vSphere AutoLab

v1.1a



www.labguides.com

Your lab on your laptop



Table of Contents

vSphere AutoLab 3

Acknowledgements 4

Lab Outer Platform 5

Outer platform setup 6

VMware Workstation Setup 6

VMware ESXi Setup 8

Deploy on ESXi 8

VMware Fusion 4 Setup 10

VMware Fusion 5 Setup 12

VMware Player Setup 16

The Virtual Machines 17

Build the Lab 18

Task 1 – Prepare the prebuilt VMs 18

Task 2 – Build DC, Windows Infrastructure 21

Task 3 – Build VC, vCenter 24

Task 4 – Build ESX servers 25

Task 5 – Populate vCenter 26

Task 6 - Add vSphere Management Assistant 27

Lab Build Time 27

Shutting the lab down 28

Accessing the built lab 29

As Built Documentation 30

IP Addressing 30

UserIDs 31

Databases 31

Host Network 31

Host and Shared Storage 31

Rebuild Process 33

Veeam ONE Installation 34

Veeam Backup & Replication installation 45

VMware View Installation 49

VMware vCloud Director Installation 50

Troubleshooting 54

AutoLab Changes and Futures 57

# vSphere AutoLab

This AutoLab kit is designed to produce a nested vSphere 5.1, 5, or 4.1 lab environment with minimum effort. Prebuilt shell VMs are provided along with automation for the installation of operating systems and applications into these VMs. The AutoLab download contains only freely redistributable software and empty VMs; you must bring your own licensed software installers to complete the build. The lab build was originally created to aid study towards VCP5 certification however it has many other possible uses. The AutoLab has grown to allow evaluation and testing of additional software, Veeam management and backup software, VMware View and VMware vCloud director.

The project lives at [www.labguides.com](http://www.labguides.com) and updates will occur there. Details of the insides of the AutoLab will be published at [www.ProfessionalVMware.com](http://www.ProfessionalVMware.com) as time allows.

These instructions are not intended for the absolute beginner. They will allow someone with moderate server infrastructure knowledge to rapidly build a vSphere lab.

### About Me

I’m [Alastair Cooke](https://twitter.com/#!/DemitasseNZ); I live in New Zealand and work as a trainer, consultant and writer for my own company [Demitasse](http://www.demitasse.co.nz). I host the [APAC Virtualization Podcast](http://apacvirtual.com) to help build a community with a focus around the region I work in and also host the APAC edition of the [ProfessionalVMware #vBrownBag podcast](http://professionalvmware.com/brownbags/).

This is version 1.1a. A fair amount of testing has been done, but there will be things that don’t work in your environment as they do in mine. Please let me know how you find the AutoLab and what needs improving or adding. You can email Nick or myself through [feedback@labguides.com](mailto:feedback@labguides.com?subject=AutoLab%20Feedback).

# Acknowledgements

This project would not have come to the world without the contributions of numerous others.

First of all my Wife, without whom I wouldn’t be where I am and who is very understanding of my need to do this work despite giving it away for free. Thanks Tracey.

[Nick Marshall](https://twitter.com/#!/nickmarshall9) [virtualnetworkdesign.com](http://www.virtualnetworkdesign.com) A previous project with Nick led to this project. His testing and QA on the lab build through its various stages were invaluable.

[Grant Orchard](https://twitter.com/grantorchard) for the vSphere 5.1 build.

[Damian Karlson](https://twitter.com/#!/sixfootdad) [damiankarlson.com](http://damiankarlson.com) for adding vCloud Director to the AutoLab, documenting Fusion 5 setup, bug fixes, and adding features.

[James Bowling](http://www.twitter.com/vSential) [www.vsential.com](http://www.vsential.com/) for documenting the full Fusion 4 setup for the AutoLab.

[Ariel Antigua](http://arielantigua.com/weblog/) for hanging out on the [AutoLab support forum](http://www.labguides.com/forums/index.php) and helping out when people have troubles.

[Veeam](http://www.veeam.com/) Aside from producing some great software that supports virtualization and then giving plenty of it away for free, Veeam is also sponsoring the AutoLab, making it much easier for us to develop new versions that support more products.

[FreeNAS](http://www.freenas.org/) The storage platform for the lab. Having an open source storage option makes the lab possible. This lab uses version 8.2.

[FreeSCO](http://www.freesco.org/) The router on a floppy. Another open source project that does great things and asks little in return.

[JouninTFTPd](http://tftpd32.jounin.net/) More free software; this time the file transfer tool of the PXE environment used to build the ESXi servers.

[VMware](http://www.vmware.com) for having such a great virtualization platform. I was amazed how much less resource the lab takes to run on ESXi than VMware Workstation.

[Microsoft](http://www.microsoft.com) for providing the operating system we most often need to virtualize.

[Labguides.com forum](http://www.labguides.com/forums/) moderators, help you through issues another example of community helping community.

This is the beta test crew, the ProfessionalVMware #vBrownBag team as well as some Kiwi and Australian helpers.

|  |  |  |  |
| --- | --- | --- | --- |
| [Cody Bunch](https://twitter.com/#!/cody_bunch) | [ProfessionalVMware.com](http://ProfessionalVMware.com) | [David Manconi](https://twitter.com/#!/dmanconi) |  |
| [Damian Karlson](https://twitter.com/#!/sixfootdad) | [damiankarlson.com](http://damiankarlson.com) | [Grant Orchard](https://twitter.com/#!/grantorchard) |  |
| [Josh Atwell](https://twitter.com/#!/Josh_Atwell) | [www.vtesseract.com](http://www.vtesseract.com) | [Tim Gleed](https://twitter.com/#!/timgleed) |  |
| [Michael Webster](https://twitter.com/#!/vcdxnz001) | [longwhiteclouds.com](http://longwhiteclouds.com) |  |  |

# Lab Outer Platform

### Required Hardware

The core lab can run on a single PC, a dual core 64bit CPU and a minimum of 8GB RAM is required, along with around 100GB of free disk space. This is the minimum; more RAM and a newer CPU will help a lot especially if you need to keep the lab running and use the PC for other things. An i7 CPU, 16GB of RAM and an SSD (120GB+) will make the lab host machine more responsive and reduce the amount of waiting during labs.

To make the labs as available as possible we have designed a configuration that will work with a less capable host. The lab was developed on a laptop purchased in 2009 and upgraded to 8GB of RAM, a large SSD and a second hard disk. Running the lab VMs off the SSD makes a big difference to the usability of the lab; it has also delayed the replacement of my old laptop.

As the AutoLab has grown there are parts that extend beyond vSphere; these components will require more RAM or for one of the ESXi hosts to be shut down to make RAM available. Running the labs under ESXi rather than Workstation does allow more efficient use of RAM, but requires a second PC for ESXi management. Another option you may choose to allow VMware Workstation to swap some VM RAM; for me this makes performance intolerable but others have had good results.

### Lab Virtualization Platform

All of the elements of the lab were developed in virtual machines under VMware Workstation v8; this version of workstation fully supports running ESXi server in a virtual machine on suitable physical hardware. The labs can also be run under VMware Player, VMware Fusion, and VMware ESXi 5. This document is based on deployment on VMware Workstation although there are directions on setting up Fusion and ESXi as lab platforms.

### Additional software

In addition to the AutoLab kit, lab host, and its virtualization software you will need a few other pieces of software. Below is a list with download links where these are freely available. For the older vSphere versions and PowerCLI you will need an account with VMware or a good contact at VMware or a VMware partner. The version 4.1 components are only required if you plan to build a 4.1 environment, usually so you can run the upgrade to 5.0 or 5.1.

* Required software for vSphere & vCloud lab:
  + From [MyVMware.com](http://MyVMware.com)
    - vCenter and ESXi installers for the versions you want to use; 4.1, 5.0 and 5.1 are supported.
    - VMware PowerCLI installer; use the same version as the vSphere & vCenter deployments.
    - vCloud binary and vShield (aka vCloud Networking & Security) appliance; 1.5 & 5.1 are supported.
  + VMware Tools’s windows.iso
    - VMware Workstation: located at C:\Program Files (x86)\VMware\VMware Workstation\
    - VMware Fusion: Finder -> Applications -> Show Package Contents of VMware Fusion -> Contents/Library/isoimages
    - VMware Player & ESXi: downloaded from [packages.vmware.com](https://packages.vmware.com/tools/esx/latest/windows/index.html)
  + Microsoft [Windows Server 2008 R2](http://technet.microsoft.com/evalcenter/dd459137.aspx) 180 day trial DVD, ISO file
  + Microsoft Windows 2003 Server 32bit CDROM, ISO file
  + [Veeam ONE and Veeam Backup & Replication installers](http://www.veeam.com/downloads)
* Required software for VMware View
  + From [MyVMware.com](http://MyVMware.com)
    - View 5.0 or 5.1 installers
* Optional software for vSphere & vCloud lab
  + VMware CLI for vSphere 5.1
  + [Microsoft SQL Server 2008 R2 SP1 - Express Edition Management Studio](http://go.microsoft.com/fwlink/?LinkID=228417&clcid=0x409)
  + VMware vMA
* Optional software for VMware View
  + Windows XP ISO

# Outer platform setup

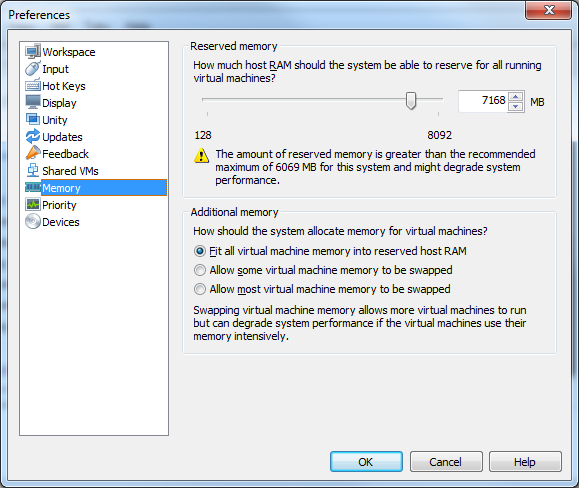
In the next few pages we will look at setting up the virtualization platform that will host the AutoLab, VMware Workstation, ESXi and Fusion. We also look at VMware Player but as I don’t have a spare machine to run Player the instructions are pretty thin; feel free to document what works for you and send to Alastair Cooke for inclusion.

You only need to follow the instructions for your chosen platform, but make sure you follow all the instructions for that platform.

## VMware Workstation Setup

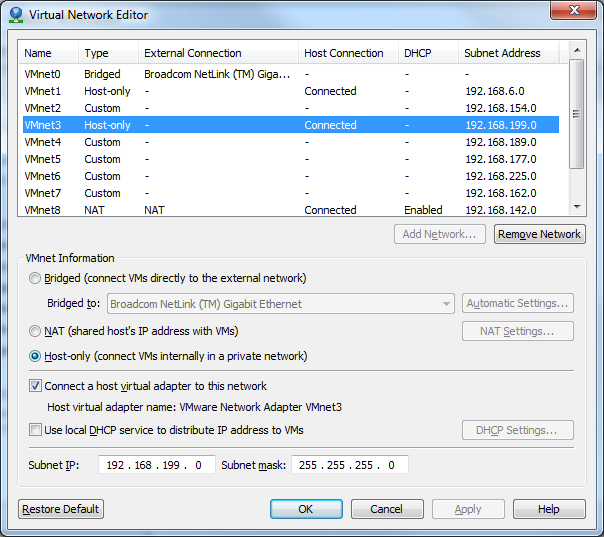
The VMware Workstation build is designed to work with a host with 8GB of RAM and VMware Workstation version 8.0. The host operating system must be 64bit and all the CPU virtualization features must be enabled in the BIOS in order to be able to run the 64bit VMs. Placing the lab files on a fast disk and having a host with more RAM will make the lab run faster.

The main setup required is to reserve 7168MB of RAM for VMs and choose to **Fit all virtual machine memory into reserved host RAM**. To configure both of these settings, go to the **Edit** menu **Preferences…** item.



If your lab host has more than 8GB of RAM select a larger value and allocate more RAM to the VMs.

The other requirement is to configure the lab network. Under the **Edit** menu is the Virtual Network Editor. Select the **VMnet3** object. If that network isn't present, click 'Add Network' button to add it. Make sure **Host Only** is selected in **VMnet Information** and that **Use Local DHCP Service to distribute IP addresses to VMs** is not selected. The **Subnet IP** should be 192.168.199.0 with a **Subnet Mask** of 255.255.255.0. You may use the option **Connect a host network adapter to this network** to allow your PC direct access to the lab network, otherwise all network access will be through the router VM. Unless you are going to run multiple copies of the lab, each on a different VMNet but the same IP range, it is easiest to have the host connect to the network.



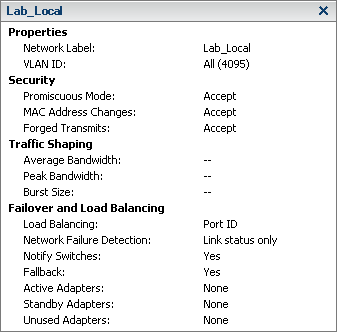
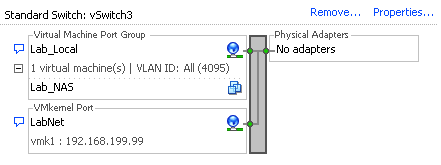
### Checklist

* VM RAM configured
* VMnet3 setup

# VMware ESXi Setup

The lab runs extremely well under ESXi; in my lab the basic vSphere cluster consumes a peak of 4.5GB of RAM during build and about 40GB of disk space, although 220GB is allocated. A higher performance disk system also reduces the lab build time as this is mainly disk IOPS constrained.

The lab will usually use a portgroup on an Internal Only Standard vSwitch, i.e. one with no physical NICs attached. The default portgroup name is **Lab\_Local** and this portgroup must be setup to allow Promiscuous mode and must be on VLAN ID 4095. When the vApp is imported you may choose to use a different portgroup for each vApp; the portgroups should be on their own vSwitch to avoid conflicts. Your ESXi server should also have a VMkernel port on this network, IP 192.168.199.99. This will allow access to the Build share on the NAS VM.



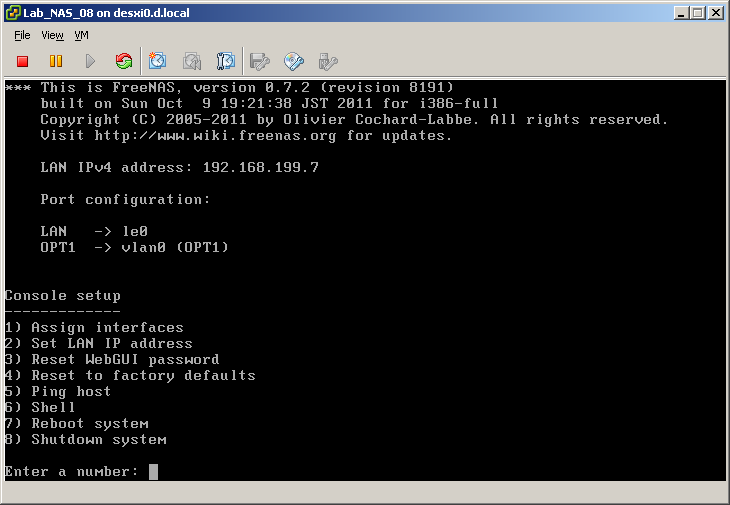
The router VM also connects to your main network, the default configuration calls this network External.

To enable the nested ESXi servers to run 64bit VMs your outer ESXi server needs to pass the Intel VT or AMD V feature into the VMs. Open a command prompt on your ESXi server, either through SSH or on the local console, and type the following command

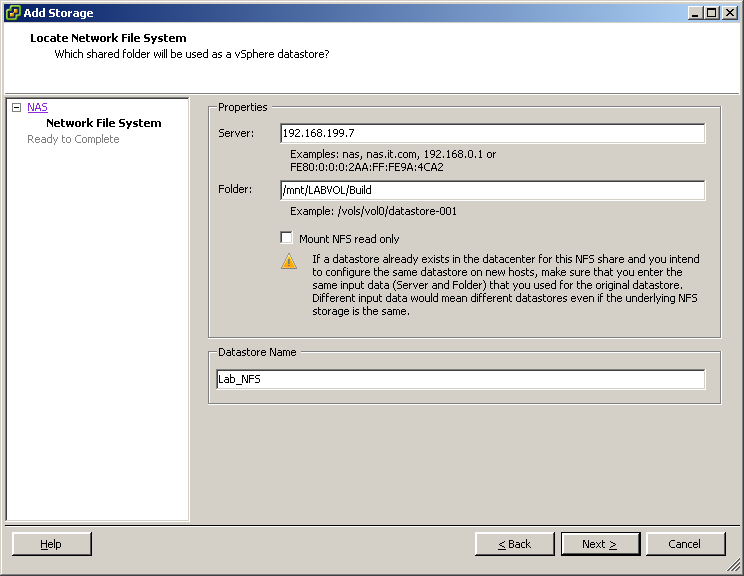
echo 'vhv.allow = "TRUE"' >> /etc/vmware/config

## Deploy on ESXi

The lab is distributed as a single ova file, this contains the NAS VM. Deploy the ova and power on the NAS VM.



Once the NAS has booted create a new NFS datastore pointing to the Build share, server **192.168.199.7** and Folder **/mnt/LABVOL/Build** as shown below



Once the datastore is created use the vSphere Client datastore browser and browse to **\Automate\ShellVMs** where you will find the remaining lab VM folders. The VMs must not be run from this location as they won't perform well and the datastore will quickly run out of space.

* If you have vCenter then register each VM then migrate it to its proper datastore. Do not power the VMs up on the Lab\_NFS datastore as it will perform poorly and then run out of space.
* If you do not have vCenter you will need to use the datastore browser to copy the VMs from the Lab\_NFS datastore before registering the VMs. The copy takes quite a while as it appears not to respect the thin provisioned disks.

Finally the VMs need to have their CDROM and floppy drives attached to media images. You may need to copy the boot floppy images from the Build share, in **\Automate\BootFloppies** to another datastore. The floppy images match the VM names, apart from vCloud which doesn't need a floppy.

### Checklist

* Lab\_Local Portgroup
  + VLAN 4095
  + Promiscuous mode allowed
* VHV allowed on physical host
* VMkernel created (optional)
* Lab NFS datastore accessible (optional)

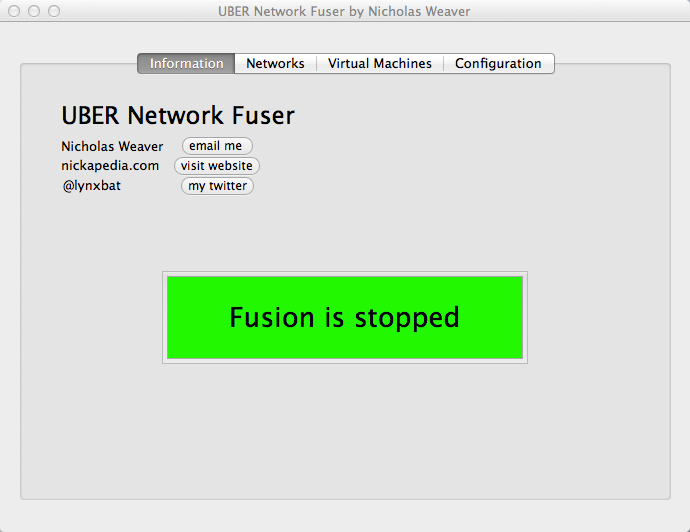
## VMware Fusion 4 Setup

Huge thanks to [James Bowling](http://www.vSential.com) [@vSential](http://www.twitter.com/vSential) for documenting the whole Fusion 4 setup process.

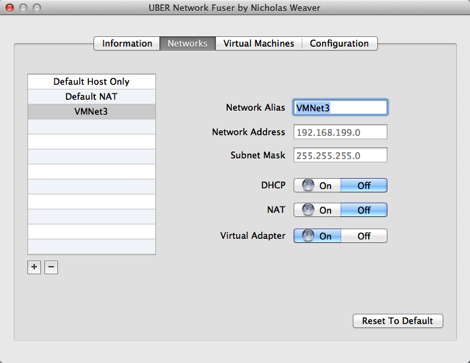
The Fusion build is designed to work with a host with 8GB of RAM and VMware Fusion version 4.0. The host operating system must be 64bit and all the CPU virtualization features must be enabled in the BIOS in order to be able to run the 64bit VMs. Placing the lab files on a fast disk and having a host with more RAM will make the lab run faster.

### Configure Lab Network

Unlike the VMware Workstation setup, configuring the network is not available within the VMware Fusion 4 settings. To create the required network we will open Nick Weaver’s [UBER Network Fuser](http://nickapedia.com/2012/01/10/breaking-new-ground-an-uber-tool-for-the-mac/) (UNF), as seen here:

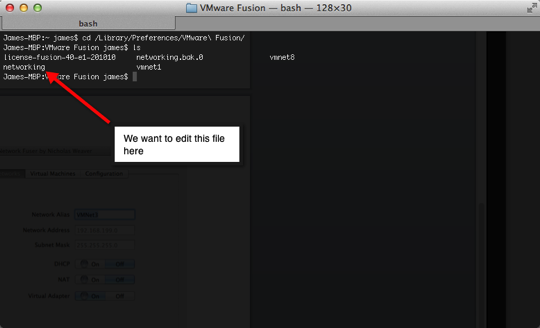


Make sure that you have VMware Fusion stopped as you will not be able to make any changes to anything within UNF. Once you have UNF open select the Networks section where you will be presented with a list of two networks, Default Host Only and Default NAT. Click on the ‘+’ to add a new network. Change the name of the network to **VMnet3** and hit enter. One thing you will notice is that the subnet has been selected at random. Don’t worry…this is by design. We will fix this in a minute. Before we do that, you will want to set DHCP and NAT to off, then set Virtual Adapter to on, like so:

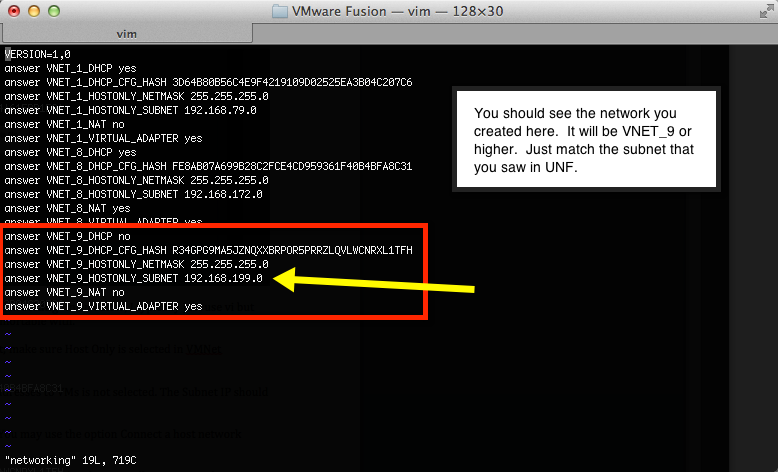


Next we will actually change the subnet. To do this, open a terminal and do the following:

cd /Library/Preferences/VMware\ Fusion



Now we need to open the file **“networking”** in an editor. I typically use vi but use whatever editor with which you’re most comfortable. You will want to change the VNET\_X\_HOST\_ONLY\_SUBNET to reflect the required 192.168.199.0. See the screenshot if you are unsure:



Once you make the change save the file and go on to the next step in deploying the AutoLab. Don’t misplace UNF as you will need it again after we place the VMs in the appropriate place.

### Placement of AutoLab Virtual Machines

In the typical VMware Workstation setup you could potentially place the VMs anywhere on your machine. To simplify the configuration in VMware Fusion and use with UBER Network Fuser (UNF) we will place the VMs in the default VMware Fusion Virtual Machines directory. This is typically:

**HD -> Users -> username -> Documents -> Virtual Machines**

Simply copy all of the folders from the extracted AutoLab zip file into the above directory. Wait…you aren’t done just yet. VMware Fusion creates virtual machines in directories that are named like so:

**esxi.example.vmwarevm**

If you notice it uses an extension of **“.vmwarevm”** to allow the association with VMware Fusion and this is the only way UNF will be able to see the VMs. Rename the folders according to the list below (or see the section on Fusion Scripts make things a bit quicker):

* **CS1.vmwarevm**
* **CS2.vmwarevm**
* **DC.vmwarevm**
* **Host1.vmwarevm**
* **Host2.vmwarevm**
* **NAS.vmwarevm**
* **Router.vmwarevm**
* **SS.vmwarevm**
* **V1.vmwarevm**
* **VBR.vmwarevm**
* **VC.vmwarevm**
* **vCloud.vmwarevm**

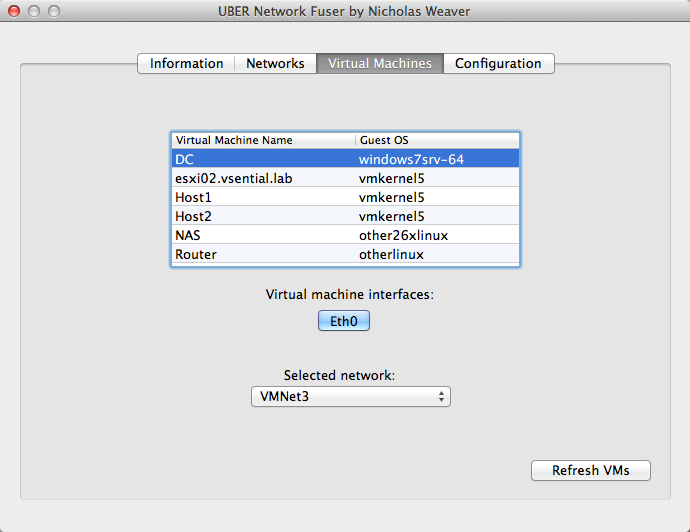
Alternatively, you can run this script in order to quickly change all of the virtual machine folder names.

ls -dF ~/Documents/Virtual\ Machines/Lab\_Local/\*/ | grep -v ".vmwarevm" | awk 'BEGIN{FS="//"} {print "mv "$1"/ "$1".vmwarevm/"}' | bash

### Modify Virtual Machine Network Settings

Now that we have renamed all of the folders we need to open UNF again so we can associate our VMnet3 network that we created at the beginning with each vmnic on each VM. This is simple but needs to be done on each VM and each vmnic. Once you have opened UNF, click on **Configuration.** Make sure that your paths are set appropriately, if they are not then correct them.

Next click on **“Virtual Machines”**, you should see a list of virtual machines now. If you don’t, hit **Refresh VMs** and it should populate. Select one of the VMs in the list and you can change the network tied to that vmnic by clicking on the vmnic.

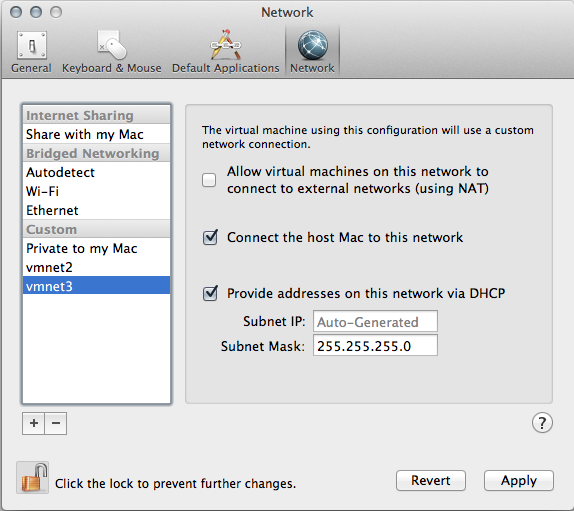


Do that for each VM and each interface that needs to be tied to the new network. Once this is complete you can go ahead and move on to the typical steps taken for deploying the AutoLab.

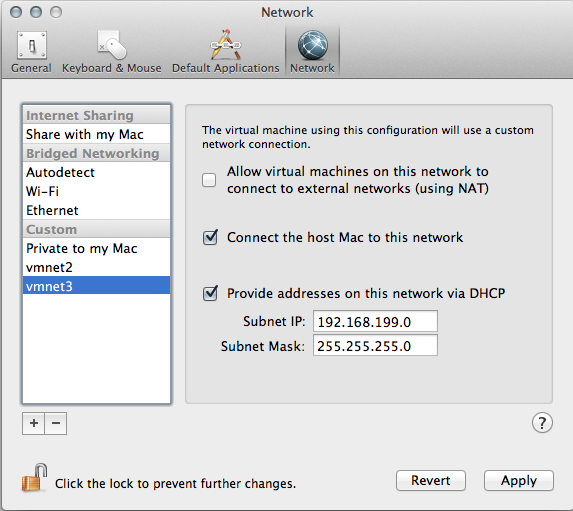
## VMware Fusion 5 Setup

### Configure the Lab Network

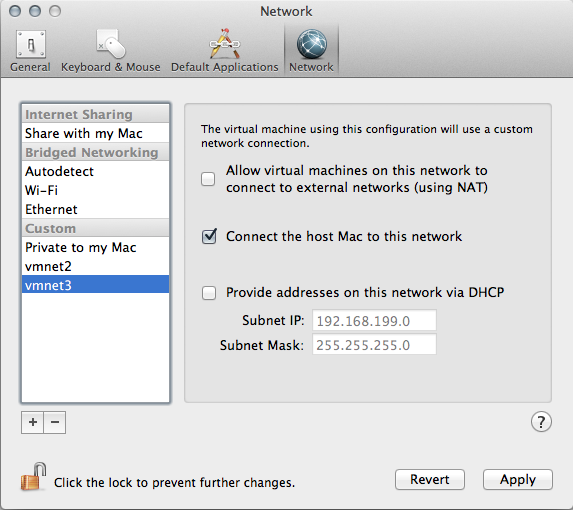
* Open VMware Fusion menu > Preferences > Network tab
* Unlock the window if necessary
* Click the plus sign at the bottom left to create vmnet3



* In the Subnet IP field replace Auto-Generated with 192.168.199.0



* Uncheck 'Provide addresses on this network via DHCP'



* The vmnet3 interface on the host Mac will get an IP set to 192.168.199.1 automatically.

### Placement of AutoLab Virtual Machines

In the typical VMware Workstation setup you could potentially place the VMs anywhere on your machine. To simplify the configuration in VMware Fusion and use with UBER Network Fuser (UNF) we will place the VMs in the default VMware Fusion Virtual Machines directory. This is typically:

**HD -> Users -> username -> Documents -> Virtual Machines**

Simply copy all of the folders from the extracted AutoLab zip file into the above directory. Wait…you aren’t done just yet. VMware Fusion creates virtual machines in directories that are named like so:

**esxi.example.vmwarevm**

If you notice it uses an extension of **“.vmwarevm”** to allow the association with VMware Fusion and this is the only way UNF will be able to see the VMs. Rename the folders according to the list below (or see the section on Fusion Scripts make things a bit quicker):

* **CS1.vmwarevm**
* **CS2.vmwarevm**
* **DC.vmwarevm**
* **Host1.vmwarevm**
* **Host2.vmwarevm**
* **NAS.vmwarevm**
* **Router.vmwarevm**
* **SS.vmwarevm**
* **V1.vmwarevm**
* **VBR.vmwarevm**
* **VC.vmwarevm**
* **vCloud.vmwarevm**

Alternatively, you can run this script in order to quickly change all of the virtual machine folder names. This script assumes that you are using OSX 10.8.3 Terminal and that you are not running as root.

cd ~/Documents/Virtual\ Machines/Lab\_Local/; ls -dF \* | grep -v -e "vmwarevm" -e "txt" | awk 'BEGIN{FS="/"} {print "mv "$1"/ "$1".vmwarevm/"}' | bash

### Modify Virtual Machine Network Settings

There are two ways you can do this, as follows:

The first is to go to the Virtual Machines folder (typically under your user’s Documents folder), Show Package Contents of each AutoLab virtual machine, edit the vmx file, and change ‘vmnet3’ to ‘VMnet3’.

The second is to run the script below. This script assumes that you are OSX’s Terminal and that you are not running as root. It also assumes that you followed the Fusion 5 Lab network setup instructions correctly. ☺

find ~/Documents/Virtual\ Machines/Lab\_Local -name "\*.vmx" -print0 | xargs -0 sed -i "" 's/VMnet3/vmnet3/g'

## VMware Player Setup

I also don’t have a PC that could run the lab and doesn’t have Workstation installed. The Workstation VMs should be fine, but like Fusion the network editor isn’t there. A quick Google showed [the answer](http://www.carbonwind.net/Virtualization/VMware-Player-Networking-Options/VMware-Player-Networking-Options.htm). Luckily it looks like it’s included in the installer but not installed by default. If the above link fails then Google for **VMware player "Virtual network editor"**

## The Virtual Machines

The VMs in the download package are setup so you can run them under VMware Workstation on an 8GB physical machine. The Windows VMs will be heavily overcommitted in the guest OS however this provides better VM performance than allowing VMware Workstation to page these VMs out to a hard disk. If you have sufficient RAM you can increase the allocation to the VMs or allow VMware Workstation to swap some RAM. Unless you have a lot of RAM in your lab host you are unlikely to be able to run everything at the same time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Group** | **Function** | **Min. RAM** | **Ideal RAM** |
| DC | Core | Domain Controller | 384MB | 1GB |
| VC | Core | Virtual Centre | 1.25GB | 2.5GB |
| NAS | Core | Shared Storage | 128MB | 512MB |
| Router | Core | In and outbound access | 12MB | 16MB |
| Host1 | Core | ESXi Server | 2GB | 4GB |
| Host2 | Core | ESXi Server | 2GB | 4GB |
| VMA |  | Command line management | 128MB | 600MB |
| V1 | Veeam | Veeam ONE | 1GB | 2GB |
| VBR | Veeam | Veeam Backup & Replication | 1GB | 2GB |
| CS1 | View | Connection Server 1 | 1GB | 2GB |
| CS2 | View | Connection Server 2 | 1GB | 2GB |
| SS | View | Security Server | 512MB | 1 GBB |
| vCloud | vCloud | vCloud Director | 1.5GB | 3 GB |

The configuration of the operating system inside the VMs is documented in the “As Built Documentation” section

With all thirteen lab VMs running at their minimum RAM my AutoLab build host uses just under 16GB of RAM, more RAM may be required to run a full vCloud environment at the same time as View.

# Build the Lab

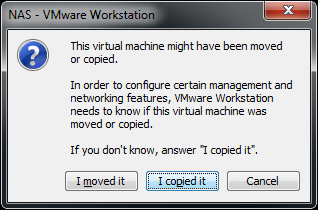
These steps use the full set of automation to build a complete lab environment. The steps should be completed in order with each step completing before starting the next step. If you are building the lab to train on then you may want to use the minimum automation and manually complete steps. The build steps are the same irrespective of the outer virtualization platform, i.e. Workstation, ESXi or Fusion.

You may choose not to use all of the vSphere build automation; I suggest you start with a complete fully automated build to make sure that all of the parts are in the right place. Once your first lab build is complete it’s a fairly simple matter to rebuild with less automation so you can manually complete the tasks that you wish to learn.

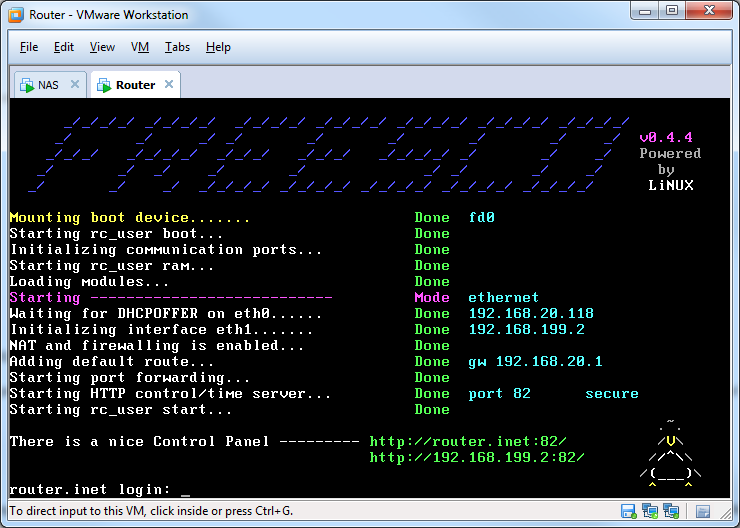
## Task 1 – Prepare the prebuilt VMs

Extract the vSphere AutoLab archive to a folder and open all the VMs with VMware Workstation or Fusion.

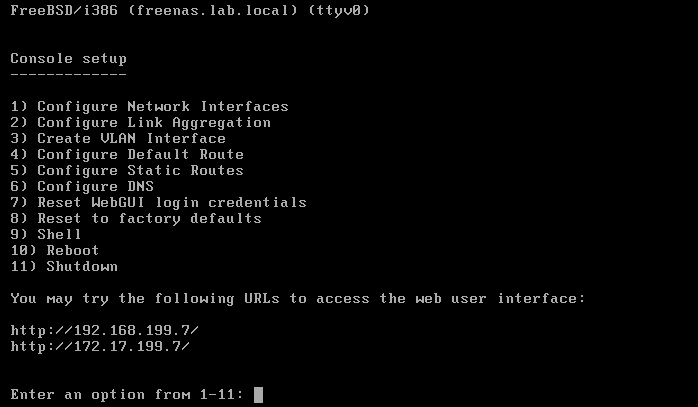
When you power on each VM you may be asked whether you moved or copied the VM. Always answer “I copied it” for these VMs, this way a new UUID and MAC address is assigned for each VM and makes running multiple isolated copies of the lab possible.



If you are using VMware Workstation or Fusion then you may not need the Router VM, it is used to allow outbound connectivity from the AutoLab network and inbound management. If your PC has an IP address on the AutoLab network then you can safely leave the Router powered off and move on to the NAS VM. If you are building the lab on ESXi then you will need the router to allow access to the lab. Power on the Router VM, wait for it to boot to the logon prompt. The router published the windows share “Build” from the NAS through its external interface; this is the IP address at the end of the line “Waiting for DHCPOFFER on eth0”. This is useful if you don’t have a PC connected to the Lab network such as deploying a lab on ESX.



Power on the NAS VM, wait for it to boot to the logon prompt



If you watch the console of the NAS VM you may see messages about not having enough RAM for ZFS, this does not cause any issues and can be ignored

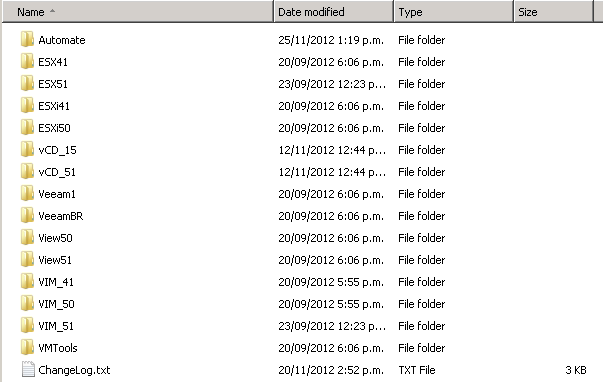
If your PC has an IP address on the VMNet3 network, it is usually 192.168.199.1. Ping 192.168.199.7 which is the NAS, if this succeeds open the window share \\192.168.199.7\Build.If the ping fails then you can access though the NAS using the external IP address of the router. In the example above the external address is 192.168.20.118 (it’s on the **Waiting for DHCPOFFER on eth0** line) so the share is [\\192.168.20.118\Build](file:///\\192.168.20.118\Build).

*VMware Fusion note: Using Finder, press Command+K to open the “Connect to Server…” dialog box. Enter smb://192.168.199.7 and connect as Guest. If you connect as a named user or connect using NFS, your user permissions will write the files in a way that renders them unaccessible. (This can be fixed with chown & chmod if you’ve already made this mistake.)*

### Build Share

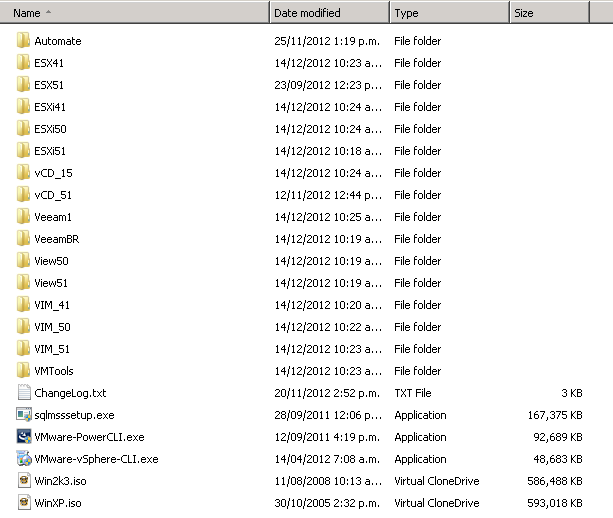
The build share is the central repository for the installers and scripts that are used inside the AutoLab, all of the software that will be used must be on this share before the rest of the builds begin. There are folders for each piece of software; most of these folders need to have the ISO files extracted into them. The only ISO files that are required are the Windows 2003 and XP install ISOs placed in the root folder; all sub folders should contain files extracted from ISOs or ZIP files.

Here are the contents of an empty build share:

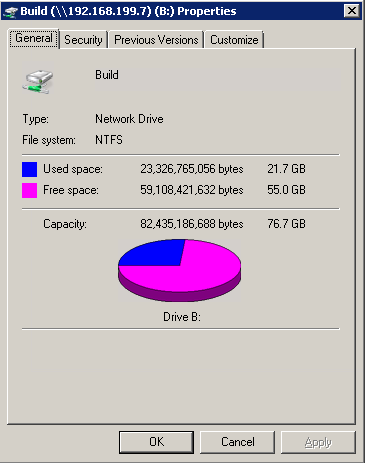


* The ESX and ESXi folders need to have the ESX and ESXi installer ISO extracted into them.
* The vCD folders should contain the vCloud director installer binary and vShield OVA as well as the Oracle installer rpm package, the vCloud install section below has links to download sources. The vCD\_51 folder is a placeholder at present as it’s not yet included in the AutoLab.
* The Veeam folders are really for organization, as I haven’t managed to automate the Veeam software installs.
* The View folders should hold the View Agent, Composer and Connection server installers for the View version.
* The VIM folders need to have the vCenter installers extracted into them.
* The VMTools folder must contain the extracted contents of the Windows VMware Tools ISO for the outer virtualization platform, for VMware Workstation the ISO can be found in **C:\Program Files (x86)\VMware\VMware Workstation\windows.iso**
* In the root folder of the build share should be the following files:
  + VMware-PowerCLI.exe - the installer for PowerCLI renamed. Do not use a newer version of PowerCLI than your vCenter as this will cause scripts to fail.
  + Win2K3.ISO – Windows 2003 Server 32bit install ISO, used to create an unattended Win2K3 install ISO in the VC build and then install windows in a nested VM
  + WinXP.ISO – Windows XP 32bit with SP3 install ISO, also used to create unattended install ISO for nested VM for View
* You may also place the following optional files are not required for the AutoLab but if they are found they will be used.
  + SQLManagementStudio\_x64\_ENU.exe– Microsoft SQL Server 2008 R2 SP1 - Express Edition Management Studio installer, will be installed on DC so you can do database troubleshooting. Download from <http://go.microsoft.com/fwlink/?LinkID=228417&clcid=0x409>
  + VMware-vSphere-CLI.exe - so you can play with command lines without using the VMA

A fully populated Build share looks like this:

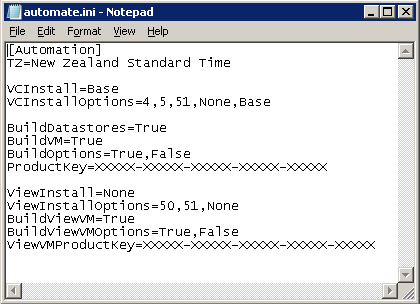


In my lab the fully populated Build share contains 22GB of files, if you don’t have all of the vSphere versions or other software in your lab then your Build share will be smaller.



### Automation Control

In the Automate folder on the build share you will find Automate.ini, this file controls the level of build automation. The VCInstall, BuildDatastores, BuildVM and Product Key lines answer the questions in the VC server build and Add Hosts scripts. The TZ line allows the automatic setting of the Time Zone in the Windows VMs, it uses the TZUtil command, to get the right text for this run “tzutil /g” on your PC and paste the result in place of my time zone. The View section allows the automation of the View connection servers and security server along with the setup of the nested Windows XP VM that can be used as a desktop for View. There are some additional settings for View, check the View section later in this document.



If you want a minimally configured Windows build in the VC machine then change the VCInstall line to VCInstall=None, only VMware Tools will be installed

If you want the VC machine to have the SQL Native client installed and a few other steps that aren’t directly the vCenter install then change the VCInstall line to VCInstall=Base

If you want a fully automated vCenter install then change the VCInstall line to VCInstall=4, 5 or 51 depending on the vCenter version you want installed.

The BuildDatastores and BuildVMs entries control the AddHosts scripts; again you may choose the level of automation to allow you to focus on what you are building the AutoLab to test. The ProductKey line is used for the unattended build of a nested Windows 2003 VM.

For the initial test build I suggest having the automation fully build vSphere, datastores and VMs. This way you can be sure all the sources and setup steps work. Subsequent rebuilds can be less automated to allow you to do tasks manually.

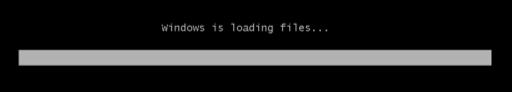
Do not move on to Task 2 until the build share is fully populated and you are happy with the automation options in the automate.ini file.

## Task 2 – Build DC, Windows Infrastructure

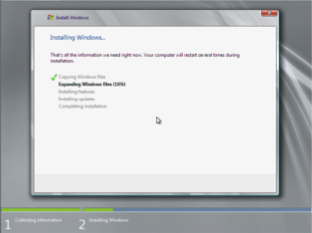
On both the VC and DC VMs make sure the CDROM drive is connected to a Windows Server 2008 R2 install ISO, connected at power on is recommended. AutoLab build v1.1 and later will work with any 2008R2 ISO, previous versions were locked to specific service pack levels. Also make sure that the right floppy image is connected, on Workstation and Fusion this is part of the package that you downloaded. On ESXi you may have to re-attach the floppies. Backup copies of the boot floppy images can be found on the build share in **\Automate\BootFloppies**.

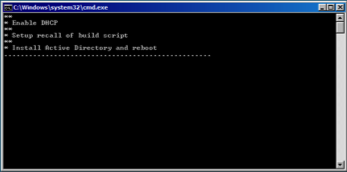
Power on the DC VM to start the unattended install. The first time you boot this VM it has a blank hard disk, so it will boot from the Windows installer CD and begin the build process. On subsequent reboots the installer will pass boot over to the hard disk unless you press a key, pressing a key at this prompt will completely rebuild the VM with no confirmation.

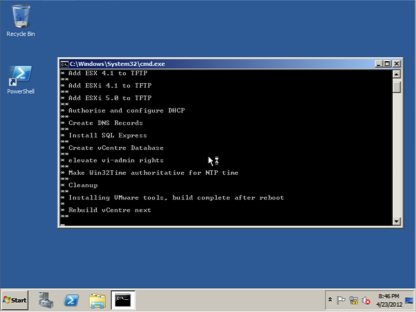




The VM will boot from the Windows Server 2008 R2 CDROM and use “autounattend.xml” file on the floppy disk image to automate the Windows install. This will take some time, go talk to your family for a while, or read some documentation to pass the time. On my laptop this takes around an hour. No input will be required from you through this process and you cannot start the other installs until it is complete.







After installing AD the VM will restart and install SQL Express and setup the PXE environment followed by installing the VMware tools. If these steps fail or does not start make sure your NAS VM is running and that you setup the Build share as outlined in task 1.

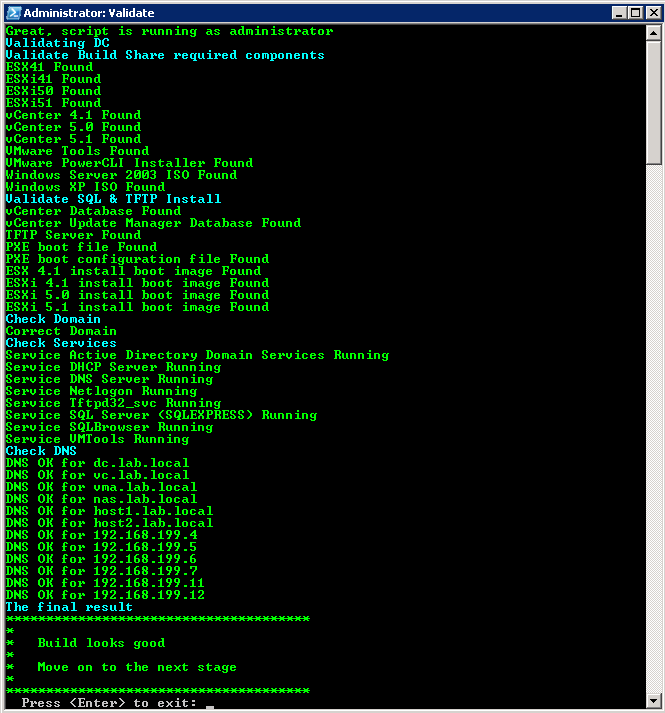


After the entire automated install completes the VM will reboot a final time, returning to the desktop as auto-logon is setup. At this point the Domain Controller is setup and ready.

If the PowerShell shortcut is missing from the desktop then the build may not be complete, check with the build log in c:\Buildlog.txt. There is also a troubleshooting section at the end of this guide.

There is a script to test that the build has completed successfully, and that the Build share was correctly populated. Open the Validate script as administrator by right clicking on the **Validate** shortcut on the desktop and selecting **Open as Administrator.**





As you would expect green is good, yellow is OK, red not so good. If the build share is not correctly populated then the DC VM will need rebuilding as most software installers come from the build share. There is a log of the build in c:\Buildlog.txt which may help; the BuildLog shortcut on the desktop opens this file.

Once the Validate script completes with green you are ready for task 3.

## Task 3 – Build VC, vCenter

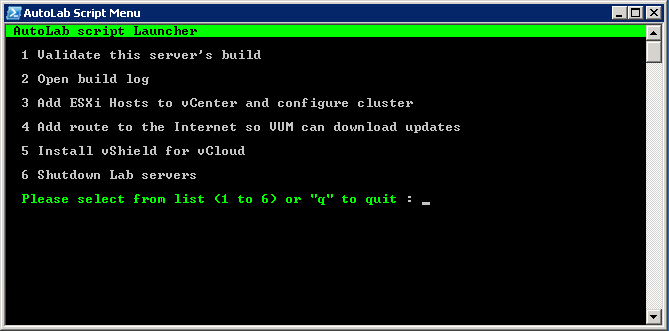
This process begins the same as the DC build, attach the ISO and appropriate floppy. Then boot from the Windows install disk, the floppy contains the “autounattend.xml” file.

Power on the VC VM and allow it to boot from the CD, as with the DC VM booting from the CD will always rebuild without any prompt. This build will take another hour, leave it alone and do something else.

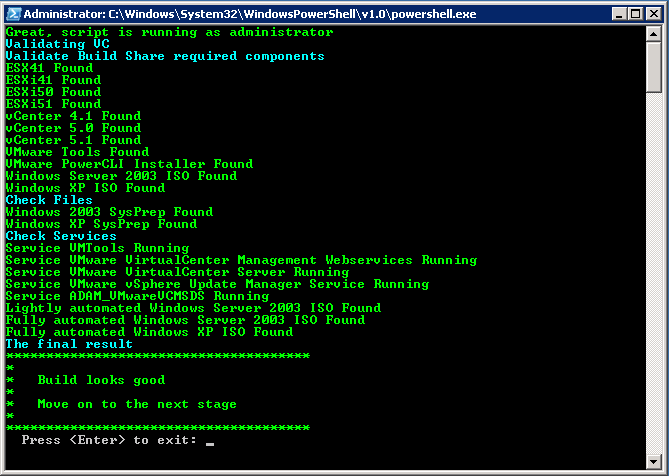
When the automation that you chose in the Automate.ini file completes the VM will be left logged on and configured with Autologon for your convenience.



There are a few scripts that can be run on the VC; they are wrapped up in the Script Menu script on the desktop. The menu script does not require elevated privileges but some of the other scripts will prompt for permission to elevate when they are launched.



The same validate script that ran on the DC can be run on the vCenter server to validate it’s build and there is a build log in C:\Buildlog.txt, both are available in the Script Menu.



The basic VC build does not include a default gateway, to prevent needless VUM downloads when you rebuild repeatedly. To access the Internet from the VC or to access the VC via the Router from your external network, you will need to run the **Add Route…** option from the script menu.

It may take a few minutes after you are able to login before all the services are started, be patient as the “VMware VirtualCenter Management Webservice” can take a few minutes to start.

The desktop shortcut named **vSphere** launches the vSphere client and automatically logs into VC as the current logged on desktop user. PuTTY has pre-configured configurations for accessing the ESXi hosts and VMA, if you deploy the VMA.

## Task 4 – Build ESX servers

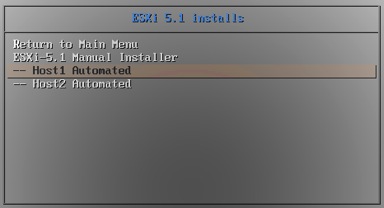
*VMware Fusion note: Unless you’re running Fusion as root, powering on the ESX servers will popup a prompt to let you know that the virtual machine is attempting to monitor all network traffic. See* [*this blog article*](http://www.jedi.be/blog/2010/12/09/automated-vmware-esx-installation-even-in-vmware-fusion/#admin_privilege) *for how to fix this behavior. For VMware Fusion 5, quit Fusion and open the OSX Terminal. Sudo or su to root as necessary and execute the following command:*

/Applications/VMware\ Fusion.app/Contents/Library/services.sh --stop  
/Applications/VMware\ Fusion.app/Contents/Library/services.sh --start

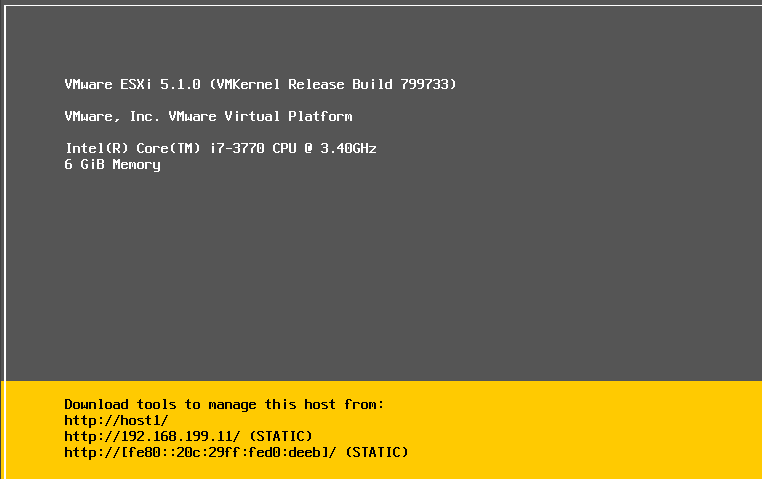
Power on the ESX VMs one at a time; wait for the build to complete on each one before starting the build on the next. If you plan to build as ESX 4.1 then change the OS type on the VMs to reflect this and increase the RAM allocation for host1 to 2.25GB to allow HA to configure successfully.

When the VMs boot they will use PXE to load a menu from the DC, use the keyboard arrow keys to choose the option for the ESXi version and server you wish to install. For each ESXi version there is a menu allowing the automated build of each ESXi host or a manual install from the network. The automated installs build with the standard IP addresses and no post build customization. The manual install behaves exactly as if you had booted from the ESXi installer ISO, asking you for all the required build information. You can find the required information in the **As Built** section of this document. For the initial test build use the automated builds for the same ESXi version as your vCenter.





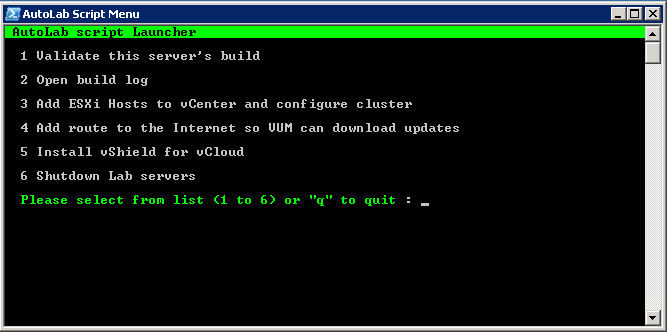
After the build your ESX server will be ready



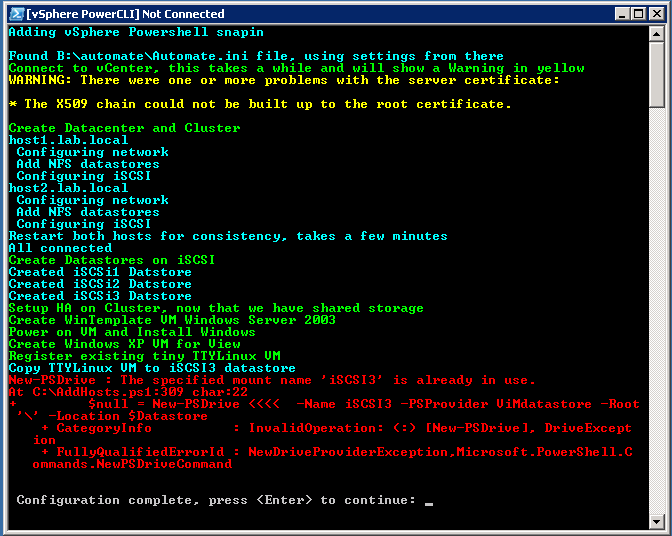
Build the second server as required. Confirm that both ESX servers have the correct static IP addresses before moving to the next stage.

## Task 5 – Populate vCenter

To automatically add the ESXi servers to vCenter, setup an HA and DRS cluster as well as networking and datastores on the ESXi servers use the **3 Add ESXi hosts…** option from the menu in the **AutoLab Script Menu** shortcut on the desktop.

The script will execute with a minimal amount of feedback, some yellow warning messages are usual, red means something has gone wrong. During the process CPU saturates on my lab host. The red here came up as I was documenting the build process, you may see the same result in your lab and safely not worry, I will look for its source ones I’ve finally shipped this version.



If you selected the options to create VMs and datastores in the Automate.ini file then the script will create these, if existing datastores and VMs exist these will be added to the vCenter inventory.

## Task 6 - Add vSphere Management Assistant

The vSphere Management Assistant (VMA) virtual appliance is not included with the AutoLab kit as it is large and is not applicable to all uses of the lab. If you are studying for the VCAP5-DCA exam or need a VMA for other purposes then you can add one to the lab. When you have imported the .ovf file edit the VM to use the VMNet3 network and 128MB of RAM. When you start the VMA appliance assign, the IP address of 192.168.199.6 and set a password; unfortunately you will not be able to use the lab default password. I use “VMw@re1!”.

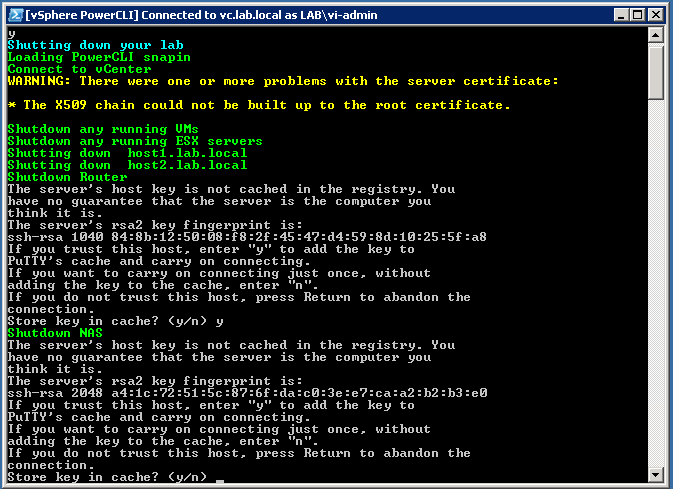
# Lab Build Time

This table gives an indication of the time to build a core vSphere lab

|  |  |
| --- | --- |
| Time | Action |
| 0:00 | Power on NAS and copy in contents of Build share |
| 0:15 | Power on VC VM, build automated |
| 1:15 | DC built  Power on VC VM, Windows build |
| 1:30 | Select vCenter 5 automated build |
| 2:00 | VC Built  Power on Host1 and select ESXi 5 build |
| 2:15 | Host1 built  Power on Host2 and select ESXi 5 build |
| 2:30 | Host2 built  Run AddHosts script |
| 3:00 | Cluster Built  Datastores built and first VM installing operating system |

# Shutting the lab down

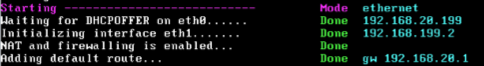
Since the lab takes up so much of the resources on a PC you will probably want to shut it down when you’re not actively working on it. The **Shutdown Lab Servers** Option from the **AutoLab Script Menu** on the VC desktop runs a PowerShell script that will quickly but cleanly power down everything except the NAS and Router VMs, which can be powered off using VMware Workstation power control. The first time the shutdown script is run after a VC rebuild you will need to confirm storing the SSH keys for the NAS, router and vCloud VMs (if these are running).



# Accessing the built lab

While the VMware Workstation console is perfectly functional you may wish to use guest OS native tools, like RDP. If your PC has an IP address on the lab subnet then you may use these directly. Alternatively the Router VM provides access to the lab from its external IP address.

Some access to the lab is published through the router. In the example below the external IP address of the router is 192.168.20.199, I use a DHCP reservation and a fixed MAC address on my router VM to keep this IP address consistent in my lab.



Windows sharing from the NAS VM is published on the normal ports so you can access the Build share through the routers external IP address.



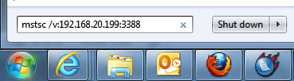
The management web interface of the NAS VM is available on the standard HTTP port 80 of the router’s external IP address.



The VC VM is available via RDP on the external IP address of the router using the default RDP port of 3389.



The DC VM is available via RDP on the external IP address of the router using port 3388.



The router also provides access to SSH access to the ESXi servers and VMA, this allows PuTTY or another SSH client to connect to these VMs from your external network.

|  |  |
| --- | --- |
| **Server** | **External port** |
| Host1 | 122 |
| Host2 | 222 |
| VMA | 22 |

The management interface of the Router is the main thing that cannot be accessed from the external network; this is at <http://192.168.199.2:82> on the internal network.

# As Built Documentation

The following information outlines the built environment when all of the automation is executed correctly. This is also useful if you are manually building to the same standard or extending the AutoLab to cover additional products.

## IP Addressing

|  |  |  |
| --- | --- | --- |
| Main network, no VLAN | | |
| Subnet | 192.168.199.0 | |
| Subnet Mask | 255.255.255.0 | |
| Gateway | 192.168.199.2 | |
| DHCP server | 192.168.199.4 | |
| DNS Zone | lab.local | |
| DHCP Scope | 192.168.199.100 – 192.168.199.199 | |
| Purpose | DNS name | IP Address |
| Host PC |  | 192.168.199.1 |
| Router (gateway) | gw | 192.168.199.2 |
| Domain Controller | dc | 192.168.199.4 |
| vCenter server | vc | 192.168.199.5 |
| vCenter Management Appliance | vma | 192.168.199.6 |
| FreeNAS | nas | 192.168.199.7 |
| Host1 Management | host1 | 192.168.199.11 |
| Host2 Management | host2 | 192.168.199.12 |
| View Connection Server 1 | cs1 | 192.168.199.33 |
| View Connection Server 2 | cs2 | 192.168.199.34 |
| View Security Server | ss | 192.168.199.35 |
| Veeam ONE server | v1 | 192.168.199.36 |
| Veeam Backup &Replication Server | vbr | 192.168.199.37 |
| vCloud Director | vcd | 192.168.199.38 |
| vCloud Proxy | vcd-proxy | 192.168.199.39 |
| vShield Manager | vshield | 192.168.199.40 |

|  |  |
| --- | --- |
| Internal Network | |
| VLAN ID | 16 |
| Subnet | 172.16.199.0 |
| Subnet Mask | 255.255.255.0 |
| Host1 VMotion | 172.16.199.11 |
| Host2 VMotion | 172.16.199.12 |
| Host1 FT Logging | 172.16.199.21 |
| Host2 FT Logging | 172.16.199.22 |
| Host2 as ESX 4.1  Service Console for HA Heartbeat | 172.16.199.42 |

|  |  |
| --- | --- |
| IP Storage Network | |
| VLAN ID | 17 |
| Subnet | 172.17.199.0 |
| Subnet Mask | 255.255.255.0 |
| Host1 IPStore 1 | 172.17.199.11 |
| Host2 IPStore 1 | 172.17.199.12 |
| Host1 IPStore 2 | 172.17.199.21 |
| Host2 IPStore 2 | 172.17.199.22 |

## UserIDs

The follow user accounts are built into the standard lab build

|  |  |  |  |
| --- | --- | --- | --- |
| System | Purpose | UserID | Password |
| Lab.local domain |  | Administrator | VMware1! |
| Lab.local domain | vSphere Administration | vi-admin | VMware1! |
| Lab.local domain | Veeam services | SVC\_Veeam | VMware1! |
| Lab.local domain | vCloud access to vCenter | SVC\_vCD | VMware1! |
| Lab.local domain | SRM services (not yet used) | SVC\_SRM | VMware1! |
| Router | administration | admin | VMware1! |
| NAS |  | admin | VMware1! |
| Windows VMs |  | Administrator | VMware1! |
| VMA |  | vi-admin | VMw@re1! |
| vCloud |  | Root | VMware1! |

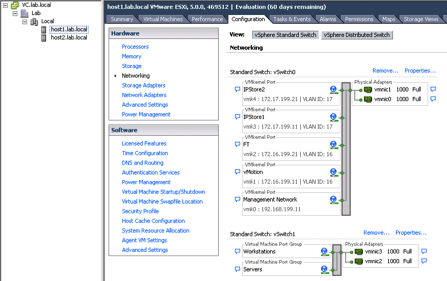
## Databases

The following databases are created in the DC build, all userIDs are SQL users

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Owner | Password | Purpose |
| vCenter | vpx | VMware1! | vCenter |
| VUM | vpx | VMware1! | Update Manager |
| ViewEvents | VMview | VMware1! | View Events |
| ViewComposer | VMview | VMware1! | View Composer |
| SRM | VMSRM | VMware1! | Site Recovery Manager |
| SRMRep | VMSRM | VMware1! | vSphere Replication |
| RSA | RSA\_USER | VMware1! | vCenter 5.1 SSO |

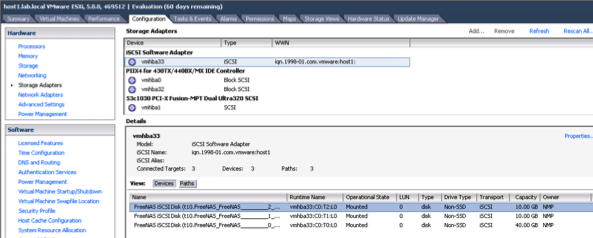
## Host Network

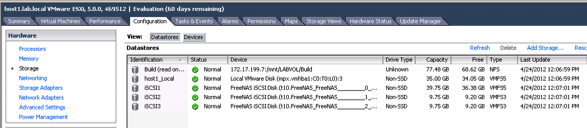
vMotion and Management Network VMkernel ports have symmetric NIC teaming on VMnic0 and VMnic1; each active on one and standby on the other. Both ports are enabled for Management traffic to provide redundant HA heart beating. There is no routing between the VLANs/subnets; the default gateway for the ESXi servers is on the 192.168.199.0 subnet.



## Host and Shared Storage

ISCSI port binding is not implemented, nor is the NIC teaming configuration required for port binding. ISCSI traffic uses the VMkernel ports on VLAN17.





# Rebuild Process

### Open Source Infrastructure

The NAS VM should not require rebuilding, nor should the router. If either of these machines requires rebuild then you should probably redeploy the entire lab kit.

### Windows Servers

The DC VM should only require rebuilding to renew licensing. If you are using a 180 day trial license then it will require rebuilding every 180 days. Rebuilding the DC VM simply requires a reboot that is interrupted at the “Press and key to boot from the CD” stage. Rebuilding the DC VM will require the rest of the lab to be rebuilt.

The VC VM will be rebuilt more frequently, to change vCenter versions or simply to refresh the lab setup. Rebuilding the VC VM simply requires a reboot that is interrupted at the “Press and key to boot from the CD” stage. The vCenter 5.1 installer does not properly overwrite an existing SSO database; consequently the DC upgrade script must be run to reset the database. There is a desktop shortcut called **Upgrade** on the DC for this script; the shortcut must be “Run as Administrator”.

After VC is rebuilt you can re-run the **Add ESXi Hosts to vCenter and configure cluster** script from the AutoLab Script Menu to re-add the ESXi servers to the new vCenter. The ESXi servers shouldn’t require rebuild and all datastores and VMs should be added to the inventory.

### ESXi Hosts

The ESX servers can be rebuilt by choosing a build option from the PXE boot menu rather than letting the timer expire.

If you are rebuilding the ESXi hosts and not vCenter then delete the cluster from the inventory before running the **Add ESXi Hosts to vCenter and configure cluster** script from the AutoLab Script Menu. The script will skip creating datastores and the WinTemplate VM if they already exist, WinTemplate will be added to the inventory as a VM or Template if it’s found in the location where the script creates it.

### Concurrent Rebuilds

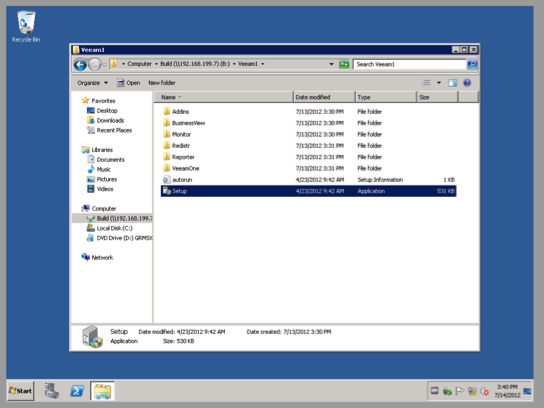
If your Lab platform has sufficient resources it is possible to concurrently rebuild the ESXi VMs, potentially while rebuilding the VC VM. The DC and NAS VMs must be built and operational for the other VMs to rebuild. Usually disk and CPU are the limiting resource for rebuilds. An SSD to store the VMs on and a Quad Core CPU will help here.

# Veeam ONE Installation

Like most management tools Veeam ONE uses a client server model. We will use a dedicated VM named V1 as the server. The environment is more interesting if there are a few nested VMs running and if there is some load in the VMs.

### Server component install on V1

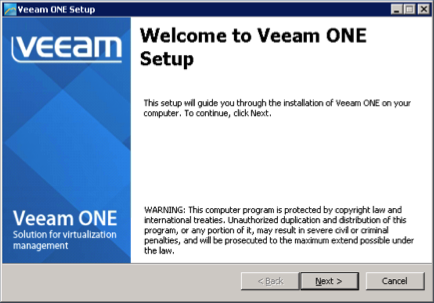
1. Build the V1 VM, this follows the usual boot from CDROM with floppy attached method used for all the Lab Windows VMs. You should have the basic AutoLab setup built with DC, VC and two ESXi servers. On an a Workstation environment with 8GB of RAM you will need to shut down one of the ESXi servers to free some RAM
2. Logon to V1 as the Veeam service account user lab\svc\_Veeam
3. On Build share locate **B:\Veeam1\setup.exe**&**Run As Administrator**

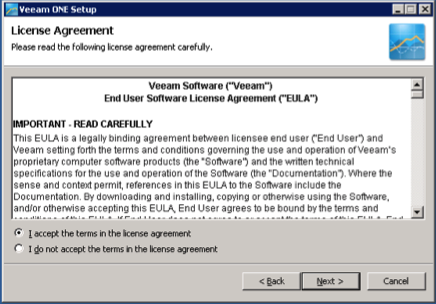


1. Select to install **Veeam ONE Server**

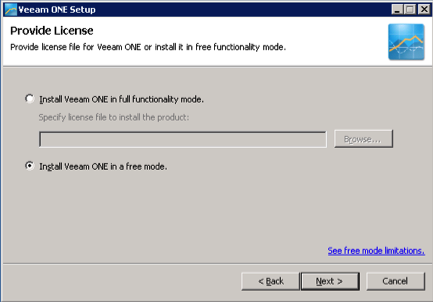


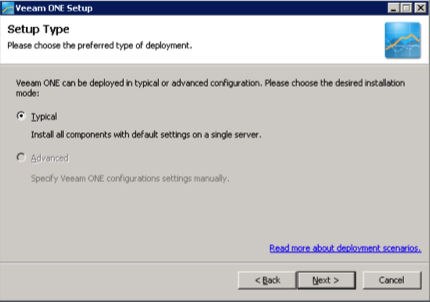
1. Click through the install wizard as usual



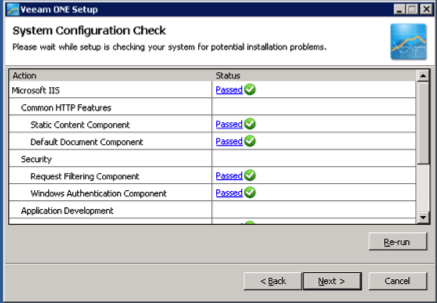


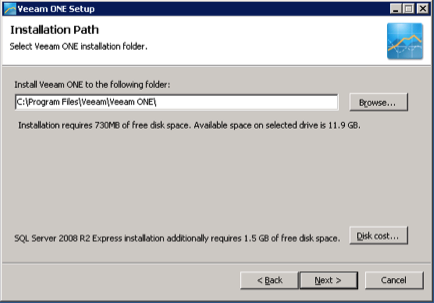
1. Since we're evaluating we will use the free edition, Select **Install Veeam ONE in a free mode** and click **Next**



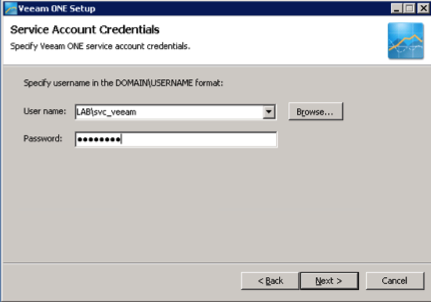


1. The Lab build has installed all of the required Windows components for you

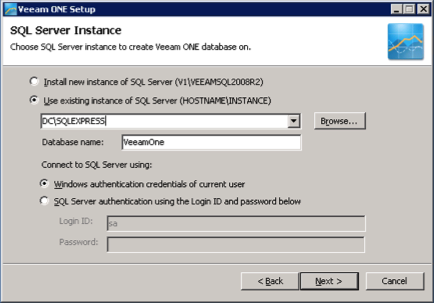


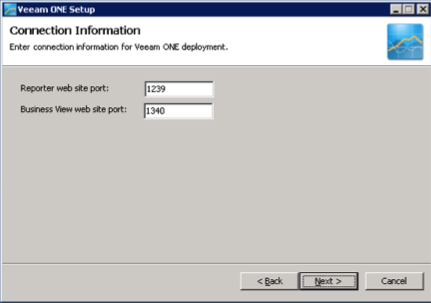


1. Enter the usual lab password VMware1! for the service account

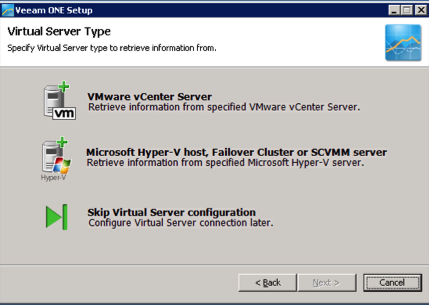


1. To minimise the RAM footprint we will use the SQL instance on the DC, the service account has rights to create the database so simply change to **Use existing instance of SQL Server** and enter the instance name of the SQL server on DC, **DC\SQLEXPRESS**

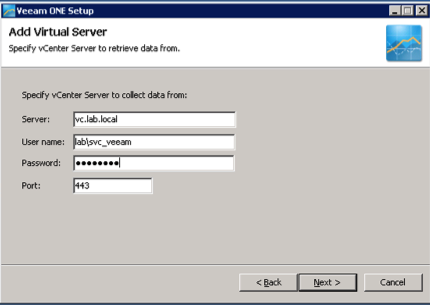


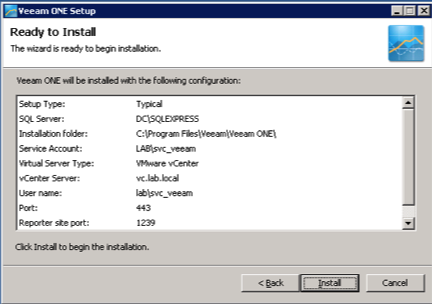


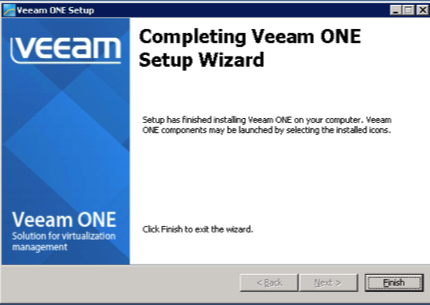
1. Click on VMware vCenter Server to add our vCenter



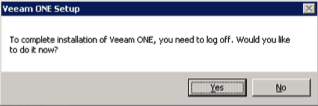
1. Enter the lab vCenter **vc.lab.local** and the Veeam service account credentials, username **lab\svc\_veeam** and password VMware1!







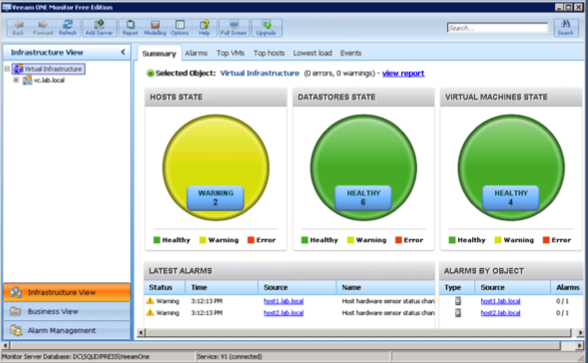
1. Once the install completes you will be prompted to log off, click **Yes**



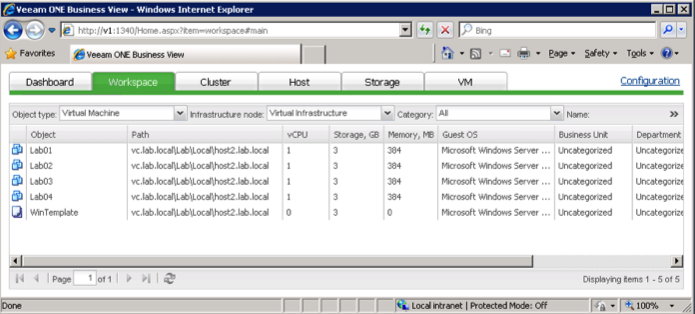
1. When you log back on you will find three new desktop shortcuts for the Veeam components



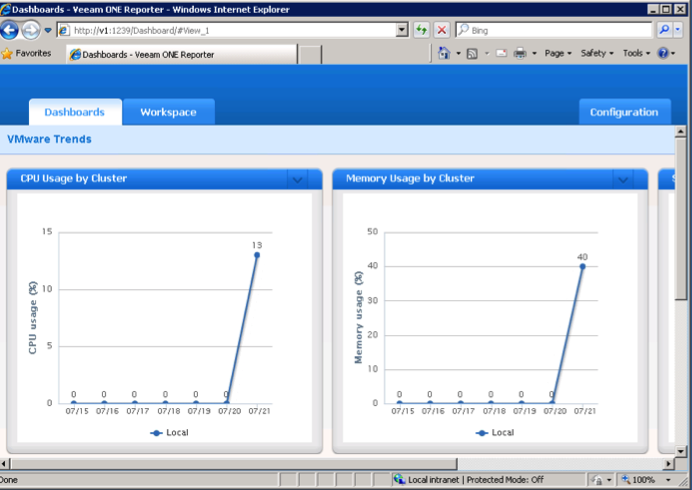
1. Use each shortcut to launch the components and make sure they operate.
2. Veeam ONE Monitor



1. **Veeam ONE Business View**, this shows some data on the **Workspace** tab



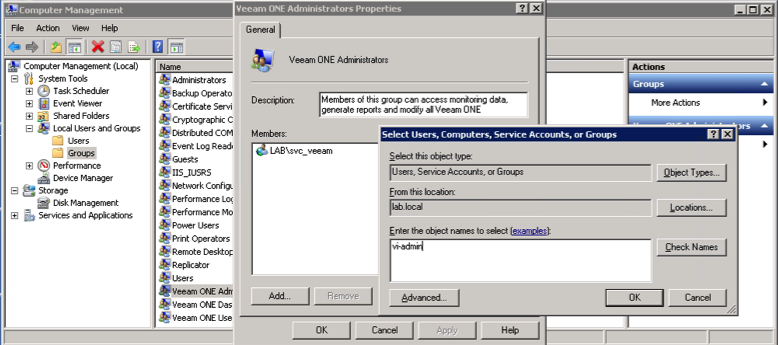
1. **Veeam One Reporter**, here the **VMware Trends** dashboard shows data immediately



1. Next setup to allow the VI-Admin user access to the Veeam products. Form the Start Menu under **Administrative Tools** select **Computer Management**
2. Under **Local Users and Groups** select the **Groups** folder



1. Double click the **Veeam ONE Administrators** group, click the **Add…** button, enter VI-Admin and click **OK** a couple of times, then close **Computer Management**

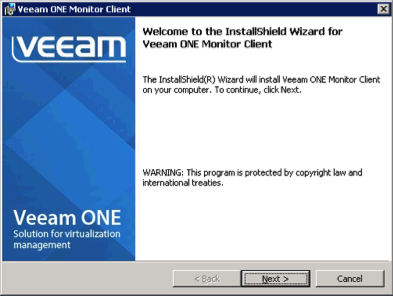


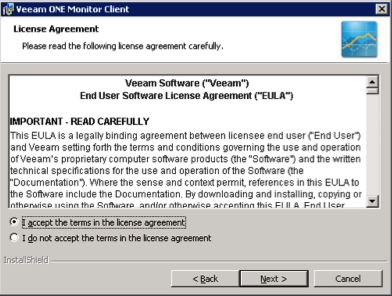
### Client component install on VC

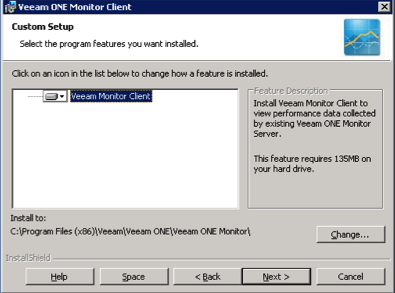
1. Logon to VC as VI-Admin
2. Again locate and **Run as Administrator** the **B:\Veeam1\setup.exe** from the Build share
3. This time choose **Veeam ONE Monitor Client**

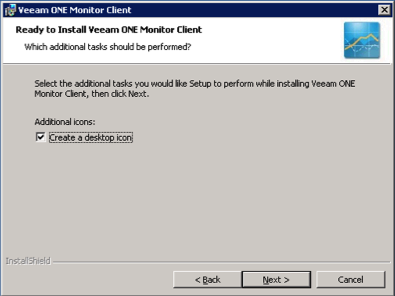


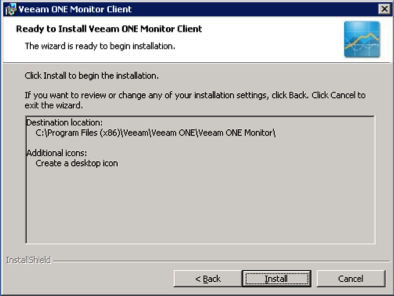
1. Click through the install wizard as usual

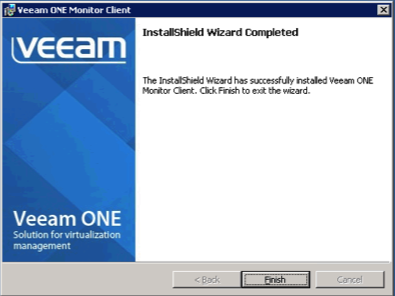








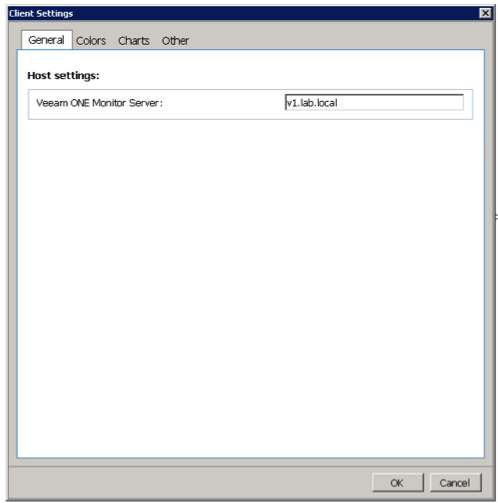




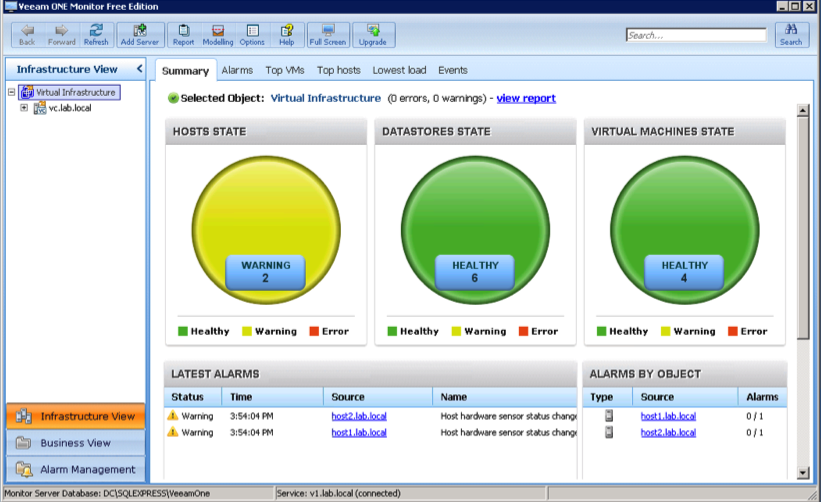
After the install completes you will have a desktop shortcut for **Veeam ONE Monitor**, the other two components are web services.



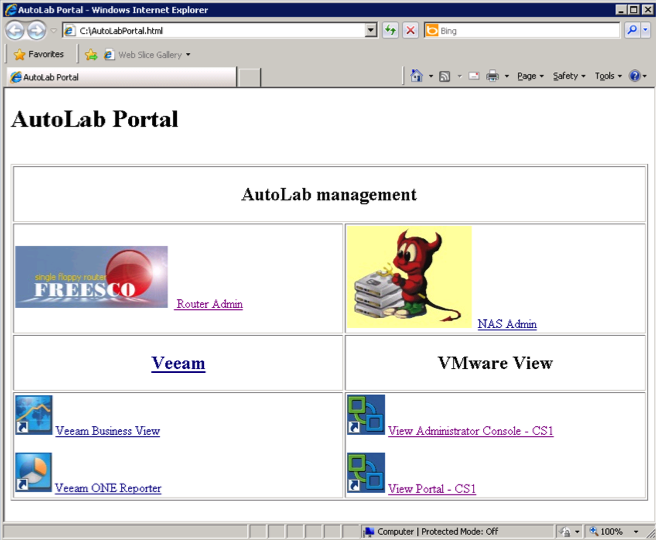
1. On the first run you will need to tell the monitor client where to find the server, enter **v1.lab.local** and click OK



1. Once the client connects it should show you exactly the same environment as you saw in the client on V1



1. Business View and Reporter are web applications; the AutoLab portal page has links to both. Simply launch Internet Explorer from the desktop shortcut then use the links to confirm access.



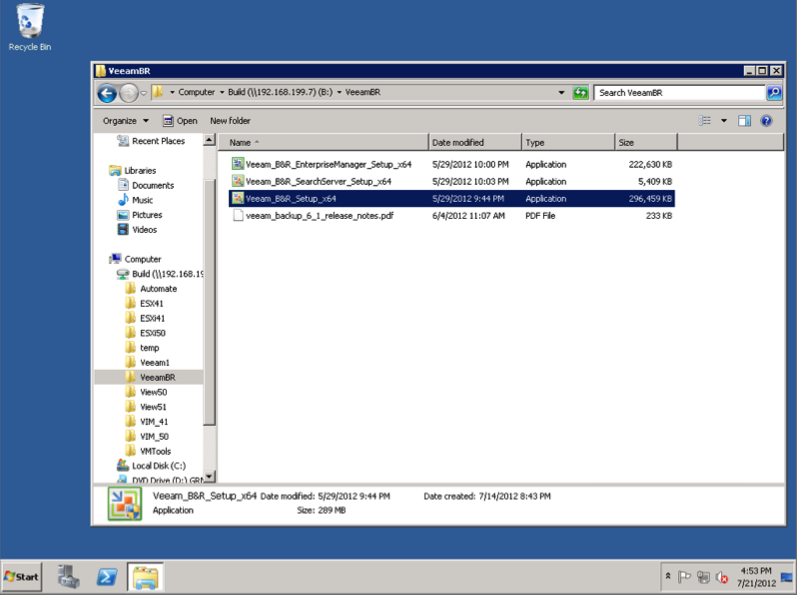
1. You can login to the web applications using the VI-Admin username and VMware1! Password

Now that you have the server and clients installed it’s time to start learning about Veeam ONE, there are lots of resources on the Veeam web site <http://www.veeam.com/virtual-server-management-one-free/resources.html>

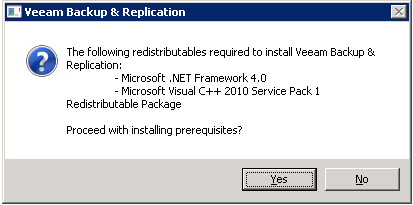
# Veeam Backup & Replication installation

### Server component install on V1

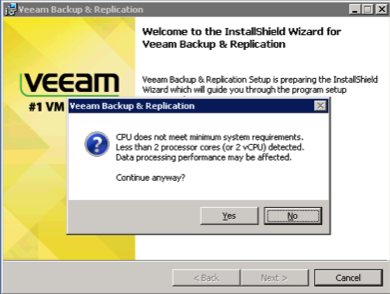
1. Build the VBR VM, this follows the usual boot from CDROM with floppy attached method used for all the Lab Windows VMs. You should have the basic AutoLab setup built with DC, VC and two ESXi servers. On a Workstation environment with 8GB of RAM you will need to shut down one of the ESXi servers to free some RAM.
2. Logon to VBR as the Veeam service account user lab\svc\_veeam
3. On the build share locate **B:\VeeamBR\Veeam\_B&R\_Setup\_x64.exe** and **Run as Administrator**

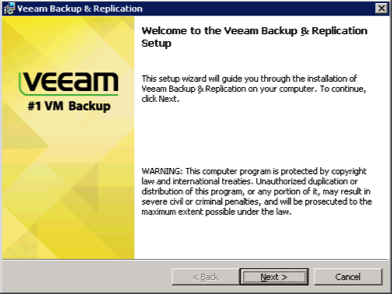


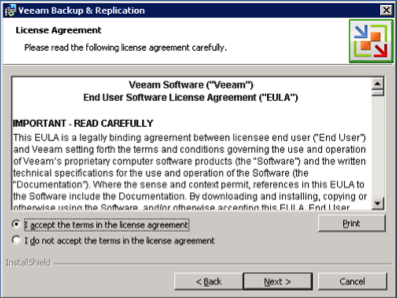
1. If you are installing Veeam Backup & Replication 6.5, you will see a warning about some pre-requisites, click **Yes** and the installer will install these for you.

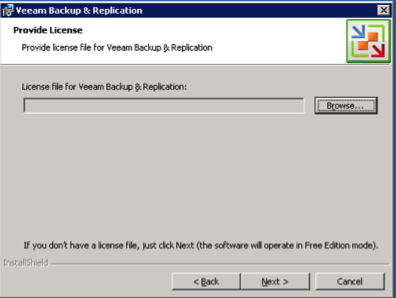


1. Accept the warning about vCPU count and proceed on through the wizard

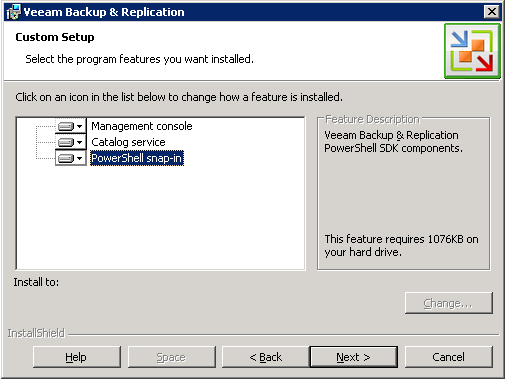




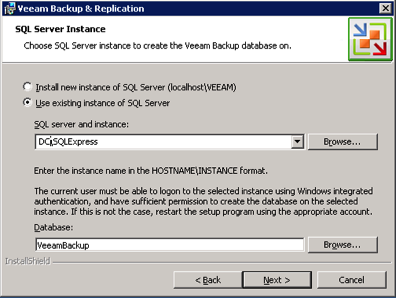




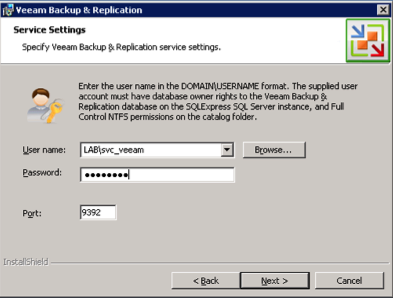
1. Since I’m a fan of PowerShell I included it’s snapin

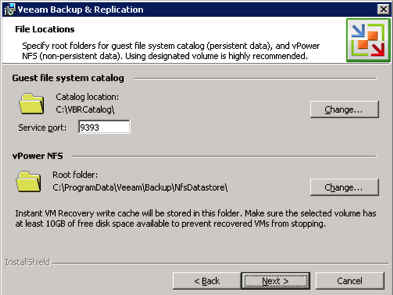


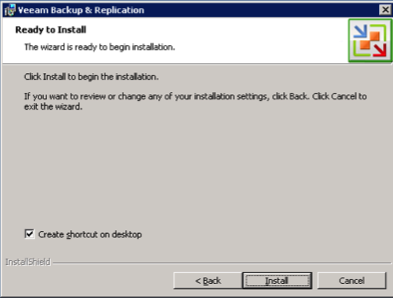
1. Select **Use existing instance of SQL Server** and enter the SQL server instance on D, **DC\SQLEXPRESS**, leave the default database name

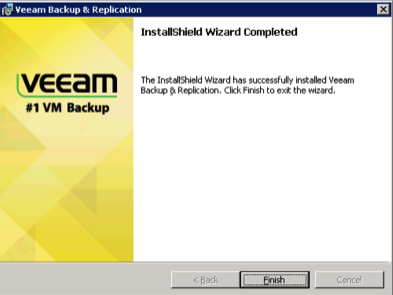


1. Enter the usual password, VMware1!, for the svc\_veeam service account









1. Once the installer has completed you will find a shortcut on the desktop



1. Now that the software is installed head on over to the Veeam web site to learn how to use it <http://www.veeam.com/vmware-esx-backup/resources.html>

# VMware View Installation

Three shell VMs are provided with the lab kit, these provide the main components of the View environment. To run these VMs under Workstation with 8GB of RAM it is easiest to power down one of the ESXi hosts.

### Automated Installation

Edit the file **B:\Automate\Automate.ini** on the build share; the following lines are relevant to the View build:

ViewInstall=None

ViewInstallOptions=50, 51, None

BuildViewVM=ask

BuildViewVMOptions=True, False, Ask

ViewVMProductKey=XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

Edit the ViewInstall line for the version of View You wish to install, make sure you have placed the files in the folder on the build share. Edit the BuildViewVM line to choose whether to build the first Windows XP VM, make sure the Windows install ISO is in the root of the build share and named WinXP.iso. Edit the ViewVMProductKey line to reflect the product key for the version of Windows XP in your WinXP.iso file. The vCenter build script will create fully unattended install ISOs for both Windows 2003 and Windows XP

The View Composer software will be installed on the VC when it is rebuilt; alternatively you can install yourself using the manual install information. If you chose to have the View VM built it will be created as part of the AddHosts script on the VC, option 3 in the **AutoLab Script Menu**.

Connection server software will be installed on both CS1 and CS2 VMs; these will be replicas for View purposes. You probably only want CS1 but replicas are useful for testing load balancing connection servers or testing Tags. With View 5.0 the vCenter server will be added along with its View composer function. View 5.1 presents some issues with certificates so the VC isn’t automatically added. View Composer domains do not appear to be able to be added automatically.

Before building the Security server VM (SS) you must set a pairing password on CS1 using the View administration page. Set the usual password of VMware1! And make sure you build SS before the password expires. The Events Database will also require configuration using the information below.

### Manual Installation

The default **B:\Automate\Automate.ini** does not automate the View install. To manually install you will want the following information:

|  |  |
| --- | --- |
| Location of install files | B:\View50 or  B:\View51 |
| First Connection Server | CS1 |
| Second Connection server | CS2 |
| Security Server | SS |
| View Composer server | VC |
| View Composer Database Server | DC\SQLEXPRESS |
| View Composer Database | ViewComposer |
| View Composer Database User | VMView |
| View Composer Database Password | VMware1! |
| View Events Database Server | DC\SQLEXPRESS |
| View Events Database Type | Microsoft SQL Server |
| View Events Database Port | 1433 |
| View Events Database | ViewEvents |
| View Events Database User | VMView |
| View Events Database Password | VMware1! |
| Table Prefix | VE\_ |

# VMware vCloud Director Installation

### Automated build

[Damian Karlson’s](http://damiankarlson.com/) ([@sixfootdad](http://twitter.com/sixfootdad)) excellent work has brought vCloud Director to the AutoLab.

|  |  |
| --- | --- |
| **Required Software** | **Source** |
| Oracle Database Express Edition 11g Release 2 for Linux x64  oracle-xe-11.2.0-1.0.x86\_64.rpm | <http://www.oracle.com/technetwork/products/express-edition/downloads/index.html>  You may need to unzip the zip file to get the rpm file. |
| vmware-vcloud-director-1.5.1-622844.bin  VMware-vShield-Manager-5.0.1-638924.ova | If you have a license: <http://www.vmware.com/go/download/vcloud-director>  If you want to use the evaluation version:  <http://www.vmware.com/go/try-vcloud-director> |
| CentOS-6.3-x86\_64-bin-DVD1.iso  CentOS-6.3-x86\_64-bin-DVD2.iso | <http://www.centos.org/modules/tinycontent/index.php?id=30>  Choose an HTTP mirror closest to you, and then navigate to /6.3/isos/x86\_64. The *CentOS-6.3-x86\_64-bin-DVD1to2.torrent* is recommended as it will download quickly, although a torrent client such as uTorrent is required. Cento 6.2 has been tested and works as well. It is also recommended that you perform a hash check of the downloaded isos to verify their integrity. |

In order to be able to create a provider virtual datacenter within VMware vCloud Director, the vSphere hosts will need to have their memory increased from 2048MB to 3372. This leaves just about 1GB of memory as available resources to vCD and also accounts for the memory used by the vShield VM. If you have an 8GB lab machine you will need to change the VMware Workstation preference to “Allow most virtual machine memory to be swapped”. You can find this setting under **Edit**>**Preferences…**>**Memory**. If your Lab machine has an SSD then this should not cause a performance problem.

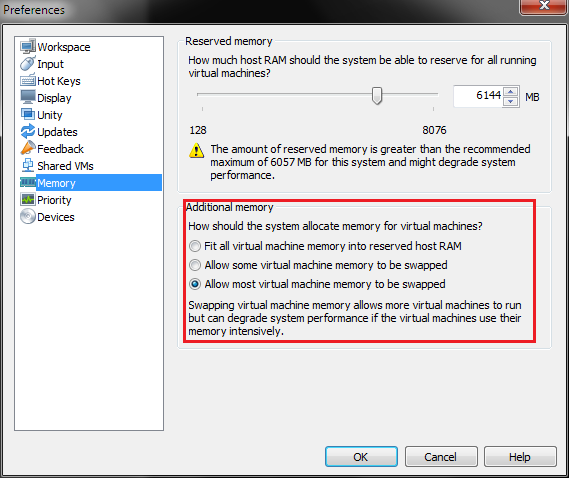
Since many vCloud labs don’t actually require the use of VMs with operating systems installed within the vCloud environment, our recommendation is to create small virtual machines that have 4MB of RAM and 4MB disks. This will allow you to work with catalogs, creating vApp templates, instantiating vApp templates, and performing power operations.

### VMware vCloud Director Installation instructions:

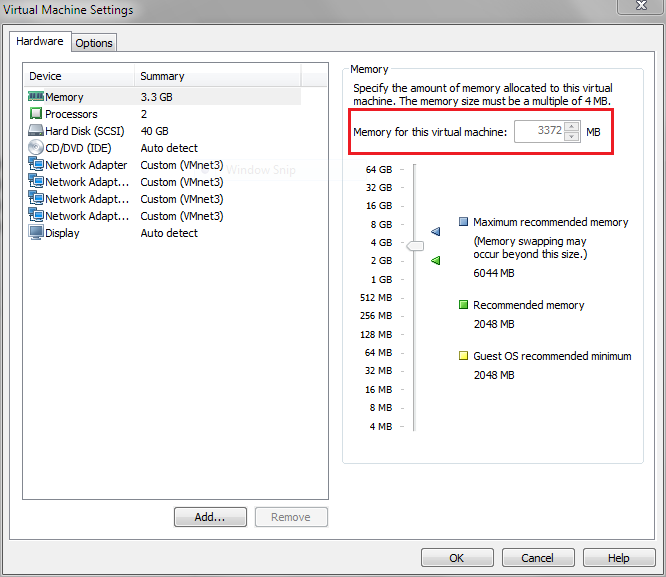
After procuring the necessary software, copy the following files to the Build folder on the NAS VM, typically [//192.168.199.7/Build](file://localhost//192.168.199.7/Build)/vCD\_15:

* oracle-xe-11.2.0-1.0.x86\_64.rpm from <http://www.oracle.com/technetwork/products/express-edition/downloads/index.html>
* vmware-vcloud-director-1.5.1-622844.bin
* VMware-vShield-Manager-5.0.1-638924.ova

Verify or change the Workstation memory preferences to “Allow most virtual machine memory to be swapped”. If you’re running Workstation on Windows 7, you may be required to run Workstation as an administrator. If you are lucky enough to have 16GB of RAM in your lab host you should be able to leave all VM memory in RAM.



Power down Host1 and Host2 and increase each host’s memory to 3372MB. Power them back on and allow them to reconnect to vCenter.



Add the vCloud VM from the AutoLab distribution folder. Mount the *CentOS-6.3-x86\_64-bin-DVD1.iso* in the vCloud VM’s CD-ROM drive. Ensure that the drive is set to “Connect at power on”.



Power on the vCloud VM and choose the “CentOS for vCloud” option from the PXE boot menu. An automated installation of CentOS, Oracle Express, and VMware vCloud Director will be performed. Installation status will be available on the startup screen during first boot.

You can verify that vCloud Director installed successfully by logging into the vCloud VM with the username **root** and the password **VMware1!** and executing **service vmware-vcd status**. If the vmware-vcd-watchdog and vmware-vcd-cell services are running, then open a web browser and go to <https://vcd.lab.local> within the AutoLab environment or <https://192.168.199.38/> from the Workstation host. You should be presented with the VMware vCloud Director Setup screen.

Note: If you try to use the DC or VC VMs to reach the vCloud Director website, you will need to connect to the Internet and install Adobe Flash on the VM in question. The DC VM is already configured with the gateway address of 192.168.199.2. The VC VM will need to have the “Add route to the Internet so VUM can download updates” script run from the AutoLab Script Menu located on VC’s Desktop. The Router VM has a virtual NIC bridged to the Workstation host’s network, and the Workstation host will need to have Internet access for the Flash download to work.

### VMware vShield installation instructions:

Connect to the VC VM and run the “Install vShield for vCloud” from the AutoLab Script Menu located on VC’s Desktop.

### VMware vCloud Director setup instructions:

Connect to <https://vcd.lab.local> from within the AutoLab environment, or <https://192.168.199.38> from the Workstation host. Follow the vCloud Director Setup wizard to complete the setup. Login to vCloud Director and “Attach a vCenter” from the Quick Start menu.

Name this vCenter:

* Host name or IP address: vc.lab.local
* Port number: 443
* User name: SVC\_vCD
* Password: VMware1!
* vCenter name: vc1

Connect to vShield Manager:

* Host name or IP address: vshield.lab.local
* User name: admin
* Password: default

Click Finish to complete attaching a vCenter. Complete the rest of the Quick Start menu as necessary.

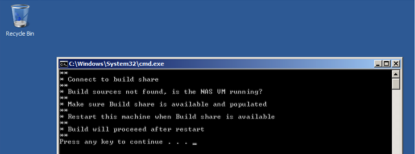
# Troubleshooting

A few things can go wrong along the way here are some we’ve seen

### DC Build stalls

If the NAS is not accessible from the DC then the build will stall after Windows and AD are installed. The VM will autologon but not start the second phase build, there will be no PowerShell shortcut on the desktop. Make sure that the NAS VM is running and accessible from the DC. Make sure the Build share is fully populated and then rebuilt the DC.

There may be a message on screen about populating the build share, make sure you have put the required files in the right places.



### Validate complains about not being run as administrator

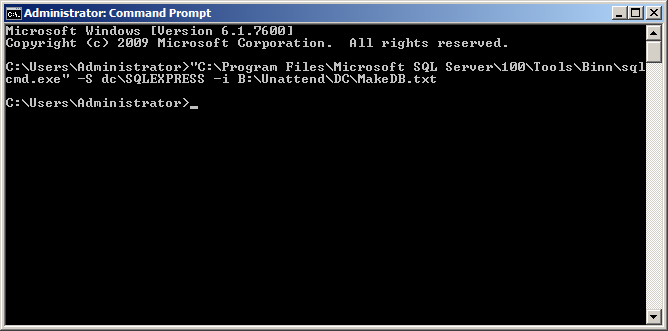
The validate script must be **Run as Administrator**, right click the shortcut and select this from the popup menu.



### DC fails Validate, databases missing

If the validate script reports that the databases are missing then the VC build will fail. Search in c:\Buildlog.txt for the status “\* Create vCenter Database” and look for errors below. If you see a “Shared memory provider: Timeout error [258]” then the error was one of timing.

To create the databases locate the file b:\Unattend\DC\Phase2.cmd and edit it with notepad. Locate the text “\* Create vCenter Database” and look a little below for the SQLCMD command line. Paste the command line into an elevated command prompt.



### VC Build Fails to create vCenter Repository

This is the result of the DC failing to create the databases, see the above error. Once the databases exist then completely rebuild the VC.

### Error installing PowerCLI on VC

Sometimes a “runtime error 1618” shows on the VC VM during install; wait a few minutes and click Retry.

### Host1 build fails –Resolved in V1.1

If the VC build script cannot identify the version of ESXi in your **B:\ESXi50** folder then it won’t setup the PXE environment correctly. On the Build share in **B:\Automate\DC** there are folders of ESXi5\_0\_RTM and ESXi5\_0\_U1 copy the contents of the appropriate folder into **C:\TFTP-Root\ESXi50** on your DC and retry the build.



### AddHosts.ps1 exits

The Add Hosts script will exit immediately if either of the ESXi servers doesn’t respond to a ping and a little later if either of the ESXi servers can’t be added to the vCenter inventory. Make sure both ESXi servers are built and available on the lab network on the correct IP addresses.

### vSphere 4.1 Build issues

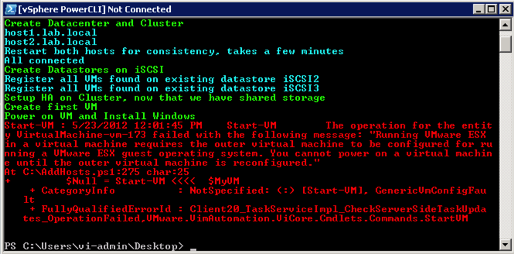
#### VUM 4.1 fails to install

The VUM 4.1 install appears not to respect the directive to overwrite its database and will fail to install if VUM 5.0 has previously created its database tables. Use the **Upgrade** shortcut on the desktop of the DC, this will recreate empty databases, then rebuild the VC.



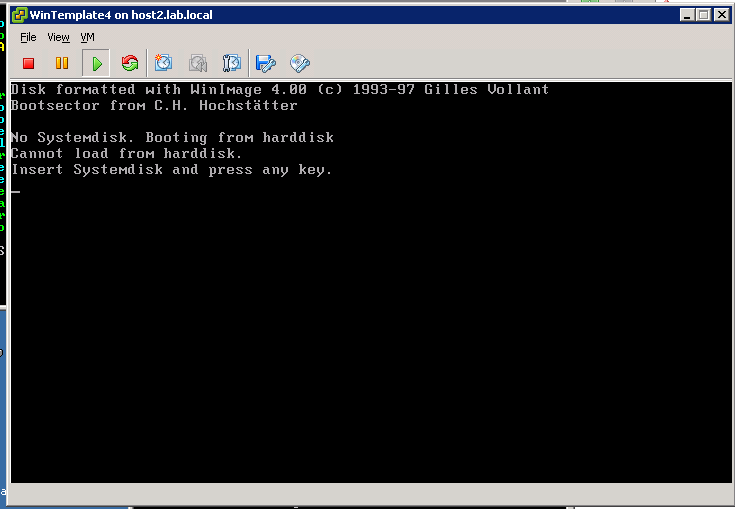
#### ESX 4.1 Cannot power on VM

The outer VM needs to be configured with a Guest OS of ESX 4.x, rather than ESX 4.x. While you’re there Host1 needs 2304MB of RAM allocated for HA to configure correctly.



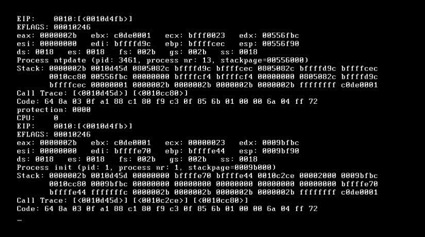
### First VM doesn’t install windows on vSphere 4 builds –Resolved in V1.1

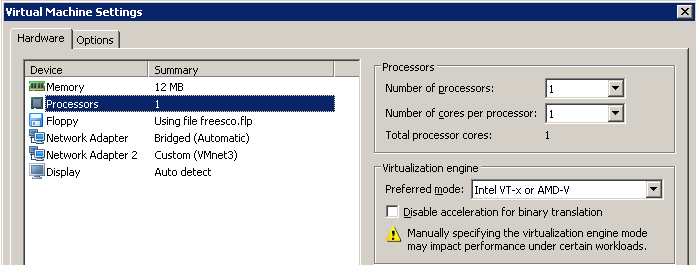
vCenter 4.1 does not respect VM boot order from PowerCLI. Use the BIOS settings in the VM to boot Hard Disk and CDROM before floppy.



### Router VM crashes on start-up –Resolved in V1.1

On some CPU types VMware Workstation or ESXi may choose to use Binary Translation for the CPU of the Router VM; this will cause the router to crash with the status message below. To force the use of hardware CPU virtualization edit the Router CM’s settings, select the CPU and change the "Virtualization Engine" "Preferred Mode" to "Intel VT-x or AMD-V"





# AutoLab Changes and Futures

### Changes

#### V1.1

* Addition of vSphere 5.1
* Removal of requirement for specific Windows 2008 installer ISO
* Various support and usability improvements

#### V1.0

* Addition of Veeam products
* Addition of VMware View 5.0 and 5.1
* Addition of vCloud Director V1.5
* Removal of Windows 2008R2 RTM support, Window VMs will use SP1 media only

#### V0.8

* vSphere 5.0 Update 1 support
* Windows Server 2008R2 SP1 support for VC & DC
* Removed requirement to download SQL client and extract deploy.cab into Build share
* Cosmetic and reliability improvements in scripts
* Support for deployment onto standalone ESXi server
* Removed suggestion that XP worked in nested VM

#### V0.5

* Initial release
* vSphere 5.0 RTM only support
* Windows 2008R2 RTM only support

### Futures

Insert your favourite disclaimer here, only things that get added to AutoLab will be added. If there’s something you really want added then build it yourself and send it to me so I can include it.

The more features the AutoLab gets the more we want it to do, here’s what’s on the list now:

* VMware Site Recovery Manager infrastructure
* More Automation
  + Time zone into nested VMs and Guest OS Customization specification
  + Option to automatically run AddHosts on vCenter build completion.
  + View configuration
    - Composer domains
    - Security server pairing password
    - View Events DB setup
    - View 5.1 configuration
* Trusted certificates to enable View 5.1 automated setup

If you solve any of these problems then let us know via [feedback@labguide.com](mailto:feedback@labguide.com) so we don’t need to duplicate your effort.