

Install & Configuring

UDA 2.0 to Deploy ESX 4.x.x

Created by Carl Thijssen

Document by Mike Laverick



Download and Install the UDA

Note:

At the time of writing the UDA is in a beta format. It has been well tested and should work. By the time the vSphere4 book is released it should no longer be a beta. With that in mind the file names used in here will undoubtedly change.

1. Download the ESX version of the UDA from <http://www.rtfm-ed.co.uk/downloads/uda20-beta-ovf.zip> to your management PC
2. **Right-click** and **extract** the **uda20-beta-ovf.zip** file
3. In vCenter select **File** in the menu and **Deploy OVF Template**
4. Select **Deploy from file** and use the **Browse** button to locate the .OVF file
5. **Click Next** to accept the description
6. Set the **UDA VM name** and **location in the inventory**
7. Select a **datastore** location
8. Select a **network** port group

Note:

Remember this appliance will be used to deploy ESX so you must put it on the same network as the Service Console network.

Occasionally, I find the .OVF format reports an error during the import process. If this happens to you might prefer to create the UDA virtual machine manually. You download and upload just the virtual disk files to the ESX host. On the RTFM website there is a zip file which just contains the virtual disks called `uda20.build2.zip`. Once the zip file is uploaded to your ESX host – you can use the command `tar` to unzip the virtual disks:

```
tar -xzf uda20-beta.zip
```

Followed by the `vmkfstools` command to rebuild them into the “thick” format

```
vmkfstools -i /root/uda20-beta/uda20_disk1.vmdk /vmfs/volumes/sanlun1/uda20-beta/uda20-beta.vmdk
```

Power On and First Run-Configuration

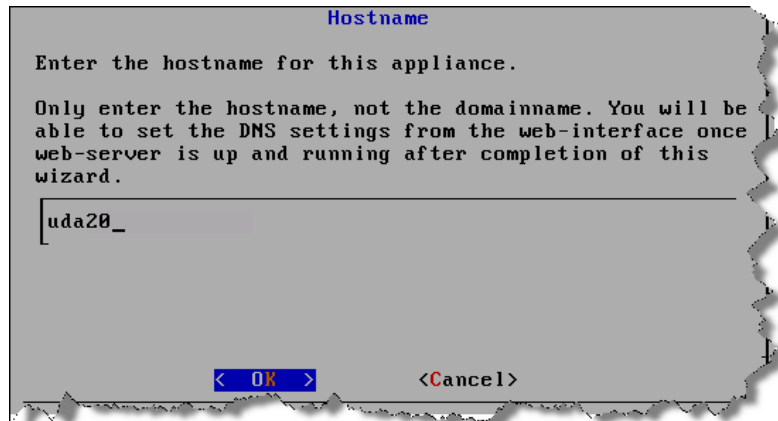
When you first power on the UDA it will boot to Linux, and inside Linux a small wizard runs to allow you to configure its various settings such as its:

- Hostname
- IP Address Settings
- DHCP Configuration (if selected)

Although you can use the tab key to navigate the wizard I’ve found using the cursor keys the best way.

1. **Power on the UDA**

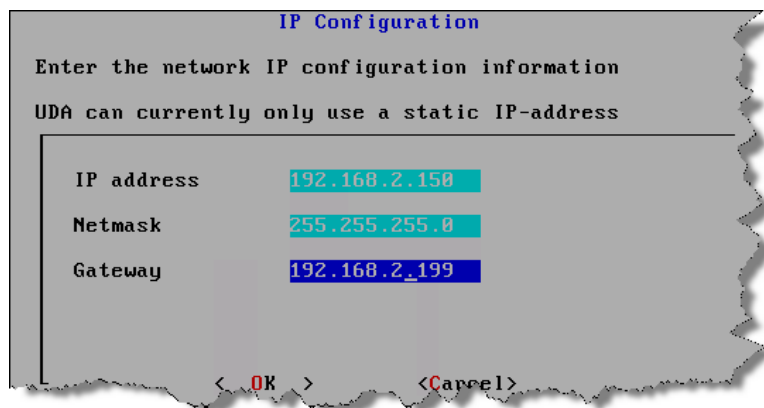
2. At the **Welcome Screen** press [ENTER]
3. In the **Hostname** dialog, type in the *hostname* for your UDA



Note:

Notice how the interface clearly states the hostname only, not FQDN.

4. Next set your **IP, Subnet Mask and Default Gateway**



Note:

As the UDA is both boot source and source of the ESX media, I used the same network range as my ESX hosts.

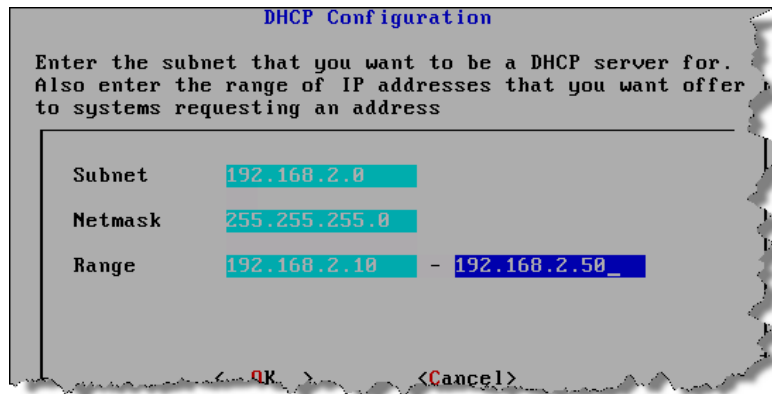
5. Next **enable the DHCP Service** in the list



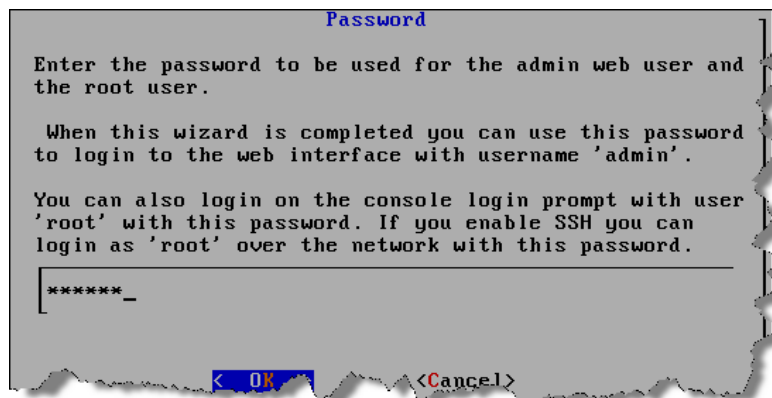
Note:

If you already have a DHCP server and merely wish the UDA server to be a PXE boot server – then skip this part. At the end I will describe how to make an existing Microsoft DHCP issue the IP address to your ESX hosts and point to the UDA as you may have corporate policies that enforce the use of authorized Active Directory DHCP Servers.

6. Next set the **Network ID, Subnet Mask and Starting and Ending Range for your DHCP Scope**



7. Next set your password for root and admin accounts



Note:

The UDA uses two user accounts – the root account for SSH and console logins, and then a more limited account called admin which is used to authenticate to the friendly web-page front-end that allows for high-level post-configuration.

8. **Confirm your password and the summary of the settings** you have provided

Note:

Once the wizard completes you can open up a web-browser on the UDA, login as admin with your password and you should see the welcome screen like so:



Note:

You may now wish to click the System menu option, to set your DNS server and DNS Domain. This will help using names rather than IP addresses in the storage aspect of the UDA

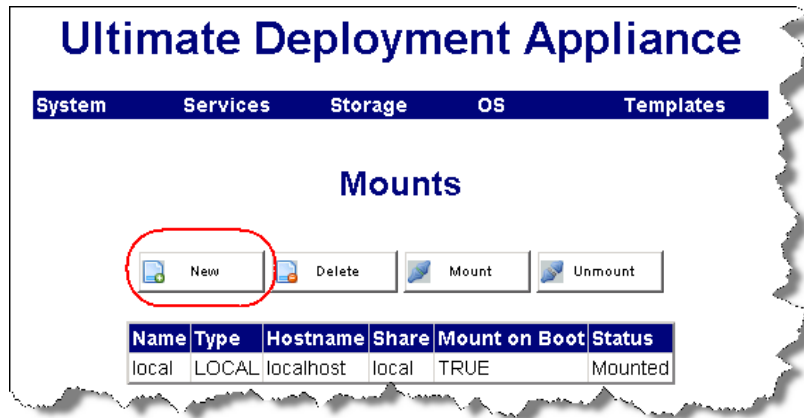
Post-Configuration of the UDA

The main post-configuration of the UDA lies in a three step process. Firstly, that's giving the UDA access to storage where your main ISO images are held – these can be virtual disk, Windows or NFS share. Secondly, once the storage has been accessed you can then “mount” the ESX4 ISO to the UDA, the UDA will automatically copy all the files required for a PXE boot from the disk to the UDA without you having to know anything about Linux, TFTP, DHCP or PXE. Lastly, you can create a template for installing ESX hosts or any other operating system that the UDA support. Templates form the outline or the basis of how a particular operating system will be installed – essentially it's the master file from which all scripted installs of ESX of a particular type can be made.

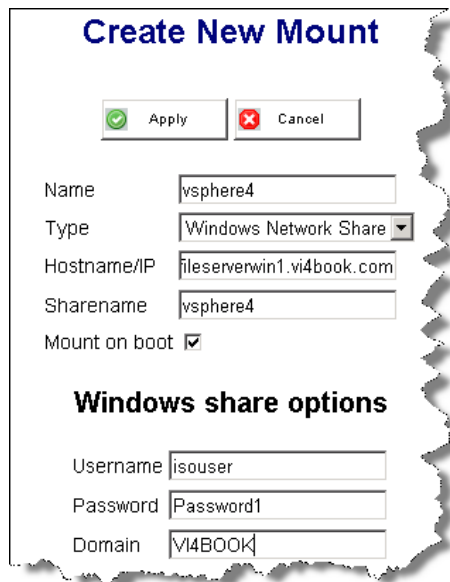
Accessing Storage (Windows Share)

If your ESX4 ISO resides on a Windows share this is how you gain access to the share via the UDA

1. Click the **Storage** menu option
2. Next click the **New** button



3. From the Type pull-down list, select **Windows Network Share**, and complete the dialog box like so:



Note:

I think most of this is self-explanatory. The field “name” is the friendly name used by the UDA to refer collectively to all these settings. It’s perhaps worth checking before you begin, that the UDA can ping the hostname, and the user specified has rights to the share, and the typical sort of IP and authentication troubleshooting you would do for Windows. If that does not help then you can resort to raw IP data as the UDA will accept that as well.

Add the OS (ESX4 Classic)

Next we need to configure an “operating system” for the UDA. It needs to know which operating system we are using – Windows, ESX3, ESX4 or GentOS. Once selected we can then indicate what flavor of that operating system we are using, ESX4.0.0, ESX4.0.1 and ESX4.5 and so on. This allows the UDA to mount any number of different distributions of a given operating system. Once you have browsed and selected the ISO in question the UDA will then “import” the critical boot files required for PXE booting to be successful for you.

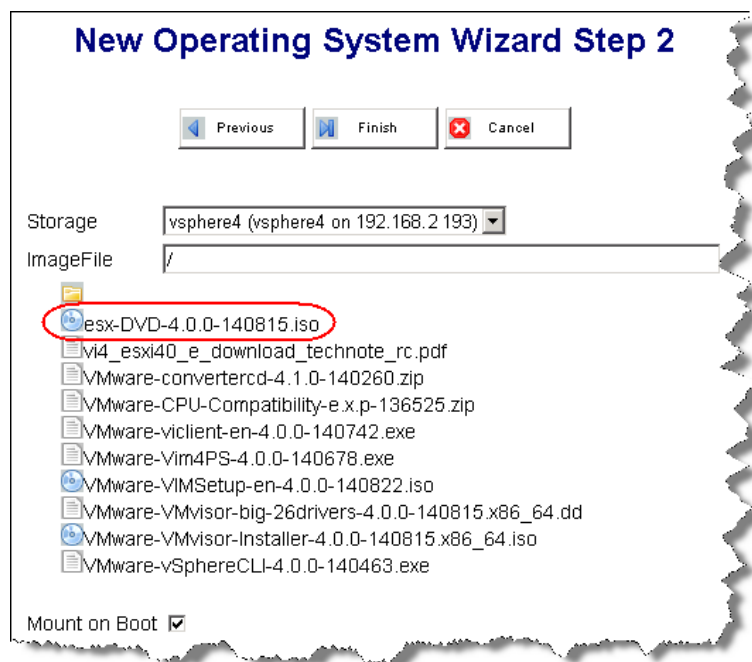
1. Click the **OS** menu option
2. Click the **New** button
3. **Select** from the pull-down list **VMware ESX Server 4.X.X**
4. Then **type in a unique flavor name** such as **esx4.0**



Note:

Flavor names can be anything you like, but you **MUST** set one for the UDA to work properly.

5. Click **Next** and then **select the ESX ISO in your mounting point**



6. Then click the **Finish** button

Note:

The UDA does **NOT** copy the entire CD from the network share – just the files required for PXE booting. The share will still need to be online and available for the UDA to function.

Add a New Template

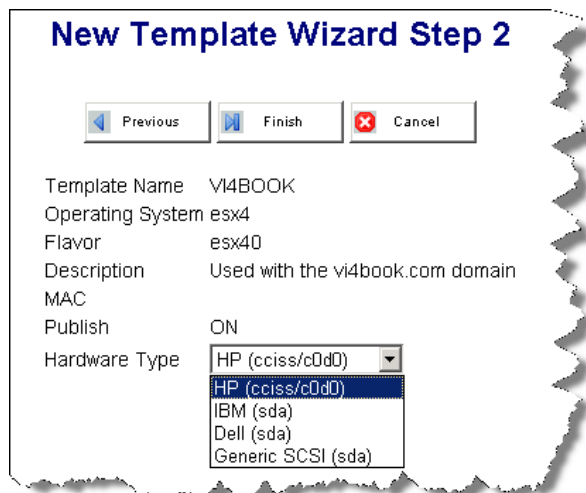
Part of the UDA takes a little time to explain but it's actually very simple. Each ESX host you have will need a text file to automate the installation – it's a scripted installation after all. However, can you

imagine having 30-40 different script files each with IP addresses, subnet masks and default gateways hard-coded to them? It would be a nightmare to maintain. It would be so much better to have one single “master” template which merely contains the install instructions with variables – and a separate file or “sub-template”, which would contain all those variables. Wouldn’t it be great if you could define (within reason) any variable you like – and for the system to automatically build a menu to select from once the main PXE boot process has completed? Well, I’m pleased to say that the UDA does all that for you! In fact in the ESX version of the UDA we have created a sample “master” template which does a complete scripted installation for you. All you need to do is change the variables.

1. Click the **Template** menu option
2. **Type in the “master” template name** such as VI4BOOK or RTFM, then from the **Operating System pull-down list select VMWare ESX 4.X.X**, and then **select the flavor used with this template.**



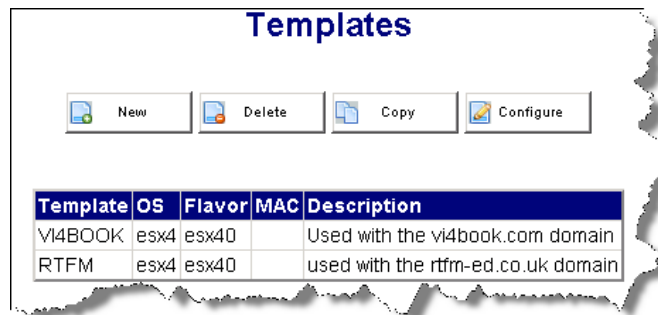
3. Click **Next**, and then select the hardware type you are using



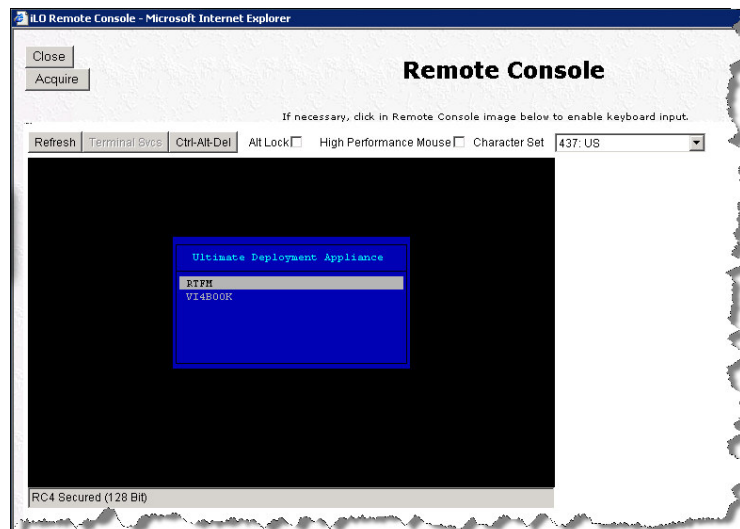
WARNING:

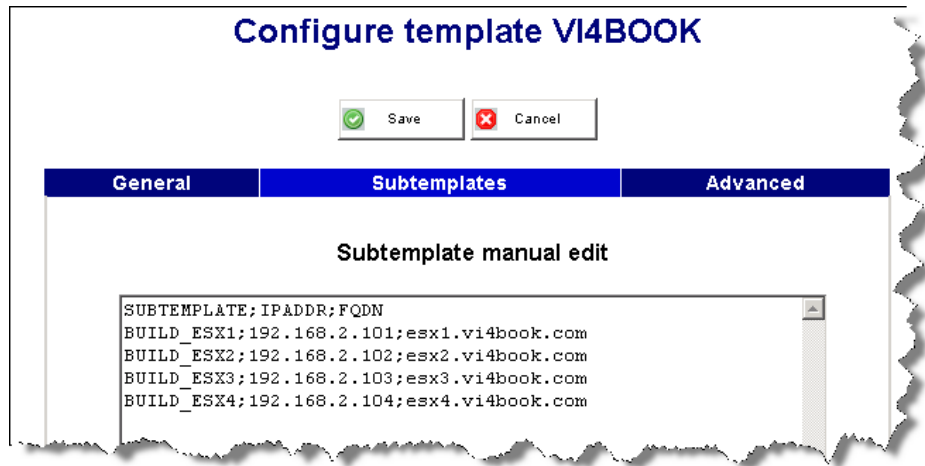
Remember HP servers with internal RAID controllers assume /dev/cciss/c0d0 for the first controller and the first disk. Dell and IBM servers will normally default to /dev/sda for local storage. The UDA will wipe any disks you set it to install ESX to, so it's important that /dev/sda isn't your Exchange Mail Store. Am I making myself clear here? If this worries you at all then either disable the ESX host's HBAs in the BIOS or use the masking features of your storage array to make sure that the ESX host does not see any shared storage during the installation.

I repeated this process to create a master template from my ESX hosts using the rtfm-ed.co.uk domain name:



These names VI4BOOK and RTFM will build a menu system inside the UDA which will be displayed whenever you carry out a PXE boot. The screen grabs below from a HP ILO Card demonstrate what the "end-user" will see





Note:

The variable SUBTEMPLATE is hard-coded and it will create a sub-menu inside my main menu of VI4BOOK with options to build_esx1 and so on. Remember the sky is the limit here; you can create as many variables as you need for any purpose. In my live UDA environment which I use to build either my rtfm-ed.co.uk hosts or my vi4book.com hosts I use these variables:

```
SUBTEMPLATE;  
IPADDR;  
HOSTNAME;  
VMOTIONIP;  
VMKISCISI;  
VMKISCISI2NDCOSPORT;  
HAHEARTBEAT
```

The UDA is good to go and if you wanted you could jump in with both feet and give it a whirl. OR, alternatively, you could hang fire and learn more about the settings in the “master” template that automates the install.

Scripted Install File Overview with Advanced %post Scripting

Below is a sample weasel script used by the UDA to automate the install. Even if you decide not to use the UDA this script would still work if you just replaced the [VARIABLES] with actual values. Most of this stuff is common sense so I will endeavor (believe it or not!) not to patronize you as we go through it. My comments are in *italics* in an “in-line” format to describe each part or add an additional explanation. A copy of the sample script can be downloaded from:

<http://www.rtfm-ed.co.uk/downloads/esx4.cfg>

Additionally, this is the sub-template file I use with it:

<http://www.rtfm-ed.co.uk/downloads/subtemplate.cfg>

VMware ESX4 template Kickstart file

VMware Specific Commands

vmaccepteula

You must include this or the script will fail.

Timezone

timezone Europe/London

Keyboard

keyboard uk

Encrypted root password: password

auth --enablshadow --enablemd5

rootpw --iscrypted \$1\$5a17\$In5zYe6YsCty76AycpGaf/

Md5Sum represents a safe way of saving passwords in plain text files without disclosing the real password. There are a number of websites and software which can generate md5sum passwords for you. However, an easy method is to create a user and set a password and then cat /etc/passwd to view the md5sum.

Reboot after install

reboot

If you don't include this the installer just stops prompting you to do a manual reboot/

Network install type

network --device=vmnic0 --bootproto=static --ip=[IPADDR] --netmask=255.255.255.0 --gateway=192.168.2.199 --nameserver=192.168.2.199 --hostname=[HOSTNAME].vi4book.com --addvmportgroup=0

This is quite straightforward I think – with vmnic0 now replacing eth0 as a way of indicating both which network card to use during the install, and also which network card will be mapped to vSwitch0. The --addvmportgroup=0 value prevents the installer from creating a virtual machine port group called “VM Network” on vSwitch0.

Firewall settings

firewall --disabled

This controls the firewall of the installer, not the ESX firewall which would be controlled by esxcfg-firewall.

Clear Partitions

clearpart --drives=[DISKTYPE] --overwritevmfs

This is quite dangerous as it destroys whatever is on the [DISKTYPE] which could be either cciss/c0d0 if it a HP Proliant Server or /dev/sda if its an IBM or Dell Server. Overwritevmfs forces the clearing of

all partitions even if they are VMFS volumes. This is also quite dangerous if you touch the wrong disk/volume/LUN

BootLoader (The user has to use grub by default)

```
bootloader --location=mbr --driveorder=[DISKTYPE]
```

This tells the installer where to put the MBR record. Without it you will receive a benign warning that it will install GRUB to the disk selected for the partition table.

Manual Partitioning

```
part /boot --fstype=ext3 --size=250 --ondisk=[DISKTYPE]
part None --fstype=vmkcore --size=100 --ondisk=[DISKTYPE]
part local_[HOSTNAME] --fstype=vmfs3 --size=20000 --ondisk=[DISKTYPE] --grow
```

This creates 4 partitions on [DISKTYPE] the boot partition, vmkcore partition and vmfs volume called local_[HOSTNAME]. The vmfs volume would not be 20MB but be the remainder of the volume.

```
virtualdisk vd1 --size=15000 --onvmfs=local_[HOSTNAME]
part swap --fstype=swap --size=1600 --onvirtualdisk=vd1 |
part /opt --fstype=ext3 --size=2048 --onvirtualdisk=vd1
part /tmp --fstype=ext3 --size=2048 --onvirtualdisk=vd1
part /home --fstype=ext3 --size=2048 --onvirtualdisk=vd1
part / --fstype=ext3 --size=5120 --onvirtualdisk=vd1 --grow
```

This partitions the first virtual disk (vd1) of the Service Console. This virtual disk is created as 15GB on the local_[HOSTNAME]. So whatever the VMFS volume is called it must have the correct name for the VMDK file to be created. We then proceed to create partitions within the virtual disk. The last partition / would not be 5120 but would grow to be the remainder of the virtual disk.

%packages

This allows you to install other packages aside from the base install. I don't use this in my scripted installation.

%post --interpreter=bash

%post is always at the end of the install process. Setting the --interpreter=bash prevents a benign warning which would state if the installer is defaulting to BASH (Bourne Again SHell). If you don't include this the commands below would still be processes as long as they were recognizable by the BusyBox or the VMkernel.

Create vSwitch1 with a port group of internal

```
esxcfg-vswitch -a vSwitch1
esxcfg-vswitch -A internal-[HOSTNAME] vSwitch1
```

This uses the esxcfg-vswitch command to create a vSwitch and Portgroup. Notice how esxcfg-vswitch -L is not being used, so network cards are NOT linked to the virtual switch.

Create a vSwitch2 with a port group of Production using vmnic1

```
esxcfg-vswitch -a vSwitch2
esxcfg-vswitch -A production vSwitch2
esxcfg-vswitch -L vmnic1 vSwitch2
```

This creates a simple virtual switch with one network card and therefore no fault tolerance.

Create a vSwitch3 with a port group of VMotion using vmnic2

```
esxcfg-vswitch -a vSwitch3
esxcfg-vswitch -A vmotion vSwitch3
esxcfg-vswitch -L vmnic2 vSwitch3
esxcfg-vmknic -a vmotion -i [VMOTIONIP] -n 255.255.255.0
```

This creates a vSwitch valid for VMotion using the esxcfg-vmknic command to set the IP address and subnet mask valid for the network.

Enable VMotion

```
vmware-vim-cmd hostsvc/vmotion/vnic_set vmk0
vmware-vim-cmd hostsvc/net/refresh
```

On it is own esxcfg-vswitch cannot enable the VMotion feature. We can do this with a high-level utility called vmware-vim-cmd. It is a supported utility and is actually a wrapper to an even more powerful utility called vimsh. Unfortunately, vimsh is not official supported by VMware – however it is exceedingly powerful if you wish to do high-level changes. Both vmware-vim-cmd and vimsh have been well documented at the [xtravirt.com](http://knowledge.xtravirt.com) website:

<http://knowledge.xtravirt.com/white-papers/scripting.html>

Create add with a port group for iSCSI/iSCSI-COS using vSwitch3

```
esxcfg-vswitch -a vSwitch4
esxcfg-vswitch -A ipstorage vSwitch4
esxcfg-vswitch -L vmnic3 vSwitch4
esxcfg-vmknic -a ipstorage -i [VMKISCSI] -n 255.255.255.0
esxcfg-vswitch -A iscsi-cos vSwitch4
esxcfg-vswif -a vswif1 -p iscsi-cos -i [VMKISCSI2NDCOSPORT] -n 255.255.255.0
```

This creates a vSwitch valid for enabling and using the ESX Software iSCSI initiator. If you remember from the storage chapter both the Service Console and the VMkernel need a valid network path to the iSCSI Target for LUN Discovery, CHAP and I/O to work. Here esxcfg-vwif is used to set the IP and Subnet Mask of the Service Console port (vwif1)

HA Heartbeat Port on VMotion Switch

```
esxcfg-vswitch -A ha-heartbeat vSwitch3
esxcfg-vswif -a vswif2 -p ha-heartbeat -i [HAHEARTBEAT] -n 255.255.255.0
```

This creates a third Service Console port to allow a HA Heartbeat network to function.

Set-up iSCSI Software Emulator

```
esxcfg-swiscsi -e  
vmkiscsi-tool -D -a 172.168.3.100 vmhba34  
esxcfg-swiscsi -s
```

This enables the ESX iSCSI Software Initiator connecting to an iSCSI Target with the IP of 172.168.3.100

Connect to a NAS...

```
esxcfg-nas -a nas-iso -o nfs.vi4book.com -s /iso
```

This mounts an NFS share to the ESX host. The name "nas-iso" is the friendly "datastore" name that users will see in the vSphere Client, whereas /iso is the actual name of the NFS Share

VLAN Example

```
# esxcfg-vswitch -a vSwitch2  
# esxcfg-vswitch -A accounts vSwitch2  
# esxcfg-vswitch -A rnd vSwitch2  
# esxcfg-vswitch -A sales vSwitch2  
# esxcfg-vswitch -L vmnic4 vSwitch2  
# esxcfg-vswitch -L vmnic5 vSwitch2  
# esxcfg-vswitch -v 10 -p accounts vSwitch2  
# esxcfg-vswitch -v 20 -p rnd vSwitch2  
# esxcfg-vswitch -v 30 -p sales vSwitch2
```

This is a sample vSwitch with multiple portgroups enabled for VLAN Tagging and multiple network cards for fault-tolerance. Accounts, rnd, and sales aren't great port group names – but I've used them to make sure it's clear that -v 10 sets the VLAN value, and that accounts, rnd and sales are just friendly port group names.

Add 2nd/3rd DNS settings

```
echo nameserver 192.168.3.200 >> /etc/resolv.conf  
echo nameserver 192.168.3.201 >> /etc/resolv.conf
```

The installer does not accept more than one DNS server, so here I'm using the ECHO command to pass two IP addresses for my secondary and territory DNS servers.

Create a local user for SSH Access - Default password is password

```
useradd -p '$1$Rg69B9QA$JUtgStBrjNFbyzyP9zTsf0' -c "Mike Laverick" lavericm
```

This creates a user called "lavericm" with a password of "password" and friendly name of "Mike Laverick". There should be no real need to create users on an ESX host because you can use WinBind which is part of the Samba Project to allow users to login using their LDAP credentials. Additionally, you can use "sudo" to prevent any need to disclose the ESX "root" password. In fact there's only one time when you have to disclose the root account – and that when you add ESX into vCenter.

DANGEROUS: Allow ROOT access using SSH

```
sed -e 's/PermitRootLogin no/PermitRootLogin yes/' /etc/ssh/sshd_config >
```

```
/etc/ssh/sshd_config.new
mv -f /etc/ssh/sshd_config.new /etc/ssh/sshd_config
service sshd restart
```

In production systems I would not lower the security to allow the root to directly SSH (or PuTTY into) the ESX host. You would have no traceability of who did what and when. However, in certain lab environments where security concerns may be of less significance you can weaken the security. SED is a text manipulation tool which can search for and replace lines of text within a text file. It does that by searching an existing file, and outputting the change to a new file. We can then take this new file and overwrite the original, using the service command to restart SSHD to have those changes take effect.

Enable the SSH client (Out/From an ESX hosts)

```
esxcfg-firewall -e sshClient
```

I quite like the ability to SSH and SCP from one ESX host to another, so on the firewall I enable the sshClient which opens port 22 outbound. SSH on port 22 is already open inbound on the firewall.

Enabling NTP Time Configuration

```
echo restrict 127.0.0.1 > /etc/ntp.conf
echo restrict default kod nomodify notrap noquery nopeer >> /etc/ntp.conf
echo server 0.uk.pool.ntp.org >> /etc/ntp.conf
echo server 1.uk.pool.ntp.org >> /etc/ntp.conf
echo server 2.uk.pool.ntp.org >> /etc/ntp.conf
echo server 3.uk.pool.ntp.org >> /etc/ntp.conf
echo fudge 127.127.1.0 stratum 10 >> /etc/ntp.conf
echo driftfile /var/lib/ntp/drift >> /etc/ntp.conf
```

Create the Step-Tickers File

```
echo server 0.uk.pool.ntp.org >> /etc/ntp/step-tickers
echo server 1.uk.pool.ntp.org >> /etc/ntp/step-tickers
echo server 2.uk.pool.ntp.org >> /etc/ntp/step-tickers
echo server 3.uk.pool.ntp.org >> /etc/ntp/step-tickers
```

Handle the Service Management

```
esxcfg-firewall -e ntpClient
service ntpd start
chkconfig --level 3 ntpd on
hwclock --systohc
```

As you can see this enables time synchronization for the ESX host. This is well documented on many Linux forums and I have nothing to add that hasn't been said countless times before.

SSH Legal Message...

```
echo >> /etc/banner
echo This is a private system. >> /etc/banner
```

```
echo Do not attempt to login unless you are an authorized user. >> /etc/banner
echo Any authorized or unauthorized access and use, may be monitored >> /etc/banner
echo and can result in criminal or civil prosecution under applicable law. >> /etc/banner
echo >> /etc/banner
echo The United Kingdom of England, Ireland, Scotland and Wales >> /etc/banner
echo Computer Misuse Act 1990 >> /etc/banner
echo http://www.opsi.gov.uk/acts/acts1990/Ukpga_19900018_en_1.htm >> /etc/banner
echo >> /etc/banner
echo Banner /etc/banner >> /etc/ssh/sshd_config
```

Here I am creating a standard logon warning when anyone connects with SSH. I borrowed the disclaimer from a HP ILO Login Page, and looked up the legislation in the UK that governs the Misuse of Computers. I would recommend you consult your legal advisor for an appropriate legally binding message relative to your location in the world.

Enabling Microsoft Windows DHCP with UDA

In some environments it may not be possible to use the UDA's built-in DHCP daemon. It is possible to disable the DHCP Service on the UDA, and configure a Microsoft DHCP server to take over the role. This is quite an easy configuration change:

To Disable DHCP On the UDA

1. Logon as Admin on the UDA's web-admin tool
2. Click the **Services** link
3. Select the **DHCP** link
4. Click the **Configure** button
5. Disable the option for **Start DHCP on boot**
6. Click the **Apply** button
7. Click the **Services** link again
8. Select the **DHCP** link again
9. Click the **Stop** button

To Enable DHCP On the Windows DHCP

1. Configure the following **Scope Options** or **Server Options**:
2. **Enable** the option **066 Boot Server Host Name**, and set the empty string value to be the **IP address of your UDA**
3. **Enable** the option **067 Bootfile Name**, and set the empty string to be, **pxelinux.0**