AutoLab 3.0 vSphere Deployment Guide

AutoLab

vSphere AutoLab

This AutoLab kit is designed to produce a nested vSphere 6.7 or earlier (back to 4.1) lab environments 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 tallow evaluation and testing of additional software like VMware View.

The project lives at http://www.labguides.com and updates will occur there.

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

This is version 3.0. 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. One place to look for help are the forums on Reddit, https://www.reddit.com/r/AutoLab/

Please let me know how you find the AutoLab and what needs improving or adding. You can email Nick and me through feedback@labguides.com.

How to use this guide

The AutoLab has many parts and many options, the flexibility come at the cost of being hard to explain. There are a few essential steps:

1. Get and read this guide, congratulations you're in the right place
2. Choose your lab outer platform
3. Build and configure your outer platform
4. Download the right AutoLab package
5. Download the licensed software
6. Build the Lab
7. Play

These phases are outlined in the following sections, make sure to read carefully as some instructions are really important.

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 A previous project with Nick led to this project. His testing and QA on the lab build through its various stages were invaluable. Also for setting up the labguides site.

Grant Orchard for the vSphere 5.1 build.

Damian Karlson for adding vCloud Director to the AutoLab, documenting Fusion 5 setup, bug fixes, and adding features.

James Bowling for documenting the full Fusion 4 setup for the AutoLab.

Ariel Antigua for hanging out on the AutoLab support forum and helping out when people have troubles.

Alex Lopez for making the Windows Server 2012 build better.

FreeNAS The storage platform for the lab. Having an open source storage option makes the lab possible.

pfSene is the router that links the private AutoLab network to your normal network. Another open source project that does great things and asks little in return.

JouninTFTPd More free software; this time the file transfer tool of the PXE environment used to build the ESXi servers.

VMware 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 for providing the operating system we most often need to virtualize.

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

Cody Bunch, David Manconi, Damian Karlson, Grant Orchard, Josh Atwell, Tim Gleed, Michael Webster, Mark Dunnett, Shane Williford, Chas Setchell and some more who I've unintentionally forgotten to list.

Table of Contents

Contents

AutoLab 3.0 vSphere Deployment Guide 1
vSphere AutoLab 2
How to use this guide 3
Acknowledgements 4
Table of Contents 5
2. Choose Your Lab Outer Platform 6
3. Build and configure your outer platform 7
4. Download the right AutoLab package 17
5. Download the licensed software 27
6. Build the Lab 28
Lab Build Time 40
Shutting the lab down 41
Accessing the built lab 42
As Built Documentation 44
Rebuild Process 47
Tuning AutoLab for RAM 48
VMware View Installation 49
Running Multiple AutoLabs 51
Troubleshooting 52
AutoLab Version Changes 55
AutoLab Futures 57
VMware vCloud Director Installation 58
Veeam ONE Installation 64
Veeam Backup & Replication installation 75

2. Choose Your Lab Outer Platform

Required Hardware

The core lab can run on a single PC, a dual core 64bit CPU and a minimum of 24GB RAM is required, along with around 200GB of free disk space. The core lab does not include vCloud, View or Veeam VMs; these will need more RAM and disk.

The main things you will want to upgrade are RAM and moving to a large SSD. Both these will make the lab build faster and more responsive to use. More cores and higher CPU clock speed will not hurt but they aren't the main factor.

Each successive release of vSphere requires more RAM, the first version of AutoLab only required 8GB of RAM to run a two node cluster. A three node vSphere 6.0 cluster with VSAN enabled will need a host with 32 GB of RAM. Take a look at the RAM tuning section to see what can be done to fit your required vSphere build into less RAM.

Lab Virtualization Platform

VMware Workstation

I use VMware Workstation to develop AutoLab, so it the platform that gets the most testing. My current build machine has 32GB of RAM and a 480 GB SSD. It takes under two hours to build a core lab on this machine, which isn't used for anything other than developing AutoLab.

VMware Fusion

My MacBook Pro has 16GB of RAM, which is barely sufficient to run a core lab provided that I shut down absolutely everything else and am very patient while it builds.

VMware Player

This is the free option, no licensing cost for Player. I sometimes use Player to demonstrate AutoLab in classrooms when I'm teaching vSphere courses.

VMware ESXi

The better memory management of ESXi is awesome; you need a third less RAM on ESXi than Workstation. I tried to make it easy to have multiple AutoLab instances on a single ESXi server. This way a single ESXi server can provide labs for a whole team.

Ravello

Ravello Systems have a hypervisor that runs on top of a public cloud instance. If you don't have access to physical hardware and don't need a permanent lab their platform can run AutoLab on a pay per hour basis. The core lab with three ESXi hosts costs under $3 per hour to run.

3. Build and configure your outer platform

In the next few pages we will look at setting up the virtualization platform that will host the AutoLab: VMware Workstation, ESXi, Fusion and player

You only need to follow the instructions for your chosen platform, but make sure you follow all the instructions for that platform. The options for outer platform covered are:

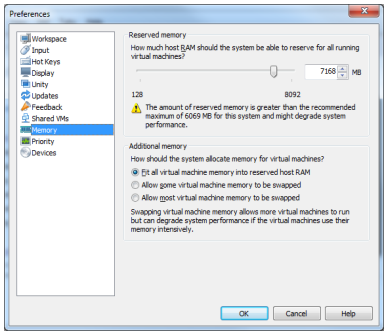
- VMware Workstation
- VMware Fusion
- VMware ESXi
- VMware Player
- Ravello

If you plan to deploy multiple copies of the AutoLab on the same outer ESXi host there are some special considerations which will be covered later in this guide.

VMware Workstation Setup

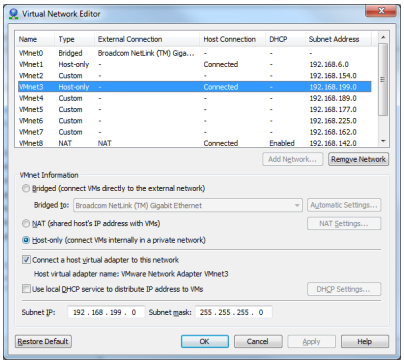
The VMware Workstation build is designed to work with a host with at least 24GB of RAM and VMware Workstation version 10.0 or later. The host operating system must be 64bit and all the CPU virtualization features must be enabled in the BIOS in order to be able to run the 64bit VMs. Placing the lab files on a fast disk (an SSD is highly recommended) and having a host with more RAM will make the lab run faster.

The main setup required is to reserve almost all of the 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 possible, allocate all but 1GB of RAM to the VMs.

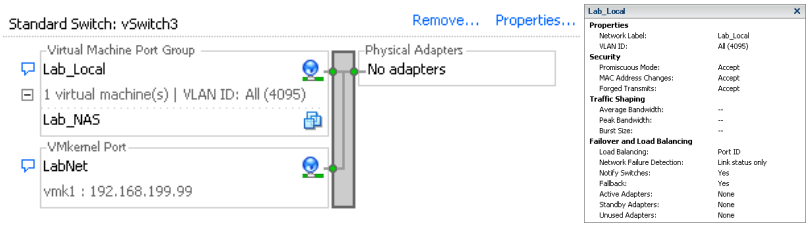
The other requirement is to configure the lab network. Under the **Edit** menu is the Virtual Network Editor. Select the **VMnet3** object. If that network isn't present, click **Add Network** button to add it. Make sure **Host Only** is selected in **VMnet Information** and that **Use Local DHCP Service 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. It is easiest to have the host connect to the network.



VMware ESXi Setup

The lab runs extremely well under ESXi, with a lower RAM footprint than on other platforms. 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 set to a VLAN ID that is unique on the switch. Your ESXi server can 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 from the outer ESXi server.



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

When you come to populate the build share and connect to the built VMs you will use the Router VM to provide access into the lab network, as discussed in the accessing the lab section.

VMware Fusion Setup

Huge thanks to James Bowling @vSential for documenting the whole Fusion 4 setup process and Damian Karlson @sixfootdad for updating for Fusion 5 Professional.

The Fusion build was designed to work with a host with 8GB of RAM and VMware Fusion version 4.0 or later. 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.

If you do not have Fusion 5 professional you can use its network editor, skip the Uber Network Fuser instructions below and follow the Fusion 5 Professional procedure.

Fusion (not Professional)

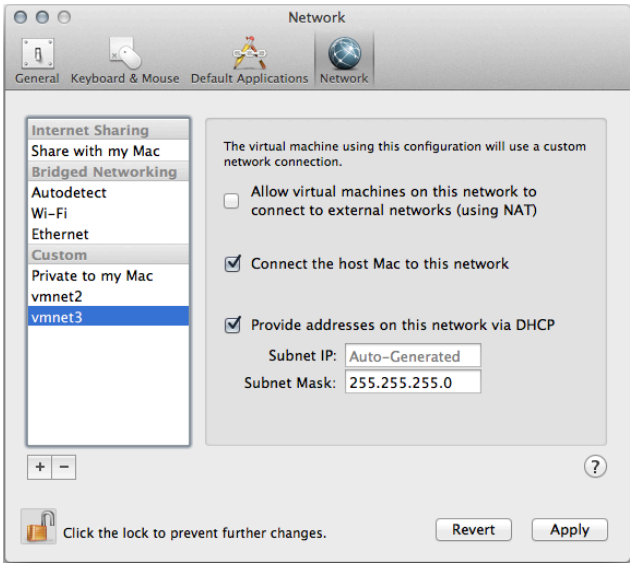
Need to document, would you like to contribute this documentation?

Fusion Professional - Configure the Lab Network

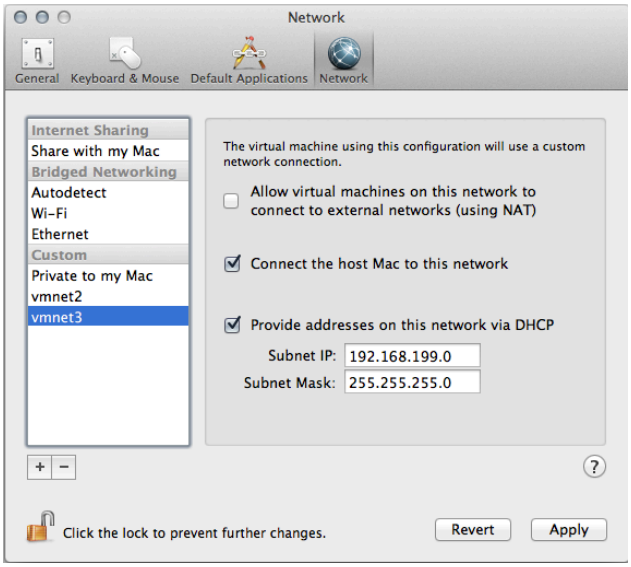
Open VMware Fusion menu > Preferences > Network tab

Unlock the window if necessary

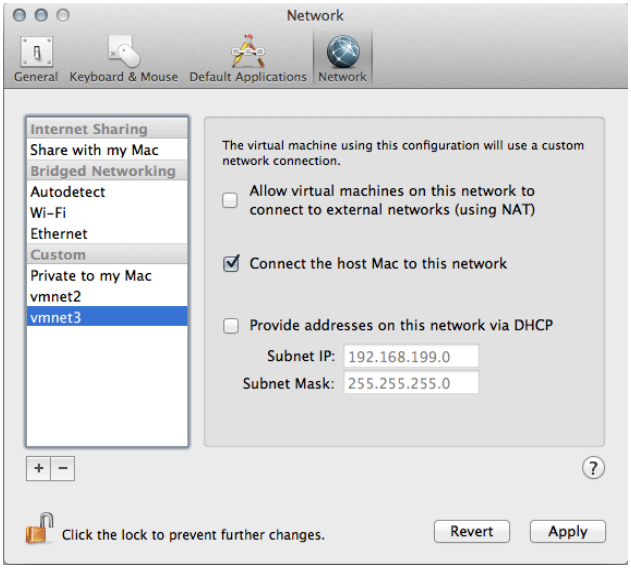
Click the plus sign at the bottom left to create vmnet3



In the Subnet IP field replace Auto-Generated with 192.168.199.0



Click Apply then uncheck 'Provide addresses on this network via DHCP' and click Apply again.



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

VMware Player Setup

VMware Player 6 & later

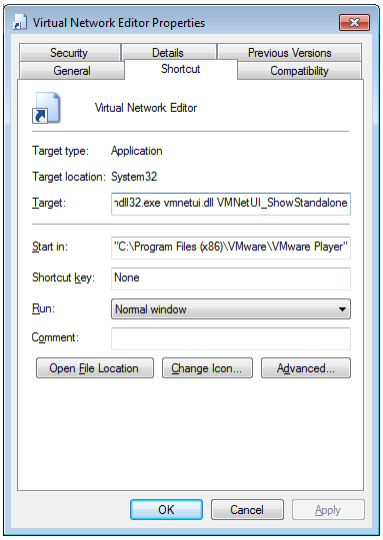
Need to document, would you like to contribute this documentation?

VMware Player 5.0.2

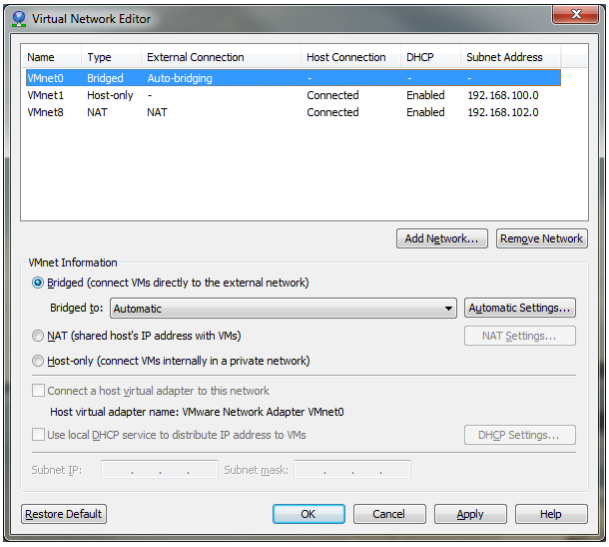
Install VMware Player

Find the program directory, where Player is installed. On my PC is was C:\Program Files (x86)\VMware\VMware Player

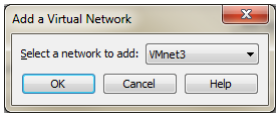
Create a shortcut with the command “rundll32.exe vmnetui.dll VMNetUI_ShowStandalone” with a working directory of the Player program directory you found above.



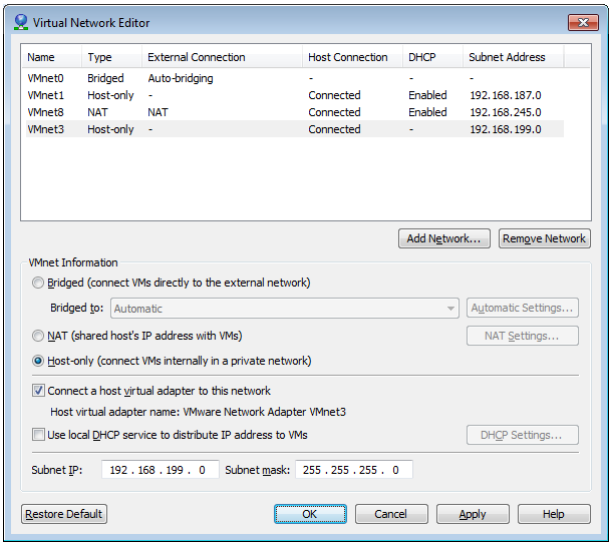
Use this shortcut to launch the Virtual Network editor.



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. It is easiest to have the host connect to the network.



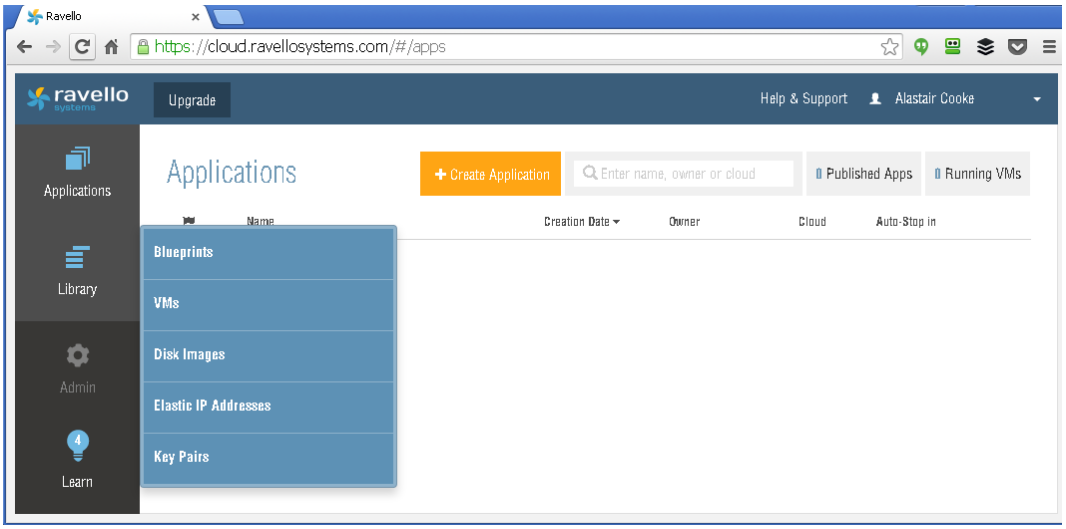
Ravello

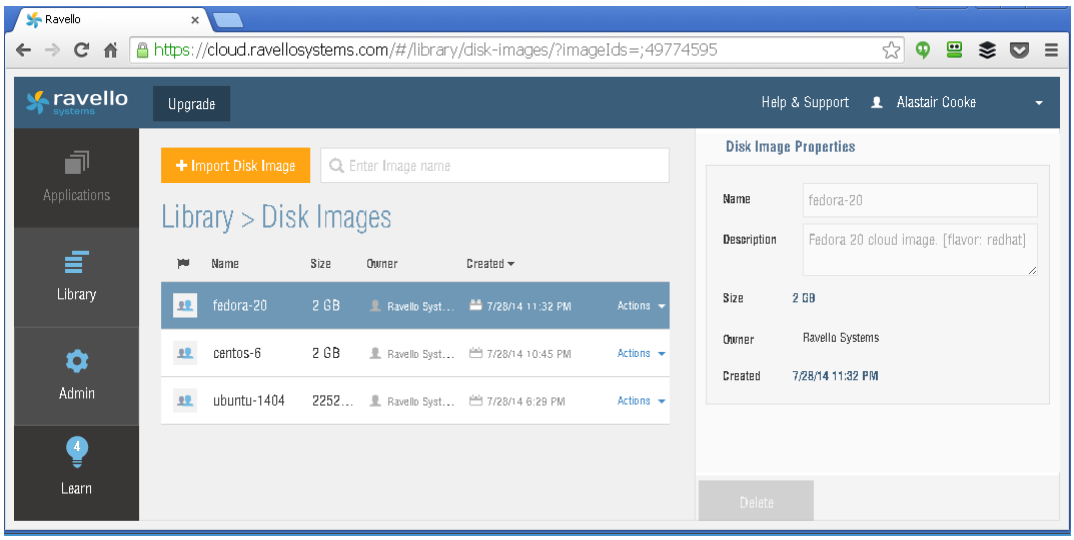
To use AutoLab on Ravello you first need to signup for a Ravello account at www.ravellosystems.com

The AutoLab Ravello blueprint includes all the VM setup and networking. If you do not have the AutoLab blueprint in your Library then log a support request with Ravello to have it added. You must then add your Windows Server and vSphere ISOs to your library.

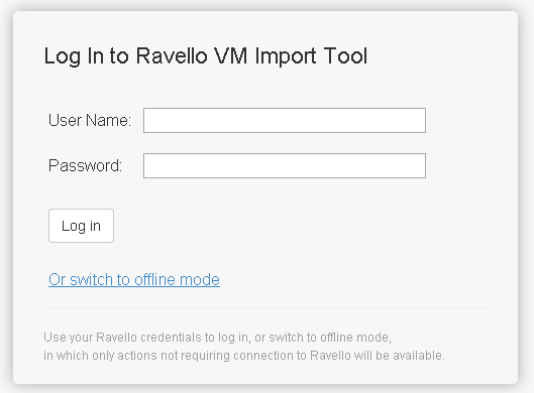
First copy the ISOs to your local disk, you cannot upload from a network share. You will need the Windows Server 2012 or 2016 iso as well as the ESXi and vCentre install ISOs.

Click the “Library” and select “Disk Images”.

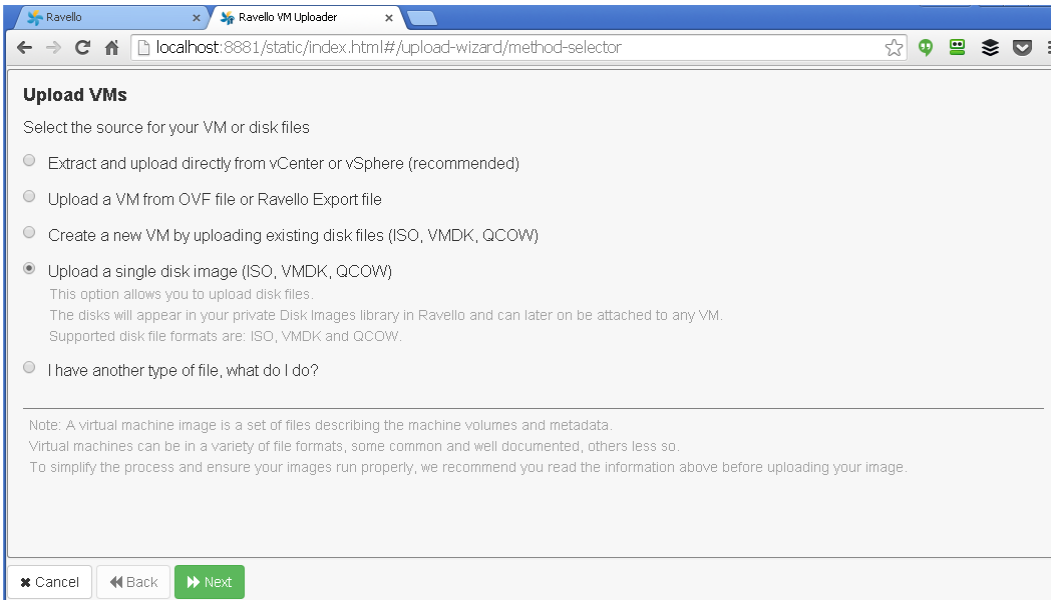




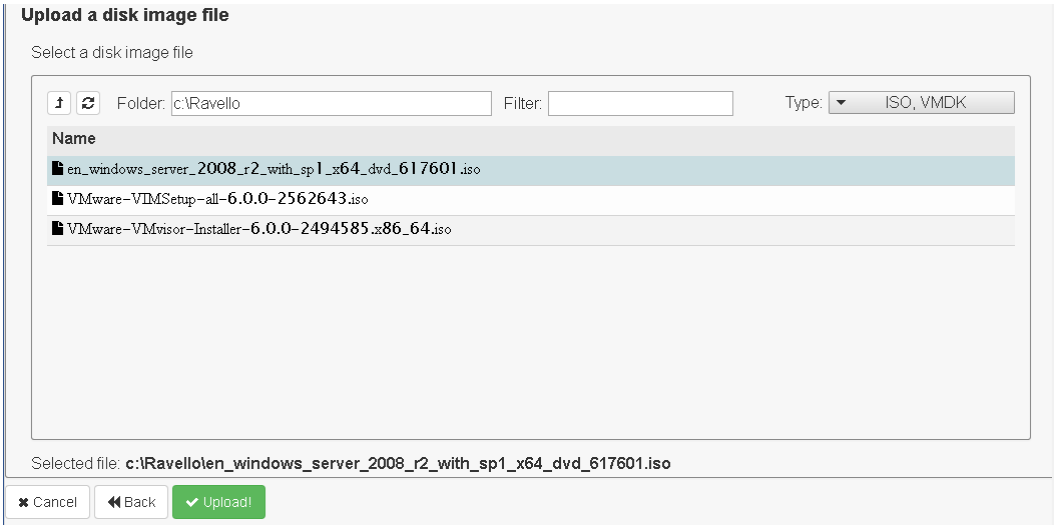
Next click the “+Import Disk Image” button, you will need to install the Ravello upload tool and then log into the upload application.



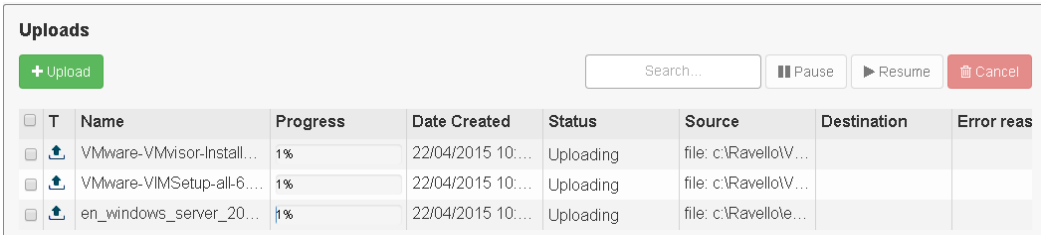
Choose to "**Upload a single disk image (ISO, VMDK, QCOW)**"



Browse for and select the ISO file, then click "Upload"



Repeat for the other ISOs, you may upload all three ISOs in parallel.



Once the three ISOs are uploaded you can close the uploader. The uploads may take some time, depending in the speed of your Internet connection.

The uploader tool does create a service that is always running, you may want to set the service to manual start.

4. Download the right AutoLab package

The AutoLab packages live at www.labguides.com/AutoLab. If you are not using ESXi then download the Workstation archive. There are two ESXi packages: one needs the ability to run vApps, so vCentre and a DRS cluster, the other is less simple to deploy but works with an ESXi environment. For each platform there is a slightly different way to get the VMs running.

VMware Workstation or VMware Player

Simply extract the ZIP file into the folder where you would like the VMs to live. Open the folder for the VM you need and double click the .vmx file. This will open the VM in Workstation or Player.

(other platforms are on the next few pages)

VMware Fusion

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. Rename the folders by adding ".vmwarevm" to folder names.

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

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

Modify Virtual Machine Network Settings Fusion (Not Professional)

Need to document, would you like to contribute this documentation?

Modify Virtual Machine Network Settings Fusion Professional

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 ESXi stand alone or with DRS

If you have vCenter in your lab environment then you will be able to deploy the multi-VM OVF. This requires either a DRS cluster or a standalone ESXi server managed by vCentre. vApps with multiple VMs cannot be deployed to a cluster that has HA and not DRS.

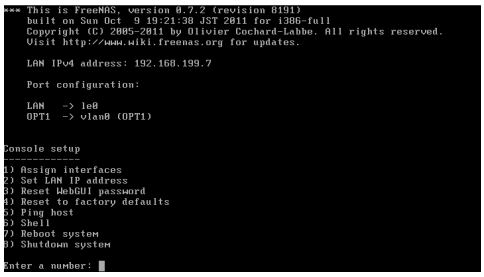
Import the OVF, give the vApp a unique name and select the required datastore and portgroups. The **Lab_Local** portgroup should map to the portgroup you created above and the **External** portgroup should map to you normal production network from which you access the vSphere environment. The router VM will consume one DHCP IP address from the external network.

To build the lab you will need to power on the VMs one at a time, you can safely ignore the warning that this is not the preferred way to handle vApps.

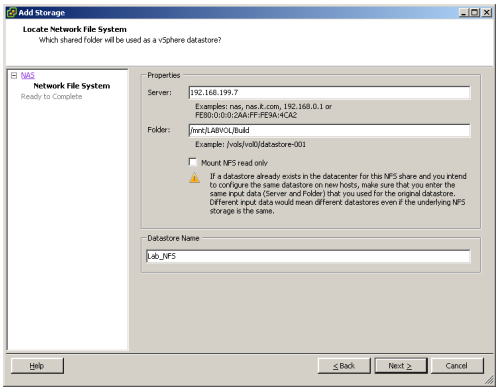
VMware ESXi HA cluster without DRS

If your ESXi environment is not able to run multi VM vApps then you will need to follow these instructions. If you can deploy they multi-VM vApp then do, it is a lot less work. You may want to deploy a temporary vCentre server just to enable you to deploy the multi-VM vApp, then remove the temporary vCentre before building AutoLab.

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



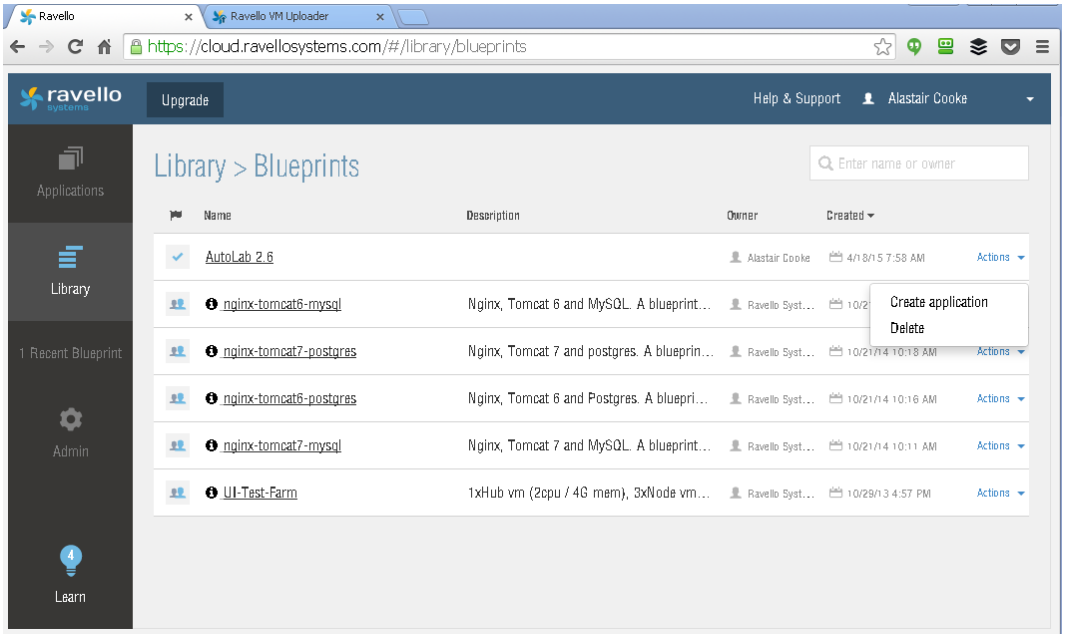
Once the datastore is created use the vSphere Client datastore browser and browse to **\Automate\ShellVMs** where you will find the remaining lab VM folders. The VMs must not be run from this location as they won't perform well and the datastore will quickly run out of space.

If you have vCenter you can register each VM then migrate it to its proper datastore before you power it on. 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.

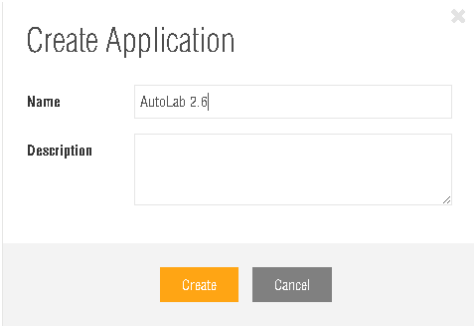
Finally the VMs need to have their CDROM and floppy drives attached to media images. You may need to copy the boot floppy images from the Build share, in **\Automate\BootFloppies** to another datastore. The floppy images match the VM names, apart from vCloud which doesn't need a floppy.

Ravello

In the Ravello portal select “Blueprints” from the Library menu. Click the “Actions” link on the “AutoLab 3.0” blueprint and select “Create Application”



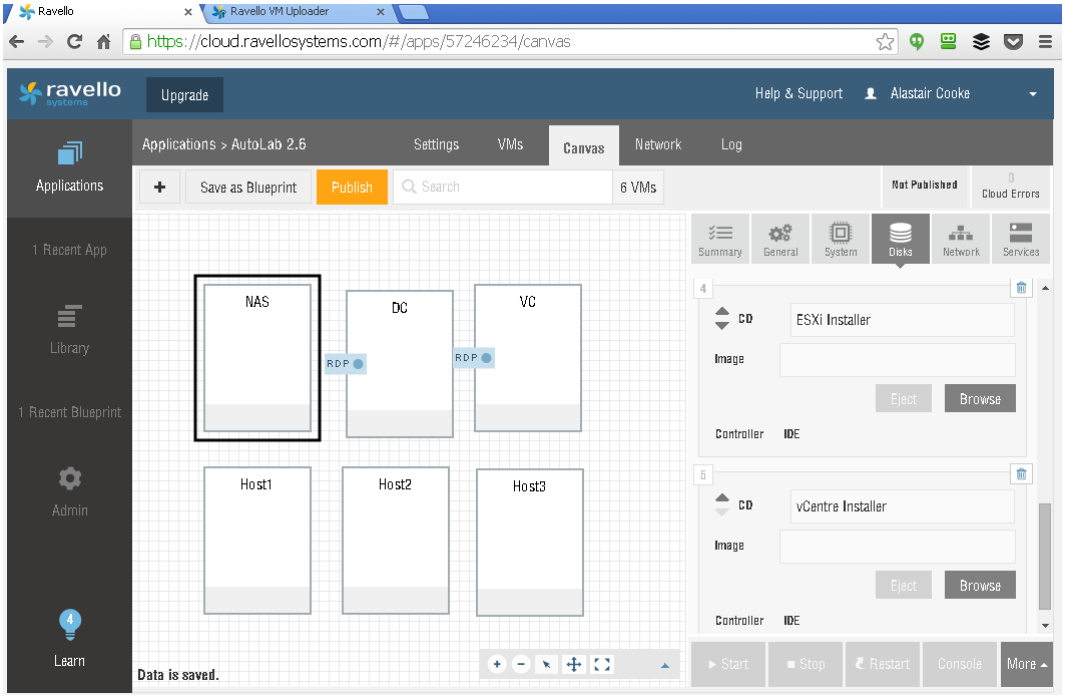
Give the application a name and maybe a description, this is useful if you have multiple AutoLab instances built.



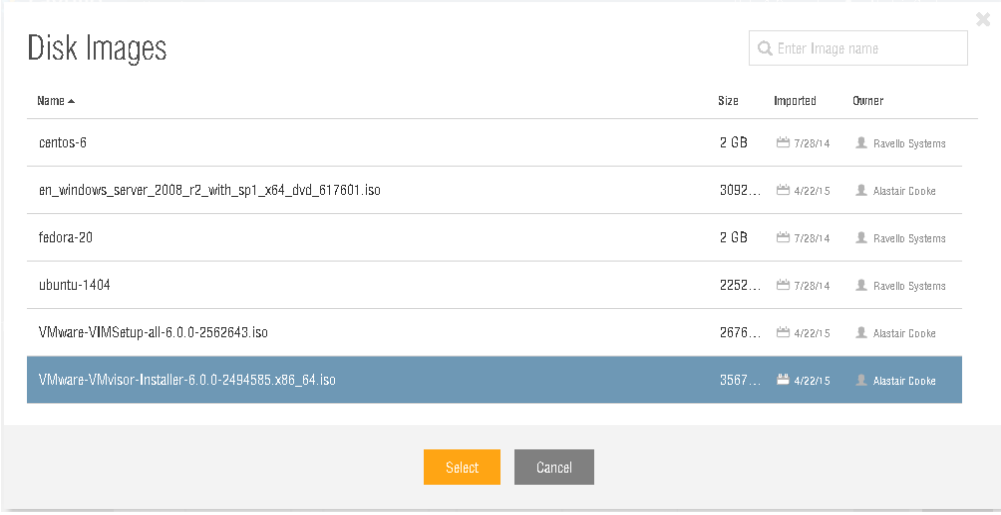
With the Ravello platform in the cloud it doesn’t make sense to access the Build Share directly, over the Internet. To get the vSphere installers into the AutoLab we must attach the ISOs to the NAS before we start the NAS VM.

On the Ravello platform you do not need the router VM, this is provided by the Ravello platform.

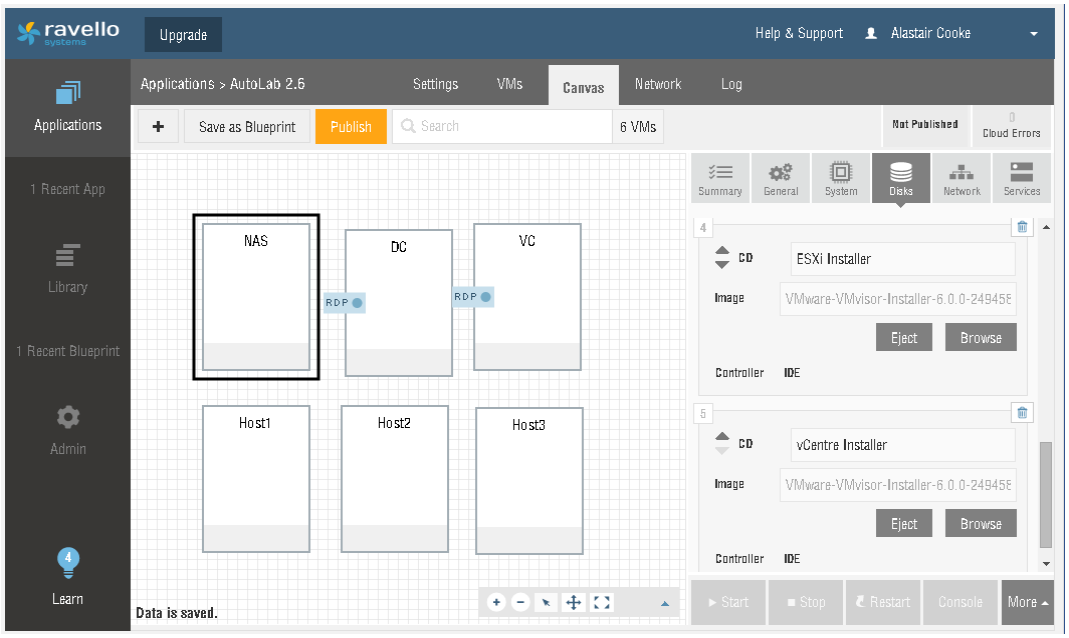
When the Application is deployed, select the NAS VM on the Canvas. The click the Disks tab and scroll down to the CDROM devices.



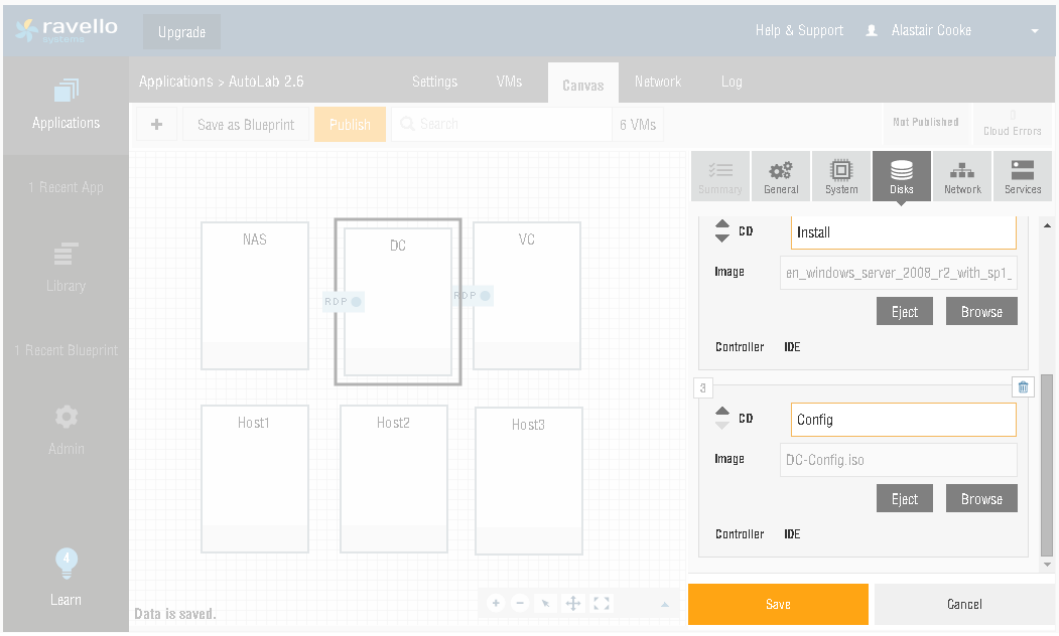
Click browse to select one of the ISOs you uploaded. Attach the ESXi installer (VMware-Vmvisorxxxx.iso) to one CDROM and vCentre installer (VMware-VIMSetupxxx.iso) to the other CDROM.



When you are done with the VM it should have two ISOs connected, then click Save and wait until you see “Data is saved” in the bottom left.

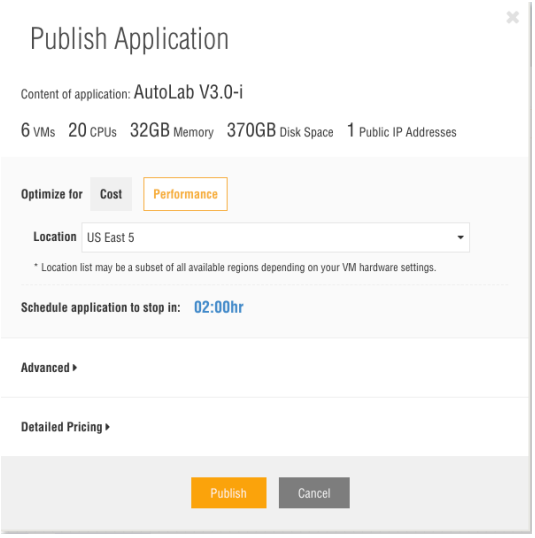


Next attach the Windows Server 2012 or 2016 ISO to both the DC and the VC. It is best to add the ISOs now, although you can do this just before the VMs are powered on if you forget.



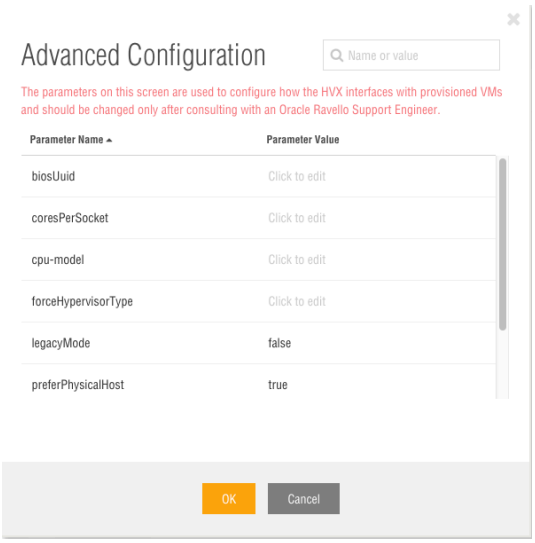
Once you have saved the changes to the NAS, DC and VC you should re-check that the changes really have saved. If you move a bit fast then one change may overwrite another.

When you are sure that the ISOs are connected click Publish.

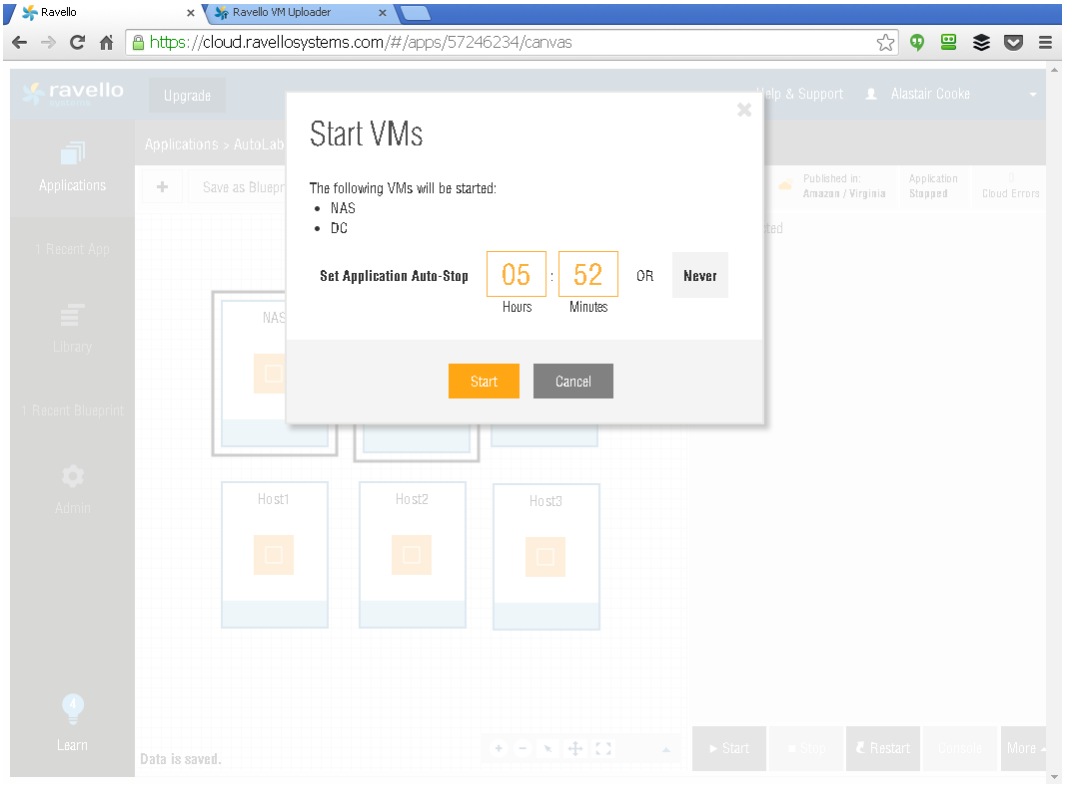


I usually Optimize for Performance and choose Amazon as the cloud platform, this usually costs under $3.00 per hour. Make sure to open the Advanced option and uncheck the "Start all VMs automatically" option. For the build we need to control when the VMs start.

The VMs in the blueprint are all set to prefer to use the Ravello hypervisor on bare metal, which delivers superior performance. If you choose to optimize for performance then you will be limited to Locations where this is available. To turn off this preference, open the General tab on each VM and click "Advanced Configuration" then change the setting "preferPhysicalHost" to false. Repeat this process for each of the six VMs.



Next power on the DC and NAS at the same time. Click to select both and then click Start. I usually select an Auto-Stop time of six hours for lab builds, it usually takes under four hours but you want contingency for things not going according to plan. Like forgetting to come back and power on the next set of VMs.



Provided the right ISOs are attached the DC build will complete in a little over an hour. This brings you quite a way into stage 6. Build the Lab. In particular the next stage is running the Validate script in the DC.

The virtual machines

The VMs in the download package are setup so you can install with as little RAM as possible. To build the core lab (2 x ESXi, vCentre & supporting VMs) you will need quite a bit of RAM. For vSphere 6.0 anything less than 20GB of physical RAM will need careful management. vSphere 5.5 can be built with around 12GB of RAM. To build with VSAN, View or vCloud you will need more RAM, and if you want to run a few nested VMs you will need a bunch more RAM. Take a look at the Tuning AutoLab for RAM section for details of what can be done with less RAM.

VM Name	Section	Role	Minimum RAM	Ideal RAM
DC	Core	Domain Controller	512MB	1GB
VC	Core	Virtual Centre	8GB	12GB
NAS	Core	Shared Storage	512MB	512MB
Router	Core	In and outbound access	256MB	256MB
Host1 **Host2** **Host3**	Core	ESXi Server	4GB	8GB or more
CS1 & **CS2**	View	Connection Serve	1GB	2GB
SS	View	Security Server	512MB	1GB
vCloud	vCloud	vCloud Director	1.5GB	3 GB or more
V1	Veeam	Veeam ONE	1GB	2GB or more
VBR	Veeam	Veeam Backup & Replication	1GB	2GB or more

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

5. Download the licensed 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, evaluation versions are fine. For the older vSphere and PowerCLI versions you will need an account with VMware or a good contact at VMware or a VMware partner. The older version components are only required if you plan to build an environment and then run the upgrade, i.e. 5.5 to 6.0 or 6.7.

Core Lab components

- vCenter and ESXi
  - Versions you want to use; 4.1 to 6.7 are supported.
- VMware PowerCLI installer
  - Use the same version as the oldest vSphere & vCenter version you will deploy.
- VMware Tools windows.iso
  - For VMware Workstation: located at C:\Program Files (x86)\VMware\VMware Workstation\
  - For VMware Fusion: Finder -> Applications -> Show Package Contents of VMware Fusion -> Contents/Library/isoimages
- Microsoft Windows Server 2012 R2 or 2016 180 day trial DVD, ISO file. This must be the trial ISO, not a full product ISO.

Optional Components

- View 5.0 to 7.5 installers
- VMware vCLI for vSphere
- Microsoft SQL Server 2008 R2 SP1 - Express Edition Management Studio
- VMware vMA
- Microsoft Windows Server 2008 R2 180 day trial DVD, ISO file
- Microsoft Windows 2003 Server 32bit CDROM, ISO file
- Windows 10 ISO
- Windows 8 ISO
- Windows 7 ISO
- Windows XP ISO
- vCloud binary and vShield (aka vCloud Networking & Security) appliance versions 1.5 & 5.1 are supported.
- Veeam ONE and Veeam Backup & Replication installers

6. Build the Lab

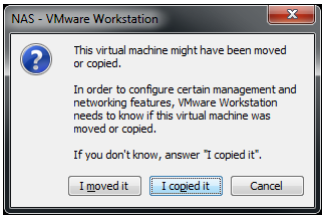
These steps use the full set of automation to build a complete lab environment. The steps should be completed in order with each step completing before starting the next step. The build steps are the same irrespective of the outer virtualization platform, i.e. Workstation, ESXi or Fusion.

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, there is a section later about rebuilding.

Task 1 – Prepare the prebuilt VMs

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

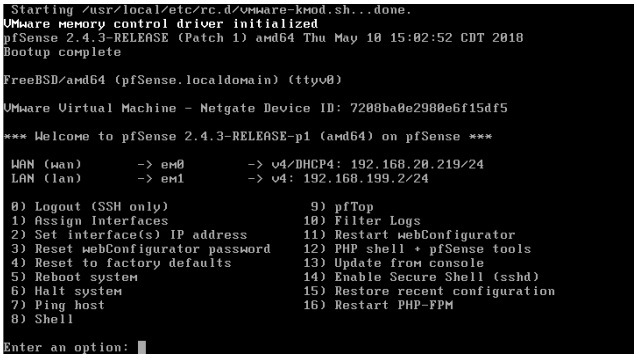
When you power on each VM you may be asked whether you moved or copied the VM. Always answer "I copied it" for these VMs, this way a new UUID and MAC address is assigned for each VM and makes running multiple isolated copies of the lab possible.



Router

The Router VM is used to allow outbound connectivity from the AutoLab network and inbound management. The Router is required for the Windows Server 2012 evaluation to activate, without Internet access the Windows Evaluation will fail to install. If you are building the lab on ESXi then you will need the router to allow access to the lab. If you are building on Ravello then there is no Router VM, it's role is filled by the Ravello platform.

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 "WAN (wan) -> em0 -> v4/DHCP". This is useful if you don't have a PC connected to the Lab network such as deploying a lab on ESX.



NAS

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.

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 inaccessible. (This can be fixed with chown & chmod if you've already made this mistake.)

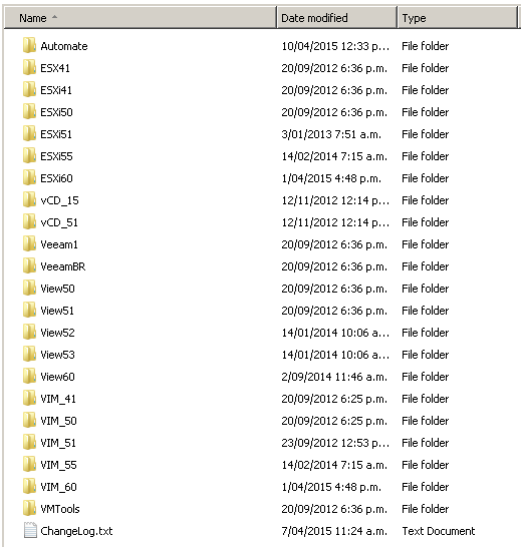
On **Ravello** the build share is populated by attaching the vSphere install ISOs to the NAS VM before it is powered on. If you miss this stage then attach the ISOs and restart the NAS VM. On **Ravello** you can power on the DC at the same time as the NAS, by the time the DC needs files from the NAS it will have finished copying from the ISOs. The DC will take approximately an hour to build, you can open the VM console, or just go do something else while you wait.

Populate the 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 2012, 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:



Core lab requirements:

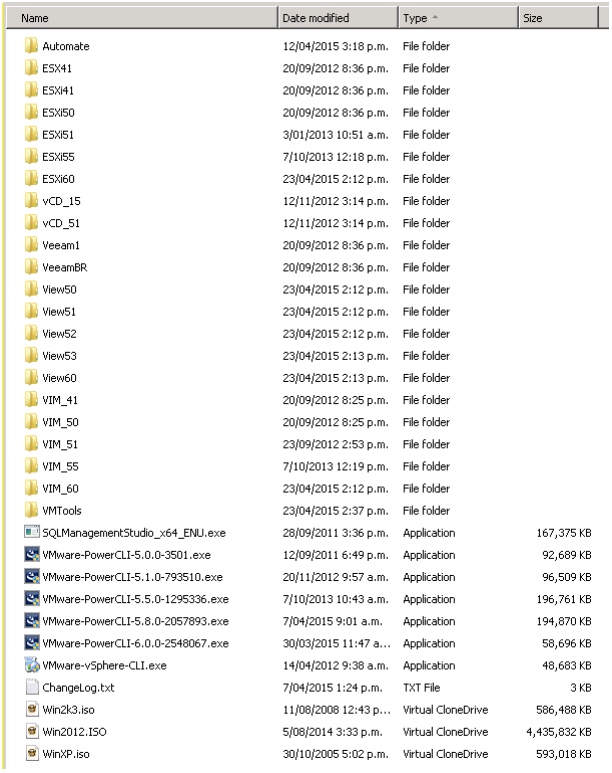
The **ESXi** folders need to have the ESXi installer ISO placed in or extracted into them. If you are using Fusion then I suggest you copy in ISOs, there can be issues with file naming if you copy the extracted files.
The **VIM** folders need to have the vCenter installers extracted into them.
VMware-PowerCLI-xxxx.exe - the installer for PowerCLI renamed. Do not use a newer version of PowerCLI than your vCenter as this will cause scripts to fail.

Additional Components:

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 **View** folders should hold the View Agent, Composer and Connection server installers for the View version.
WinXP.ISO – Windows XP 32bit with SP3 install ISO, also used to create unattended install ISO for nested VM for View
Win7.ISO – Windows 7 64bit evaluation install ISO, to install windows in a nested VM
Win8.ISO – Windows 8 64bit evaluation install ISO, to install windows in a nested VM
Win10.ISO – Windows 10 64bit evaluation install ISO, to install windows in a nested VM
Win2K3.ISO – Windows 2003 Server 32bit install ISO, to create an unattended Win2K3 install ISO in the VC build and then install windows in a nested VM
Win2008.ISO – Windows 2008 Server 64bit evaluation install ISO, to install windows in a nested VM
Win2012.ISO – Windows 2012 Server 64bit evaluation install ISO, to install windows in a nested VM

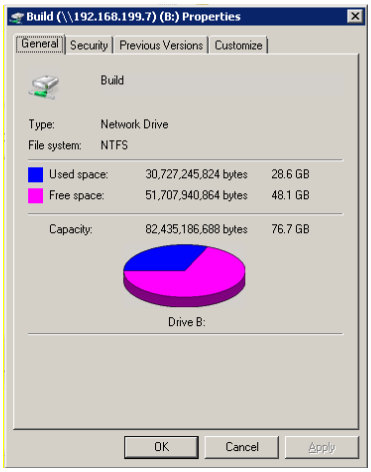
Win2016.ISO – Windows 2016 Server 64bit evaluation install ISO, to install windows in a nested VM
SQLManagementStudio_x64_ENU.exe – Microsoft SQL Server 2008 R2 SP1 - Express Edition Management Studio installer, will be installed on DC so you can do database troubleshooting. Download from
http://go.microsoft.com/fwlink/?LinkID=228417&clcid=0x409
VMware-vSphere-CLI.exe - so you can play with command lines without using the VMA
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 Provided your AutoLab has Internet access the correct VMTools will be downloaded when the DC builds.
The **Veeam** folders are really for organization, as I haven't managed to automate the Veeam software installs.

A fully populated Build share looks like this:



Name	Date modified	Type	Size
Automate	12/04/2015 3:18 p.m.	File folder	
ESX41	20/09/2012 8:36 p.m.	File folder	
ESXi41	20/09/2012 8:36 p.m.	File folder	
ESXi50	20/09/2012 8:36 p.m.	File folder	
ESXi51	3/01/2013 10:51 a.m.	File folder	
ESXi55	7/10/2013 12:18 p.m.	File folder	
ESXi60	23/04/2015 2:12 p.m.	File folder	
vCD_15	12/11/2012 3:14 p.m.	File folder	
vCD_51	12/11/2012 3:14 p.m.	File folder	
Veeam1	20/09/2012 8:36 p.m.	File folder	
VeeamBR	20/09/2012 8:36 p.m.	File folder	
View50	23/04/2015 2:12 p.m.	File folder	
View51	23/04/2015 2:12 p.m.	File folder	
View52	23/04/2015 2:12 p.m.	File folder	
View53	23/04/2015 2:13 p.m.	File folder	
View60	23/04/2015 2:13 p.m.	File folder	
VIM_41	20/09/2012 8:25 p.m.	File folder	
VIM_50	20/09/2012 8:25 p.m.	File folder	
VIM_51	23/09/2012 2:53 p.m.	File folder	
VIM_55	7/10/2013 12:19 p.m.	File folder	
VIM_60	23/04/2015 2:12 p.m.	File folder	
VMTools	23/04/2015 2:37 p.m.	File folder	
SQLManagementStudio_x64_ENU.exe	28/09/2011 3:36 p.m.	Application	167,375 KB
VMware-PowerCLI-5.0.0-3501.exe	12/09/2011 6:49 p.m.	Application	92,689 KB
VMware-PowerCLI-5.1.0-793510.exe	20/11/2012 9:57 a.m.	Application	96,509 KB
VMware-PowerCLI-5.5.0-1295336.exe	7/10/2013 10:43 a.m.	Application	196,761 KB
VMware-PowerCLI-5.8.0-2057893.exe	7/04/2015 9:01 a.m.	Application	194,870 KB
VMware-PowerCLI-6.0.0-2548067.exe	30/03/2015 11:47 a...	Application	58,696 KB
VMware-vSphere-CLI.exe	14/04/2012 9:38 a.m.	Application	48,683 KB
ChangeLog.txt	7/04/2015 1:24 p.m.	TXT File	3 KB
Win2k3.iso	11/08/2008 12:43 p...	Virtual CloneDrive	586,488 KB
Win2012.ISO	5/08/2014 3:33 p.m.	Virtual CloneDrive	4,435,832 KB
WinXP.iso	30/10/2005 5:02 p.m.	Virtual CloneDrive	593,018 KB

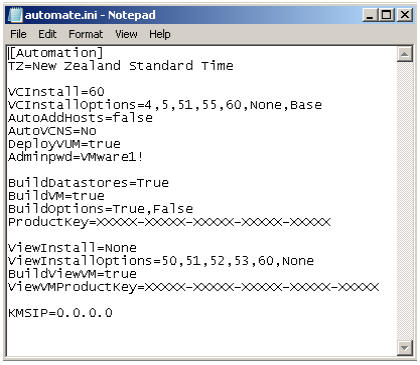
In my lab the fully populated Build share contains 30+GB of files, if you don't have all of the vSphere versions or other software in your lab then your Build share will be smaller.



Automation Control

In the Automate folder on the build share you will find Automate.ini, this file controls the level of build automation. The file has the following entries:

TZ: 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.
VCInstall: Alows the automatic installation of vCenter in the VC VM, the VCInstallOptions line shows valid choices. None simply installs the VMTools. Base also installs SQLNative client and makes a couple of other convenient changes. Version numbers mean automatically install vCenter.
AutoAddHosts: If set to true runs the add hosts script when the VC build completes. The ESXi servers must be built prior to VC being built.
AdminPWD: Sets a new default password for all accounts. This should be set to something other than VMware1! For good security, particularly if you access this lab over the Internet. Unfortunately many special characters are stripped by PowerShell as the string is read, avoid using any either $ or `
BuildDatastores, **BuildVM** and **ProductKey**: These control how much the script to add hosts to the vCenter server will do, anything you want to do yourself you can turn off. These settings don't affect the VC build. The ProductKey line is for your Windows 2003 Server product key for use with nested VMs.
ViewInstall, **BuildViewVM** and **ViewVMProductKey**: Automatically install View, the version to install, whether to build a Windows XP VM and what product key to use in the Windows XP VM.



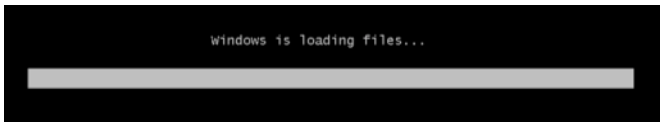
For the initial test build I suggest having the automation fully build vSphere, datastores and VMs. This way you can be sure all the sources and setup steps work. Subsequent rebuilds can be less automated to allow you to do tasks manually.

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.

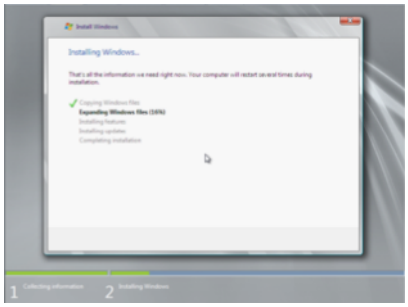
Task 2 – Build DC, Windows Infrastructure

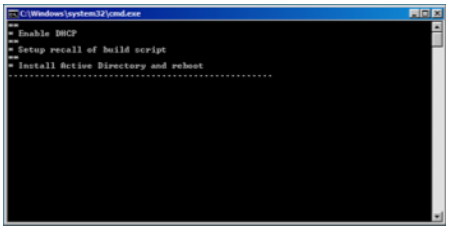
On both the VC and DC VMs make sure the CDROM drive is connected to a Windows Server 2012 R2 or 2016 evaluation ISO, connected at power on is recommended. Also make sure that the right config ISO image is connected, on Workstation and Fusion this is part of the package that you downloaded. On ESXi you may have to re-attach the config ISO to the second CDROM drive. Backup copies of the config ISO images can be found on the build share in **\Automate\BootFloppies**.

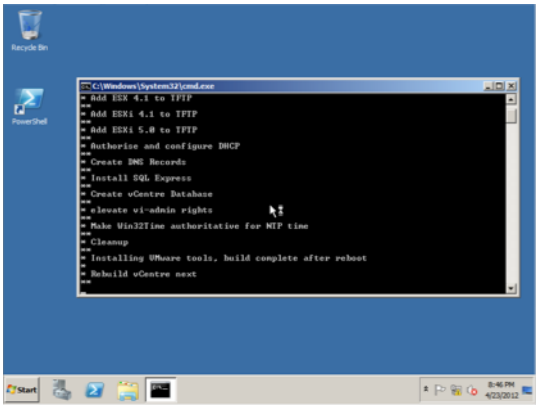
Power on the DC VM to start the unattended install. The first time you boot this VM it has a blank hard disk, so it will boot from the Windows installer CD and begin the build process. On subsequent reboots the installer will pass boot over to the hard disk unless you press a key. Pressing a key at this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



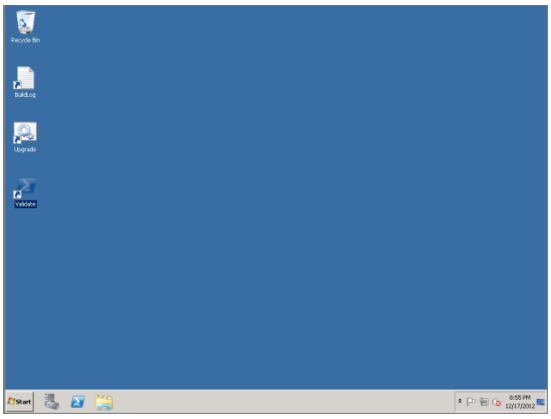
The VM will boot from the Windows Server CDROM and use “autounattend.xml” file on the config ISO image to automate the Windows install. This will take some time, go talk to your family for a while, or read some documentation to pass the time. On my laptop this takes around an hour. No input will be required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







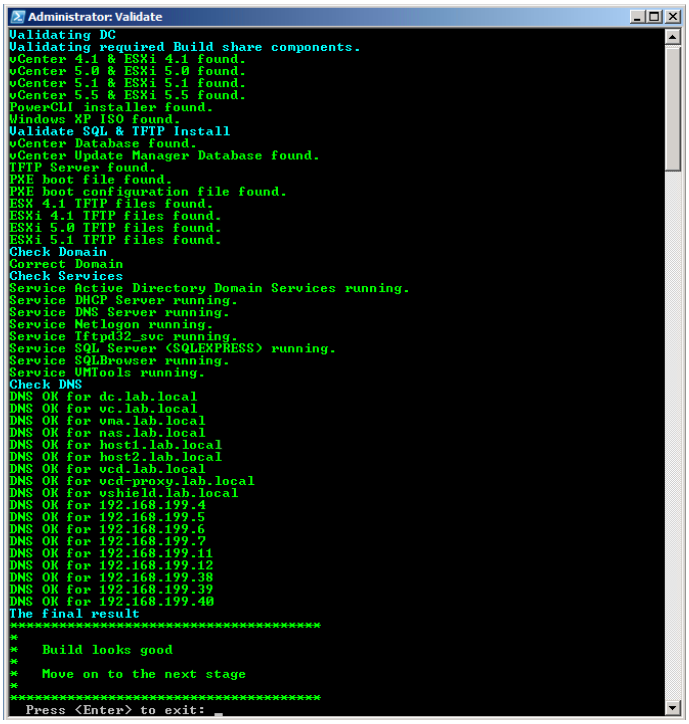
After installing AD the VM will restart and install SQL Express and setup the PXE environment followed by installing the VMware tools. If these steps fail or does not start make sure your NAS VM is running and that you setup the Build share as outlined in task 1.



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 called Validate 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 which may help you resolve any issues.

There is a script to test that the build has completed successfully, and that the Build share was correctly populated. Double click the Validate icon on the desktop.



As you would expect green is good, yellow is OK, red not so good. If your Autiomate.ini file does not change the default password you will be asked to set a new default password and the existing accounts will have their password updated. 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.

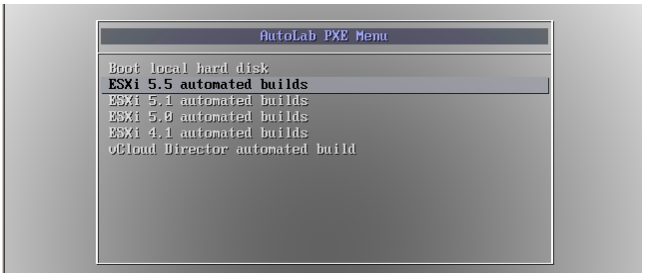
Once the Validate script completes with green you are ready for task 3.

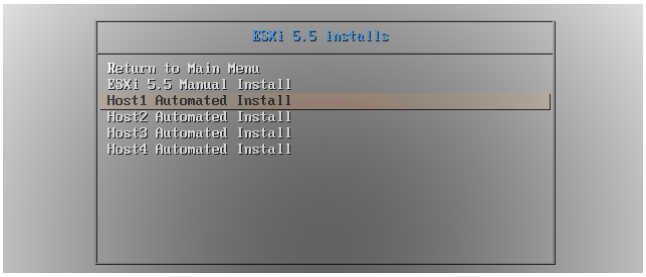
Task 3 – Build ESXi servers

Power on the ESX VMs one at a time, once the first starts the automated build you can move on to the second. If you plan to build as ESX 4.1 then change the OS type on the VMs to reflect.

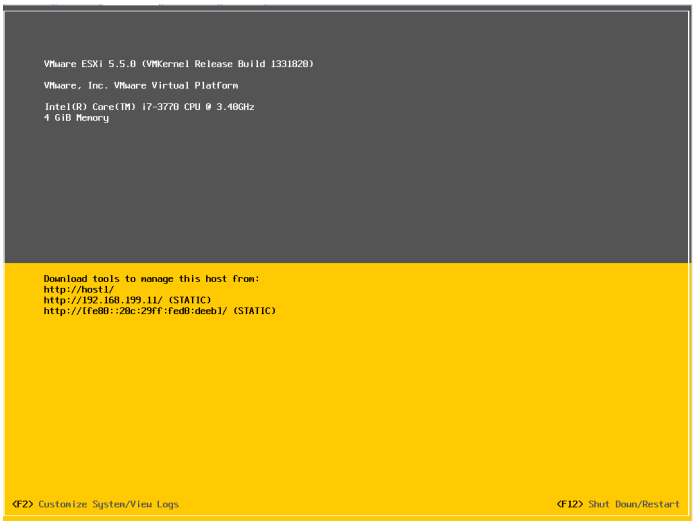
When the VMs boot they will use PXE to load a menu from the DC, use the keyboard arrow keys to choose the option for the ESXi version and host number you wish to install. For each ESXi version there is a menu allowing the automated build of each ESXi host or a manual install from the network. The automated installs build with the standard IP addresses and little 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



If your physical machine has less than the recommended amount of RAM then refer to the memory optimization section.

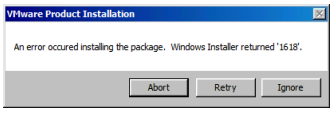
Build the second ESXi host as required. Confirm that both ESX servers have the correct static IP addresses before moving to the next stage.

Task 4 – Build VC, vCenter server

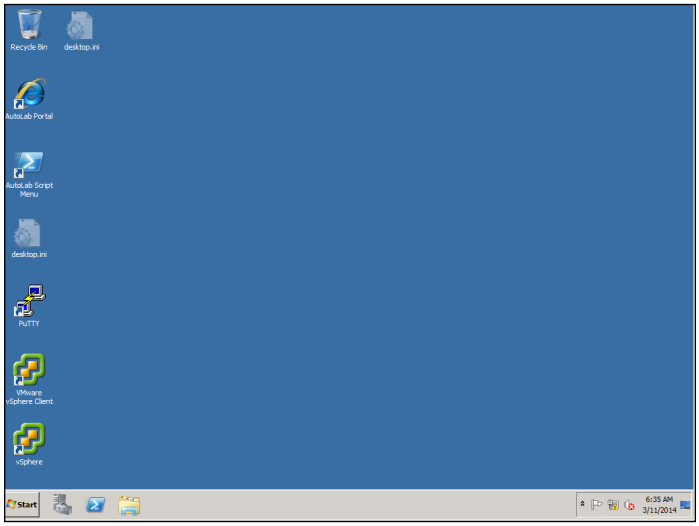
This process begins the same as the DC build, attach the ISOs\. Then boot from the Windows install disk, the config ISO contains the “autounattend.xml” file.

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.

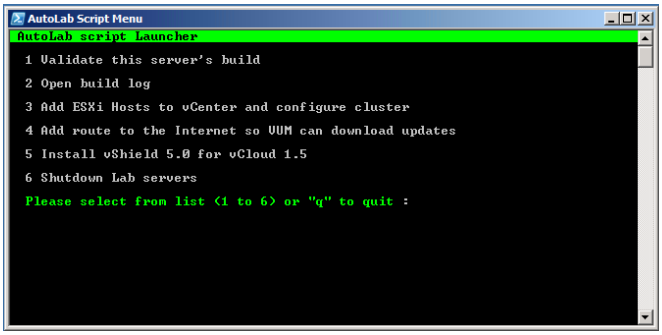
During the VC build you may see a Windows Installer error 1618, you can safely ignore this.



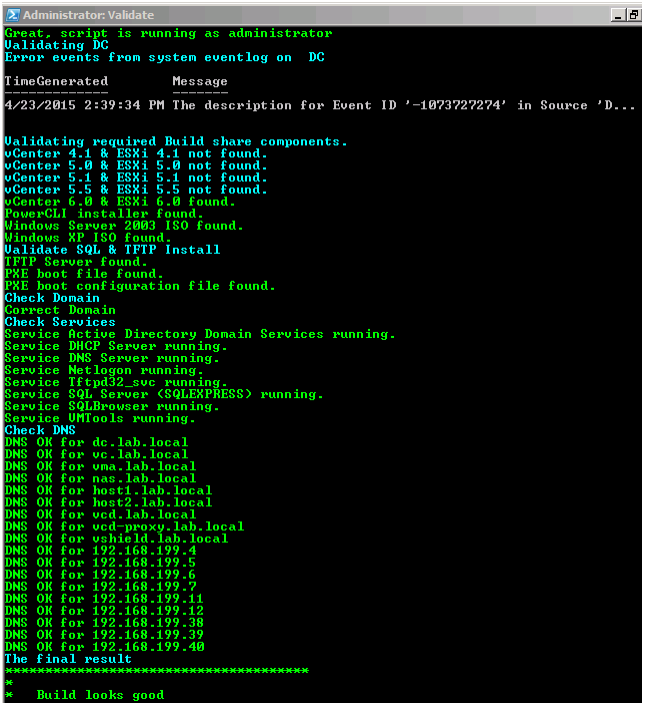
When the automation that you chose in the Automate.ini file completes the VM will be left logged on and configured with Autologon for your convenience.



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.



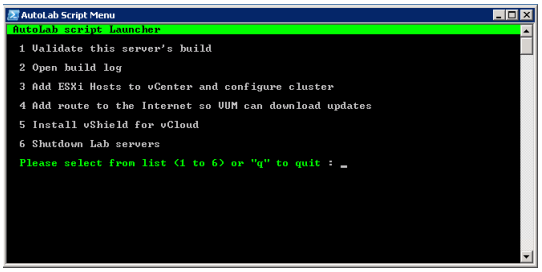
It may take a few minutes after you are able to login before all the services are started, be patient as the "VMware VirtualCenter Management Webservice" can take a few minutes to start so don't worry of validation fails on that, give it five more minutes.

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.

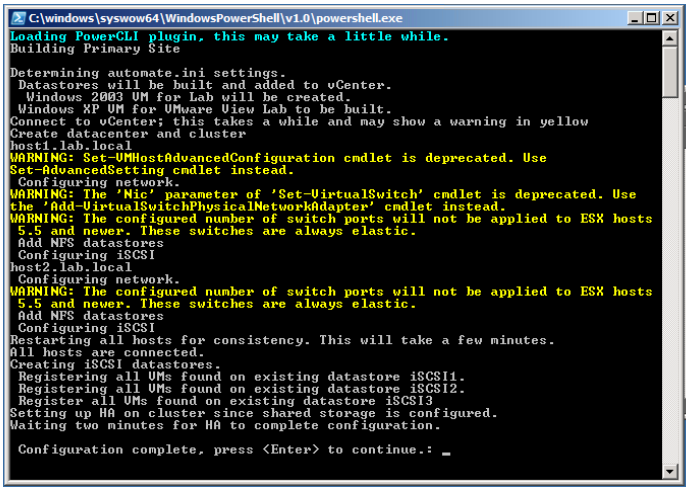
Task 5 – Populate vCenter

If you put the AutoAddHosts=true line in your automate.ini file then this has already been done as part of the VC build, otherwise you can do this now.

To add the ESXi servers to vCenter, setup an HA and DRS cluster as well as networking and datastores on the ESXi servers use the **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.



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.

Lab Build Time

This table gives an indication of the time to build a core vSphere lab using a computer with a quad core i7, lots of RAM and an SSD, your mileage will vary.

0:00 – Start, 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

2:00 - VC Built, power on Host1 and select ESXi 5 build

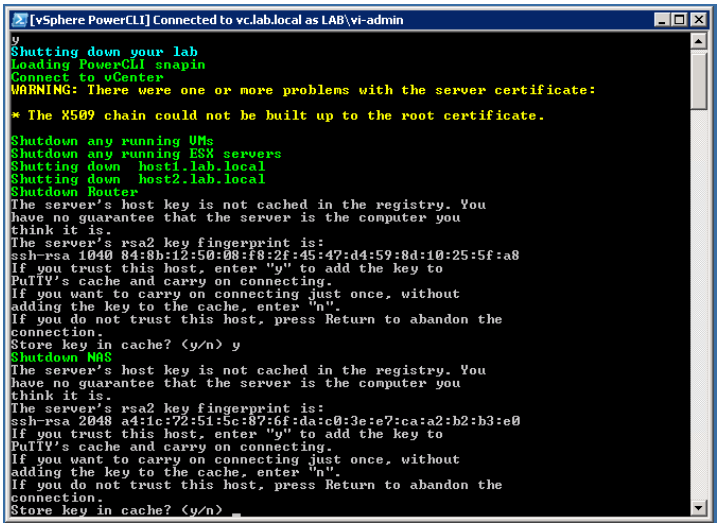
2:15 - Host1 built, power on Host2 and select ESXi 5 build

2:30 - Host2 built, run AddHosts script

3:00 - Cluster Built, datastores built and first VM installing operating system

Shutting the lab down

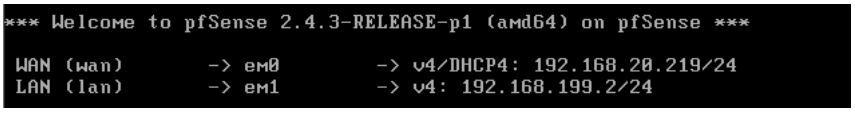
Since the lab takes up so much of the resources on a PC you will probably want to shut it down when you're not actively working on it. The Shutdown Lab Servers option from the AutoLab Script Menu on the VC desktop runs a PowerShell script that will quickly but cleanly power down everything except the 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).



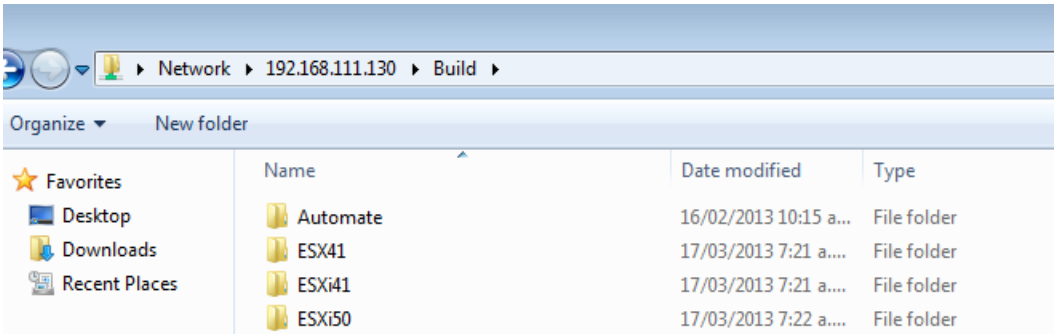
Accessing the built lab

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 192.168.111.219, I use a DHCP reservation and a fixed MAC address on my router VM to keep this IP address consistent in my lab.

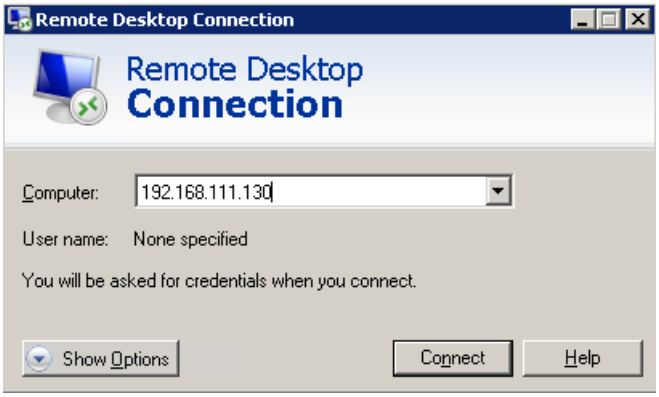


Windows sharing from the NAS VM is published on the normal ports so you can access the Build share through the routers external IP address.

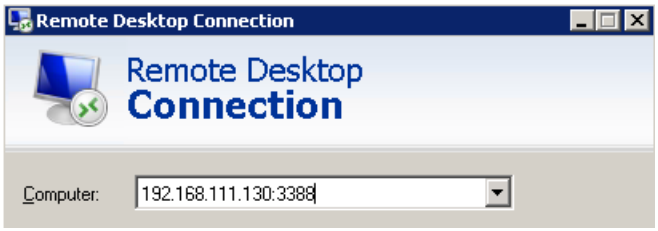


The management web interface of the NAS VM is available on the standard HTTP port 80 of the router's external IP address.

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.

Host1 on port 122
Host2 on port 222
VMA on port 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

VLANID	None	
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	
Host physical PC	192.168.199.1	
Router (gateway)	192.168.199.2	gw.lab.local
Domain Controller	192.168.199.4	dc.lab.local
vCenter server	192.168.199.5	vc.lab.local
vCenter Management Appliance	192.168.199.6	vma.lab.local
FreeNAS	192.168.199.7, nas.lab.local	
Host1	192.168.199.11	host1.lab.local
Host2	192.168.199.12	host2.lab.local
View Connection Server 1	192.168.199.33	cs1.lab.local
View Connection Server 2	192.168.199.34	cs2.lab.local
View Security Server	192.168.199.35	ss.lab.local
Veeam ONE server	192.168.199.36	v1.lab.local
Veeam Backup &Replication Server	192.168.199.37	vbr.lab.local
vCloud Director	192.168.199.38	vcd.lab.local
vCloud Proxy	192.168.199.39	vcd-proxy.lab.local
vShield Manager	192.168.199.40	vshield.lab.local

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 4.1Service 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, Windows accounts have their password set from the Automate.ini file on the build share.

	Username	Password
Lab.local domain	Administrator	From Automate.ini
vSphere Administration	LAB\vi-admin	From Automate.ini
Veeam services	LAB\SVC_Veeam	From Automate.ini
vCloud access to vCenter	LAB\SVC_vCD	From Automate.ini
SRM services (not yet used)	LAB\SVC_SRM	From Automate.ini
Router web admin	admin	VMware1!
NAS administration	admin	VMware1!
vCloud server	root	VMware1!
VMA	vi-admin	VMw@re1!
Ada Lovelace	ada	From Automate.ini
Alan Turing	alan	From Automate.ini
Charles Babbage	charles	From Automate.ini
Grace Hopper	grace	From Automate.ini

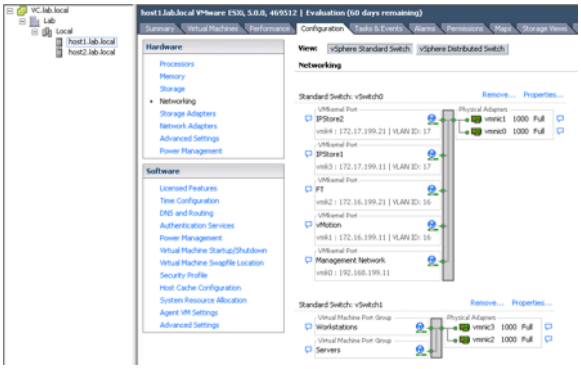
Databases

The following databases are created in the DC build, all userIDs are SQL users, all passwords are **VMware1!**

DB Name	Password	Owner	Function
vCenter	VMware1!	vpx	vCenter
VUM	VMware1!	vpx	Update Manager
ViewEvents	VMware1!	VMview	View Events
ViewComposer	VMware1!	VMview	View Composer
SRM	VMware1!	VMSRM	Site Recovery Manager
SRMRep	VMware1!	VMSRM	vSphere Replication
RSA	VMware1!	RSA_USER	vCenter 5.1 SSO

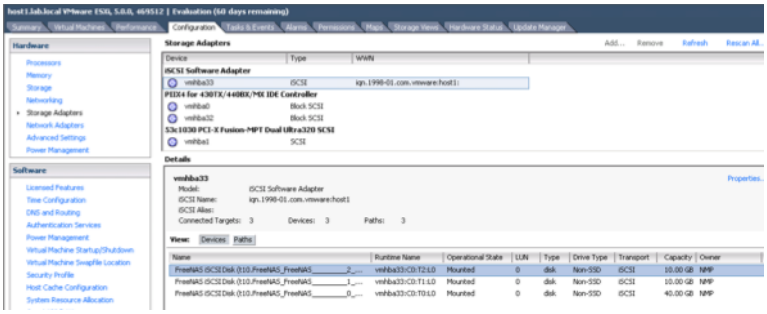
Host Network

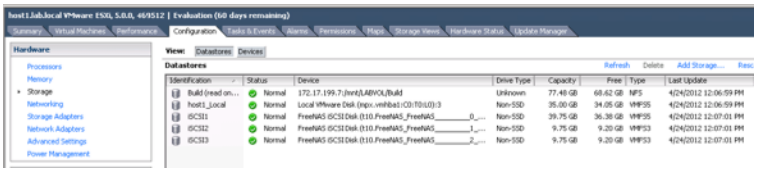
vMotion and Management Network VMkernel ports have symmetric NIC teaming on VMnic0 and VMnic1; each active on one and standby on the other. Both ports are enabled for Management traffic to provide redundant HA heart beating. There is no routing between the subnets; the default gateway for the ESXi servers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





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 from the download.

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". The upgrade script will also update the PXE configuration for new ESXi versions added to the build share. After running the upgrade script you must rebuild the VC.

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.

Tuning AutoLab for RAM

RAM is expensive, just like in your production vSphere it is liely to be the scarce resource in your AutoLab. The Ravello platform allows you rent a platform with a lot of RAM, it is a good option if you need short term lab with a lot of resource.

You will find that the nested cluster cannot hold a lot of VMs, the lab includes a tiny Linux VM called TTYLinux which you can clone to get a few VMs. Take a look at **http://ttylinux.net/** to learn more about TTYLinux.

vSphere 6.0

ESXi 6.0 requires 4GB of RAM to install and 3.5GB of RAM to operate after it is installed, meaning two ESXi 6.0 host will require 7GB of RAM.

vCentre 6.0 will not install with less than 8GB of RAM, however after install it can use as little as 4GB of RAM. If your physical host has 16 GB of RAM you may be able to build 2 ESXi hosts and then VC by allowing some VM RAM to be swapped. After building vCentre drop its RAM to 4GB.

vSphere 5.5

As ESXi 5.5 requires 4GB of RAM to install the ESXi VMs are configured for 4GB each. To fit in the 8GB minimum host these VMs need to be reduced to 2GB each after ESXi is installed.

If you have 16GB of RAM you can run all three host VMs with 4GB each. If you have 32GB then you can increase them to 6GB and get VSAN to work.

vSphere 5.0 and earlier

Older vSphere versions are much less demanding and will allow a core lab on a physical host with only 8GB of RAM.

ESXi servers can have 2GB of RAM configured and vCentre 1.5GB.

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=75, 70, 60, 53, 52, 51, 50, None
BuildViewVM=ask
BuildViewVMOptions=True, False
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 Add Hosts to vCenter 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 usually only want one Connection server in the lab but replicas are useful for testing load balancing connection servers or testing Tags. With View 5.0 the vCenter server will be added to the View configuration along with its View composer function. View 5.1 and later presents some issues with certificates so the VC isn't automatically added to View although it is still installed. View Composer domains do not appear to be able to be added automatically using PowerShell, so you will need to set this up yourself.

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, again this does not appear to be automatable using PowerShell.

Manual View 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**

Configuration item	Value
Server Functions	
First Connection Server	CS1
Second Connection server	CS2
Security Server	SS
View Composer	
Server	VC
Database Server	DC\SQLEXPRESS
Database	ViewComposer
Database User	VMView
Database Password	VMware1!
View Events Database	
Database Server	DC\SQLEXPRESS
Database Type	Microsoft SQL Server
Database Port:	1433
Database Name	ViewEvents
Database User	VMView
Database Password	VMware1!
Database Table Prefix	VE_

Running Multiple AutoLabs

One aim in AutoLab was to make it easy for a team to share one ESXi server and each team member to have their own AutoLab. Each instance of AutoLab will require it's own PortGroup with a unique VLAN ID. If each AutoLab instance can be contained within one ESXi host then an internal-only vSwitch can be used. If AutoLab instances will span multiple ESXi hosts, say a DRS cluster, then the prortgroup must be connected via physicla NICs & swicthes with VLANs. The router VMs will need to connect to the same external network but will each use only a single IP address for the whole AutoLab instance.

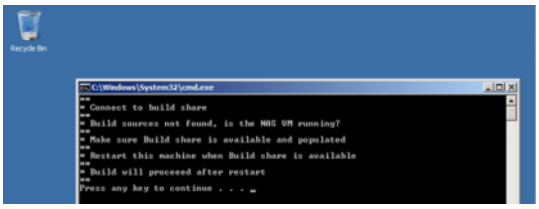
You will want to use the OVF deployment and leave the vApp in place, that way you can have the same VM name in each AutoLab instance. Do keep in mind that the VM folders on the datastores will be the same, so if two instances share a datastore the VM folder names won't all match the VM names.

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. There may be a message on screen about populating the build share, make sure you have put the required files in the right places. Make sure the Build share is fully populated and then rebuilt the DC.



Windows Server 208 vSphere Web Client

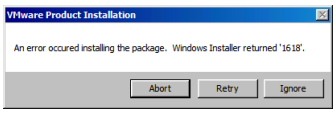
The web client does not work with the version of Internet Explorer that is installed with Windows Server 2008 R2. Use Windows Update to install Internet Explorere 11, there is no need to install any of the other updates unless you like to be patched. Be carefull of free space on the Windows VMs, their disks are quite small.

Ravello, wrong ISOs

If you attached the wrong ISOs to the NAS on the Ravello platform before you powered it on then there may not be vSphere installers on the Build share. Attach the correct ISOs and restart the NAS VM.

Error installing PowerCLI on VC

A "runtime error 1618" shows on the VC VM during install. This is a purely cosmetic error; either ignore it or wait a few minutes and click Retry.

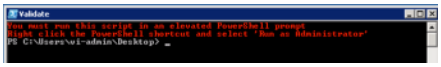


AddHosts.ps1 exits

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.

Validate complains about not being run as administrator

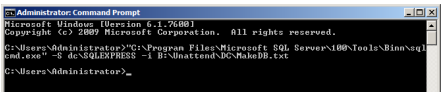
The validate script must be Run as Administrator, right click the shortcut and select this from the popup menu.



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.

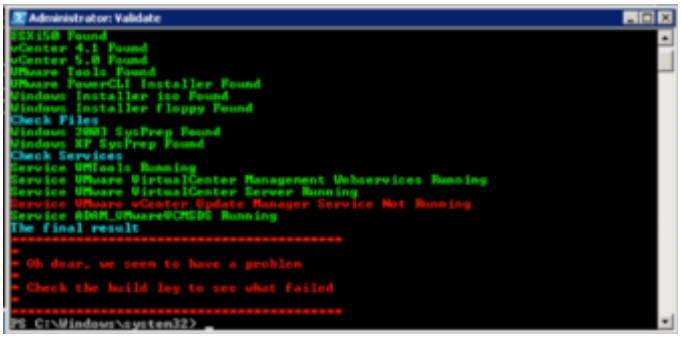


VC Build Fails to create vCenter Repository

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.

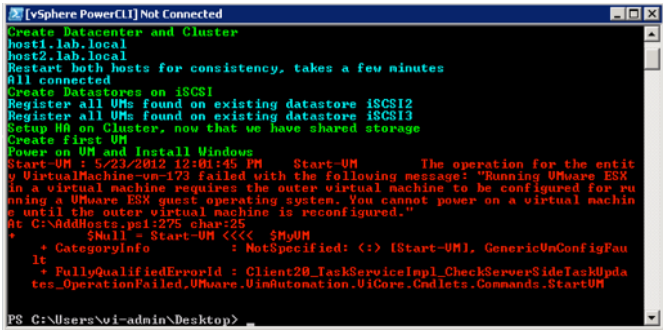
VUM 4.1 fails to install

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.

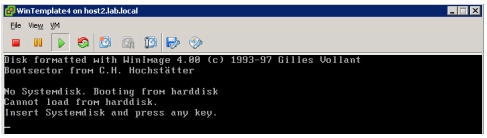


Host1 build fails –Resolved in V1.1

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.

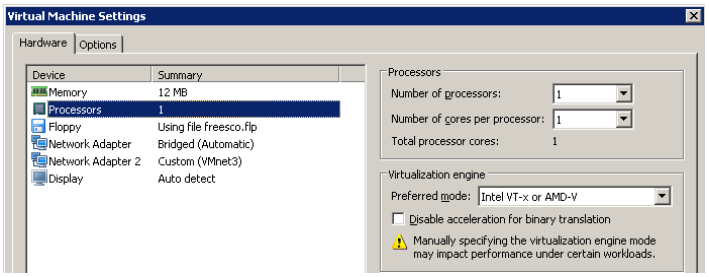
First VM doesn't install windows on vSphere 4 builds –Resolved in V1.1

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 Version Changes

V3.0

Support for vSphere 6.5 and 6.7

Support for View 7.0 and 7.5

Support for Windows server 2016

Nested Windows Server 2008, Windows Server 2016, Windows 7, Windows 8, Windows 10

pfSense firewall/router

V2.6

Support for vSphere 6.0

Support for Ravello

Nested Windows Server 2012

V2.0

Support for Windows Server 2012

Support for View 6.0

Removal of VLANs

V1.5

Support for vSphere 5.5
Support for View 5.2 & 5.3
Ground work for SRM support

V1.1a

Addition of vCloud 5.1
Cleaned up requirement to have vCenter 5.0 installer for successful install
Added support for completing vSphere 5.1 Install, Configure, Manage course labs using AutoLab, workbook to be released shortly.

V1.1

Addition of vSphere 5.1
Removal of requirement for specific Windows 2008 installer ISO, SP1 or RTM OK
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

AutoLab 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**
    - In progress
- **Time zone into nested VMs and Guest OS Customization specification**
- **vCentre Appliance based AutoLab (no Windows)**
- **View configuration**
    - Security server pairing password

If you solve any of these problems then let us know via feedback@labguide.com so we don't need to duplicate your effort.

VMware vCloud Director Installation

Automated build

Damian Karlson's (@sixfootdad) excellent work has brought vCloud Director to the AutoLab.

Required Software Source

Oracle Database Express Edition 11g Release 2 for Linux x64 from 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.

Physical host configuration

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.

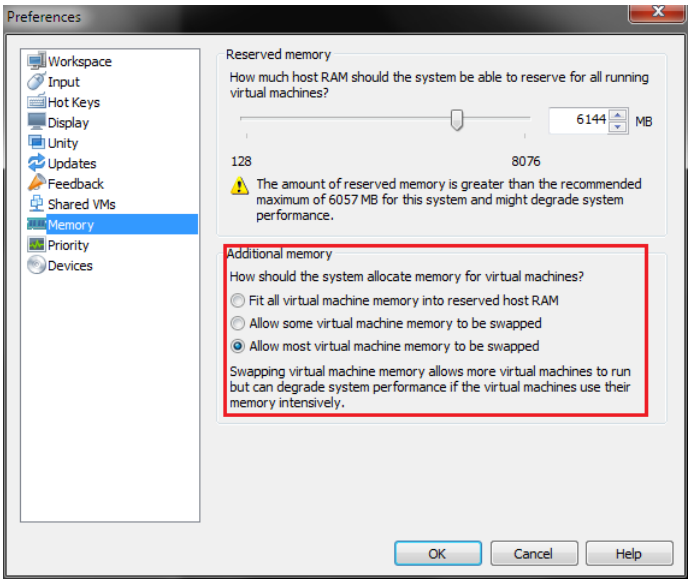
Since many vCloud labs don't actually require the use of VMs with operating systems installed within the vCloud environment, our recommendation is to create small virtual machines that have 4MB of RAM and 4MB disks. This will allow you to work with catalogs, creating vApp templates, instantiating vApp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

VMware vCloud Director Installation instructions:

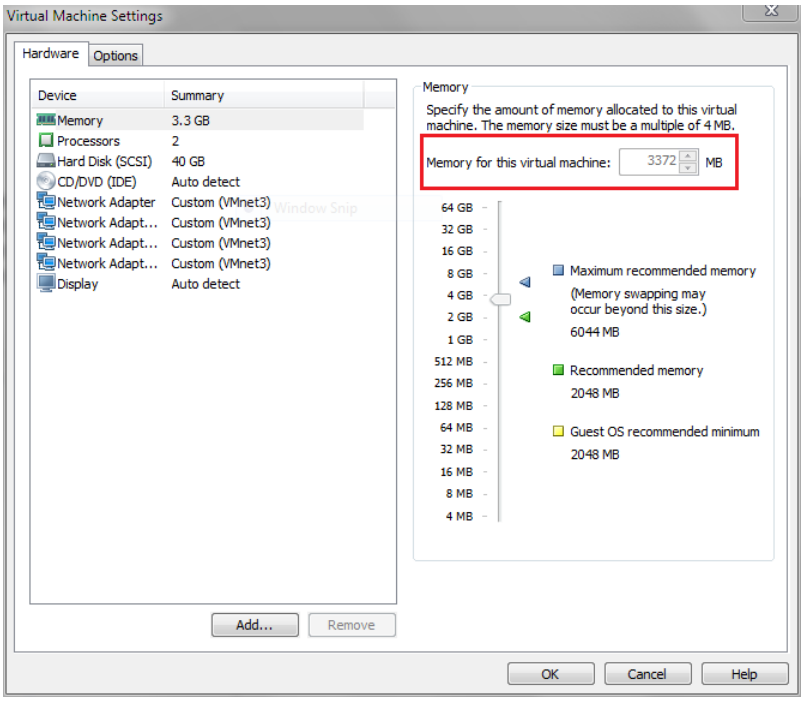
After procuring the necessary software, copy the following files to the Build folder on the NAS VM, typically //192.168.199.7/Build/vCD_15:

oracle-xe-11.2.0-1.0.x86_64.rpm
vmware-vcloud-director-1.5.1-622844.bin
VMware-vShield-Manager-5.0.1-638924.ova

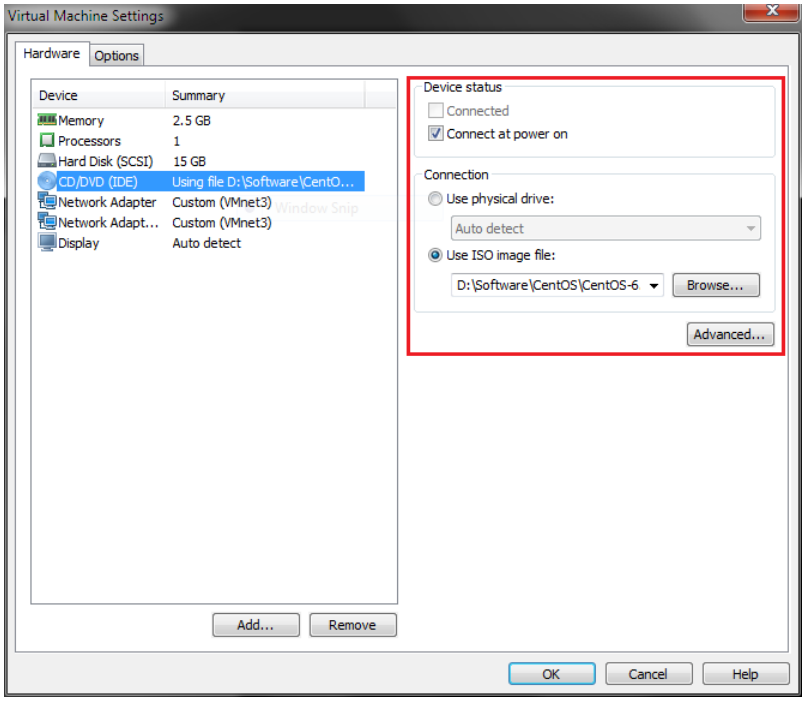
Verify or change the Workstation memory preferences to "Allow most virtual machine memory to be swapped". If you're running Workstation on Windows 7, you may be required to run Workstation as an administrator. If you are lucky enough to have 16GB of RAM in your lab host you should be able to leave all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

vShield 5.0 for Cloud 1.5

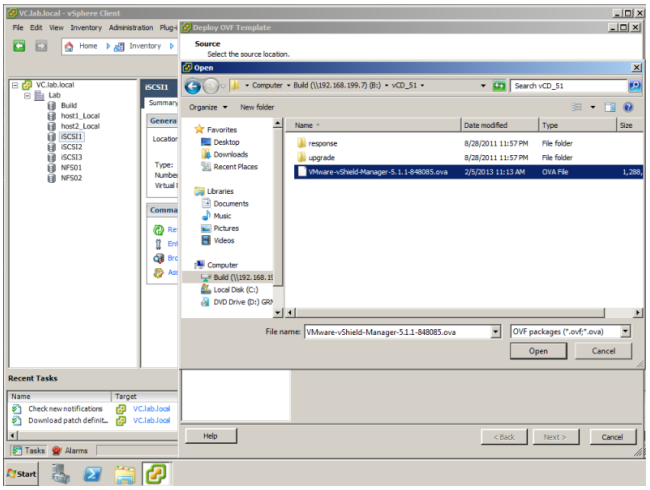
This setup is highly automated, in the VC VM and run the **Install vShield 5.0 for vCloud 1.5** option from the **AutoLab Script Menu** located on VC's Desktop.

vShield 5.1 for vCloud 5.1

This version integrates with SSO and the Lookup service, which is not supported by the PowerCLI commandlets so the install is all manual, follow this process:

Open the vSphere client and select **Deploy OVF Template…** option from the **File** menu

Browse to **B:\vCD_51\VMware-vShield-Manager-5.1.2-943471.ova** and click through the rest of the Deploy OVF Template wizard. The iSCSI1 datastore should have sufficient free space provided you thin provision the VM.

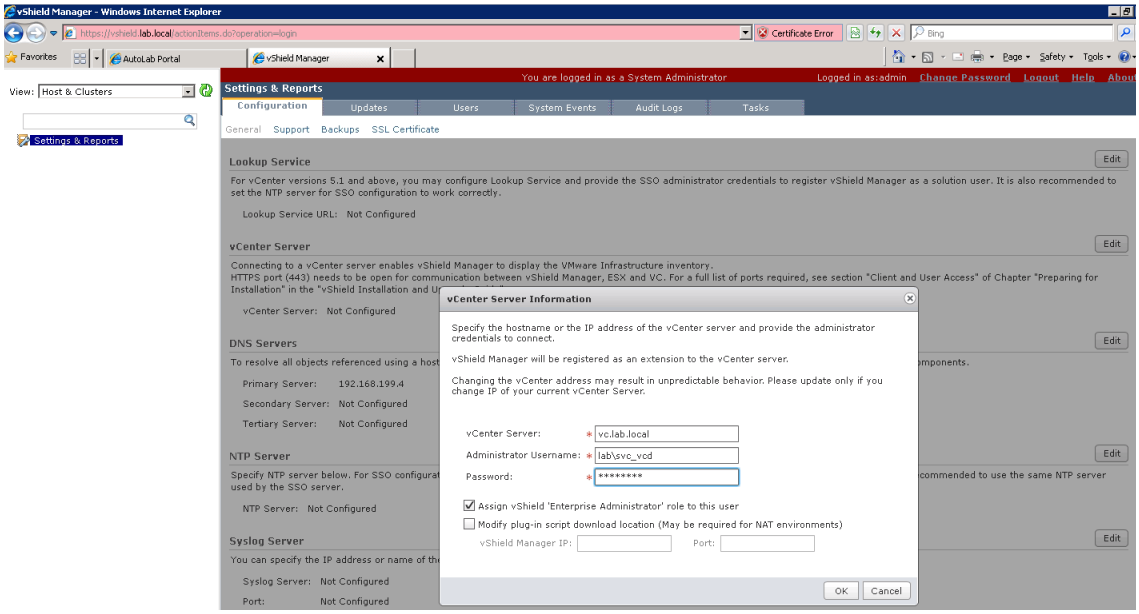


Once imported reduce the RAM configured on the VM, 512MB is a minimum, if you can run your ESXi hosts with 4GB of RAM then use 1GB for vShield.

Power on the VM and wait for it to boot, when the login prompt appears login as **admin** with password **default**. Then enter privileged mode by using the **enable** command, same password. Next run **setup** and enter the IP address information:

IP Address: 192.168.199.40
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.199.2
Primary DNS: 192.168.199.4
Secondary DNS: 192.168.199.4
DNS domains: lab.local

Apply the settings and log out of the console. Wait a couple of minutes for the services to restart then use a web browser to connect to https://vshield.lab.local, accept the certificate warning and logon as **admin** with password **default** again.



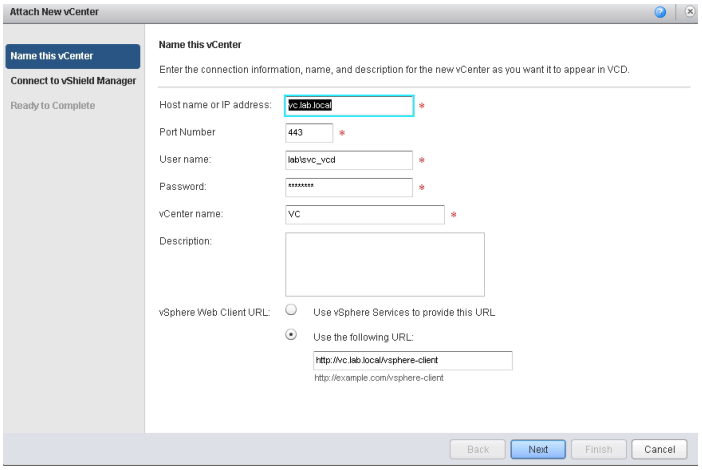
Edit the vCenter connection information, you will need to accept the vCenter server's thumbprint.

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.

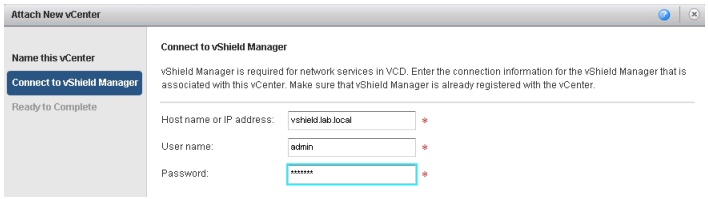
Name this vCenter:

Host name or IP address: vc.lab.local
Port number: 443
User name: SVC_vCD
Password: VMware1!
vCenter name: vc1
vSphere Web Client URL: http://vc.lab.local/vsphere-client



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.

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.

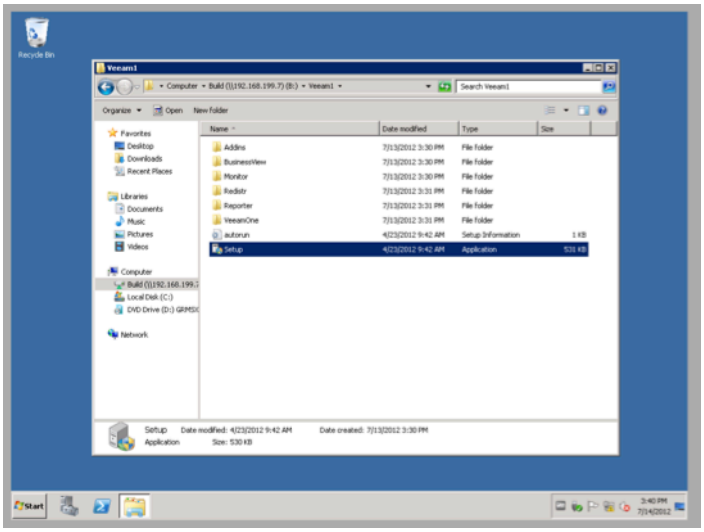
Only the Windows 208 R2 RTM install media will work, SP1 media will fail at the Windows component install phase of Windows setup.

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

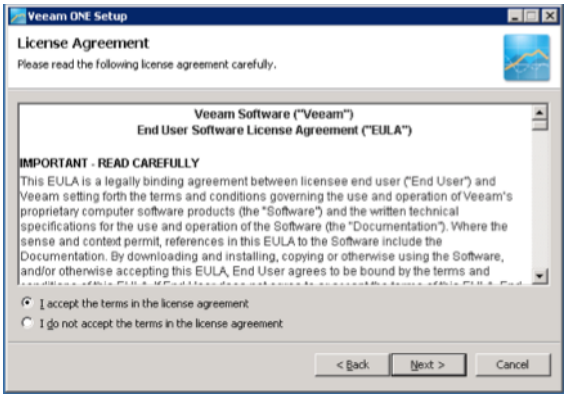
2. Logon to V1 as the Veeam service account user lab\svc_Veeam

3. On Build share locate **B:\Veeam1\setup.exe** & **Run As Administrator**

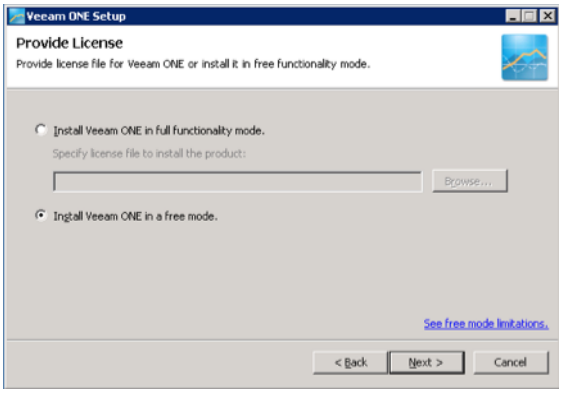


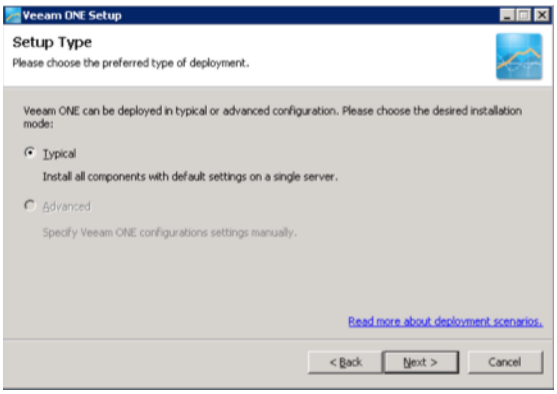
4. Select to install **Veeam ONE Server**

5. Click through the install wizard as usual

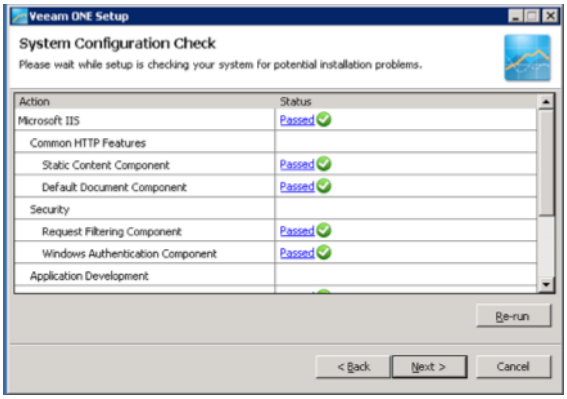


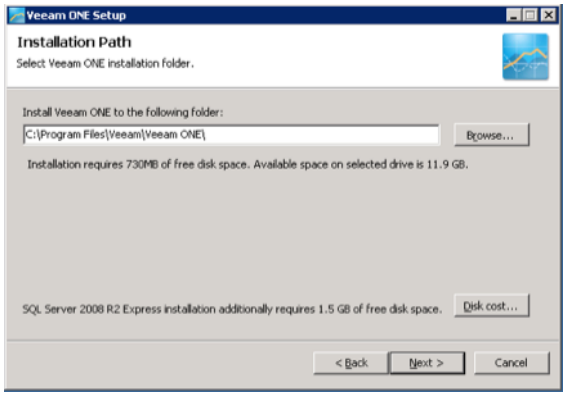
6. Since we're evaluating we will use the free edition, Select **Install Veeam ONE in a free mode** and click **Next**



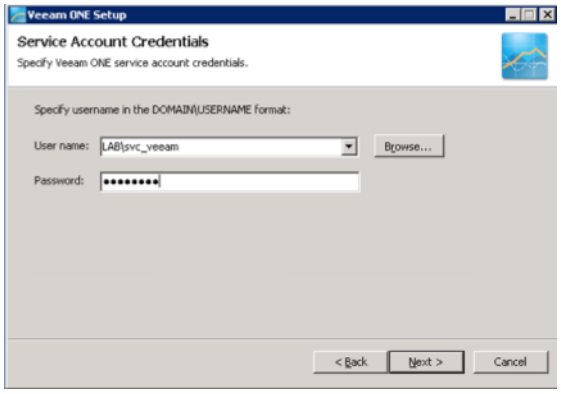


7. The Lab build has installed all of the required Windows components for you

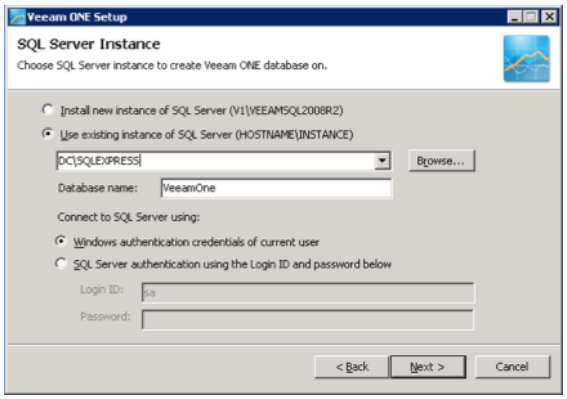


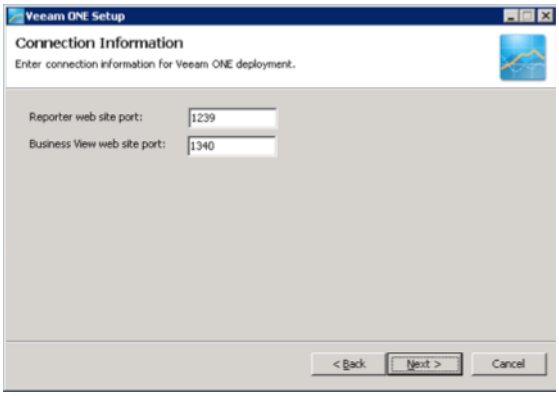


8. Enter the usual lab password **VMware1!** for the service account

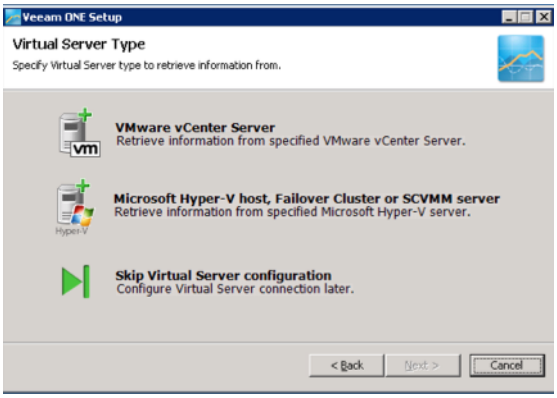


9. To minimise the RAM footprint we will use the SQL instance on the DC, the service account has rights to create the database so simply change to **Use existing instance of SQL Server** and enter the instance name of the SQL server on DC, **DC\SQLEXPRESS**

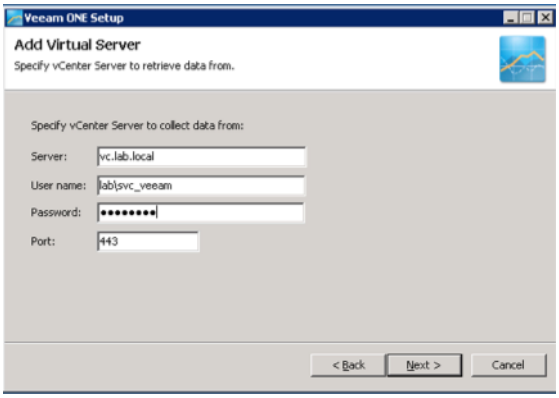


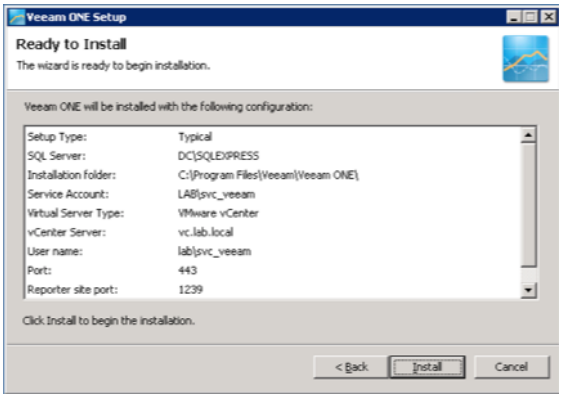


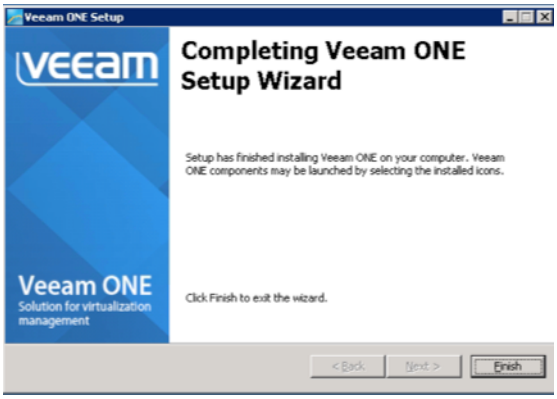
10. Click on VMware vCenter Server to add our vCenter



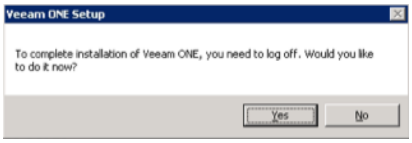
11. Enter the lab vCenter **vc.lab.local** and the Veeam service account credentials, username **lab\svc_veeam** and password **VMware1!**







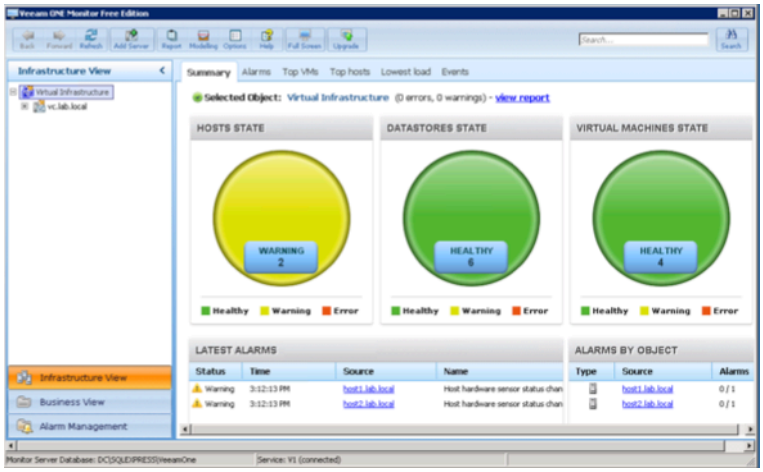
12. Once the install completes you will be prompted to log off, click **Yes**



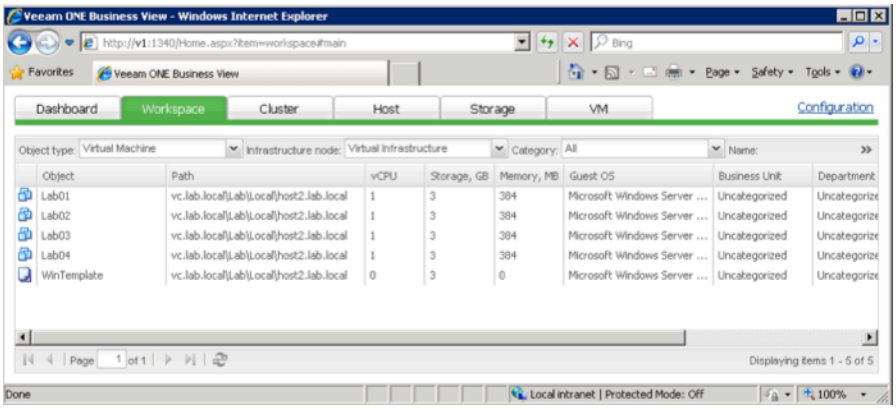
13. When you log back on you will find three new desktop shortcuts for the Veeam components

14. Use each shortcut to launch the components and make sure they operate.

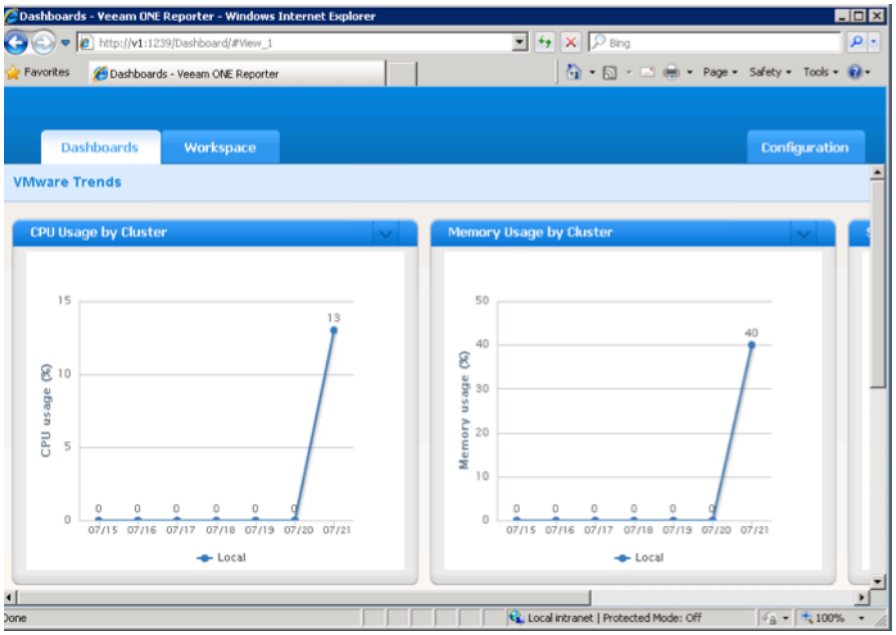
15. **Veeam ONE Monitor**



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

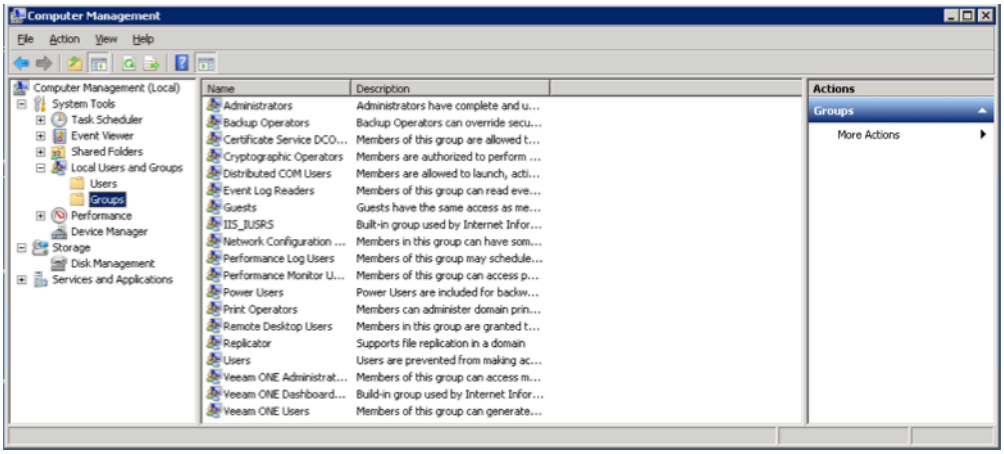


17. **Veeam One Reporter**, here the **VMware Trends** dashboard shows data immediately

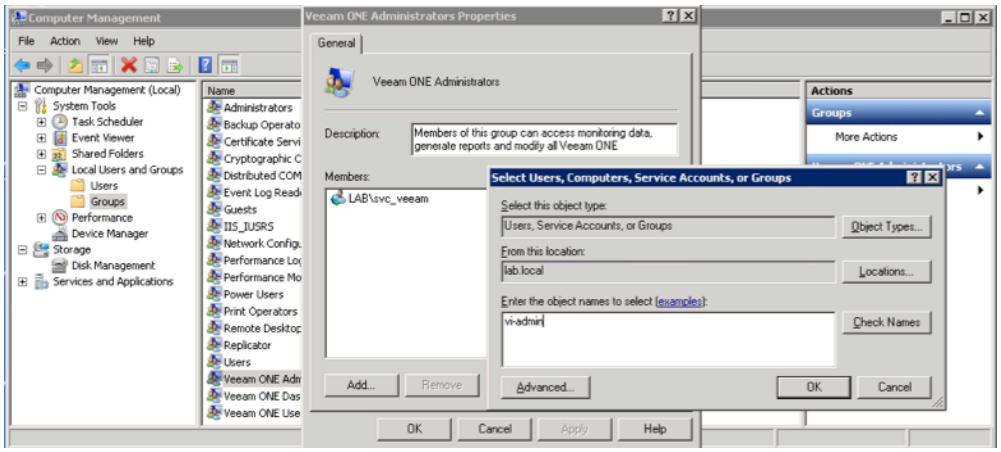


18. Next setup to allow the VI-Admin user access to the Veeam products. Form the Start Menu under **Administrative Tools** select **Computer Management**

19. Under **Local Users and Groups** select the **Groups** folder



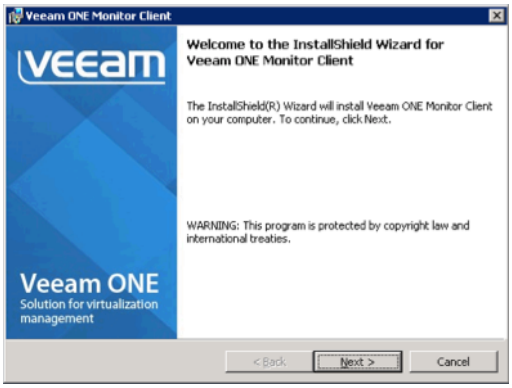
20. Double click the **Veeam ONE Administrators** group, click the **Add…** button, enter VI-Admin and click **OK** a couple of times, then close **Computer Management**

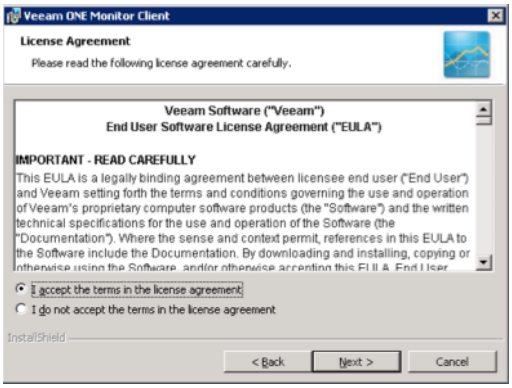


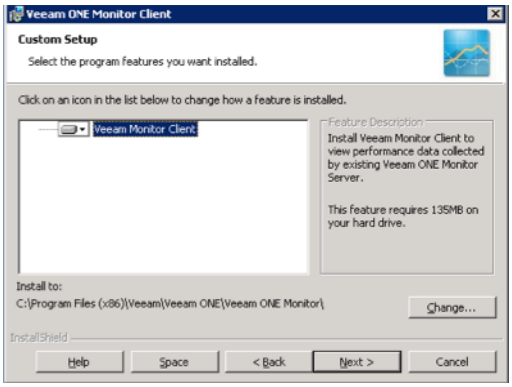
Client component install on VC

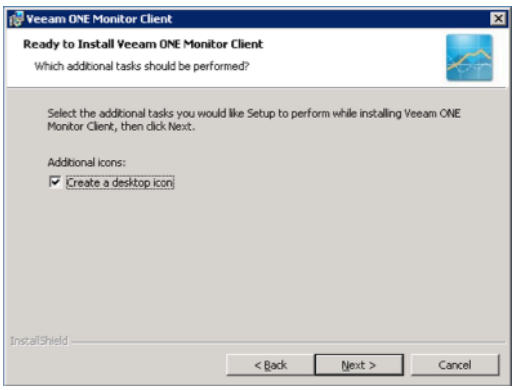
1. Logon to VC as VI-Admin
2. Again locate and **Run as Administrator** the **B:\Veeam1\setup.exe** from the Build share
3. This time choose **Veeam ONE Monitor Client**

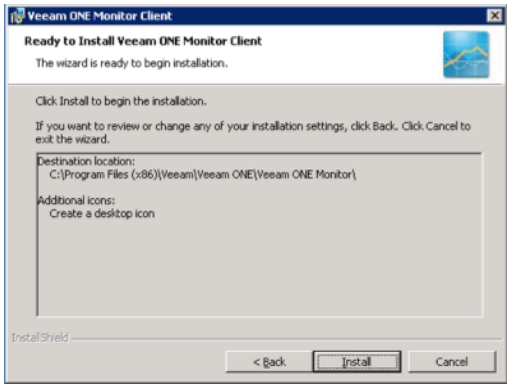
4. Click through the install wizard as usual

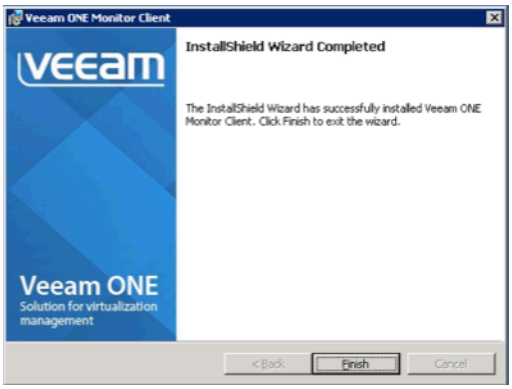

Veeam ONE Monitor Client
Welcome to the InstallShield Wizard for Veeam ONE Monitor Client
The InstallShield(R) Wizard will install Veeam ONE Monitor Client on your computer. To continue, click Next.
WARNING: This program is protected by copyright law and international treaties.
Veeam ONE
Solution for virtualization management
< Back
Next >
Cancel


Veeam ONE Monitor Client
License Agreement
Please read the following license agreement carefully.
Veeam Software ("Veeam")
End User Software License Agreement ("EULA")
IMPORTANT - READ CAREFULLY
This EULA is a legally binding agreement between licensee end user ("End User") and Veeam setting forth the terms and conditions governing the use and operation of Veeam's proprietary computer software products (the "Software") and the written technical specifications for the use and operation of the Software (the "Documentation"). Where the sense and context permit, references in this EULA to the Software include the Documentation. By downloading and installing, copying or
I accept the terms in the license agreement
I do not accept the terms in the license agreement
InstallShield
< Back
Next >
Cancel


Veeam ONE Monitor Client
Custom Setup
Select the program features you want installed.
Click on an icon in the list below to change how a feature is installed.
Veeam Monitor Client
Feature Description
Install Veeam Monitor Client to view performance data collected by existing Veeam ONE Monitor Server.
This feature requires 135MB on your hard drive.
Install to:
C:\Program Files (x86)\Veeam\Veeam ONE\Veeam ONE Monitor\
Change...
InstallShield
Help
Space
< Back
Next >
Cancel

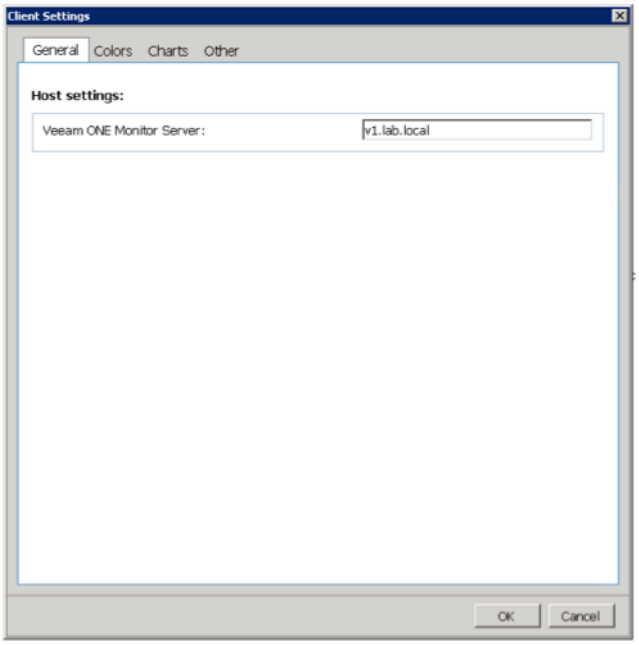

Veeam ONE Monitor Client
Ready to Install Veeam ONE Monitor Client
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing Veeam ONE Monitor Client, then click Next.
Additional icons:
Create a desktop icon
InstallShield
< Back
Next >
Cancel



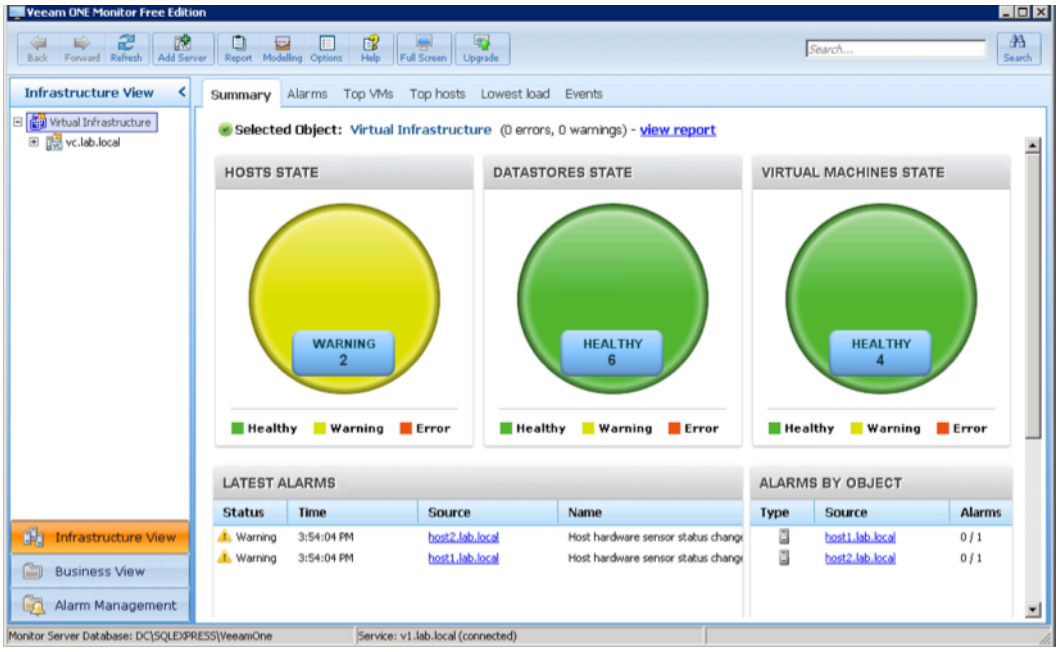


After the install completes you will have a desktop shortcut for **Veeam ONE Monitor**, the other two components are web services.

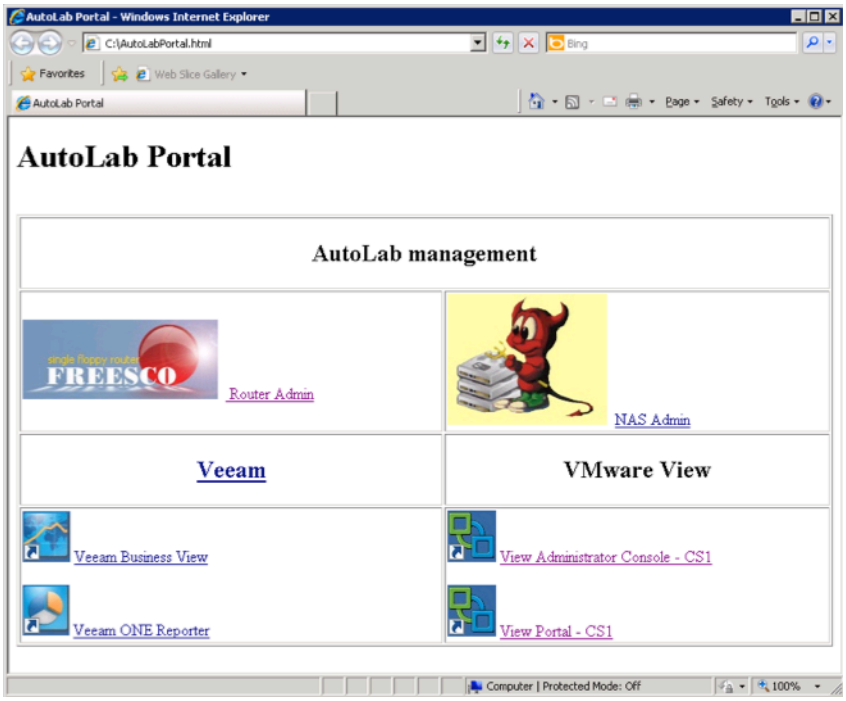
5. On the first run you will need to tell the monitor client where to find the server, enter **v1.lab.local** and click **OK**



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



7. 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.



8. 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

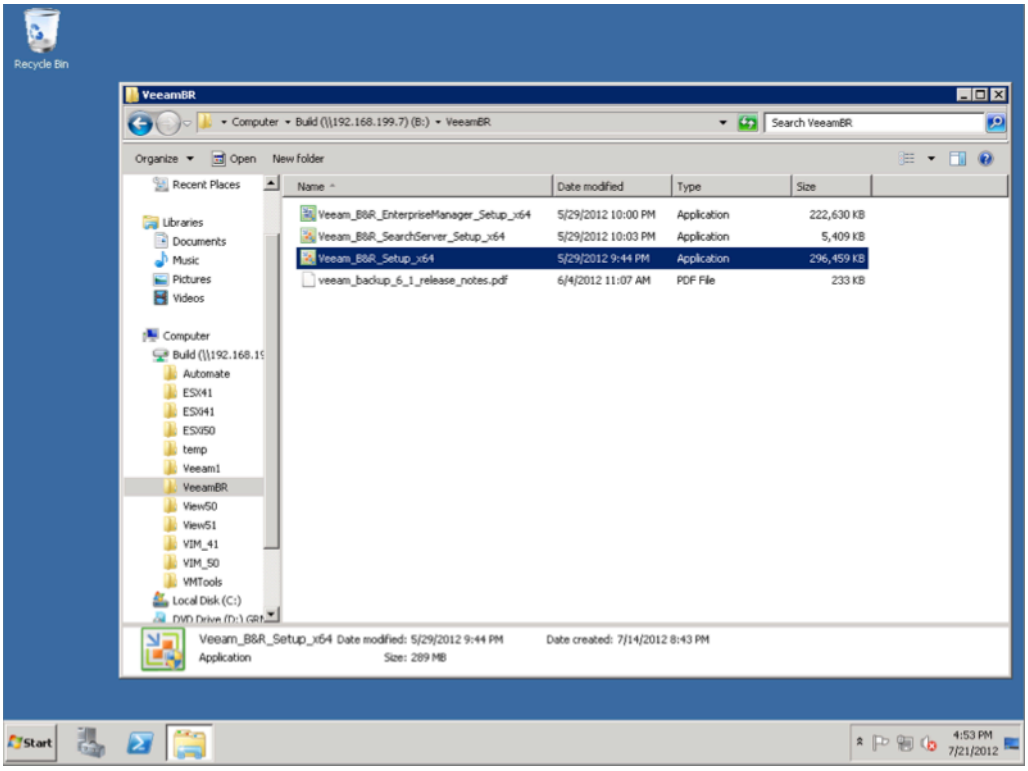
Only the Windows 208 R2 RTM install media will work, SP1 media will fail at the Windows component install phase of Windows setup.

Server component install on VBR

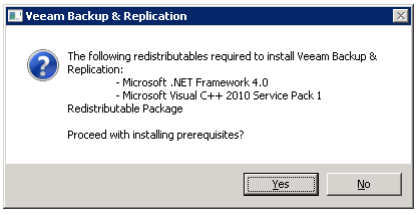
1. Build the VBR VM, this follows the usual boot from CDROM with floppy attached method used for all the Lab Windows VMs. You should have the basic AutoLab setup built with DC, VC and two ESXi servers. On a Workstation environment with 8GB of RAM you will need to shut down one of the ESXi servers to free some RAM.

2. Logon to VBR as the Veeam service account user lab\svc_veeam

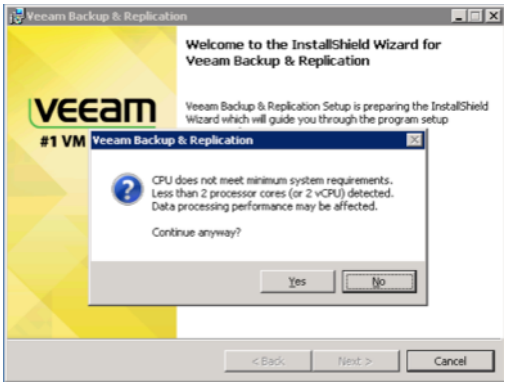
3. On the build share locate **B:\VeeamBR\Veeam_B&R_Setup_x64.exe** and **Run as Administrator**

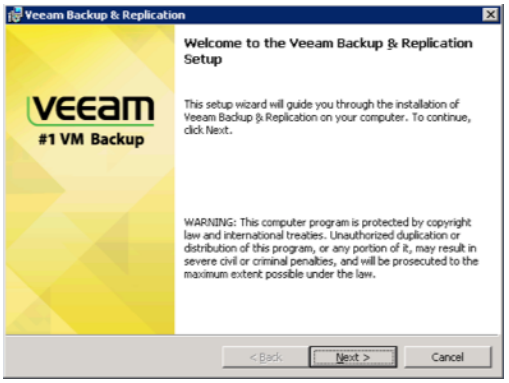


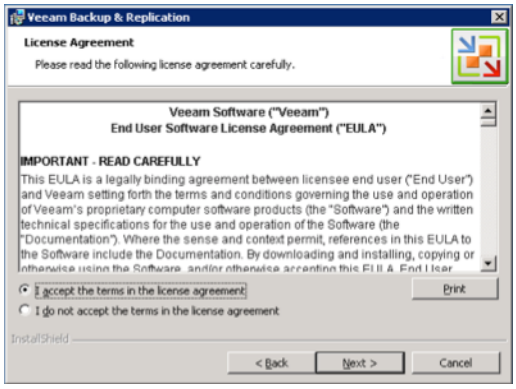
4. If you are installing Veeam Backup & Replication 6.5, you will see a warning about some pre-requisites, click **Yes** and the installer will install these for you.

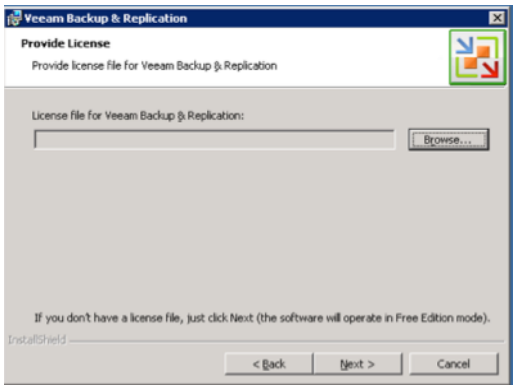


5. Accept the warning about vCPU count and proceed on through the wizard

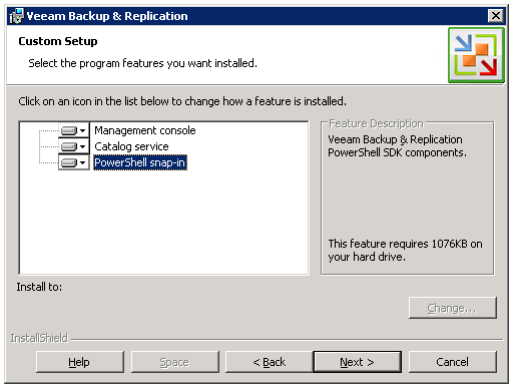




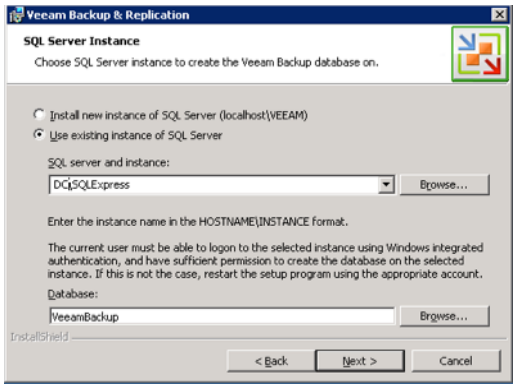




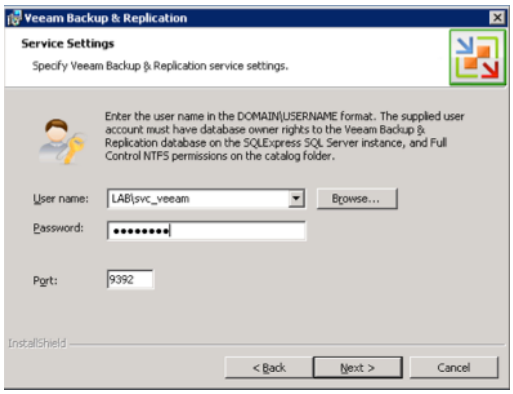
6. Since I'm a fan of PowerShell I included it's snapin

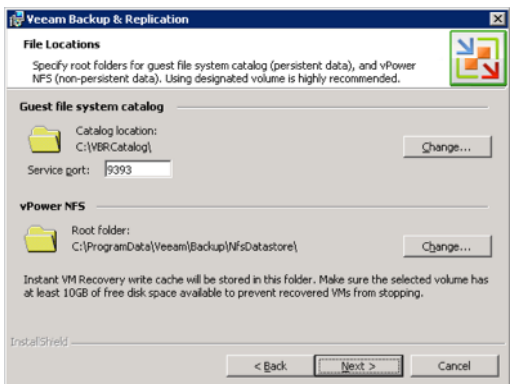


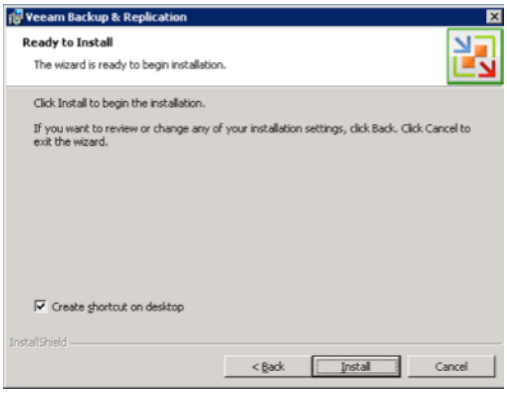
7. Select **Use existing instance of SQL Server** and enter the SQL server instance on DC, **DC\SQLEXPRESS**, leave the default database name

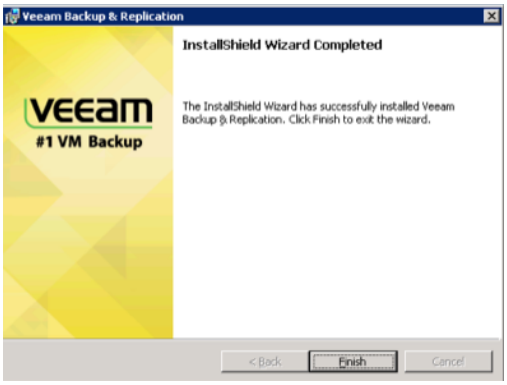


8. Enter the usual password, **VMware1!**, for the **svc_veeam** service account









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

10. 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