

CHAPTER **4**

**创造你的第三朵云 -  
vNetwork**



## 4-1 认识 vNetwork

传统的实体网络架构，会在服务器与 PC 上面安装实体网卡，并接上实体交换机，通过网线连接，以此形成局域网，彼此交换数据和提供服务。

那么，在虚拟化的环境下，如何让 VM 与外界通信呢？如果一台实体服务器只有一个网卡，但服务器里却有 10 个 VM，要如何分配带宽的优先级？如果有数个实体网卡，又如何让 ESX/ESXi host 里的 VM 做到分流、容错、安全性？这些问题，就是本章要讨论的内容。

### 虚拟化网络环境的基本概念 – Virtual Switch

虚拟化做到了 Server Consolidation，将很多的 VM 集中在一台实体的服务器上，达到资源共享。但无论如何，VM 一定要通过网络才能提供服务，如何让众多的 VM 能够通过实体网卡与外界联系是非常重要的。于是就有了虚拟交换机 (Virtual Switch) 的概念。

Virtual Switch (也称为 vSwitch) 是由 ESX 所虚拟出来的，功能类似一台实体的 Layer 2 交换机，拥有大部分 L2 Switch 的功能，例如 VLAN。但它毕竟是虚拟出来的，所以跟实体交换机并不完全相同。

有了 Virtual Switch 后，接着是实体网卡所扮演的角色。

在虚拟化的环境里，实体网卡会被虚化成 Virtual Switch 上的一个 uplink port，它不再单纯只是一张网卡，而是一个“通道”，你不能在这个实体网卡上指定 IP，因为它的功能是要提供“路”给 VM 来走，使 VM 可以与外界进行沟通。当你的“通道”很多的时候，我们可以让不同的 VM 走不同的路，某一条路走不通了（网卡损坏或线路问题），VM 可以改走不同的道路。

如果不能在实体网卡上面指派 IP，那我们早先在安装 ESX 时所输入的 IP 地址又是什么（192.168.1.101/102）？答案是当时所指定的 IP 是在 Service Console 上的虚拟网卡，别忘了 COS 也是一个 VM，它也需要有一个 IP 地址，才能让我们远程联机管理 ESX host。而当时选择的实体网卡只是让你决定一个通道，要让 COS 走哪一条路与外界沟通。

在图 4-1 的范例中，左侧有 3 个虚拟机，假设每个 VM 都配置了一个虚拟网卡，然后将 VM 的虚拟网卡接上虚拟交换机 (vSwitch)，再来将实体网卡也接上同一个 Virtual Switch，这样 VM 便可以通过 vSwitch 的 uplink port (实体网卡) 来提供服务给 Client 端了。

### vNetwork 的几个名词解释

**vNetwork Standard Switch:** 简称 vSS，就是 VI3 时代的 Virtual Switch (VS)，因为在 vSphere4 后新增了 vDS，所以将一般的虚拟交换机改称为 vSS。承接图 4-1，

我们可以将中间的 Virtual Switch 称为一个 vSS，在一个 ESX/ESXi host 中，我们可以视情况所需创造出许多 vSS，如图 4-2 所示。

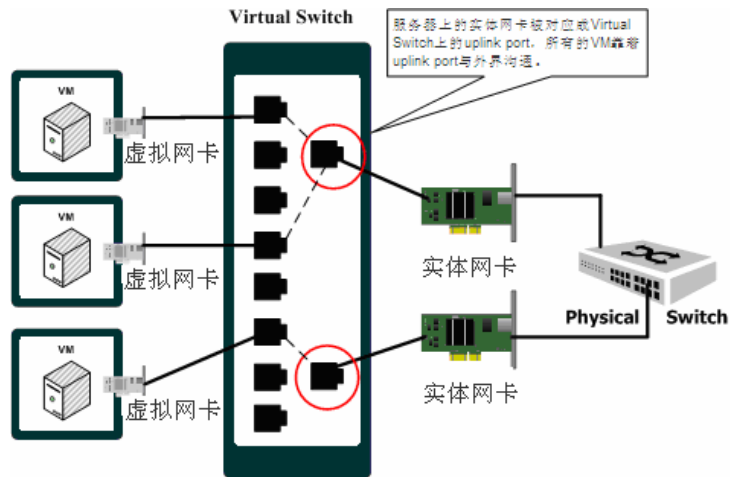


图 4-1

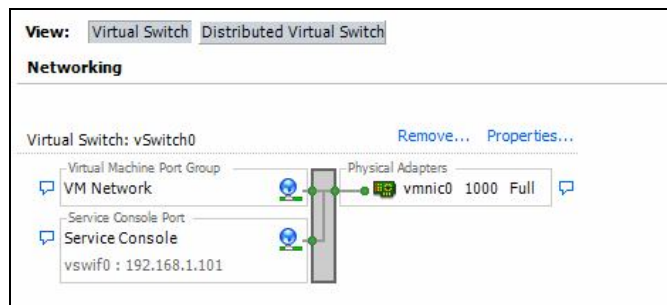


图 4-2 vSS

**vNetwork Distributed Switch:** 简称 vDS 或 DVS (Distributed Virtual Switch)，这是新的功能，可以让虚拟交换机看起来是一个横跨不同 ESX host 的大型 Switch，便于管理员统一针对它进行设置，简化了以往 Virtual Switch 必须在每个 ESX host 上产生，并且每个单独 host 进行设置的管理困扰，如图 4-3 所示。

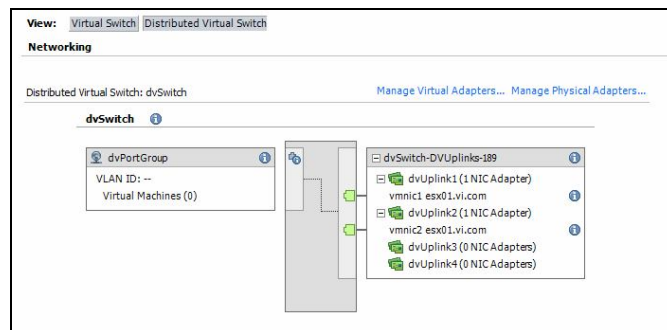


图 4-3 vDS

**Virtual Switch ports:** 每一个虚拟交换机都可以指定网络端口的数量，在 ESX/ESXi 4 中，一个 vSwitch 最多可以拥有 4088 个 ports。

**vmnic:** 你一定以为 vmnic 是虚拟网卡吧？错了，vmnic 指的是实体的网卡，编号从 vmnic0 开始，如果实体服务器有 6 个网络端口，就会看到 vmnic0~vmnic5，如图 4-4 所示。

Device	Speed	Configured	Switch	MAC Address
<b>82545EM Gigabit Ethernet Controller (Copper)</b>				
vmnic5	1000 Full	1000 Full	None	00:0c:29:11:e5:55
vmnic4	1000 Full	1000 Full	None	00:0c:29:11:e5:4b
vmnic3	1000 Full	1000 Full	None	00:0c:29:11:e5:41
vmnic2	1000 Full	1000 Full	None	00:0c:29:11:e5:37
vmnic1	1000 Full	1000 Full	None	00:0c:29:11:e5:2d
vmnic0	1000 Full	1000 Full	vSwitch0	00:0c:29:11:e5:23

图 4-4

**Virtual NIC:** 也可以叫 vNIC。Virtual NIC 指的才是虚拟网卡。在 ESX/ESXi 4 里，一个 VM 最多可以虚拟出 10 个网卡，而每个 vNIC 都拥有自己的 MAC Address。由于实体网卡 (vmnic) 是 vSwitch 上的 uplink port，所以真正的 IP 地址指定是在 VM 的 vNIC 上。

Q: 虚拟网卡的 MAC Address 是否可以改变呢?

网卡的 MAC Address 前三个字节是 Company ID，可以辨识出网卡的制造厂商。VMware 是虚拟网卡的制造厂商，所以也会有 Company ID。VMware 的 vNIC MAC Address 的前六码为 00-0C-29 和 00-50-56 两组。

00-50-56 这组是可以手动更改的，范围为 00:50:56:00:00:00 ~ 00:50:56:3F:FF:FF。但是如果自定义不同的 MAC address (不属于 VMware 的 Company ID 范围)，则需要 Guest OS level (Windows 或 Linux) 这一层进行修改。

在图 4-4 中，看到的实体网卡 vmnic0 ~ vmnic5 MAC Address 都是 00:0C:29 开头，为什么呢？因为我们的 ESX 是以 VMware Player 安装成 VM。ESX 上所有的实体网卡都是虚拟出来的，当然就是 vNIC 的 MAC Address。

了解了 vNIC 和 vmnic，现在可以将图 4-1 重新绘制为图 4-5，更加简洁。

**NIC Teaming:** 当多个 vmnic 指派给一个 vSwitch 时，就表示接在这个 vSwitch 上面的 VM 有了“不同路线”的支持机制。前面提到 uplink port 是 VM 的“通路”，如果只有一条路大家共享，可能就会发生堵车，或是此路不通就无路可走的情况。如果有两个以上的 vmnic，则可以做到平时分流、损坏时互相支援的机制，例如图 4-6，

在一个实体网卡损坏的情形下 VM3 仍然能够自动切换到另一条路持续地提供服务。

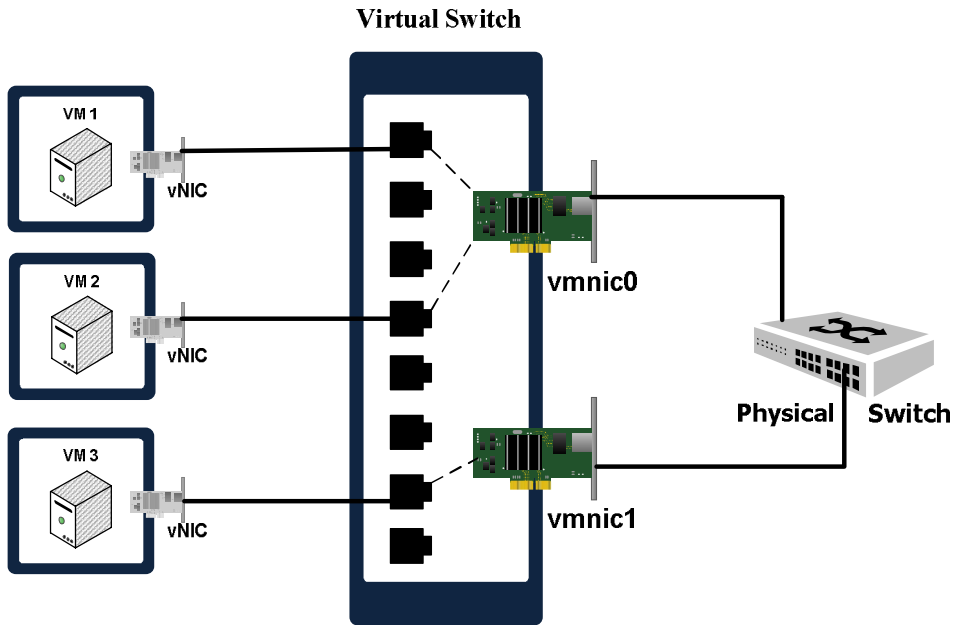


图 4-5

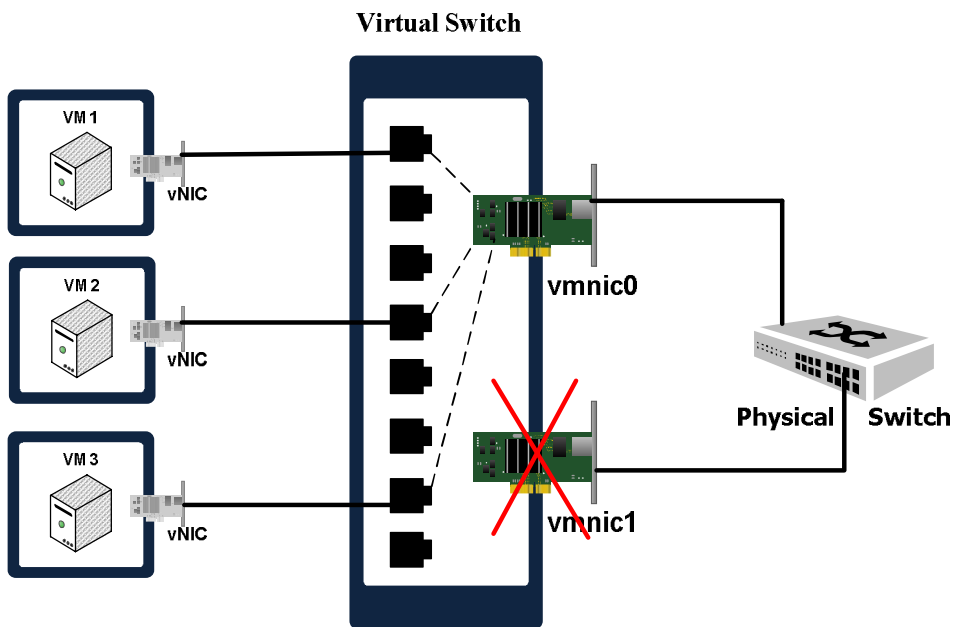


图 4-6

Q: NIC teaming 要如何设置才能达到负载均衡机制?



基本上不需要设置，当一个 vSwitch 拥有多个 vmnic 时，会自动成为 NIC teaming。VMware ESX/ESXi 的网络负载均衡有三种方式：

- **Originating virtual port ID**: VMkernel 以 vSwitch port 为 hash 标的，决定哪个 port 走哪条路，当 vNIC 使用那个 port 时就会走已经选定好的 vmnic。
- **Source MAC hash**: VMkernel 以 VM 的 vNIC MAC Address 为 hash 标的。
- **IP base hash**: 以 VM 的 IP 和 client 的 IP 两者一起为 hash 标的。

默认值是 Port ID base，原则上不需要特别设置负载均衡方式，Port ID base 与 Source MAC 负载均衡模式类似，而第三种 IP base 则适用于特定的环境。前两种负载均衡都是让不同的 VM 走不同的路，IP base 则可以让一个 VM 同时走很多条路（vmnic），前提是实体 Switch 必须支持 802.3ad（LACP）的功能，如图 4-7 所示。

\*注意，负载均衡只管 VM 从 uplink port 出去的流量，没有办法管从外面进来的。

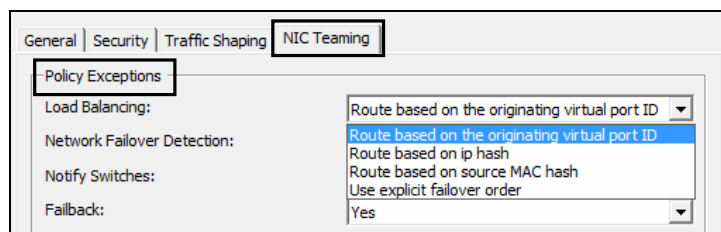


图 4-7

实际中，ESX/ESXi host 配置多个实体网卡进行 NIC teaming 时，通常会避免将属于同一张实体网卡上的多个网络端口 teaming 在一起。也就是说，要让 vSwitch 的 uplink port 实现 NIC teaming，请尽量让不同实体网卡的网络端口在同一个 vSwitch 上。例如图 4-8，服务器上有两张双网络端口的实体网卡和一个 on board 网卡，在进行 vmnic 的 NIC Teaming 时，就可以将不同实体网卡的网络端口指派在同一个 vSwitch 上，或是内置网络端口与插卡的网络端口在同一组，成为某个 vSwitch 的 uplink ports。

这种做法的好处是不会因为一张实体网卡坏掉而导致 vSwitch 的 uplink 都失效，因为 Teaming 在一起的 vmnic 分属不同的实体网卡。Vmware 还建议实体网卡最好也分属不同的厂家、不同的型号。这样还可以避免因为韧体和芯片设计瑕疵、制造质量瑕疵等问题导致服务器上的实体网卡同时发生状况的不幸情形。

**Port Group**: 在一个 vSwitch 里，我们可以将一些 VM 组织起来，成为一个 port group，然后针对整个 port group 应用网络原则与设置，例如 VLAN、Security 与 Traffic Shaping。一个 vSwitch 可以包含多个 port group，例如图 4-9。

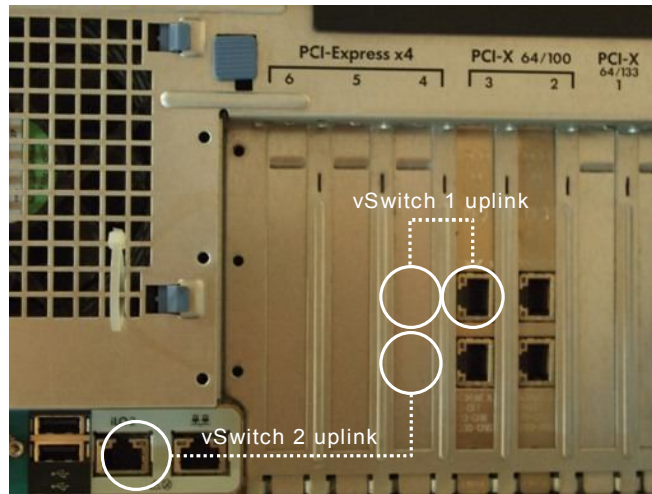


图 4-8

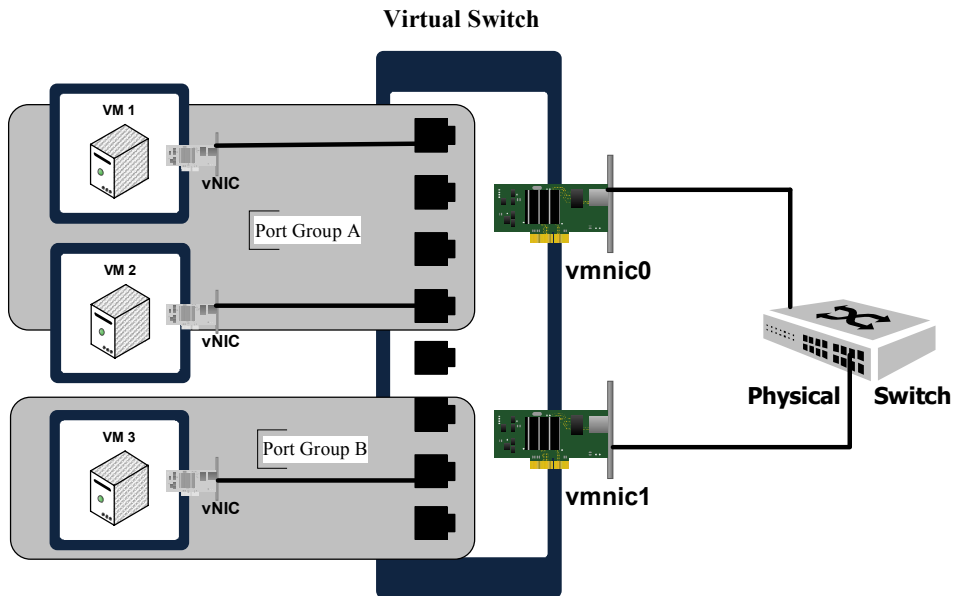


图 4-9

**Traffic Shaping:** 可以针对 Port Group (PG) 进行流量的控管，例如，PG A 为 Production VMs，PG B 是 Test VM，相形之下不是那么重要，当网络带宽紧张时，就可以对 PG B 的 VM 限制流量，使它不要使用超过固定带宽（图 4-10）。

**VLAN:** VLAN 在企业网络环境中应用很普遍，一般用来解决网络性能（只有同样 VLAN 的成员可以收到 Ethernet Frame）以及安全性（有效隔离 Broadcast Domain）的问题。在一个 vSwitch 上可以对不同的 Port Group 定义不同的 VLAN (802.1Q)，与实体 VLAN 相同，分属不同 Port Group 的 VM 彼此之间无法互通(除非有 Routing)。VLAN 的配置在虚拟环境下有三种不同方式：

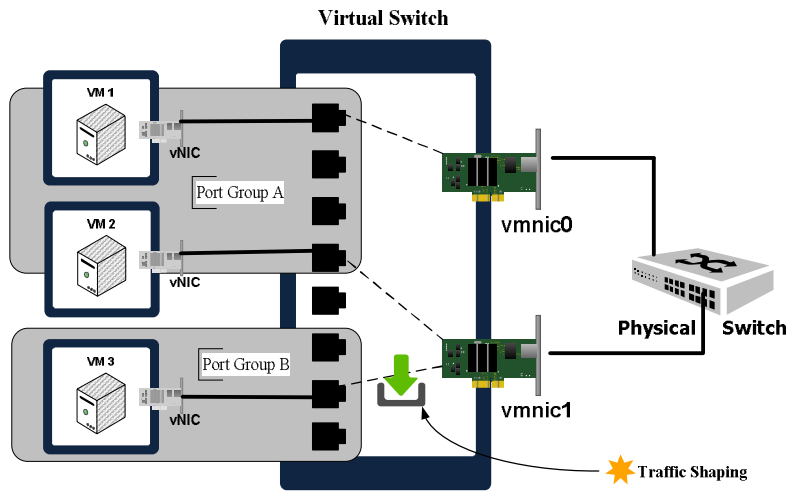


图 4-10

- VST (Virtual Switch Tagging)**：在 vSwitch 上定义 VLAN 的做法，称为 VST。各个 Port Group 都可以给一个 VLAN ID (1~4094)，由于通过 uplink port 载送不同的 VLAN ID，所以 vmnic 必须接在实体 Switch 的 trunk port 上，而不能是属于实体 Switch 的某个 VLAN port，请参考图 4-11。采用 VST 的方式做 VLAN，因为实际上 vSwitch 是由 VMkernel 在运行，VMkernel 必须要做 tagged 和 untagged 的操作，所以会消耗掉一些实体 ESX host 的性能。好处是不同的 Port Group 有不同的 VLAN tag，均能通过 vmnic 载送，VM 在切换与配置方面较灵活。

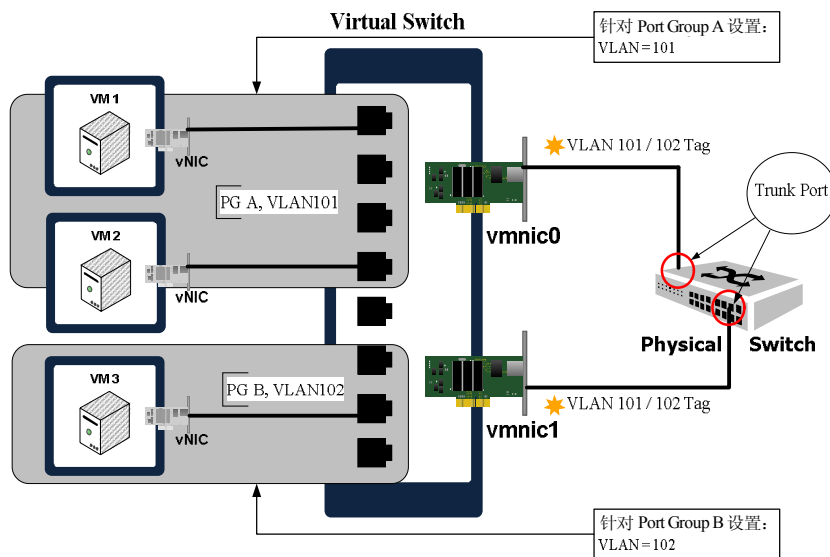


图 4-11

- EST (External Switch Tagging)**：实体交换机设置 VLAN，然后 vSwitch 不做操作。这种做法称为 EST，不用在 vSwitch 上给 VLAN ID，vmnic 不用



接在 trunk port 上。接在实体 Switch 上的 vmnic 属于哪一个 VLAN，通过该 vmnic uplink 的 VM 就会直接属于该 VLAN 成员，例如图 4-12。

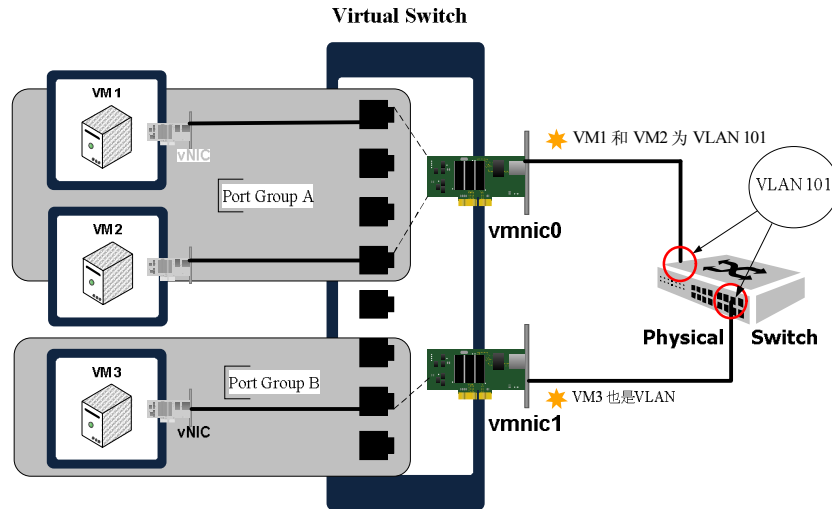


图 4-12

- **VGT (Virtual Guest Tagging)**: EST 与 VST 都是由 Switch 来 tagged/untagged 的, OS 层不需要认得 802.1Q Frame, 但 VGT 则是由 Guest OS 自行负责 tagged 和 untagged 的, 而不通过 Switch level 来做 VLAN, 一般而言较少采用这种方式。

但所有的 VM 一定要在同一个 vSwitch 里面, 然后再区分 Port Group 吗? 当然不是。你也可以在 ESX host 里有多个 vSwitch。注意 vNetwork 并没有制式的配置, 只要符合需求即可, 例如图 4-13。

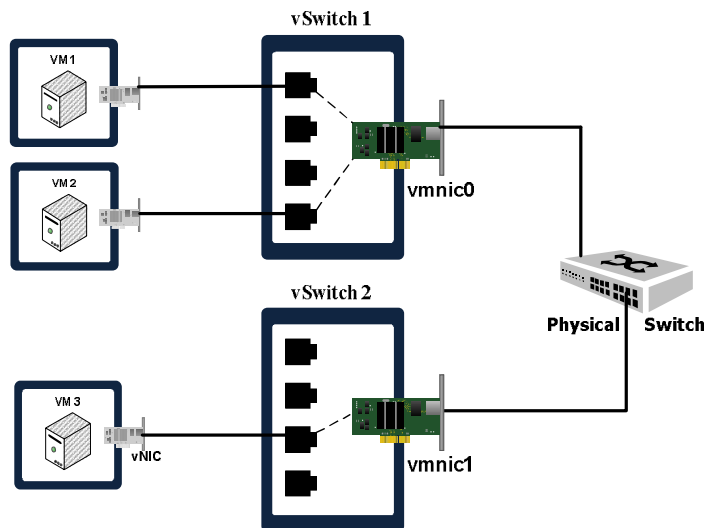


图 4-13

图 4-13 的配置, 在一个 ESX/ESXi host 上创建 2 个 vSwitch, 每个 vSwitch 各只

有一个 Port Group,确实地将 VM3 与 VM1/2 的 uplink 分开来,这样便不需要针对 Port Group 应用细部设置。但是这样配置的缺点是,因为实体网卡不足(只有两个 vmnic),每个 vSwitch 的 uplink 都没有负载平衡的功能和网络 fail over 的机制,因为一个 vSwitch 就只分配到一个 vmnic,无法运行 NIC Teaming。

Q: 可以将一个 vmnic 指派给两个 vSwitch 使用吗?



不行,我们可以将多个 vmnic 给一个 vSwitch,达到 NIC Teaming 的功能,但是却没有办法将一个 vmnic 分给多个 vSwitch 来使用。

也就是说,今天这个 vmnic 是某个 vSwitch 的 uplink port,它就不能同时又属于另一个 vSwitch 的 uplink port。当然,你可以随意更动哪个 vmnic 要属于哪一个 vSwitch,这个部分是可以调整的。

Q: 那么一部实体的 ESX host 应该需要多少实体网卡呢?



建议是越多越好,规划分配较有弹性,在一个硬件网络资源不足的情况下,无可避免地会造成一些安全性或性能问题,因为大家都要使用少数的 uplink port。举例来说,如果你的 ESX 有 8 个实体网卡,就可以做到 2 for COS、3 for VMkernel (IP Storage、vMotion、FT)、3 for VM port group 的配置。

当然,假如有 10GbE 的网络那就更好用了,整个带宽提升很大,通通一起载送都没有问题,顶多再针对每种 Connection Type 和 port group 使用 Traffic Shaping 限制不同的流量。假如两个 10GbE 更妙,一个专门给 VM 使用,一个专门给 COS 和 VMkernel,然后两个 uplink 再设置互援(为彼此的 Standby)即可。

Q: 如果一个 vSwitch 没有 uplink port,会造成什么样的情况?



在创建一个 vSwitch 时,如果默认不勾选任何一个 vmnic,则表示这个 vSwitch 没有任何 uplink,是一个 Internal Only 的 vSwitch。

这样的 vSwitch,接在上面的 VM 只能自成一个内部的 LAN,没办法跟外界沟通,因为没有 uplink 可以走出去。但是也因为不与外界沟通,没有实体 TCP/IP 会产生的碰撞(collision)问题,内部网的 VM 彼此之间传送数据,性能会非常好。

另一种情况是,可以用一个拥有双虚拟网卡(vNIC)的 VM,分别将一个接往有 uplink 的 vSwitch,另一个 vNIC 接上 Internal vSwitch,形成一个 NAT 的环境。这样 Internal vSwitch 上的所有 VM 都是 NAT clients,都可以通过 NAT Router VM 来与外界的实体网络沟通,并受到保护。

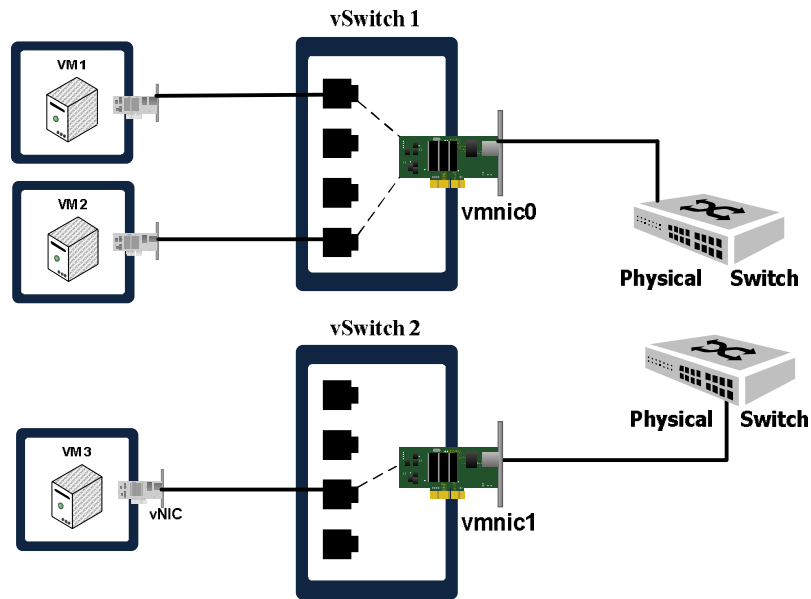


图 4-14

不同 vSwitch 上的 vmnic 连接不同实体 Switch 可以吗？行。那么同一个 vSwitch 的多个 vmnic 分别连接不同实体 Switch 呢？也行。

怎样去运用 vSwitch，使其与实体网络结合，并没有一定标准配置的答案。所以，我们在规划虚拟网络环境时，要针对 vmnic 的实际数量、VM 的用途、VLAN 的设置、实体交换机的设置等进行通盘考虑，规划最适合公司环境的配置方法。

### vSwitch 的 Connection Type

在创建 vSwitch 时，首先必须从三种 Connection Type (Virtual Machine、VMkernel、Service Console) 中选择一种创建，如图 4-15 所示。

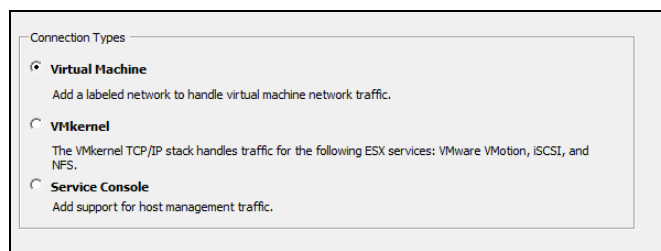


图 4-15

这并不是说从此以后该 vSwitch 就只能有一种 Connection Type，可以视需要而定，随时在 vSwitch 里再增加不同的 Connection Type。

那么 Connection Type 的用途是什么呢？我们一个一个来看。

**Virtual Machine Port Group:** 选择这一种的话，vSwitch 将会产生 VM 的 Port Group，专门给 VM 来连接使用。如图 4-16 所示，初期只会有一个 Port Group 可以放

VM，如果有需要可随时新增不同的 Port Group。请大家再回头看图 4-5，就是只有一个 Virtual Machine Port Group 的情况，而图 4-9，则是一个 vSwitch 包含了两个 Port Group。

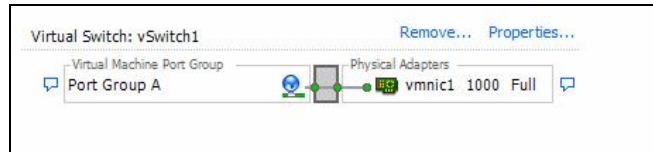


图 4-16

**VMkernel Port:** 许多人不是很了解何时该创建 VMkernel Port 以及它的用途。请各位将它想象成：VMkernel 也拥有虚拟的网卡，需要一个 IP，接上 vSwitch 走 uplink 去对外访问，如图 4-17 所示。有两种情形，必须在 vSwitch 上创建 VMkernel Port：

- **vMotion:** vMotion 在线转移 VM 的行为，其实是将 ESX host 的内存数据状态通过 IP 网络从 host A 传送到 host B，所以双方 host 的 VMkernel 都需要有 IP 地址，并且经由 vSwitch 通过 vmnic uplink 互通，即可传送 VM 内存状态（vMotion 的运行原理及限制将在第 8 章进行说明）。
- **访问 iSCSI/NFS Storage:** 如果存放 VM 文件的 Storage 是 iSCSI/NFS，用的是 IP SAN 而不是 FC SAN，那么也需要创建 VMkernel Port，让 VMkernel 可以识别到 IP 存储设备（详情见第 5 章）。

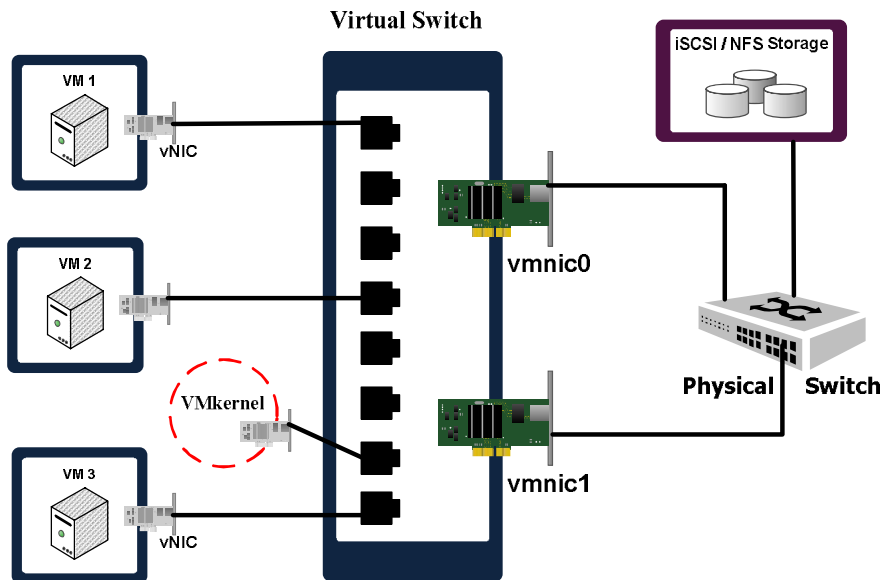


图 4-17 VMkernel Port 示意图

而某些时候，我们也会需要多个 VMkernel Port，这样的做法是为了将 vMotion 与 iSCSI/NFS 的流量隔开，避免通通经由同样的 vmnic 造成性能与安全性的问题。请各位想象，就好似在 VMkernel 上装了多个虚拟网卡，分接到不同的 vSwitch 上，各

走各的路，例如图 4-18。

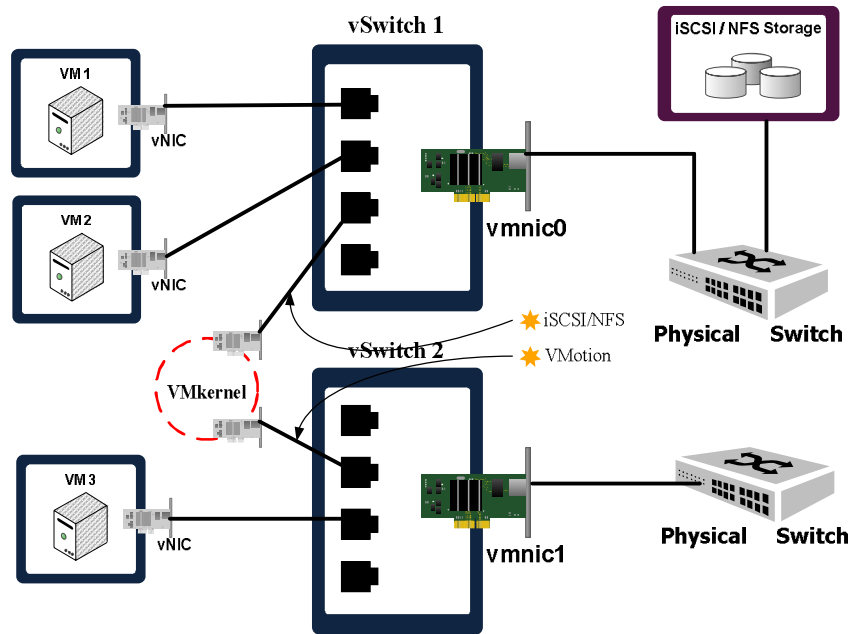


图 4-18 vMotion 与 IP Storage Traffic 分开

**Service Console Port:** 当 ESX 安装完成时，Host 其实就已经有第一个 vSwitch 了，叫做 vSwitch0。这个 vSwitch 默认会有两种 Connection Type：一个是给 VM 用的，默认叫做 VM Network 的 Port Group；另一个则是 Service Console Port。图 4-19 在 Service Console 下有一个 vswif0：192.168.1.101，这个 IP 指的并不是 vSwitch 上的 IP，vSwitch 本身是没有 IP 的，此 IP 是 COS 的虚拟网卡的 IP 地址。

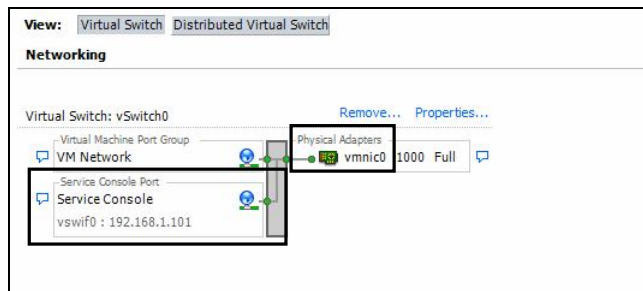


图 4-19

**注：**安装 ESX 才会有 Service Console port，安装 ESXi 则没有。

不知道大家还有没有印象，在安装 ESX 时，要求输入一个 IP 地址，然后选择 vmnic，这个 IP 其实是指定在 COS 上，也就是说，安装好 ESX，其实就创建好了 Service Console Port，否则，我们要如何用 vSphere Client 连到 ESX host 进行管理呢？所以，各位可

以想象成 COS 有一个虚拟网卡,接上了 vSwitch,然后就可以通过 vmnic 联机到 COS,如图 4-20 所示。

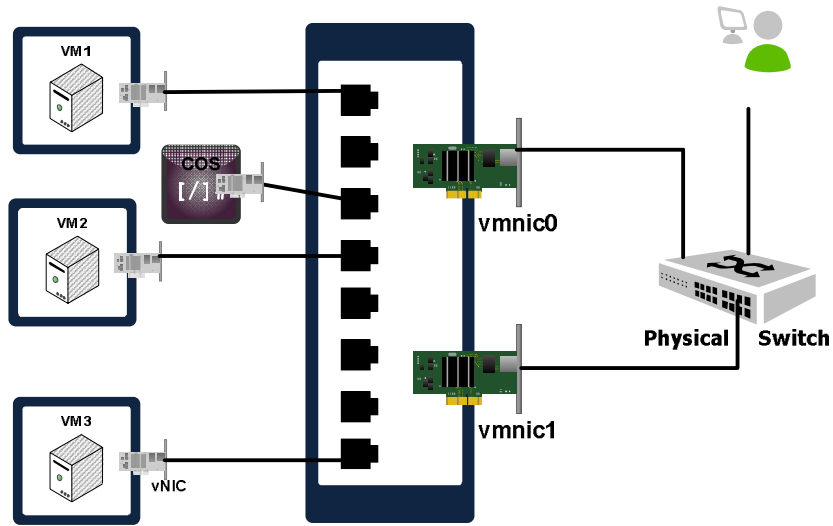


图 4-20

某些时候,也会需要第二个 Service Port,走不同的 uplink。例如,VMware HA 是使用 COS 来作 Heartbeat,就需要(非必要)新增第二个 Service Console Port,以避免 COS Heartbeat 通讯的 SPOF(单点故障)情形发生。请各位想象就如同在 COS 上新增了第二个虚拟网卡,然后将它连接到不同的 vSwitch 走不同的 uplink 即可(见图 4-21)。

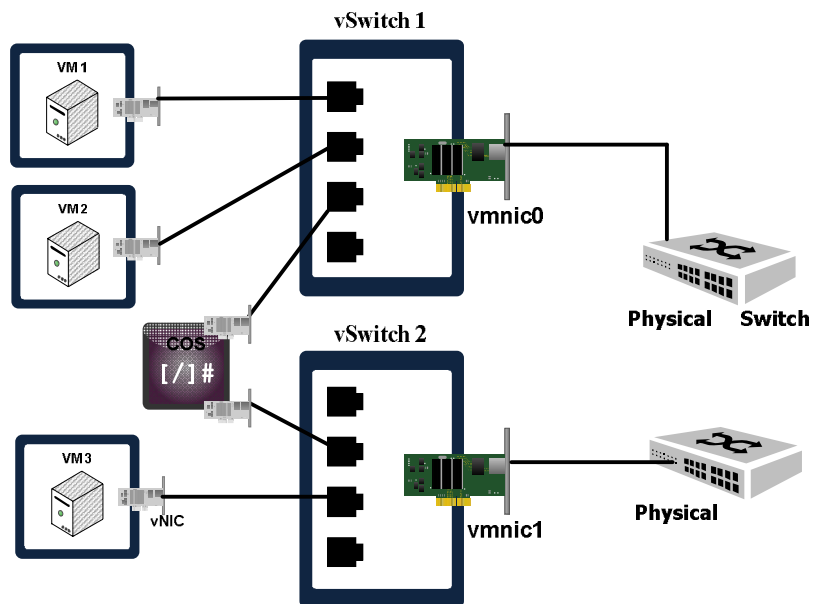


图 4-21

在一个网络资源有限的情况下（例如范例中实体网卡只有 2 个），如果要实现每一种应用，势必会是一个混合类型，也就是三种 Connection Type 都出现在同一个 vSwitch 中，并共享 vmnic，如图 4-22 所示。

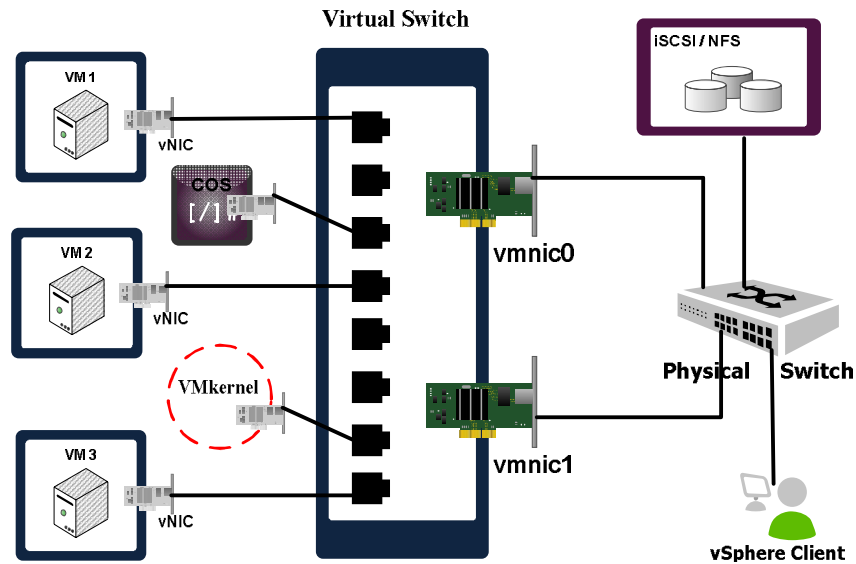


图 4-22

## 4-2 设置 vNetwork Standard Switch

俗话说：“一图解千文”。上一节用了许多辅助图片来介绍 vSS 的运作，希望能让读者对 Virtual Switch 有清晰的虚拟化网络概念。接下来，就要实际操作，针对范例的环境创建 vSwitch、设置各种 Connection Type。

### 新增 vSwitch、VM Port Group

1. 登录 vCenter，单击 **ESX01 host**，再单击 **Configuration** → **Networking**，如图 4-23 所示。

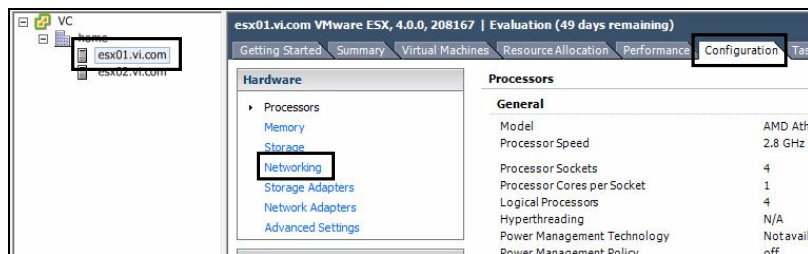


图 4-23

- 在 Networking 里，看到的就是第一个 vSwitch (vSwitch0)，这个是安装好 ESX 时就有了，默认是有一个 Service Console Port 和 VM Port Group，我们现在要将 VM Network 这个 Port Group 删除，代表 vmnic0 纯粹当成 Management Network 来使用，不掺杂其他 VM 的 Network Traffic。请选择 **Properties**，如图 4-24 所示。

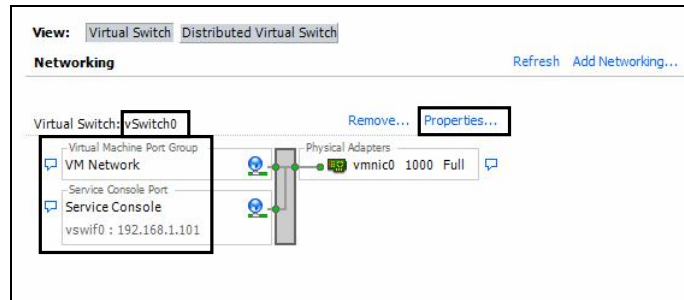


图 4-24

- 单击 VM Network 这个 Port Group，再单击 **Remove** 按钮，如图 4-25 所示。

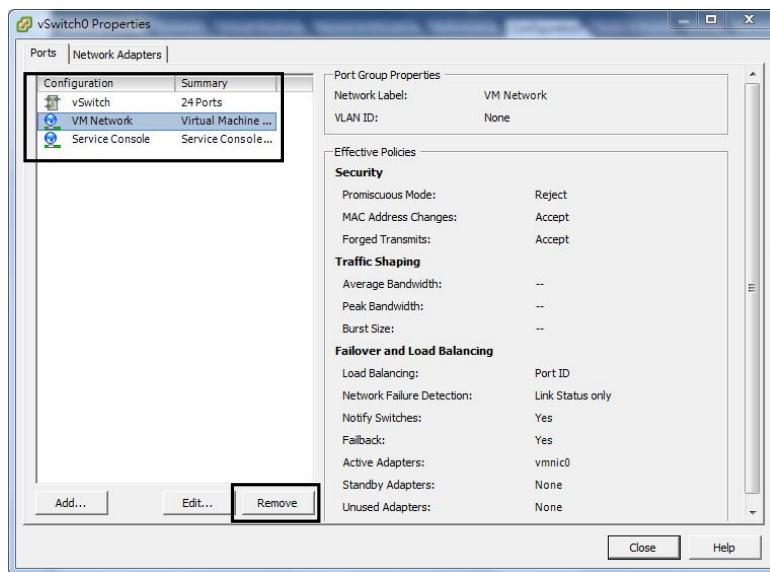


图 4-25

- 出现确认是否要删除的对话框，单击“是”按钮，如图 4-26 所示。

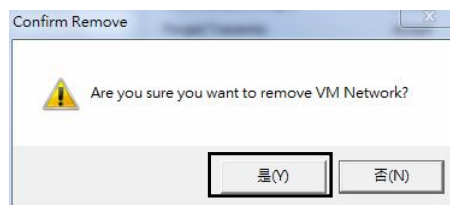


图 4-26



- 删除后，发现 VM Network Port Group 不见了，只剩下 Service Console Port，如图 4-27 所示。

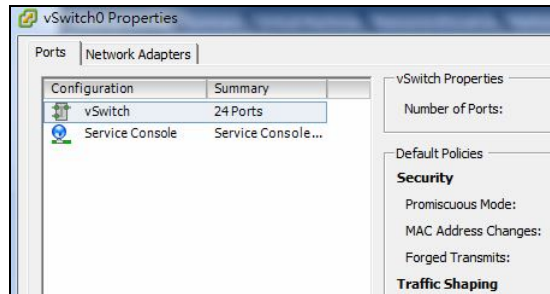


图 4-27

- 回到 vSwitch 界面，确认只剩下 Service Console 走 vmnic0 uplink。再添加一个 vSwitch，请单击 **Add Networking** 按钮，如图 4-28 所示。

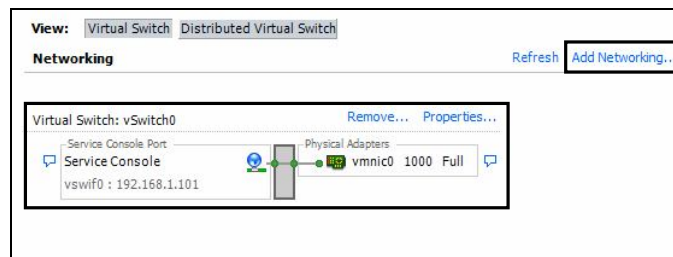


图 4-28

- 首先要选择一种 Connection Type for vSwitch，在这里选择 **Virtual Machine**，单击 **Next** 按钮，如图 4-29 所示。

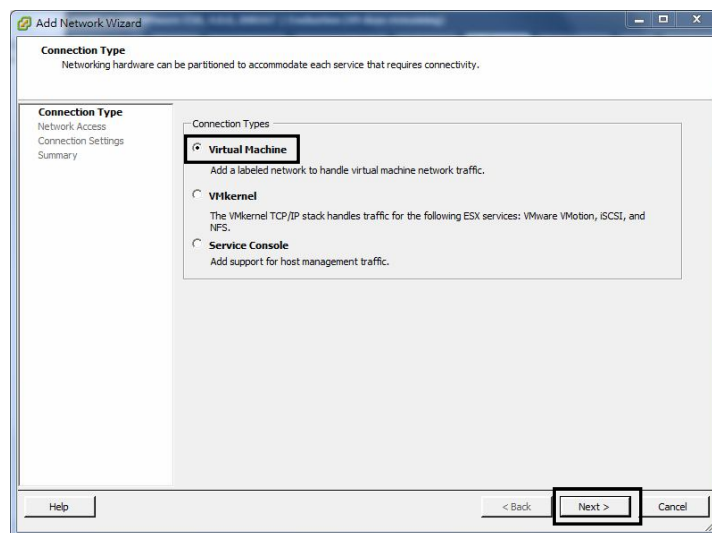


图 4-29

8. 决定哪些 vmnic 要给这个 vSwitch 当成 uplink port，这里勾选 **vmnic1**、**2**、**3** 当作 uplink，形成 NIC Teaming 的机制，单击 **Next** 按钮，如图 4-30 所示。

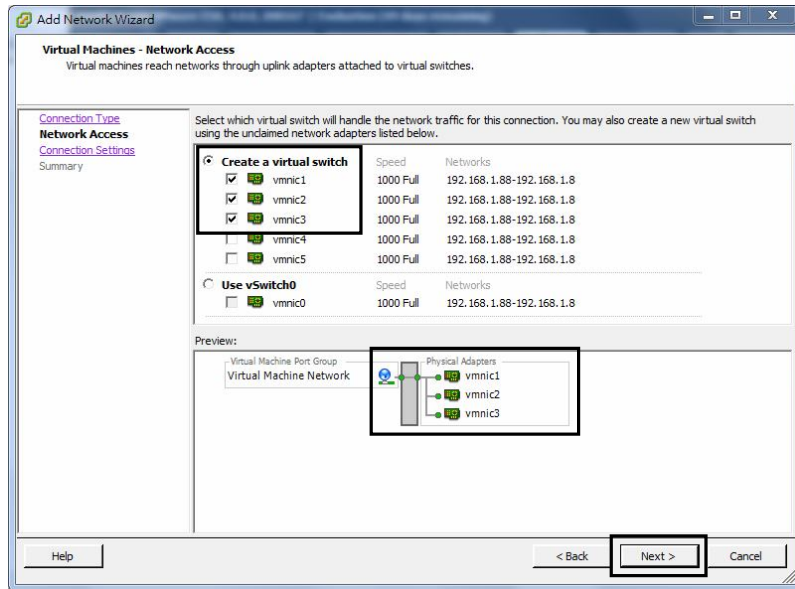


图 4-30

9. Port Group 的部分。Network Label 指的是 Port Group 的名称，范例取名为 **Production**，另外 VLAN ID 字段如果给了一个 ID，代表 vSwitch 要做 VLAN，就是前面提到的 VST，我们不在这里设置 VLAN，单击 **Next** 按钮，如图 4-31 所示。

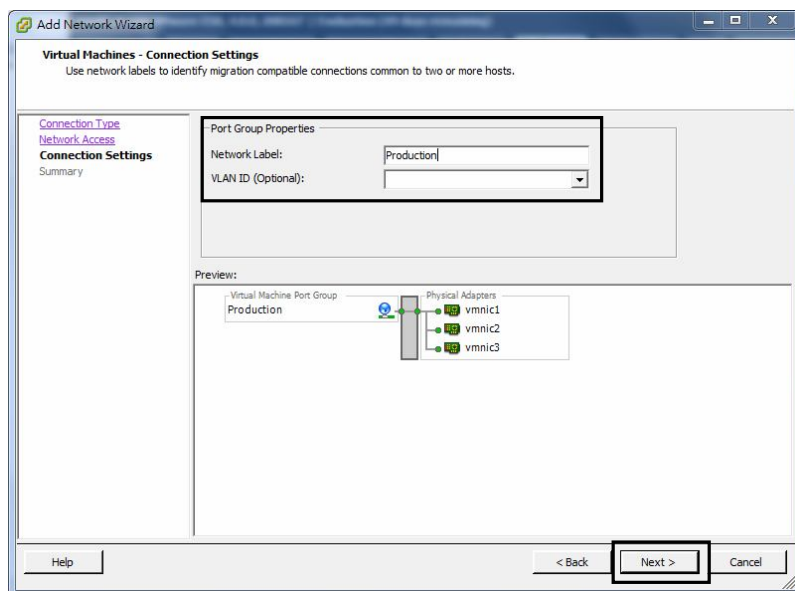


图 4-31

10. 创建好 vSwitch、vmnic 与 Port Group Name，单击 **Finish** 按钮，如图 4-32 所示。

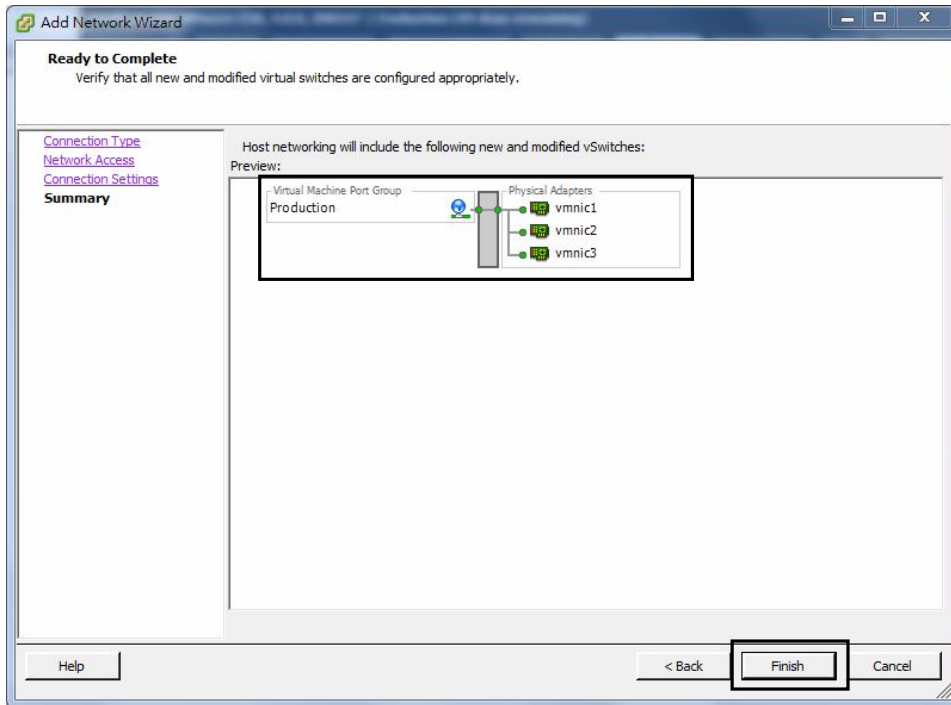


图 4-32

11. 现在看到已经出现了新的 vSwitch。接着要在 vSwitch1 上新增第二个 Port Group，请单击 vSwitch1 的 **Properties** 按钮，如图 4-33 所示。

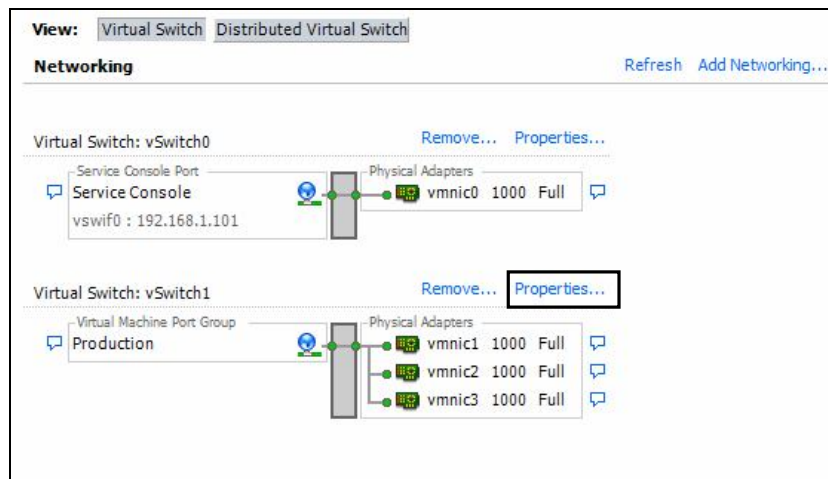


图 4-33

12. 目前 vSwitch 只有一个叫做 Production 的 Port Group，单击 **Add** 按钮，如图 4-34 所示。

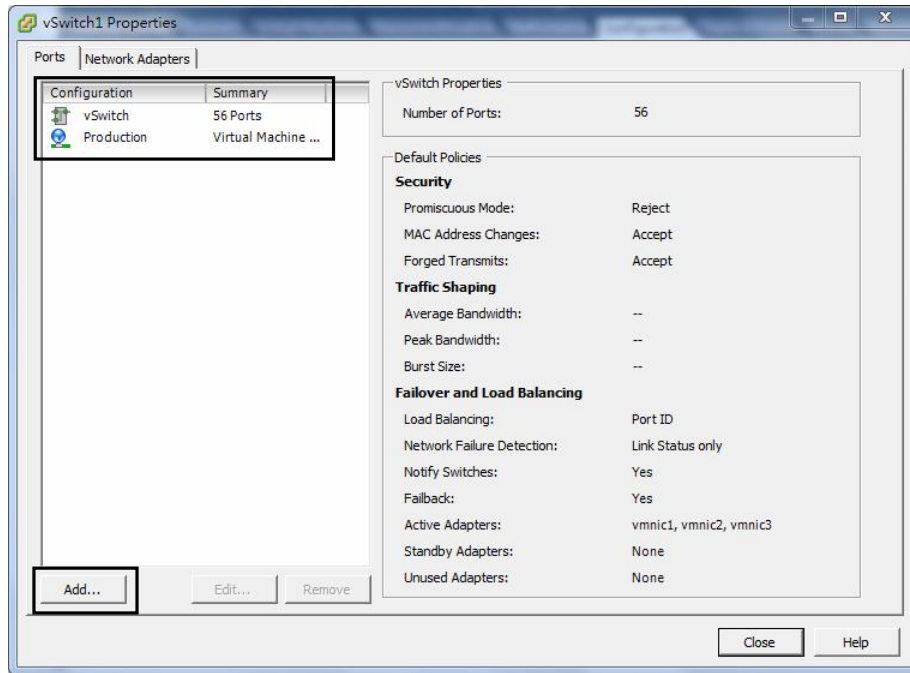


图 4-34

13. 在 Connection Type 下选择 **Virtual Machine**，单击 **Next** 按钮，如图 4-35 所示。

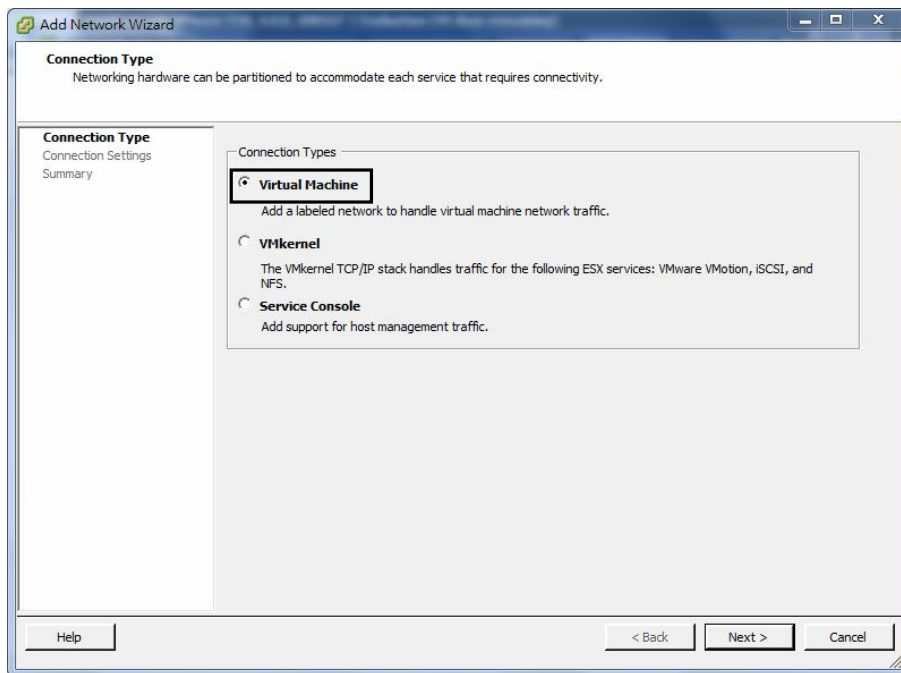


图 4-35

14. 为 Port Group 取名称为 **Test**，然后单击 **Next** 按钮，如图 4-36 所示。

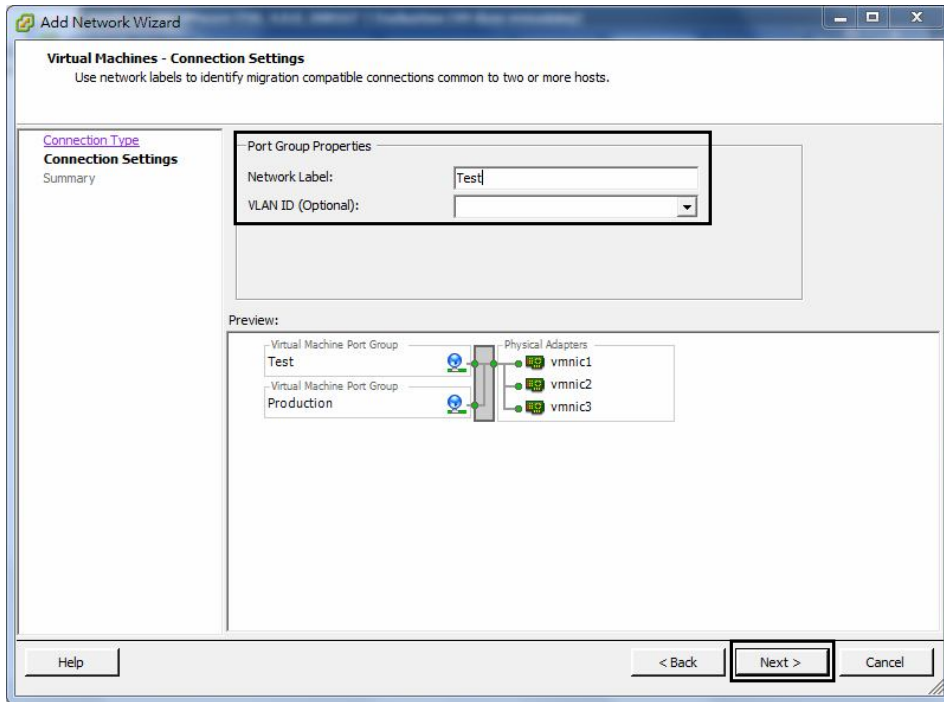


图 4-36

15. 多出了一个 Test 的 Port Group，单击 **Finish** 按钮，如图 4-37 所示。

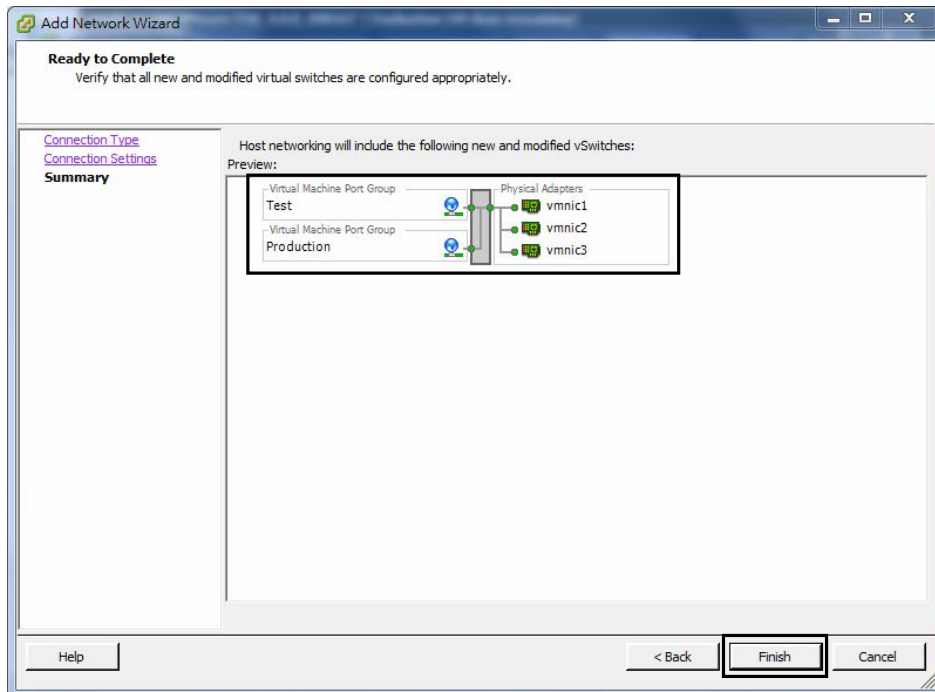


图 4-37

## 设置 Network Policy

1. 现在可以看到 vSwitch 有两个 Port Group，右侧会显示目前的设置值。这两个 PG 可以遵循 vSwitch 的默认值，即设置与 vSwitch 都相同。但是也可以给予每个 PortGroup 不同的 Network Policy，让不同的 Port Group 拥有不同的 Policy，达到网络设置的最大弹性。我们先针对 Production 来应用网络原则，请选择 Production Port Group，然后单击 **Edit** 按钮，如图 4-38 所示。

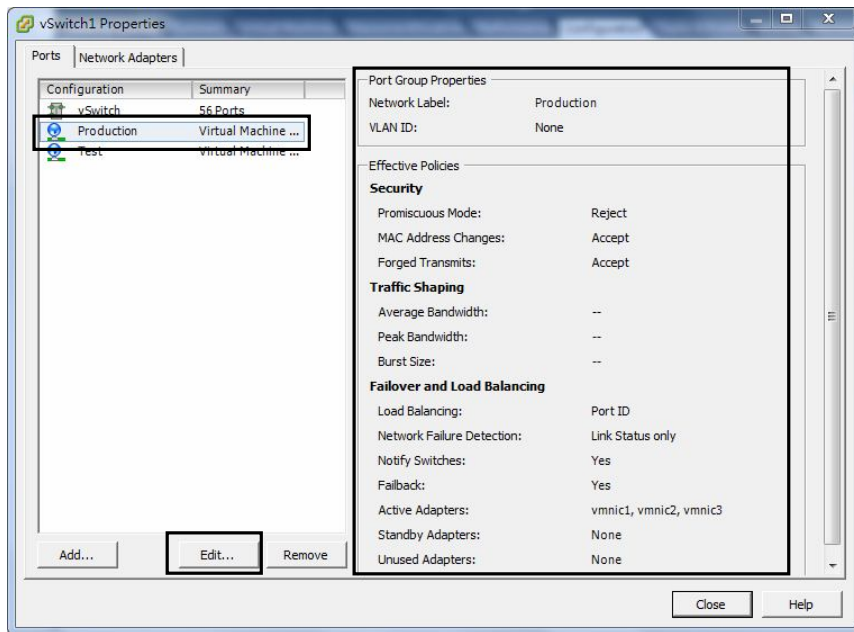


图 4-38

2. 我们要设置的是 NIC Teaming 部分，单击 **NIC Teaming** 选项卡，如图 4-39 所示。

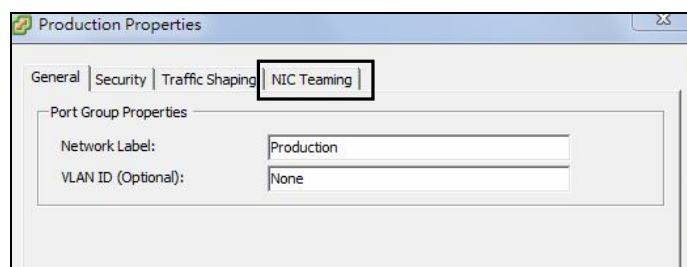


图 4-39

3. 在 Failover Order 部分，勾选 **Override vSwitch failover order**，表示不依照 vSwitch 原来的共同设置，而针对这个 Port Group 使用自己的 Failover Order，如图 4-40 所示。
4. 使用 Move Up 和 Move Down 按钮，调整 vmnic 的 Active/Standby 顺序。在此范例我们将 vmnic3 调整成 Production Port Group 的 Standby，表示平时 Production

是使用 vmnic1 和 vmnic2 这两个 uplink，一旦这两个 uplink 发生问题时，自动切换到 vmnic3。设置好 failover order，单击 **OK** 按钮，如图 4-41 所示。

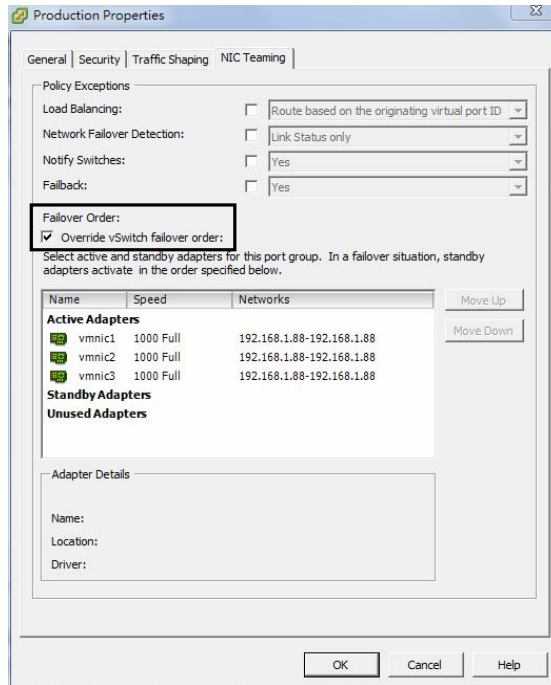


图 4-40

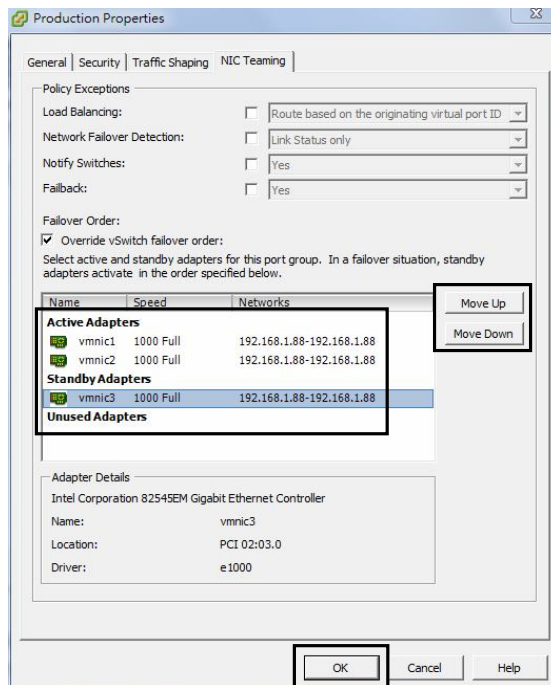


图 4-41

5. 检查右侧信息栏中的 Failover and Load Balancing, 确认 Active Adapters 和 Standby Adapters 是否显示出正确的设置, 如图 4-42 所示。

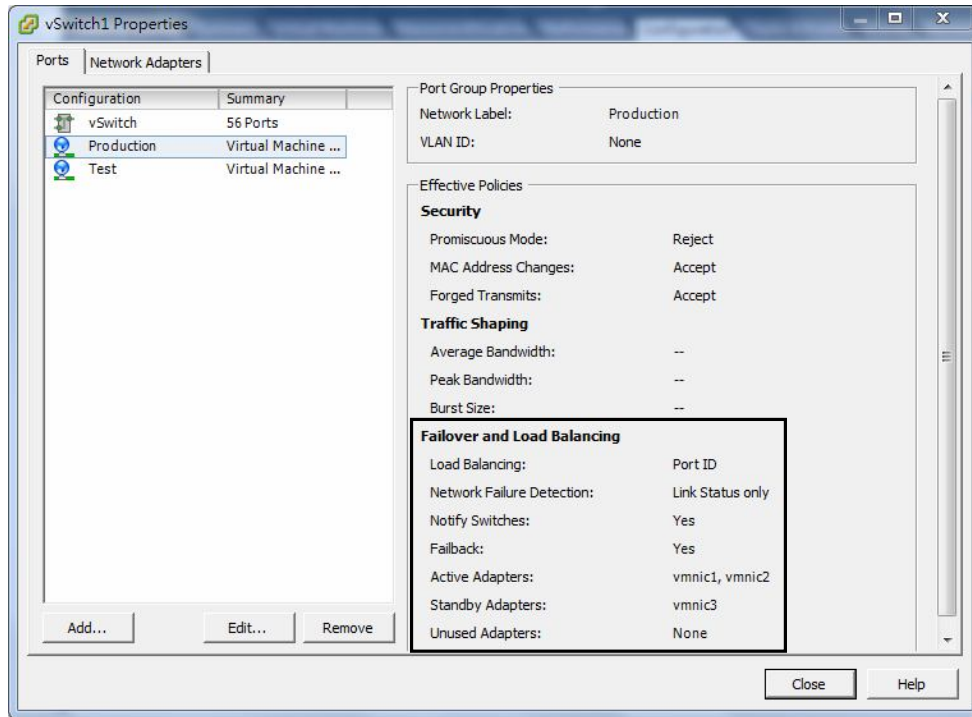


图 4-42

6. 接着要更动的是 Test Port Group。单击 **Test PG**, 单击 **Edit** 按钮, 如图 4-43 所示。

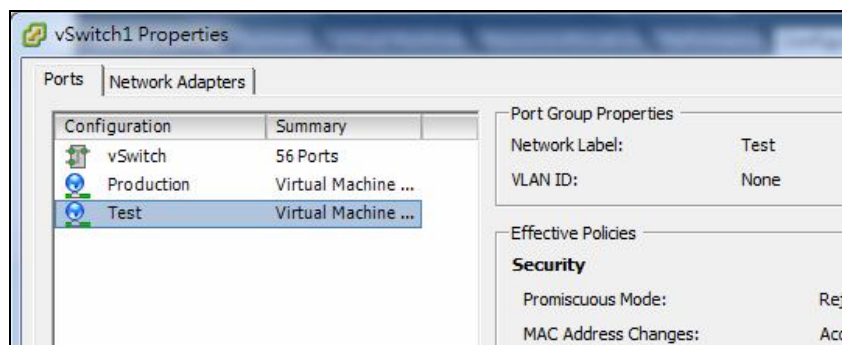


图 4-43

7. 一样到 NIC Teaming 中, 勾选 **Override vSwitch failover order**, 然后将 vmnic 调整成如图 4-44 所示: vmnic3 为 Test Port Group 的 Active Adapters, vmnic2 是 Standby, 而 vmnic1 则调整到 Unused Adapters。这代表, 我们让 Test Port Group 的 VM 平时只用 vmnic3, 故障时由 vmnic2 接手, 至于 vmnic1 则不让 Test PG 来使用。



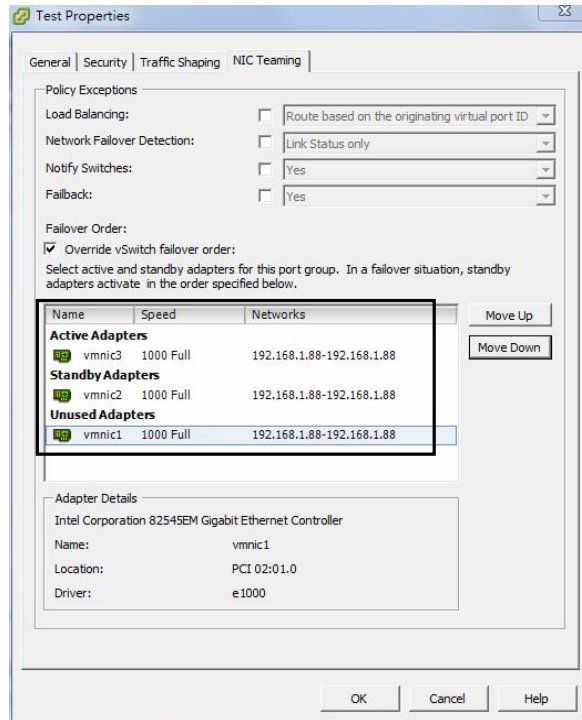


图 4-44

8. 来看一下 TEST PG 的设置吧！没有问题的话，请单击 **Close** 按钮，如图 4-45 所示。

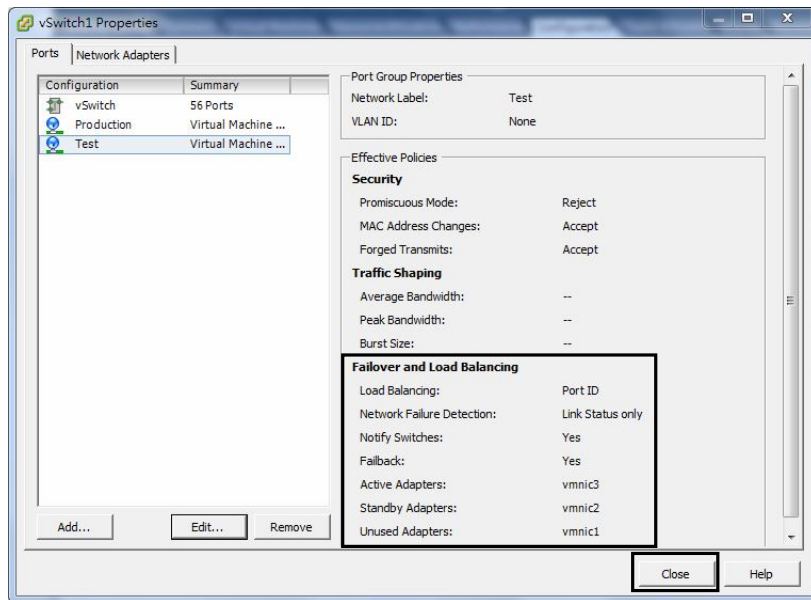


图 4-45

9. 返回到 Networking 的界面，现在 vSwitch1 已经有了两个 Port Group。接着要再新增第三个 vSwitch，单击 **Add Networking** 按钮，如图 4-46 所示。

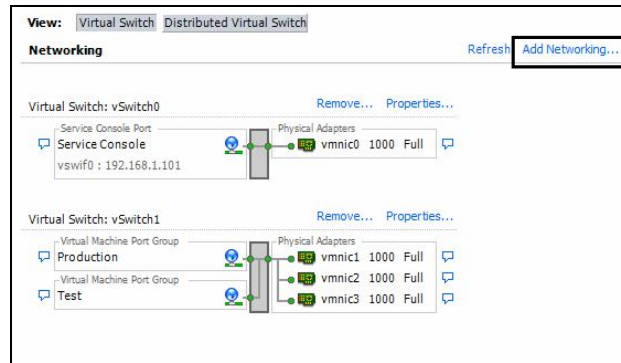


图 4-46

## VMkernel Port for iSCSI 和 vMotion

1. 这次要选择的是 **VMkernel Port**，单击 **Next** 按钮，如图 4-47 所示。

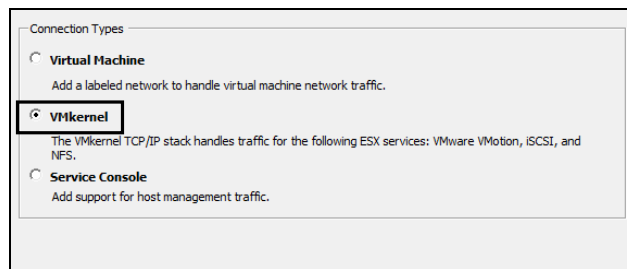


图 4-47

2. 选择 **vmnic4** 当成 vSwitch 的 uplink Port，单击 **Next** 按钮，如图 4-48 所示。

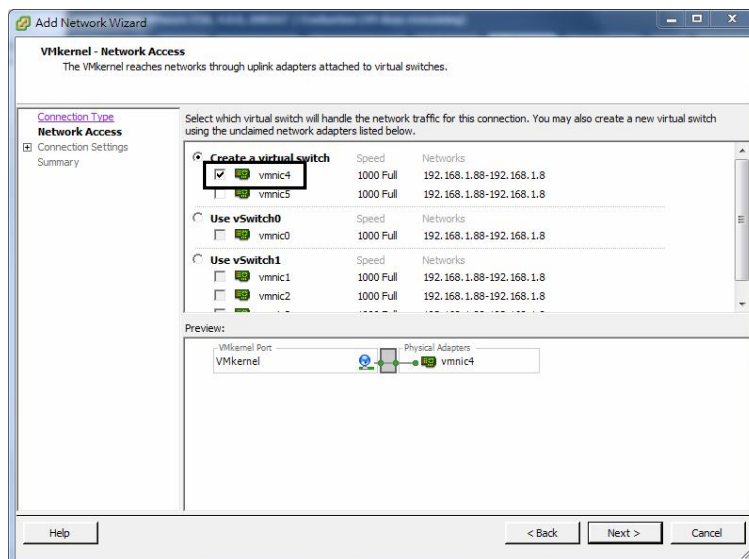


图 4-48

3. 取个名称，范例为 **for iSCSI access**，如图 4-49 所示。

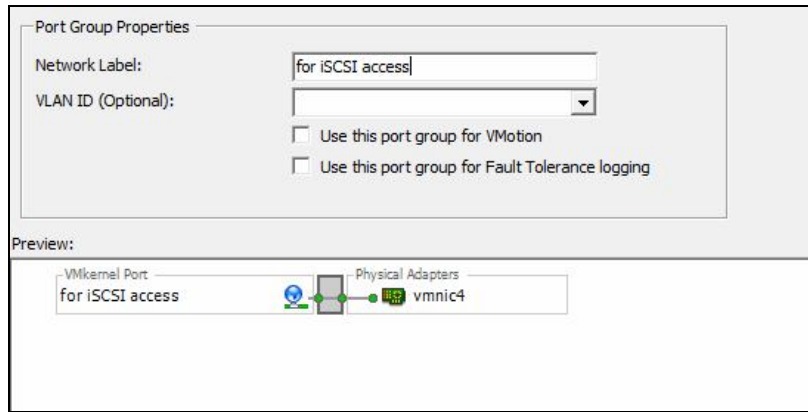


图 4-49

4. VMkernel Port 需要指定一个 IP 地址，因为 VMkernel 也要有一个 IP 才能去访问 iSCSI 的存储设备，下一章介绍 Storage 时，我们会安装 iSCSI Target。范例的 IP 地址设置为：**192.168.1.51**，Subnet Mask：**255.255.255.0**（ESX01 host 的 VMkernel Port 的 IP），如图 4-50 所示。

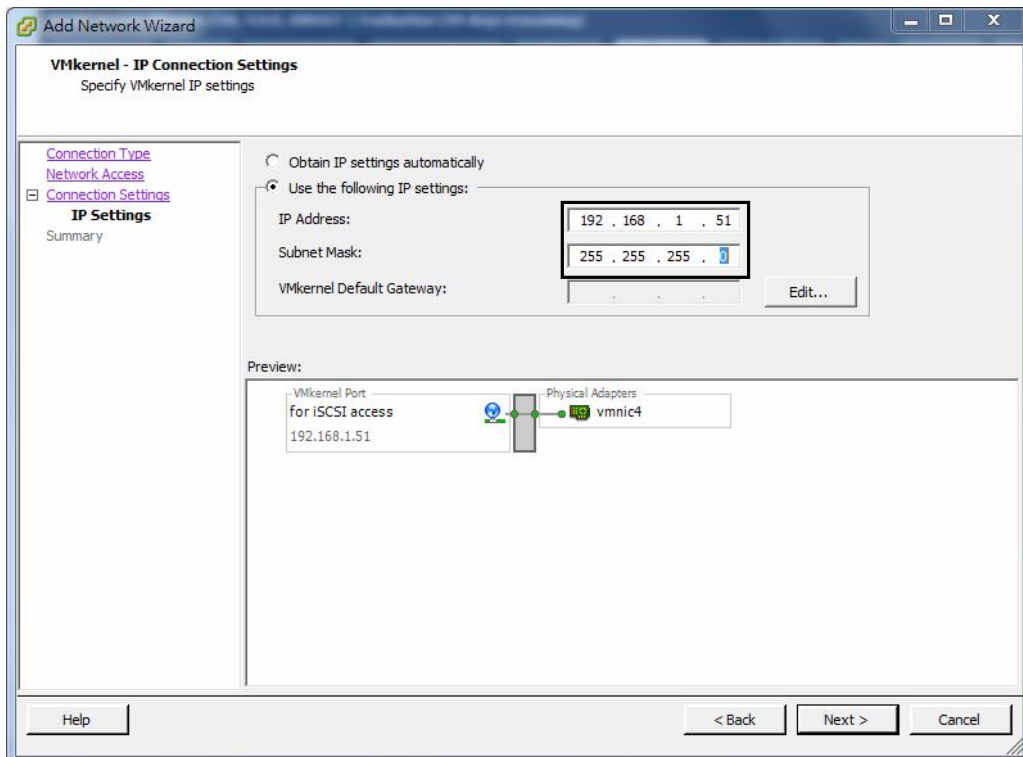


图 4-50

5. 这个 vSwitch 是给 VMkernel 访问 iSCSI 专用的，uplink 是 vmnic4，如图 4-51 所示。直接单击 **Finish** 按钮。

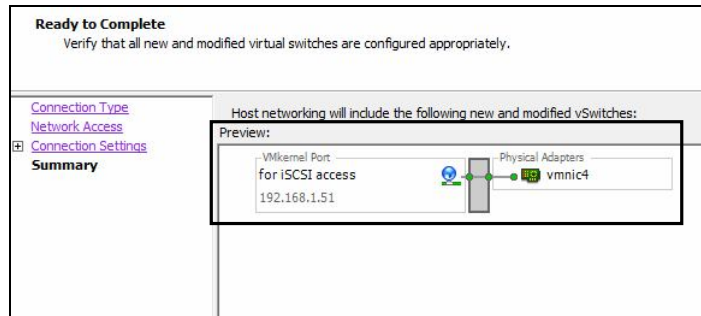


图 4-51

6. 因为没有设置 default gateway，所以出现询问对话框，本书的练习没有 default gateway，所以不用设置，单击“否”按钮即可，如图 4-52 所示。

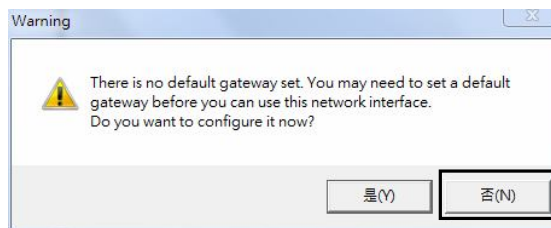


图 4-52

7. 完成了 vSwitch2 的配置，目前已经有三个 vSwitch 了。还需要一个 vSwitch，所以请单击 **Add Networking**，如图 4-53 所示。

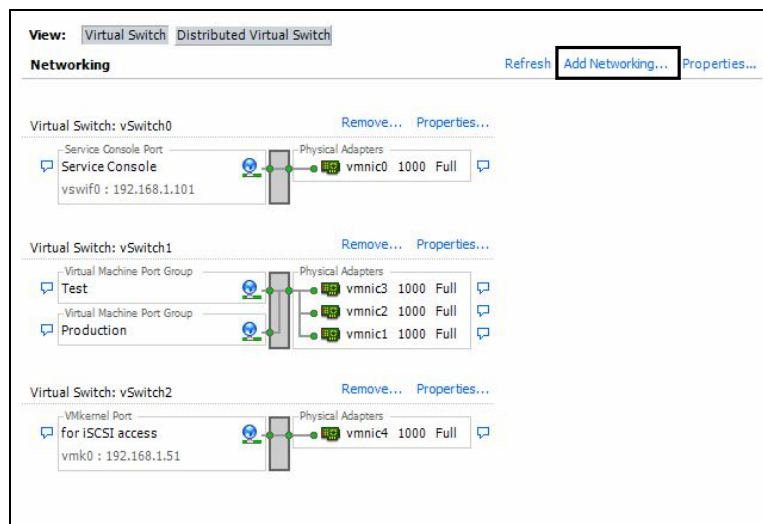


图 4-53

8. 这次选择的仍是 **VMkernel Port**，要用作 vMotion 用途，如图 4-54 所示。

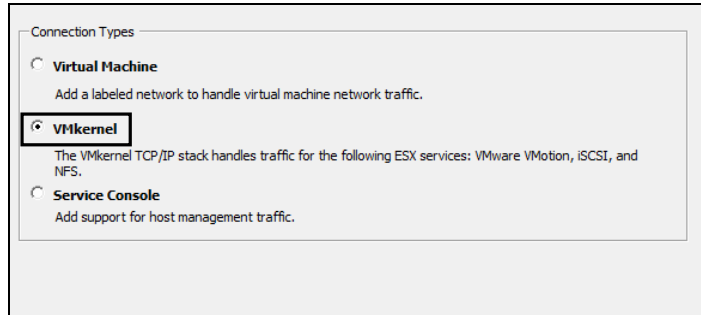


图 4-54

9. 只剩下一个 vmnic 可以分配给新的 vSwitch 了，勾选 **vmnic5**，单击 **Next** 按钮，如图 4-55 所示。

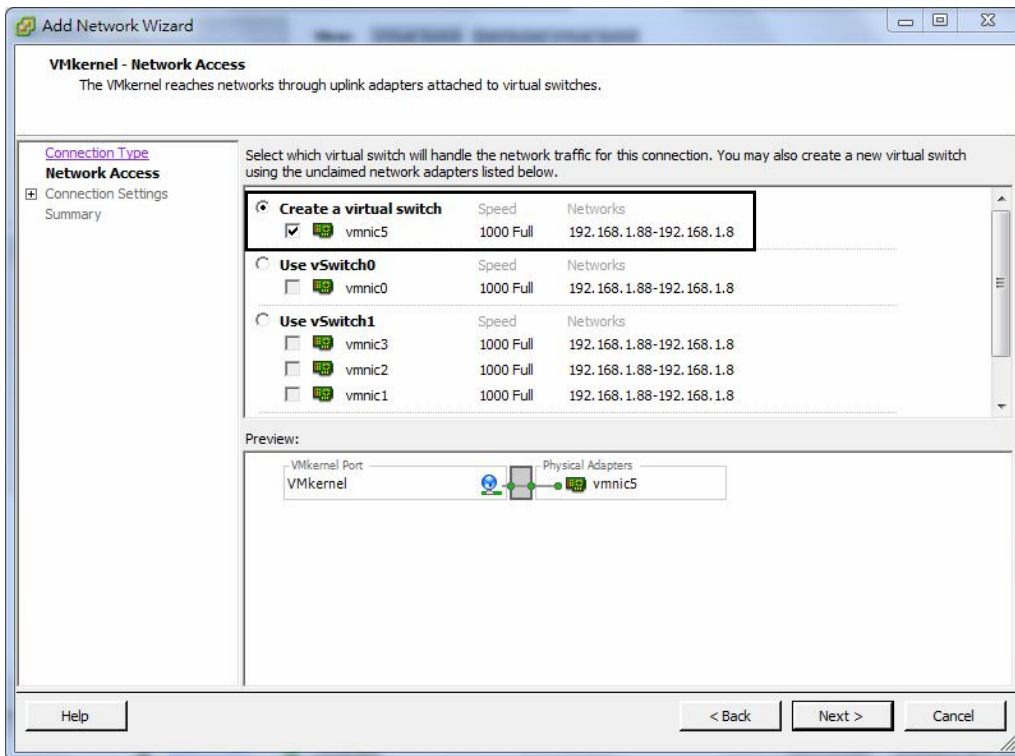


图 4-55

10. 取一个名称，请输入 **for vMotion**，并勾选下面的 **Use this port group for VMotion**，表示要以这个 uplink 来传送 vMotion，如图 4-56 所示。

11. 第二个 VMkernel 仍然要输入 IP 地址，范例是 **192.168.1.61**，Subnet Mask 是 **255.255.255.0**，单击 **Next** 按钮，如图 4-57 所示。

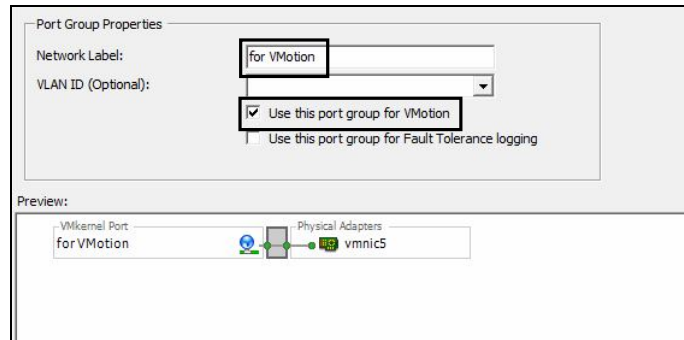


图 4-56

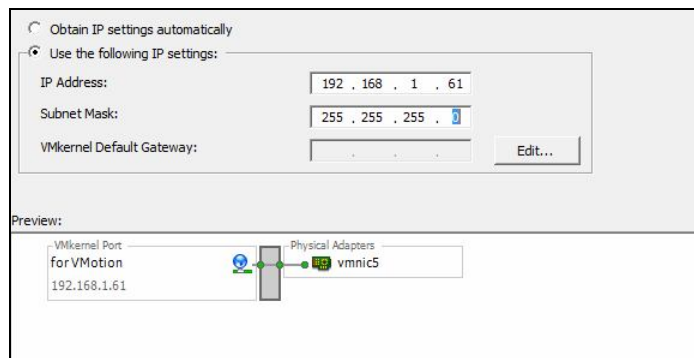


图 4-57

12. 这个 vSwitch 代表的意义是，以后的 vMotion 将使用这个 uplink，如图 4-58 所示。

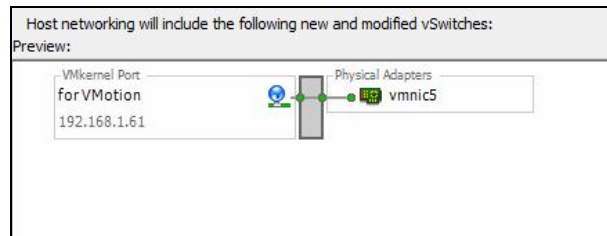


图 4-58

13. 一样不设置 default gateway，单击“否”按钮，如图 4-59 所示。

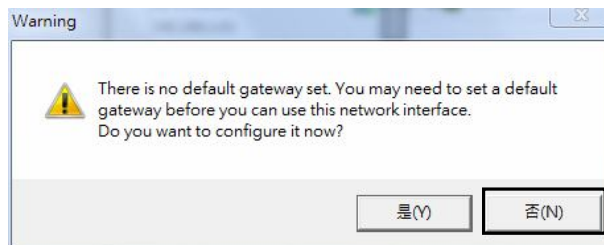


图 4-59

14. 完成了 ESX01 的所有 vSwitch 配置，如图 4-60 所示。接着请各位读者照着 4-2 节的所有步骤对着 ESX02 再设置一遍，整个网络的配置就大功告成了。  
ESX02 的 vmk0（VMkernel IP）范例为：192.168.1.52，vmk1 为 192.168.1.62。

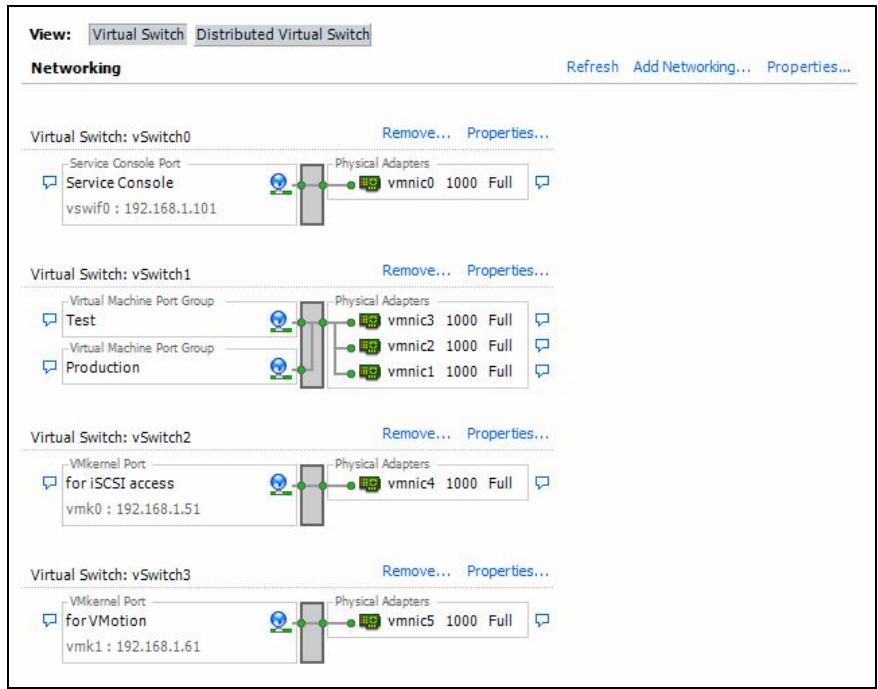


图 4-60

Q: 为何不讨论 vDS?



vDS 是横跨 host 的 vSwitch，功能强大且提高管理效率。我们只是配置两个 ESX host 的 vSwitch 就花了一些时间，如果今天有 100 个 ESX host 等着你去设置呢？每个 host 去设置 vSwitch 确实有些麻烦，vDS 可以简化设置步骤，我们可以用 vCenter 来配置逻辑上看起来是横跨多个 host 的 vSwitch，这样便可以将不同 host 上的 VM 都连接到相同的 port group 上，以便一致性地应用 Network Policy（请注意 host profile、vDS 都必须是 Enterprise Plus 版本才有提供的功能）。

此外，vNetwork API 开放给网络设备厂商自行开发自己的 vDS，提供更高级的功能，例如客户可以用 Cisco Nexus 1000V 替换掉原来 VMware 的 vDS。这样做的好处是可以将 vSwitch 的配置控制权交由网络团队来主导，避免缺乏讨论沟通、网络配置不当、网络团队无法监控造成安全死角或服务中断的问题。由于 Cisco 的 vDS 采用与实体交换机相同的命令行管理模式，方便网络设备管理人员设置，且可以与实体 Cisco 设备整合在一起，并提供了许多高级功能，例如更多元的 NIC Teaming 负载均衡方式、VN-Link，解决 Network Configuration 因 vMotion 之后，由于 uplink link 不

同，若实体 Switch 的 port policy 或 ACL 限制所产生 VM 服务中断的问题等。

还认为虚拟化只是 Data Center Server Team 的事情吗？其实虚拟化冲击了许多层面，传统的数据中心，通常是各司其职：服务器、网络设备、存储设备等，可能都是不同的团队，各有各的专业。但虚拟化涉及的层面很广泛，也有许多新思维，致使每个团队必须要互相合作才能规划出完全适合数据中心的方案。所以，虚拟化可以说与每个 IT 人员多少都会有关系。

即便是如此，笔者仍然不建议刚接触 vSphere 的人直接使用和设置 vDS。笔者见过许多例子，尚未接触过 vNetwork 的人，实体与虚拟网络之间常常造成概念混淆，如果此时再去介绍 vDS，只会乱上加乱。例如，有人听了 NIC Teaming 的作用，以为它可以像 Layer 4 Switch 一样地做到 NLB 的功能，实际上那是做不到的。Virtual Switch 只是一个虚拟的 L2 交换机，NIC Teaming 也不能控制 Inbound Traffic，怎么能要求它做到 L4 这一层的功能呢？除非 VM 使用 Windows Server 的 NLB Cluster 来配合。

vSwitch 的初始目的是，要让 VM 可以在虚拟化的环境，通过实体的网络与外界沟通，然后再利用这些 uplink port 来实现频宽分配、优先级、容错等机制，在有限的实体网卡（最多到 32 个 vmnic，不过实体服务器应该不会有这么多 slot）下，配置出最符合网络性能与需求的环境。

所以，对于刚接触虚拟化的人来说，笔者诚心地建议：请先忘了 vDS。好好地掌握 vSS 的概念、配置及应用，等到 vSphere 真的上手之后，再来看 vDS，便可轻松驾驭。

目前，ESX host 上的 6 个 vmnic（实体网卡）都用上了，各司其职，每个都有其用途。但是不代表这样的设置就完善，或是符合需求的完美配置。并不见得。例如，我们的环境并没有使用 VMware FT 的功能（vSphere in a box 硬件限制），也没有配置 VLAN，相较于企业环境与数据中心，单纯了许多，所以目前配置 6 个 vmnic 已足够使用（你可以用 VMware Player 虚拟出最多 10 个 vmnic 来给 ESX 使用）。

这里要再强调一次，每个环境的需求、实体网络配置与应用都不相同，每一种设置也各有优缺点。一个好的规划配置是优点多、缺点少，只要秉承这个方向，规划出你的 vNetwork 配置，符合公司环境的应用即可。

现在，我们已经完成了第三朵小云，即网络（vNetwork）的部分，网络的安全性、性能等配置对虚拟化之后的数据中心非常的重要，云端世界的所有服务均需要依赖网络传递，这绝对是每个企业都必须面对的课题。