**在vSphere上通过BOSH工具大规模部署Cloud Foundry**

开始部署前，硬件和软件方面的前提条件。

**软件：**

1) 64 位 Ubuntu 10.04 LTS，最好是 ISO 格式。（注意：接下来的所有操作均只针对10.04）

2) vSphere V4.1 或 V5.x（采用vSphere 作为hypervisor）

3) vSphere Client

4) vCenter（安装在 Windows 2008 R2 64 位或 Windows 2003 服务器上，物理机或虚拟机皆可）

注意：本文中提到的所有对部署和安装时间的估计如“几分钟”，“几十分钟”等均以上述硬件环境和网络环境为基础，实际情况应该略有不同。

**硬件：**

假设所有节点都是虚拟机，下表显示了所需的虚拟机数目：

|  |  |  |  |
| --- | --- | --- | --- |
| **组件** | **节点数目** | **操作系统** | **可否是物理机** |
| BOSH CLI | 1 | Ubuntu | 可以 |
| vCenter+vSphere Client | 1 | Win2008 | 可以安装在一起，也可划分成两个节点 |
| Micro BOSH | 1 | Ubuntu | 不可以 |
| BOSH | 6 | Ubuntu | 不可以 |
| Cloud Foundry | 34 \* | Ubuntu | 不可以，见下文 |
| 合计： | 43 |  |  |

注意：上表中 Cloud Foundry 的节点数目是所需的最少节点数目。此数目可能会因实际的 Cloud Foundry 部署规模而异。选择硬件配置时通常要考虑两个原则：

1) vCPU总数不应超过物理核心总数的两倍。在生产系统中，两者之比应该接近于 1。

2) 所有虚拟机的总内存应尽量小于或略微大于所有Hypervisor的物理内存。

下面例子是假设每个虚拟机有 4 GB 内存和 1 个vCPU时的硬件配置：

6 台服务器，每台服务器有 8 核 CPU和 32GB RAM。

就实验系统而言，我们曾在一台配置如下的服务器上成功部署（假定每个虚拟机有 256 MB 内存）：

1 台物理服务器：8 核 CPU，16 GB RAM。

对于生产环境，我们建议选择CPU核数和内存容量都比较大的机型，这样在同一台物理机，可以运行更多的虚拟机。同时需要考虑有比较高吞吐量的网卡和存储设备。

除了服务器之外，存储也是云平台中的一个关键要素。存储最好应有 200 GB 或更大的可用空间，以便保存所有虚拟机的映像。在生产系统中，建议采用快速的共享存储。NFS 是用来在Hypervisor间共享存储的最常用协议。在试验环境中，可以使用基于 Linux 的 NFS 服务器来代替专用存储。尽管Hypervisor中的本地磁盘在 测试环境中可以使用，但通常不建议将本地磁盘用于生产系统中。

我们最后应规划的是网络。在实验室环境中，我们可以直接将所有节点都放在同一网络中。不过，在生产系统中，出于安全和管理需要，应将 Cloud Foundry 的各个组件正确分配到VLAN中。在本文中，我们不讨论网络连接方面的细节。作为例子，我们在部署期间将采用四个VLAN：

|  |  |
| --- | --- |
| **VLAN** | **节点** |
| Management VLAN | Hypervisor和 NFS 存储 |
| CF VLAN | BOSH 虚拟机以及 Cloud Foundry 的虚拟机 |
| Service VLAN | LB ，双宿 (Dual-Homed) router |
| Public VLAN | LB，外网请求 |

Cloud Foundry 实例的安装过程分为以下四个部分：

1) 在 Ubuntu 10.04 操作系统中安装 BOSH CLI 工具。此操作系统的主机可以是物理机，也可以是虚拟机。

2) 安装Micro BOSH。Micro BOSH是一个包含 BOSH 所有组件的虚拟机。它所具备标准 BOSH 的所有功能。不过，它用来存储多个Release的磁盘空间十分有限。部署Micro BOSH的目的是为了安装 BOSH，因为BOSH 本身就是一个分布式系统。

3) 通过Micro BOSH来安装 BOSH。BOSH 通常包含 6 个结点，每个节点部署一个组件。其中一个称作blobstore的节点具有较大的磁盘，可以保存较大的Release。

4) 通过 BOSH 安装 Cloud Foundry 实例。

**重要前提条件：**

在 BOSH 和 Cloud Foundry 的整个安装过程中，都需要直接的 Internet 连接。这一点非常重要，因为部分软件代码是直接从 Internet 下载的，例如 Ruby Gem 以及一些开源软件。在虚拟机与 Internet 之间设置 Web 代理服务器将导致安装失败。注：NAT是允许的。

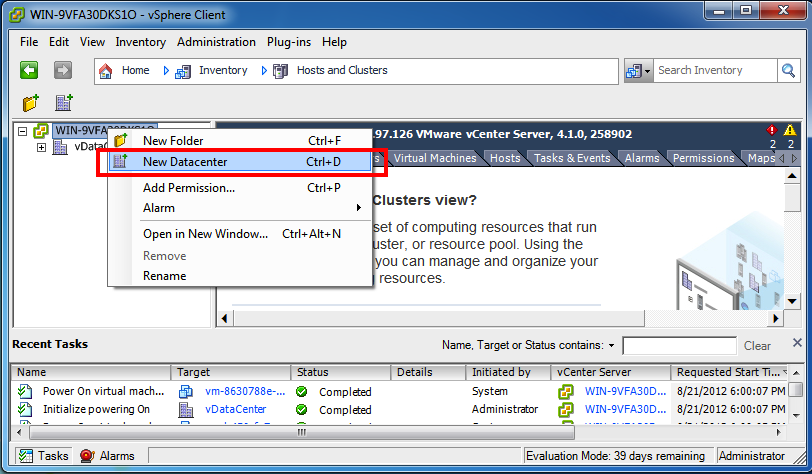
另一项前提条件是要有稳定的 Internet 连接。如果您的网络在从 Internet 下载文件时速度缓慢或者不可靠，安装可能会因出现超时或连接错误而失败。

**在 vCenter 中创建一个群集**

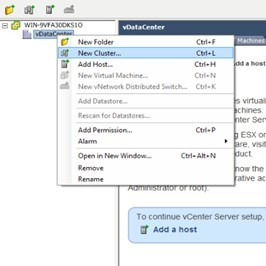
假定所有节点都是虚拟机，那么我们首先在所有裸机服务器上安装 vSphere（在本篇文章中我们采用 V5.x）。各 vSphere 服务器通过Management VLAN相连。安装完毕后，我们需在其中一个Hypervisor上创建一个虚拟机以安装 64 位 Windows 2008 R2。随后，我们需在此 Windows 2008 虚拟机上安装 vCenter。下一步是使用 vSphere Client 连接到 vCenter，以便我们可以管理这些服务器。

有关vSphere，vCenter和vSphere Client的安装使用细节，请参考VMware官方网站中的文档介绍 <http://www.vmware.com/cn/products/>

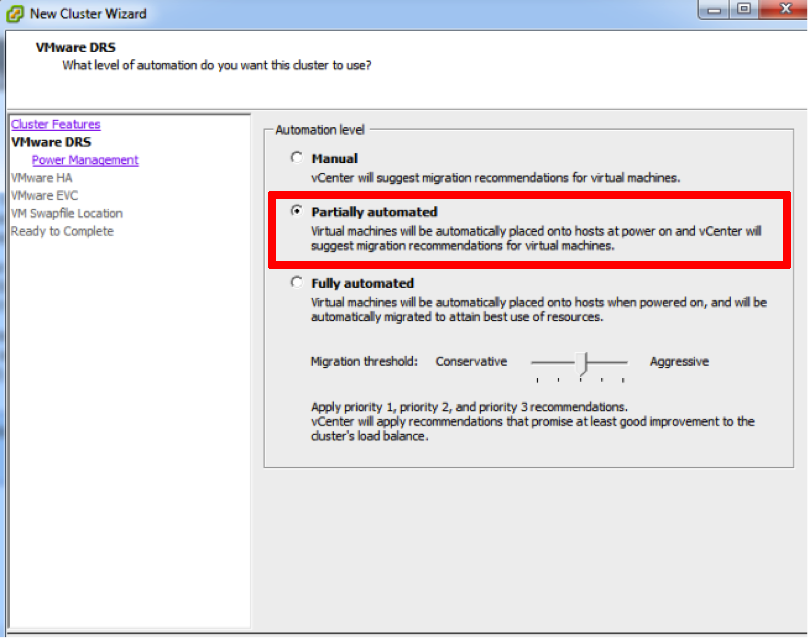
我们可以在任意 Windows机器（甚至是虚拟机）上安装 vSphere Client。之后，我们便可以通过 vSphere Client 以远程方式连接到 vCenter。首先，我们来在 vCenter 中创建一个数据中心。为此，请右键单击左窗格中的 vCenter 节点，然后选择“新建数据中心”（New Datacenter）以添加一个新的数据中心。



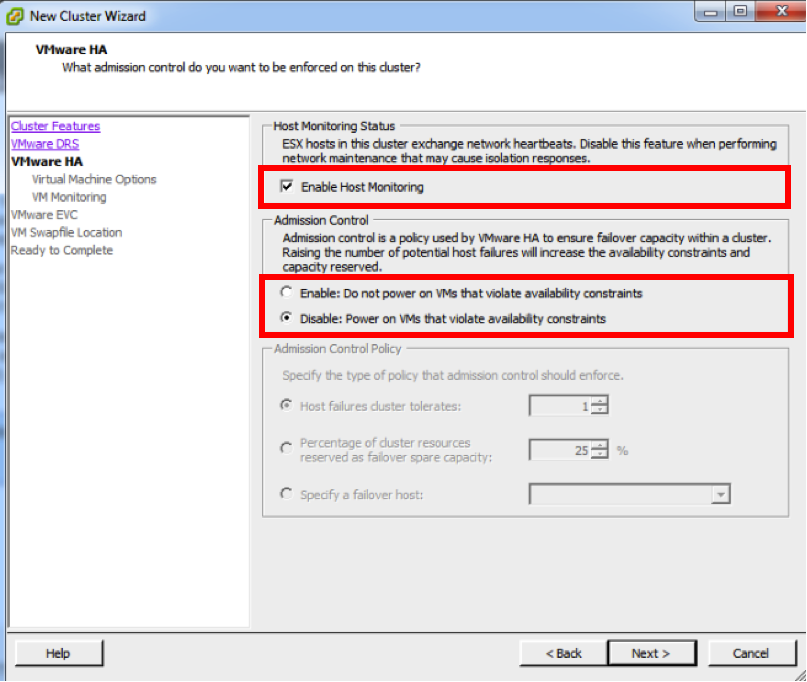
接下来，请右键单击新创建的数据中心节点，然后选择“新建群集…”(New Cluster…)。



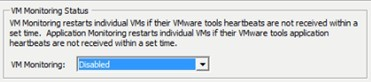
在“新建群集向导”(New Cluster Wizard) 执行期间，如果您启用了 vSphere DRS 功能，系统将要求您配置 VMware DRS。请确保“自动化级别”(Automation Level) 设置为“半自动”(Partially Automated) 或“全自动”(Fully Automated)，如下所示。如果您选择“手动”(Manual)，将会弹出一个窗口提示您输入您的选择。这种行为可能会阻止 BOSH 自动化安装。



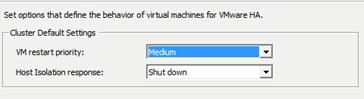
然后，请选中“启用主机监控”(Enable Host Monitoring) 复选框，再选中“禁用:启动违反可用性限制的虚拟机”(Disable: Power on VMs that violate availability constraints)：



接着，转到“虚拟机监控”(VM Monitoring) 子部分，选择“已禁用”(Disabled)：

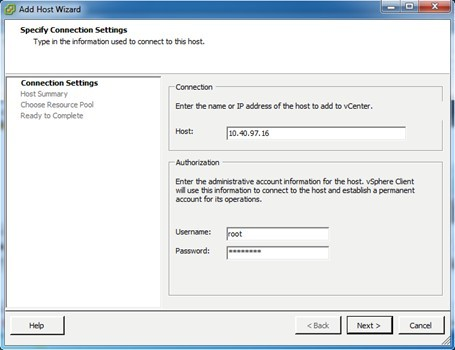


单击“下一步”(Next)，按如下所示做出选择：

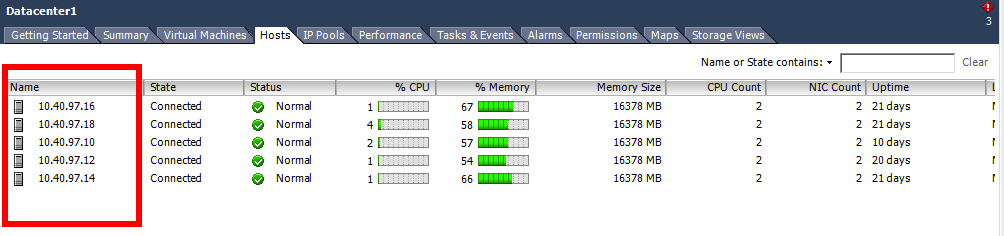


**添加 vSphere 主机**

下一步是将hypervisor放入我们刚创建的群集中。为此，请右键单击该群集节点，然后选择“添加主机…”(Add Host…)。对于每台 vSphere 服务器，输入其 IP 地址、管理员用户名和密码，然后确认您进行的配置：

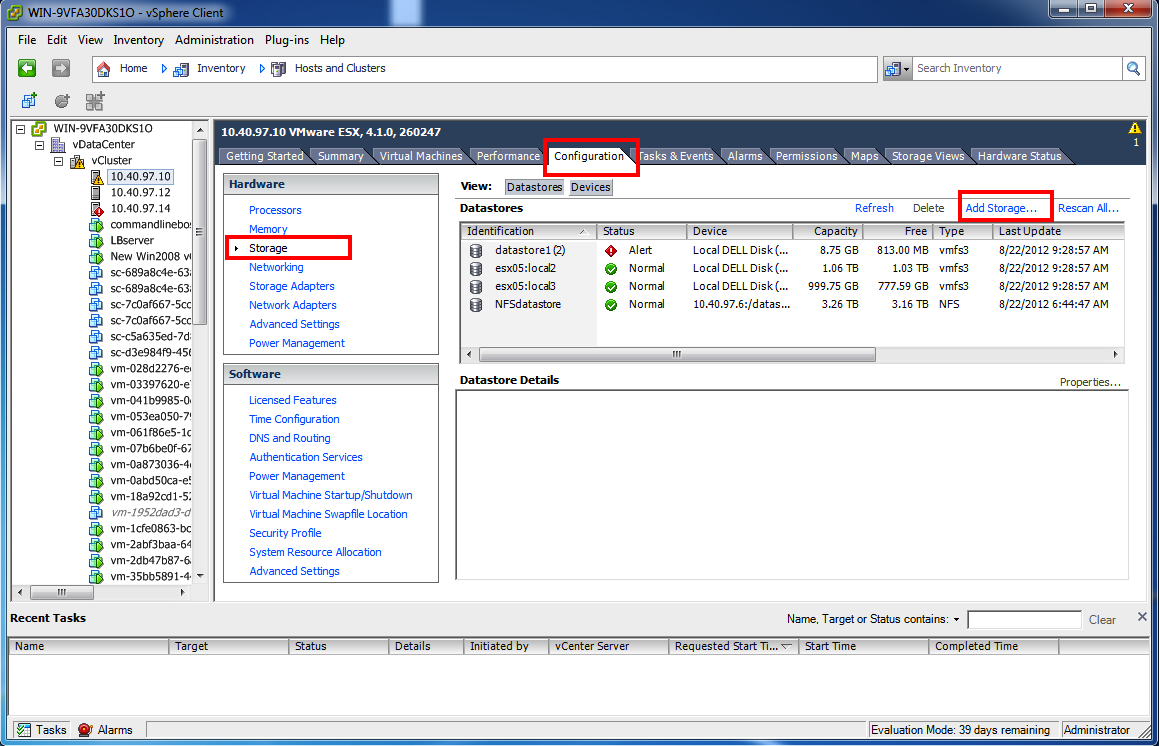


添加完所有主机后，这些主机将在“数据中心”(Datacenter) ->“主机”(Hosts) 选项卡中列出：

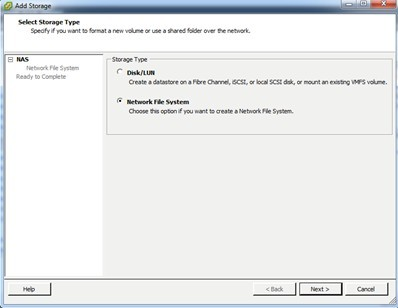


**将数据存储挂接到主机**

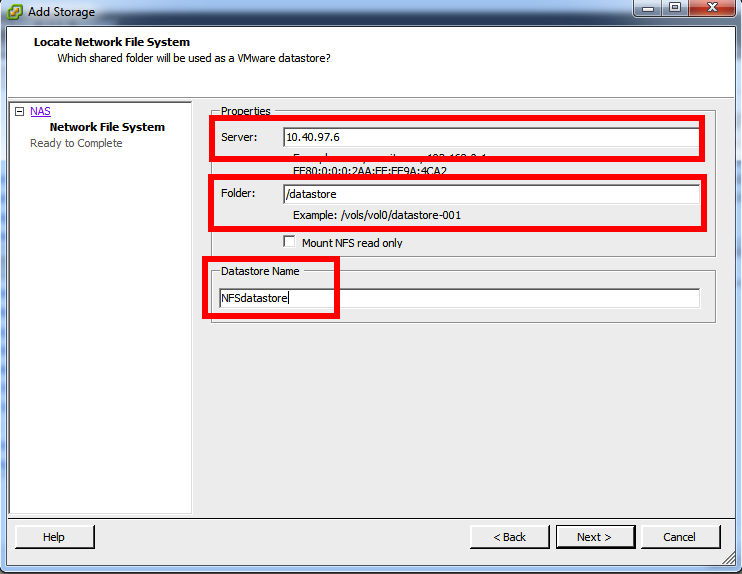
该群集中的所有主机都应共享同一 NFS 存储。对于每个主机，我们将该存储以数据存储的形式添加进来。为此，请在 vCenter 中单击相应的 vSphere 主机，然后选择“配置”(Configuration) 选项卡。选择“硬件”(Hardware) ->“存储”(Storage)。单击右上方的“Add Storage…”(添加存储…)



在“选择存储类型”(Select Storage Type) 对话框中，选择“网络文件系统”(Network File System)。

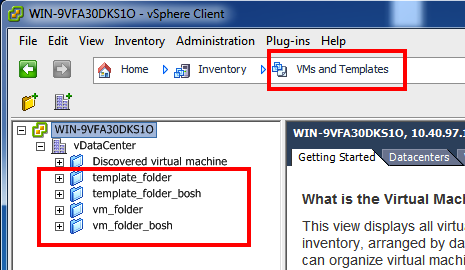


输入 NFS 存储的 IP 地址、相应的文件夹名称以及相应的数据存储名称。请务必对该群集内的所有主机都采用完全相同的数据存储名称，这一点非常重要。在本例中，我们采用“NFSdatastore”这一名称。



**为虚拟机和模板创建文件夹**

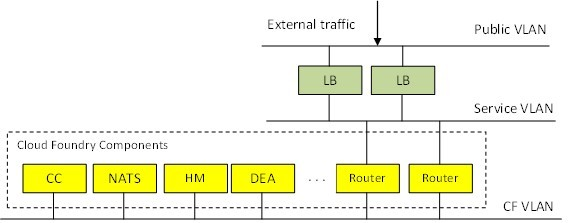
从 vCenter 的导航栏中，选择“主页”(Home) ->“清单”(Inventory) ->“虚拟机和模板”(VMs and Templates) 视图，然后按如下所示创建文件夹：



这些文件夹随后将用来对 BOSH 和 Cloud Foundry 的虚拟机进行分组。在上例中，“template\_folder\_bosh”用来存放 BOSH stemcell。“vm\_folder\_bosh”用来存放 BOSH 节点。“template\_folder”用来存放 Cloud Foundry stemcell。“vm\_folder”用来存放 Cloud Foundry 节点。随后将在部署清单文件中用到这些名称。

**网络配置**

Cloud Foundry 的虚拟机将部署到一个或多个网络中。在部署前，我们需要在 vSphere 中创建一些网络。下图显示了 Cloud Foundry 所需的网络连接。



在每个 vSphere 主机上，我们创建以下两个网络：

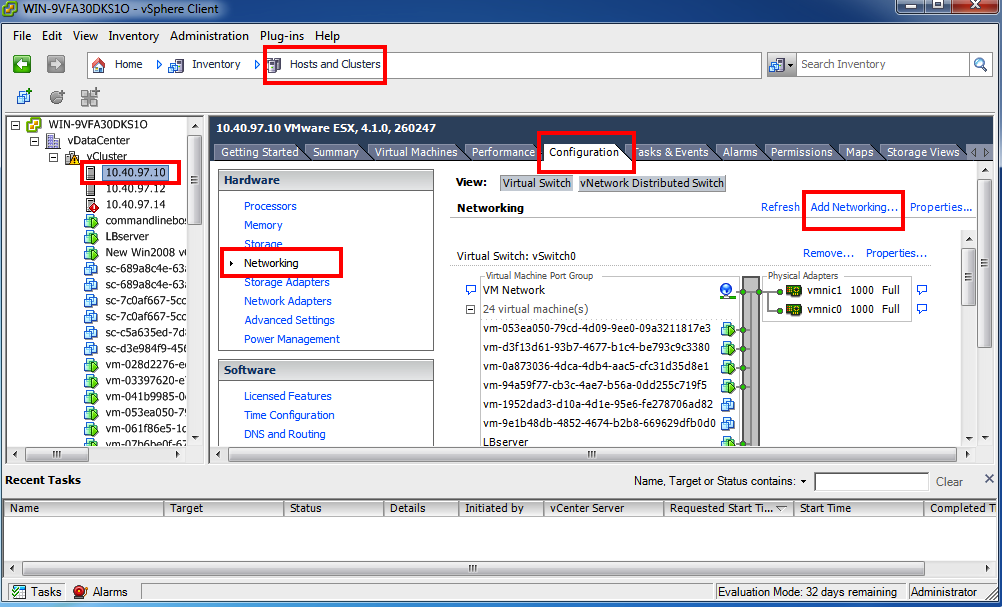
1) CF Network：映射到 CF VLAN

2) Service Network：映射到Service VLAN。

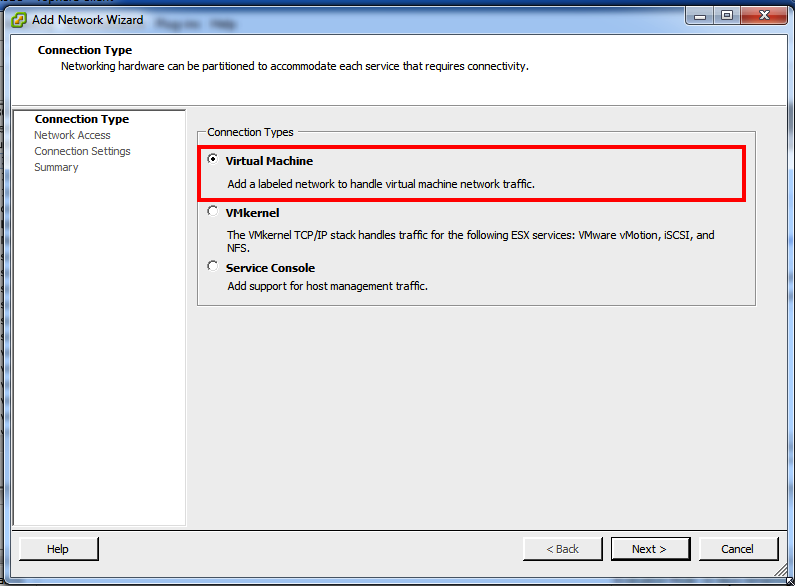
大多数虚拟机都位于 CF Network上。只有router虚拟机是双宿在Service Network和 CF Network上。

注意：在试验环境中，您可以将所有虚拟机都放在同一网络上，以便简化安装过程。因此，Hypervisor上的只有1个网络可能就绰绰有余。

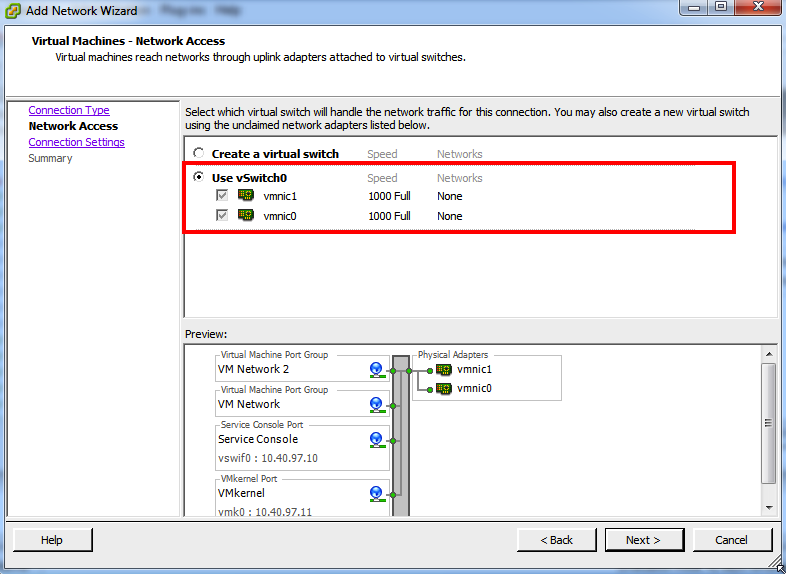
要创建网络，请选择“主机和群集”(Hosts and Clusters) 视图。选择一个主机，然后切换到“配置”(Configuration) 选项卡。然后选择“网络”(Networking)，再单击“添加网络”(Add Networking)：



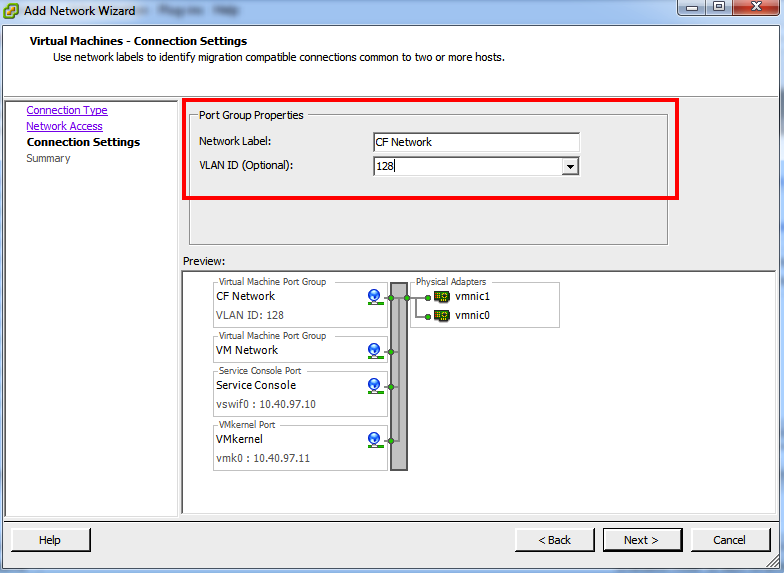
连接类型应为“虚拟机”(Virtual Machine)：



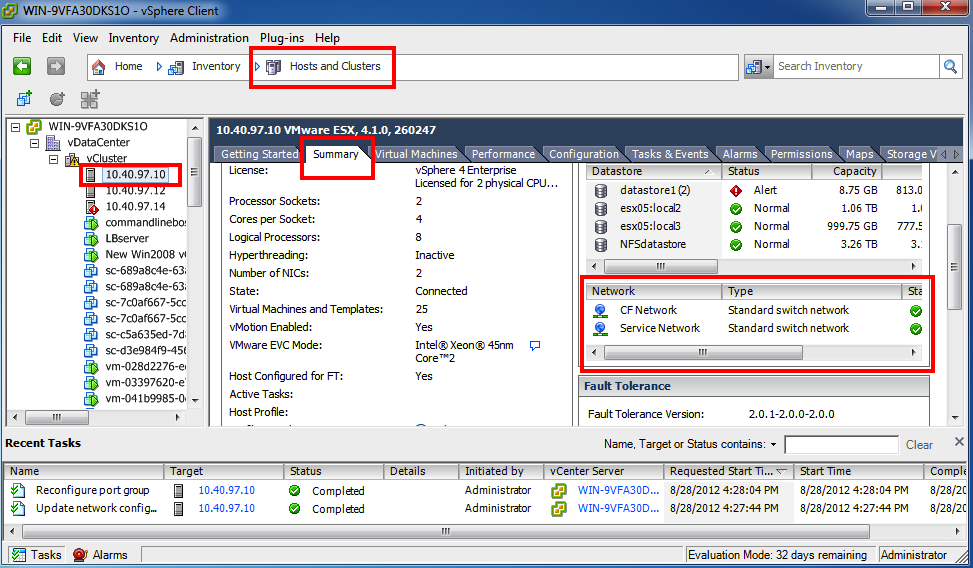
请使用现有的虚拟交换机：



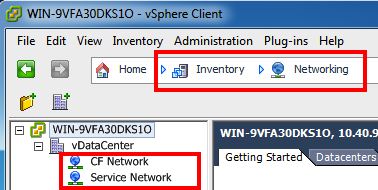
在下一步中，将网络标签重命名为“CF Network”。如果网络管理员已经指定了VLAN ID，请相应地输入 CF VLAN ID。



然后单击“完成”(Finish)，这样便完成了网络创建。重复上述步骤创建“Service Network”，直接将网络标签命名为“Service Network”即可。 务必要让同一群集内所有主机上的网络名称都保持完全相同。下图显示已经为某个主机创建了两个网络。我们将这两个网络分别命名为“CF Network”和“Service Network”。稍后在 BOSH 和 Cloud Foundry 的 yml 文件中将会用到这两个名称。

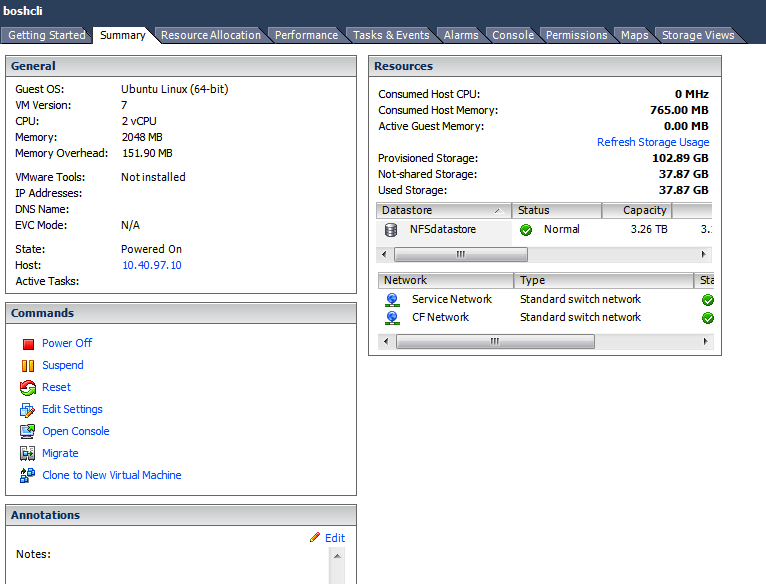


此外，如果您从“数据中心”(Datacenter) ->“清单”(Inventory) ->“网络”(Networking) 视图中进行查看的话，显示如下：



**为 BOSH CLI 创建一个虚拟机**

从 vCenter 中，我们选择该群集中的主机之一来创建一个虚拟机。为此，请单击“创建新虚拟机”(Create a new virtual machine)。我们将在此虚拟机上安装 64 位 Ubuntu 10.04 操作系统。为此虚拟机分配 2 个虚拟 CPU、2 GB 内存、20 GB 磁盘空间（或者更多）。在安装期间，一定要手动设置网络。



我们现在开始在该虚拟机上安装 BOSH CLI。为此，请登录到 Ubuntu，然后遵照以下步骤操作。（请注意，这些步骤大都摘自 BOSH 官方文档。为了您方便起见，在此将它们列出来。）

在安装前更新一下Ubuntu的apt-get：

$ sudo apt-get upgrade (takes long time, 200M+ files to download, optional for bosh install)$ sudo apt-get dist-upgrade (200M+ to download, update the kernel and image, optional)$ sudo reboot

安装BOSH CLI需要完整的Ruby环境，所以接下来我们分别介绍如何使用rbenv和rvm这两种Ruby版本管理器来安装Ruby和BOSH CLI。其中使用rvm安装的方法相对简单且易于维护，建议大家使用。

**方法一：通过rbenv安装Ruby和bosh cli**

1.安装 Ruby 的依赖项：

$ sudo apt-get install git-core build-essential libsqlite3-dev\curl libmysqlclient-dev libxml2-dev libxslt-dev libpq-dev genisoimage

2.获取最新版本的rbenv

$ git clone git://github.com/sstephenson/rbenv.git .rbenv

3.将 ~/.rbenv/bin 添加到您的 $PATH 以便能够访问rbenv命令行实用程序

$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bash\_profile

4.将rbenvinit添加到您的 shell 以启用填充程序 (Shim) 和自动完成

$ echo 'eval "$(rbenv init -)"' >> ~/.bash\_profile

5.下载 Ruby 1.9.2注意：您也可以使用适用于 rbenv 的 ruby-build 插件来构建 ruby。请参见https://github.com/sstephenson/ruby-build

$ wget http://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.2-p290.tar.gz

6.将 Ruby 解包并安装

$ tar xvfz ruby-1.9.2-p290.tar.gz

$ cd ruby-1.9.2-p290

$ ./configure --prefix=$HOME/.rbenv/versions/1.9.2-p290

$ make

$ make install

7.重新启动您的 shell 以使路径更改生效$ source ~/.bash\_profile

8.将您的默认 Ruby 设置为 1.9.2 版本$ rbenv global 1.9.2-p290

注意：使用此方法时可能需要重新安装 rake 0.8.7 gem$ gem pristine rake

9.更新rubygem并安装捆绑包。 注意：安装 gem（gem install 或 bundle install）后，请运行rbenv rehash 以添加新的填充程序

$ rbenv rehash

$ gem update –system

$ gem install bundler

$ rbenv rehash

最后，安装 BOSH CLI：

1.安装 gerrit-cli gem：$ gem install gerrit-cli

2.安装 BOSH CLI：

$ gem install bosh\_cli

$ rbenv rehash

$ bosh –version

如果一切运行顺利，最后一个命令将会显示您刚刚创建的 BOSH 版本。这表明 BOSH CLI 已安装成功。

**方法二：使用rvm安装Ruby和bosh cli：**

$ sudo apt-get install curl

\curl -L https://get.rvm.io | bash -s

在.bashrc中添加rvm的环境变量，否则退出终端后再进入，rvm不会被正常加载：

[[ -s "$HOME/.rvm/scripts/rvm" ]] && source "$HOME/.rvm/scripts/rvm"

$ source .bashrc

$ source ~/.bash-profile

输入下面指令来验证rvm是否安装成功：

$ type rvm |head -1

正确的返回结果：

rvm is a function

然后重启终端rvm就可以工作了。

执行$ rvm requirements

rvm的依赖包可以用过这条命令列出，经过测试，只要把这些依赖装好，之后的cli和bosh deploy gem的依赖，都已经自动加入了

执行rvm install 1.9.2 这个ruby的安装非常耗费时间，一般在半个小时左右

执行ruby -v 检测ruby版本

安装一些bosh cli所需的gem包：

$ gem install bundler

$ gem install rake

$ gem install gerrit-cli

$ gem install bosh\_cli

执行$ bosh --version看到版本号，说明bosh cli安装成功