可以交换转发，不同 VLAN 端口之间隔离。
3. 所以交换机其包含两层功能：**交换与隔离**。
4. Linux 的 VLAN 设备实现的是**隔离**功能，但没有交换功能。
5. 一个 VLAN 母设备（比如 eth0）不能拥有两个相同 ID 的 VLAN 子设备，因此也就不可能出现数据交换情况。
6. Linux Bridge 专门实现**交换**功能。
7. 将同一 VLAN 的子设备都挂载到一个 Bridge 上，设备之间就可以交换数据了。

总结起来，Linux Bridge 加 VLAN 在功能层面完整模拟现实世界里的二层交换机。

eth0 相当于虚拟交换机上的 trunk 口，允许 vlan10 和 vlan20 的数据通过

eth0.10，vent0 和 brvlan10 都可以看着 vlan10 的 access 口

eth0.20，vent1 和 brvlan20 都可以看着 vlan20 的 access 口