Red Hat Reference Architecture Series

Migrating from VMware ESXi 5 to Red Hat Enterprise Virtualization (RHEV) 3

John Herr, Senior Software Engineer
RHCA, RHCVA

Version 1.0
June 2012

redhat.

1801 Varsity Drive™
Raleigh NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

The GPG fingerprint of the security@redhat.com key is:
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Send feedback to refarch-feedback@redhat.com

Table of Contents

1 Executive Summary

Virtualizing infrastructure, development, and other production servers has quickly become a requirement for many companies and organizations due to the potential cost savings provided by virtualization. Cost saving can be noticed directly by reducing the number of physical servers required to be purchased or indirectly through energy and cooling savings.

Many virtualization solutions lack the features necessary to allow them to be easily deployed and maintained in enterprise environments. Other solutions that may be feature rich are very costly to purchase and implement.

At a 60-80 percent lower cost than other virtualization solutions and without compromising features, Red Hat Enterprise Virtualization provides a robust and manageable environment for virtualization. Red Hat Enterprise Virtualization is packed with features such as high availability, live migration, automatic balancing of workloads, and more.

The underlying Open Source technologies that comprise Red Hat Enterprise Virtualization are well tested and designed to meet the demands of today's mission critical enterprise environments. Virtual machines from other virtualization solutions can be migrated to Red Hat Enterprise Virtualization securely and with confidence.

This paper describes the process of migrating virtual machines from a VMware ESXi server to a Red Hat Enterprise Virtualization server using the Red Hat Enterprise Virtualization virtual-to-virtual migration tools. The goal is to demonstrate how to perform and verify functionality for the migrated virtual machines. **`virt-v2v`** is used as the primary tool to perform virtual machine migration for the following guest Operating Systems running on a VMware ESXi host:

- Windows 7
- Windows 2008 R2
- Red Hat Enterprise Linux 6

2 Red Hat Enterprise Virtualization

2.1 RHEV Hypervisor

A hypervisor is a computer software platform that allows multiple “guest” operating systems to run concurrently on a host computer. The guest virtual machines interact with the hypervisor which translates guest I/O and memory requests into corresponding requests for resources on the host computer.

Running fully virtualized guests, i.e., guests with unmodified guest operating systems, used to require complex hypervisors and previously incurred a performance penalty for emulation and translation of I/O and memory requests.

Over the last few years chip vendors Intel and AMD have been steadily adding CPU features that offer hardware enhancements to support virtualization. Most notable are:

1. First-generation hardware assisted virtualization: Removes the requirement for hypervisor to scan and rewrite privileged kernel instructions using Intel VT (Virtualization Technology) and AMD's SVM (Secure Virtual Machine) technology.

2. Second-generation hardware assisted virtualization: Offloads virtual to physical memory address translation to CPU/chip-set using Intel EPT (Extended Page Tables) and AMD RVI (Rapid Virtualization Indexing) technology. This provides significant reduction in memory address translation overhead in virtualized environments.

3. Third-generation hardware assisted virtualization: Allows PCI I/O devices to be attached directly to virtual machines using Intel VT-d (Virtualization Technology for directed I/O) and AMD IOMMU. Also, SR-IOV (Single Root I/O Virtualization) which allows special PCI devices to be split into multiple virtual devices. This provides significant improvement in guest I/O performance.

The great interest in virtualization has led to the creation of several different hypervisors. However, many of these pre-date hardware-assisted virtualization, and are therefore some-what complex pieces of software. With the advent of the above hardware extensions, writing a hypervisor has become significantly easier and it is now possible to enjoy the benefits of virtualization while leveraging existing open source achievements to date.

Red Hat Enterprise Virtualization uses the Kernel-based Virtual Machine (KVM)[1], which turns Linux into a hypervisor. Red Hat Enterprise Linux 5.4 provided the first commercial-strength implementation of KVM, which is developed as part of the upstream Linux community. RHEV 3.0 uses the RHEL 6 KVM hypervisor, and inherits performance, scalability and hardware support enhancements from RHEL 6.

1 http://www.redhat.com/promo/qumranet/

2.2 Red Hat Enterprise Virtualization

Virtualization offers tremendous benefits for enterprise IT organizations – server consolidation, hardware abstraction, and internal clouds deliver a high degree of operational efficiency.

Red Hat Enterprise Virtualization (RHEV) combines the KVM hypervisor (powered by the Red Hat Enterprise Linux kernel) with an enterprise grade, multi-hypervisor management platform that provides key virtualization features such as live migration, high availability, power management, and virtual machine life cycle management. Red Hat Enterprise Virtualization delivers a secure, robust virtualization platform with unmatched performance and scalability for Red Hat Enterprise Linux and Windows guests.

Red Hat Enterprise Virtualization consists of the following two components:

- **RHEV Manager (RHEV-M):** A feature-rich virtualization management system that provides advanced capabilities for hosts and guests.
- **RHEV Hypervisor:** A modern, scalable, high performance hypervisor based on RHEL KVM. It can be deployed as RHEV-H, a small footprint secure hypervisor image included with the RHEV subscription, or as a RHEL server (purchased separately) managed by RHEV-M.

A **host** is a physical server which provides the CPU, memory, and connectivity to storage and networks that are used for the virtual machines (VM). The local storage of the standalone host is used for the RHEV-H executables along with logs and enough space for ISO uploads.

A **cluster** is a group of hosts of similar architecture. The requirement of similar architecture allows a virtual machine to be migrated from host to host in the cluster without having to shut down and restart the virtual machine. A cluster consists of one or more hosts, but a host can only be a member of one cluster.

A **data center** is a collection of one or more clusters that have resources in common. Resources that have been allocated to a data center can be used only by the hosts belonging to that data center. The resources relate to storage and networks.

A **storage domain** is a shared or local storage location for guest image files, import/export or for ISO images. Storage domain types supported in RHEV 3.0 are NFS, iSCSI, Fibre Channel, and local disk storage.

The RHEV **network** architecture supports both guest traffic and traffic among RHEV hypervisors and the RHEV-M server. All hosts have a network interface assigned to the logical network named *rhevm*. This network is used for the communications between the hypervisor and the manager. Additional logical networks are created on the data center and applied to one or more clusters. To become operational, the host attaches an interface to the local network. While the actual physical network can span across data centers, the logical network can only be used by the clusters and hosts of the creating data center.

Figure 2.2.1: RHEV Environment provides a graphical representation of a typical Red Hat Enterprise Virtualization environment with each component listed.

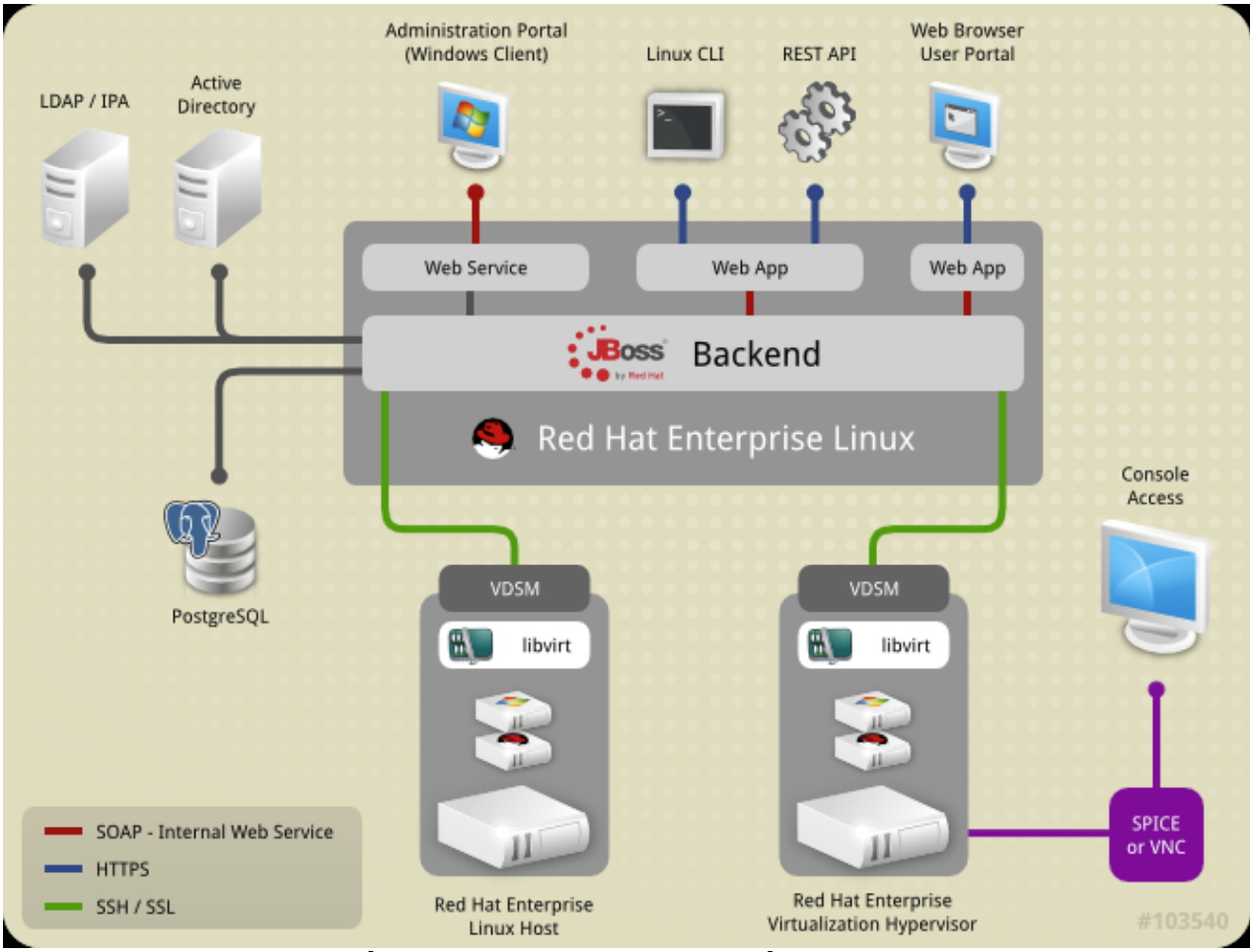


Figure 2.2.1: RHEV Environment

2.3 virt-v2v

`virt-v2v` command converts virtual machines from a foreign hypervisor to run on Red Hat Enterprise Virtualization 2.2 or later. It automatically packages the virtual machines into a file that is in the Open Virtualization Format (OVF) and uploads them to a Red Hat Enterprise Virtualization export storage domain. From the export storage domain, the OVF files can be imported into Red Hat Enterprise Virtualization using the Red Hat Enterprise Virtualization Manager.

VirtIO drivers are special paravirtualized drivers that provide enhanced performance capabilities for virtual machines running within the Red Hat Enterprise Virtualization environment. Using `virt-v2v`, the migration process becomes greatly simplified as the VirtIO drivers are injected into the virtual machine, when possible, during the migration process. The injection of the drivers is automatically handled by the `virt-v2v` command during the export of the virtual machine. The entire process is not simply just a copy or move process that takes place from one hypervisor to another.

The */var/lib/virt-v2v/virt-v2v.db* and */etc/virt-v2v.conf* files contains the configuration information for the `virt-v2v` command. These files allow the mapping of networks, specification of storage options, and the specifications of specific files and packages required for the various supported operating systems. The */var/lib/virt-v2v.db* file contains the default configurations and should not be modified. The */etc/virt-v2v.conf* file should be modified to contain custom configurations through the use of profiles.

Additional information regarding `virt-v2v` and its configuration file can be found in the virt-v2v and virt-v2v.conf man pages.

3 Reference Architecture Environment

The environment used in the writing of this Reference Architecture consists of both physical systems and virtual machines.

The following picture depicts the environment used and is followed by a description of the systems involved.

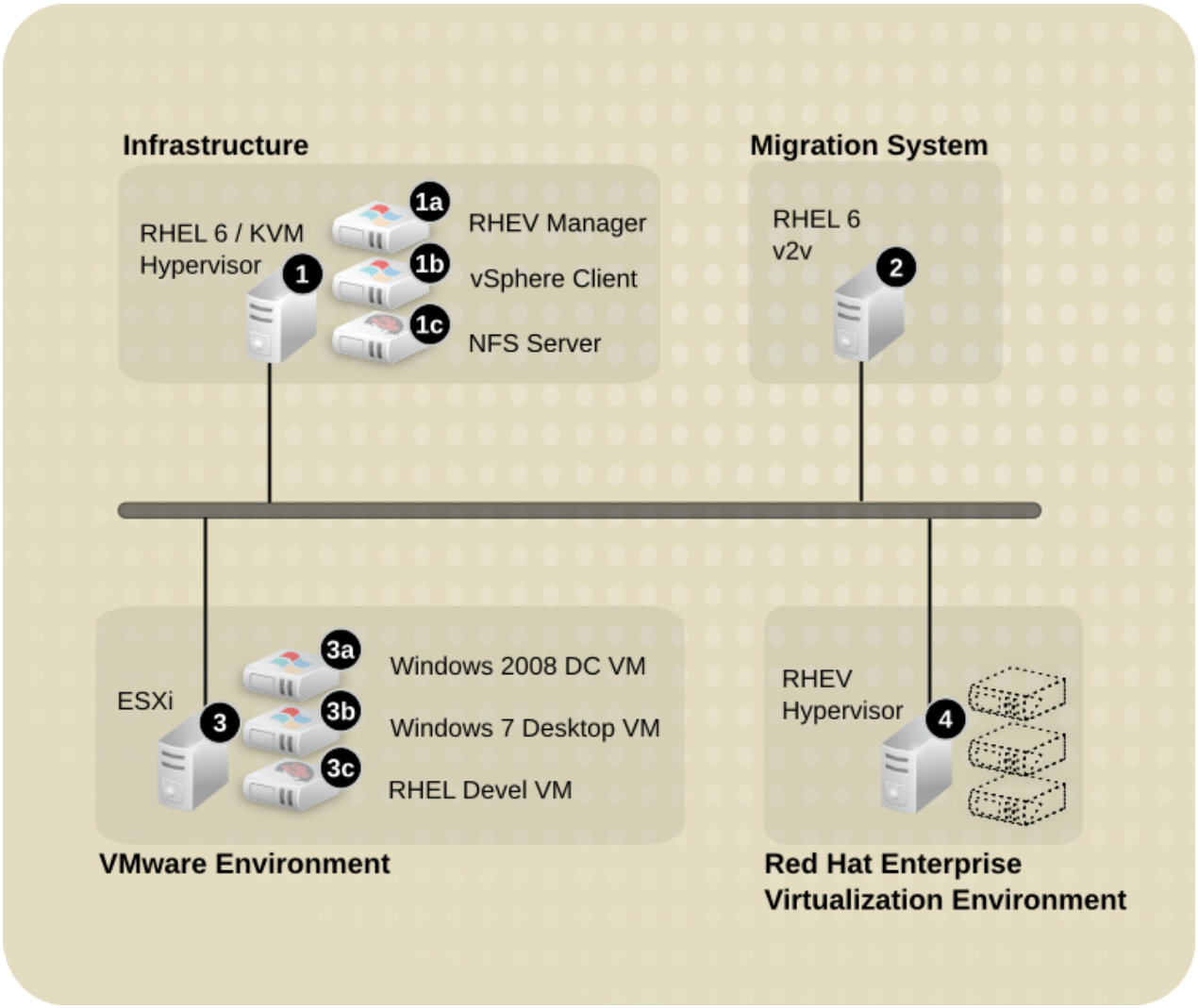


Figure 3.1: Reference Architecture Environment

Infrastructure

1 *RHEL 6 / KVM Hypervisor* - This system is a physical server in the lab environment running Red Hat Enterprise Linux 6 and the Kernel-based Virtual Machine (KVM) hypervisor. The systems that are hosted are utility or management type systems and are hosted as virtual machines to limit the hardware requirements for this Reference Architecture.

1a *RHEV Manager* - This virtual machine runs the Red Hat Enterprise Virtualization Manager and is used to manage the Red Hat Enterprise Virtualization environment.

1b *vSphere Client* - This virtual machine runs the VMware vSphere client that connects directly to the ESXi server and to access the RHEV Manager web interface.

1c *NFS Server* - This virtual machine is running Red Hat Enterprise Linux 6 and exports two directories via NFS. Both the exported directories are used by the Red Hat Enterprise Virtualization environment and Red Hat Enterprise Virtualization Manager. One exported directory is an ISO Domain and the second exported directory is an Export domain.

Migration System

2 *RHEL 6 v2v* - This is a physical system used to perform the migration of the virtual machines from the VMware environment to the Red at Enterprise Virtualization environment. This system is running RHEL 6 and the **`virt-v2v`** tools. This system has access to the Export Domain via NFS and the ESXi server via ssh.

VMware Environment

3 *ESXi* - This is a physical system running the VMware ESXi hypervisor. This system runs virtual machines that are migrated to the Red Hat Enterprise Virtualization environment during the writing of this Reference Architecture.

3a *Windows 2008 Domain Controller VM* - This virtual machine is running Windows 2008 R2 as a domain controller. This domain controller is used by the other virtual machines in this environment.

3b *Windows 7 Desktop VM* - This is a virtual machine running Windows 7 and joined to the domain controlled by the Windows 2008 DC VM.

3c *RHEL Devel VM* - This virtual machine is running Red Hat Enterprise Linux 6. This server authenticates login accounts using domain credentials.

Red Hat Enterprise Virtualization Environment

4 *RHEV Hypervisor* - This is a physical the Red Hat Enterprise Virtualization Hypervisor. This system hosts the virtual machines after they are migrated from the VMware environment.

The following table lists the specifications for systems considered part of the infrastructure.

System	Specifications
RHEL 6 / KVM [Dell PowerEdge R810]	RHEL 6.2.0.3 Kernel 2.6.32-220.17.1.el6.x86_64 Qemu-kvm 0.12.1.2-2.209.el6_2.5
	2 x Eight Core Intel Xeon X7560 CPUs @2.26GHz
	128 GB Memory
	4 x 146 GB SAS internal disk drive (RAID 5)
	2 x Broadcom Gigabit BASE-T MC Server Adapters
	1 x Broadcom 10Gigabit Dual Port SFP+ Adapter
RHEV Manager [Virtual Machine]	RHEL 6.2.0.3 Kernel 2.6.32-220.17.1.el6.x86_64
	1 x Virtual CPUs QEMU Virtual CPU version cpu64-rhel6 @ 2.26GHz
	4 GB Memory
	1 x VirtIO Disk file @ 8 GB
	1 x VirtIO Network Adapter
vSphere Client [Virtual Machine]	Microsoft Windows 7 Ultimate
	1 x Virtual CPUs QEMU Virtual CPU version cpu64-rhel6 @ 2.26GHz
	2 GB Memory
	1 x VirtIO Disk file @ 40 GB
	1 x VirtIO Network Adapter
NFS Server [Virtual Machine]	RHEL 6.2.0.3 Kernel 2.6.32-220.17.1.el6.x86_64
	1 x Virtual CPUs QEMU Virtual CPU version cpu64-rhel6 @ 2.26GHz
	1 GB Memory
	1 x VirtIO Disk file @ 40 GB 1 x VirtIO Disk file @ 120 GB
	1 x VirtIO Network Adapter

Table 3-1: Infrastructure Systems

The following table lists the physical hardware and software specifications for the system used to perform the migration.

System	Specifications
RHEL 6 V2V [HP ProLiant BL460c G6]	RHEL 6.2.0.3 Kernel 2.6.32-220.17.1.el6.x86_64
	Dual Socket, Quad Core w/hyperthreading (8 cores, 16 hyperthreads) Intel® Xeon® CPU X5550 @2.67GHz
	48 GB Memory
	2 x 146 GB SAS internal disk drive (mirrored)
	2 x Broadcom NetXtreme II BCM57711E Flex-10 10Gb Ethernet Controller

Table 3-2: Migration System

The following table lists the physical hardware and software specifications for the system running the Red Hat Enterprise Virtualization hypervisor.

System	Specifications
Red Hat Enterprise Virtualization Hypervisor [Dell PowerEdge R810]	RHEV Hypervisor-6.2-20111215.0.el6_2 Kernel 2.6.32-220.2.1.el6.x86_64 KVM 0.12.1.2-2.209.el6_2.1
	2 x Eight Core Intel Xeon X7560 CPUs @2.26GHz
	128 GB Memory
	4 x 146 GB SAS internal disk drive (RAID 5)
	1 x QLogic ISP2532-based 8Gb FC HBA 1 x 430 GB Lun
	2 x Broadcom Gigabit BASE-T MC Server Adapters
	1 x Broadcom 10Gigabit Dual Port SFP+ Adapter

Table 3-3: Red Hat Enterprise Virtualization Environment

The following table lists the physical and virtual hardware and software specifications for each system in the VMware environment.

System	Specifications
ESXi [HP ProLiant DL580 G5]	VMware ESXi 5.0.0 Build 623860
	4 x Quad Core Intel Xeon X7350 CPUs @ 2.93GHz
	64 GB Memory
	4 x 73 GB internal disk drives (RAID 5)
	2 x Broadcom NetXtreme II BCM5708 Gigabit Ethernet Controllers
	1 x Intel 82572EI Gigabit Ethernet Controller
Windows 2008 DC [Virtual Machine]	Windows Server 2008 R2
	1 x Virtual CPUs Intel® Xeon® CPU X7350 @ 2.93GHz
	4 GB Memory
	1 x Virtual Disk file @ 40 GB
	1 x Virtual Network Adapter
Windows 7 Desktop [Virtual Machine]	Microsoft Windows 7 Ultimate
	1 x Virtual CPUs Intel® Xeon® CPU X7350 @ 2.93GHz
	2 GB Memory
	1 x Virtual Disk file @ 30 GB
	1 x Virtual Network Adapter
RHEL Devel [Virtual Machine]	RHEL 6.2.0.3 Kernel 2.6.32-220.17.1.el6.x86_64
	1 x Virtual CPUs Intel® Xeon® CPU X7350 @ 2.93GHz
	2 GB Memory
	1 x Virtual Disk file @ 20 GB
	1 x Virtual Network Adapter

Table 3-4: VMware Environment

All the Red Hat Enterprise Linux systems run the **`iptables`** firewall using the default firewall rules except the *v2v3-nfs* system. This system has been configured to allow NFS to use static ports and the firewall to allow traffic from the local network to pass through the server. See **Appendix A: Securing NFS using IPTables**.

The *v2v3-W2K8-DC* is a domain controller and provides a domain user account called *refarch*. This account is used for testing the domain integration of the *v2v3-W7* and *v2v3-rhel* virtual machines.

The Red Hat Enterprise Virtualization environment used is configured for a single datacenter called *NorthAmerica*. This data center contains one cluster called **Infrastructure** that attaches to the *v2v3-rhevh* hypervisor server. The data center has three storage domains configured. The data domain, named *Data*, is a fibre channel domain. The export and the ISO domain are NFS shares and are named *Export* and *ISO* respectively.

4 Migration Overview

Migrating virtual machines from an ESXi system to the Red Hat Enterprise Virtualization environment can be done easily with some simple planning. Knowing the migration process can significantly help in the planning process. This allows better planning of system down times as well as resource availability if issues arise during a migration process. The migration process is outlined below.

Preparation - Configuration of the ESXi server and the Red Hat Enterprise Virtualization environment must be performed to allow the migrations. The configuration consists of the following.

Network Configuration – Networks that exist in the ESXi environment may not exist in the Red Hat Enterprise Virtualization Environment. Equivalent networks must be created or mapped.

Export Domain – An Export Domain must be created and defined on the Red Hat Enterprise Virtualization environment. This Export Domain is used for the temporary storage of the virtual machines during the migration process.

Guest Tools ISO – The Guest Tools ISO must be uploaded into the Red Hat Enterprise Virtualization environment.

Gather Virtual Machine Information – Information about the virtual machine, its function, and any services it offers should be noted. This helps with the migration planning as well as ensuring the virtual machine functions as required after migration.

Commit Snapshots – Any snapshots on the virtual machines must be committed before the export process starts.

Uninstall VMware Tools – VMware Tools must be uninstalled from the virtual machines prior to starting the migration process.

Export Virtual Machine – Exporting the virtual machine is done using the `virt-v2v` command and tools. This command is executed on a Red Hat Enterprise Linux server.

Import Virtual Machine – Importing the virtual machines is done using the Red Hat Enterprise Virtualization Manager interface.

Post Migration – After the virtual machines are imported into the new environment a few more post migrations steps must be performed.

First Boot – For Windows based virtual machines it is required to reconfigure the network information since a new network adapter is presented and the Windows operating system does not move the network configuration over to the new adapter.

Testing – The virtual machine and its services must be tested for functionality.

Cleaning up – The virtual machines still exist in the original ESXi environment and copies of them exist on the Export Domain. These must be addressed and removed.

5 Migration Preparation

The migration from the VMware environment to the Red Hat Enterprise Virtualization environment requires some preparation in order to make the migration process run smoothly. This section discusses these preparations.

This paper assumes that a currently installed and functioning Red Hat Enterprise Virtualization environment is the destination for the migrated virtual machines. If this is not the case, a Red Hat Enterprise Virtualization environment must be installed and configured prior to starting the migrations. If an appropriate environment is not configured, please refer to **Appendix D: References** for links on how to set up and configure an environment.

5.1 Back Up Virtual Machines

It is always recommended to have a good backup and restore plan in place in case of any unforeseen issues that may cause data loss. The migration of virtual machines is no different. Make sure there are known good backups of the systems involved in this migration and a proven process to restore the systems.

There are many tools to accomplish these tasks that include basic Linux commands to full featured backup and restore applications offered by the community and major software vendors. The backup and restoration of the data is not within the scope of this paper. However, the RHEV 3.0 Backup and Recovery using Acronis[2] Reference Architecture discusses the subject.

2 https://access.redhat.com/knowledge/refarch/2012-rhev-30-backup-recovery-using-acronis

5.2 Map and Create Networks

Networks that are used by the virtual machines in the ESXi environment may not exist in the Red Hat Enterprise Virtualization environment. These networks must be created, mapped to new networks, or both.

A network must be created if there are no networks defined in the Red Hat Enterprise Virtualization environment that match or meet the requirements of the virtual machines defined network. Running networks with VLANs or having multiple networks could be a reason for this.

If a suitable network does exist but is called a different name, then a network mapping can be created by editing the */etc/virt-v2v.conf* file. If the network definitions are simple, then a command line option to the `virt-v2v` migration command can be specified. This method allows the defining of a single default of fallback network name to use.

Creating mappings between the two environments can help avoid network issues with the virtual machines. This can easily be done by creating a table similar to **Table 5.2-1: Network Mappings**. This table is first populated with the virtual network information from the ESXi environment, then the names of the virtual networks defined in the Red Hat Enterprise Environment are placed into the table if they match the network and VLAN information already entered. The table can easily be viewed to see what mappings can be made or if new virtual networks must be created in the Red Hat Enterprise Virtualization environment.

Network	VLAN	ESXi Network Name	RHEV Network Name	Notes
10.16.136.0/21		VM Network	infrastructure	Network for Infrastructure Servers
0.0.0.0/0 (any network)			rhevm	Default RHEV Network

Table 5.2-1: Network Mappings

The data in the above table shows that the *VM Network* network on the on the ESXi server must be mapped to the *infrastructure* network in the Red Hat Enterprise Virtualization Environment. The *VM Network* could also work if connected to the *rhevm* network since the *rhevm* network is configured to pass traffic on any network (0.0.0.0/0).

If a row in the table had a network that was used and contained an entry in the *ESXi Network Name* column but not the *RHEV Network Name* column, then a new network must be created in the Red Hat Enterprise Virtualization environment.

5.2.1 Creating the Networks

In the Red Hat Enterprise Virtualization Manager interface, select the **Data Centers** tab (1). The data centers defined in the environment are listed.

Select the **NorthAmerica** data center (2). A **Detail** pane (3) appears at the bottom of the window. Select **Logical Networks** (4) in the **Detail** pane and select on the **New** button (5). A new window opens called **New Logical Network**.

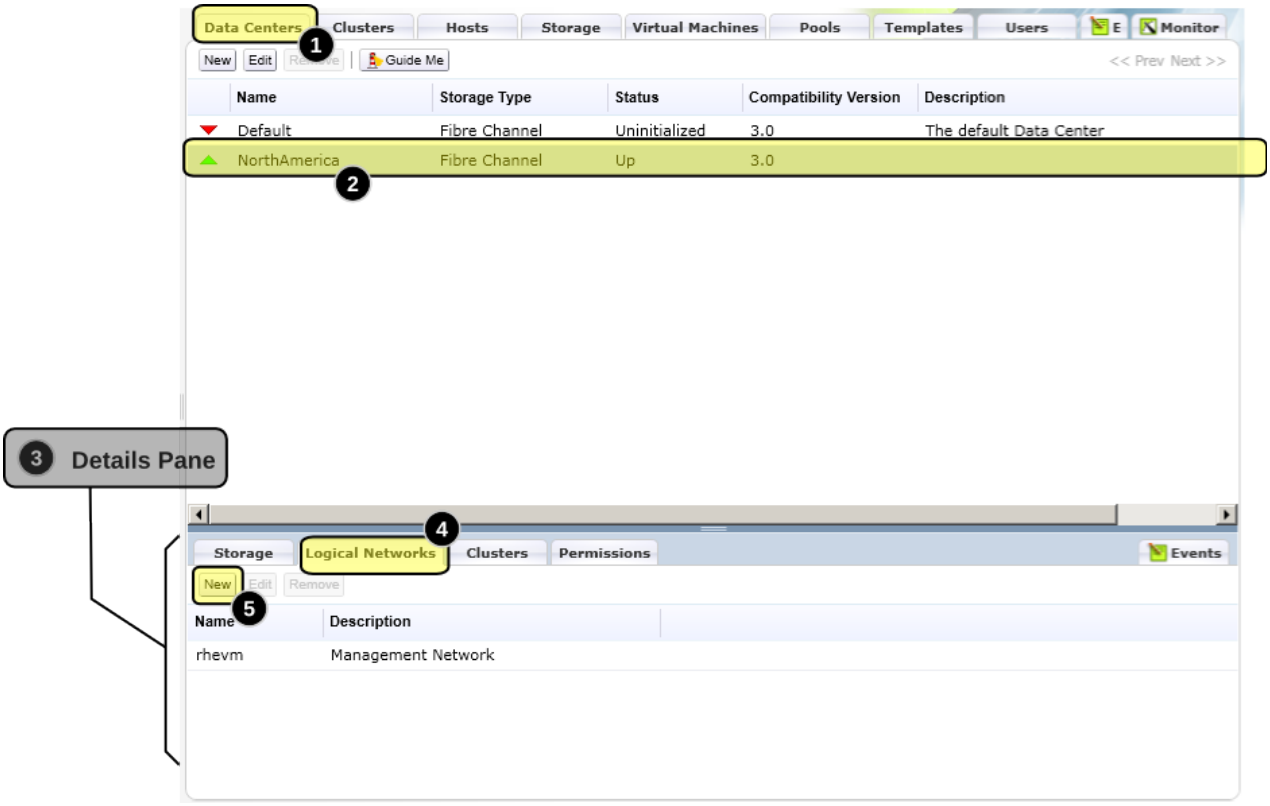


Figure 5.2.1.1: New Logical Network

The **Name** and **Description** for the new logical network is entered as shown in the following figure. Once the information is entered, check the box next to **Infrastructure** to assign the newly created network to the *Infrastructure* cluster. Select the **OK** button to complete the task. The new network is now created and listed under the **Logical Networks** tab of the **Details** pane.

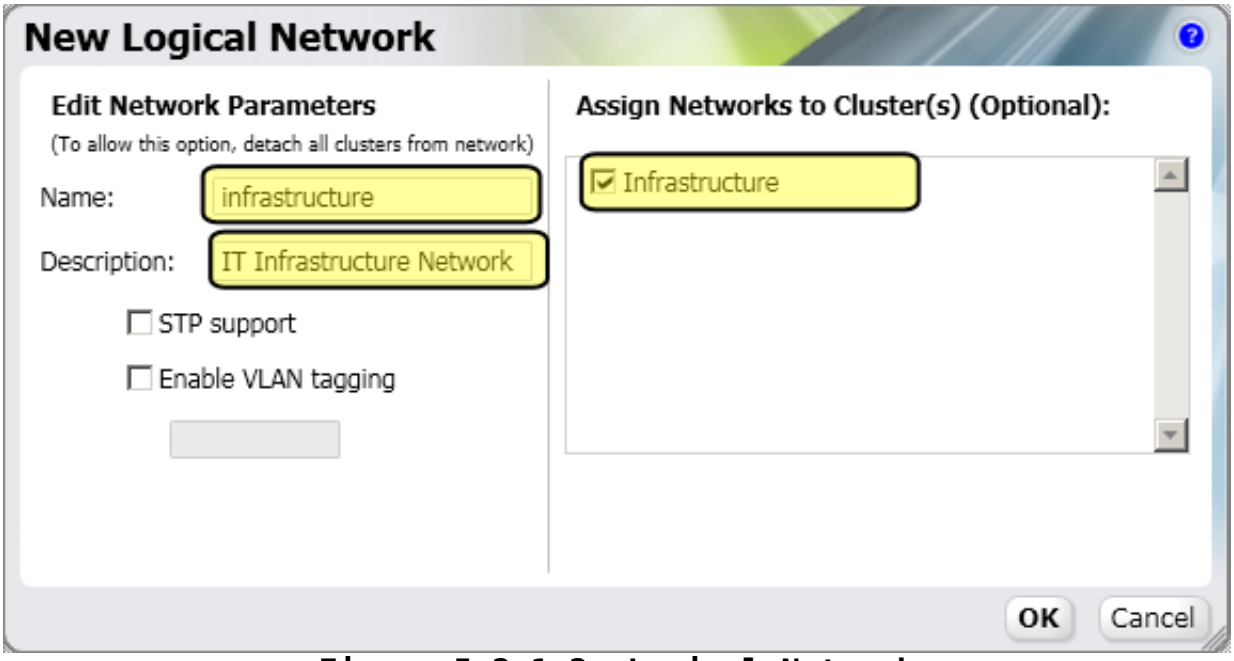


Figure 5.2.1.2: Logical Network

The network now shows up in the **Logical Network** of the **Details** pane. The network has an icon next to it that looks like a broken wrench. This is fine for now, the network must be assigned to a physical network interface.

Refer to the following figure while assigning a physical interface to the logical network. Select the **Hosts** tab (1) and select the server (2), a **Details** pane opens up. Select the **Network Interfaces** tab (3) and select an unused interface (4), this Reference Architecture uses the **eth1** interface. Select the **Add / Edit** button (5) to open the **Edit Network Interface** window.

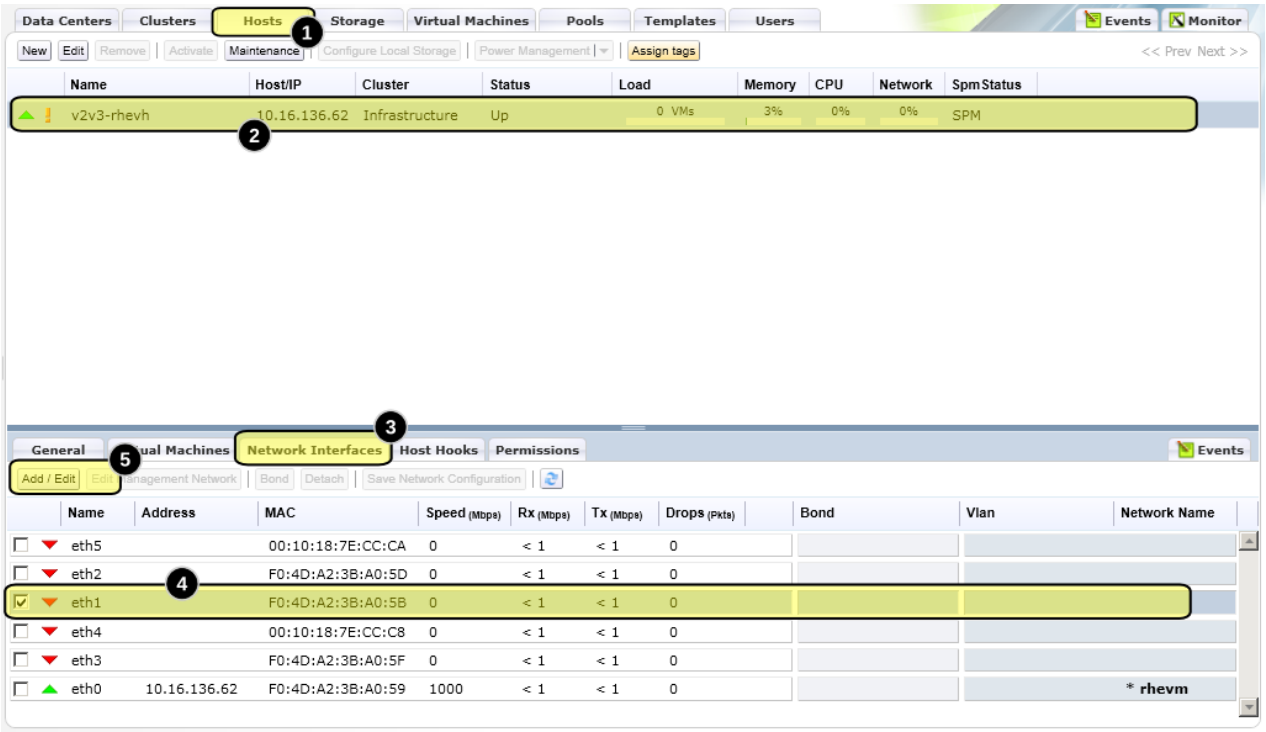


Figure 5.2.1.3: Assign Interface

The following figure depicts the **Edit Network Interface** window. In the **Edit Network Interface** window, make sure the new network called **infrastructure** is selected for the **Network** option and the radio box called **None** is selected. **DHCP** or **Static** could be selected to provide the interface with an IP address. This allows the Red Hat Enterprise Virtualization server to be accessed from this interface. This is not required for this Reference Architecture. Make sure the **Save network configuration** check box is checked. Select the **OK** button to apply the changes.

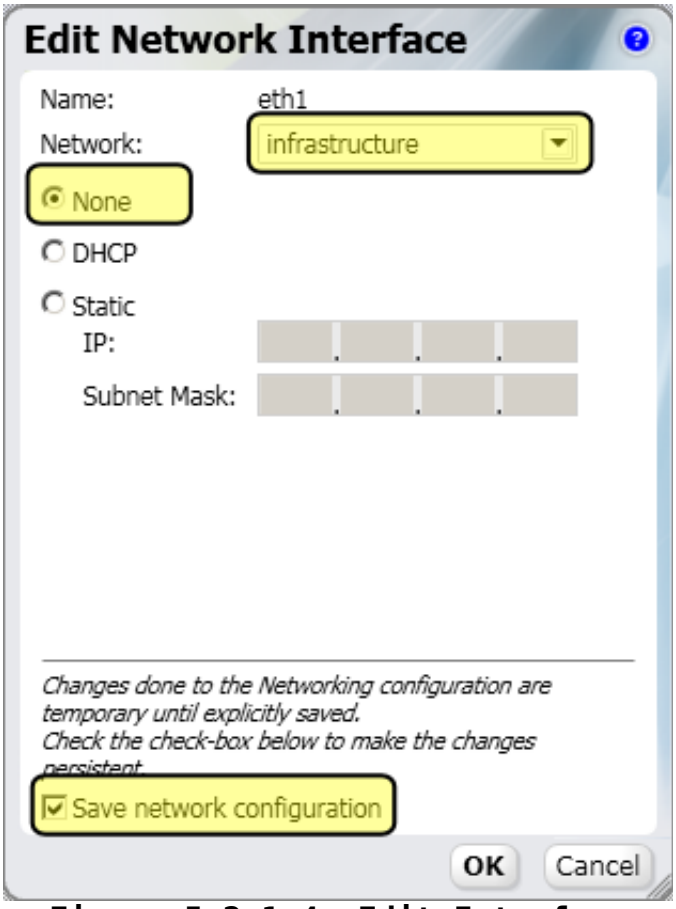


Figure 5.2.1.4: Edit Interface

The interfaces icon in the **Network Interfaces** tab changes to a green arrow pointing up. This indicates the interface is up and active.

5.2.2 Mapping the Networks

The */etc/virt-v2v.conf* configuration file allows the mapping of networks from the original virtual environment to the new virtual environment. This configuration file contains nested network sections in the following format:

```
<network type='bridge|network' name='bridge_or_network'>
   <network type='bridge|network' name='bridge_or_network'/>
 </network>
```

The outside network section defines the network on the environment being migrated from. The inside definition defines what network to map to on the destination environment.

The mapping used in this paper to map the VM Network virtual network on the ESXi system to the virtual network called infrastructure in the Red Hat Enterprise Virtualization environment looks as follows:

```
<network type='bridge' name='VM Network'>
   <network type='network' name='infrastructure'/>
 </network>
```

The ESXi virtual networks are treated as bridges by the `virt-v2v` command and the mappings in the *virt-v2v.conf* file should use a network type of bridge.

5.3 Create an Export Domain

Open the Red Hat Enterprise Virtualization Manager. Select the **New Domain** button in the **Storage** tab. This brings up a **New Domain** window as seen in **Figure 5.3.1: New Domain**.

Give the new storage domain a **Name** of ***Export***. Select **NorthAmerica** as the **Data Center**. Select **Export / NFS** as the domain function and select **v2v3-rhevh** as the host to use for access to the new export domain.

Finally, specify ***v2v3-nfs.cloud.lab.eng.bos.redhat.com:/exports/Export*** as the export path using an NFS URI. The FQDN must be used when specifying the location. Select the **OK** button and the new Export Domain is displayed in the **Storage** tab with a status of **Inactive.**

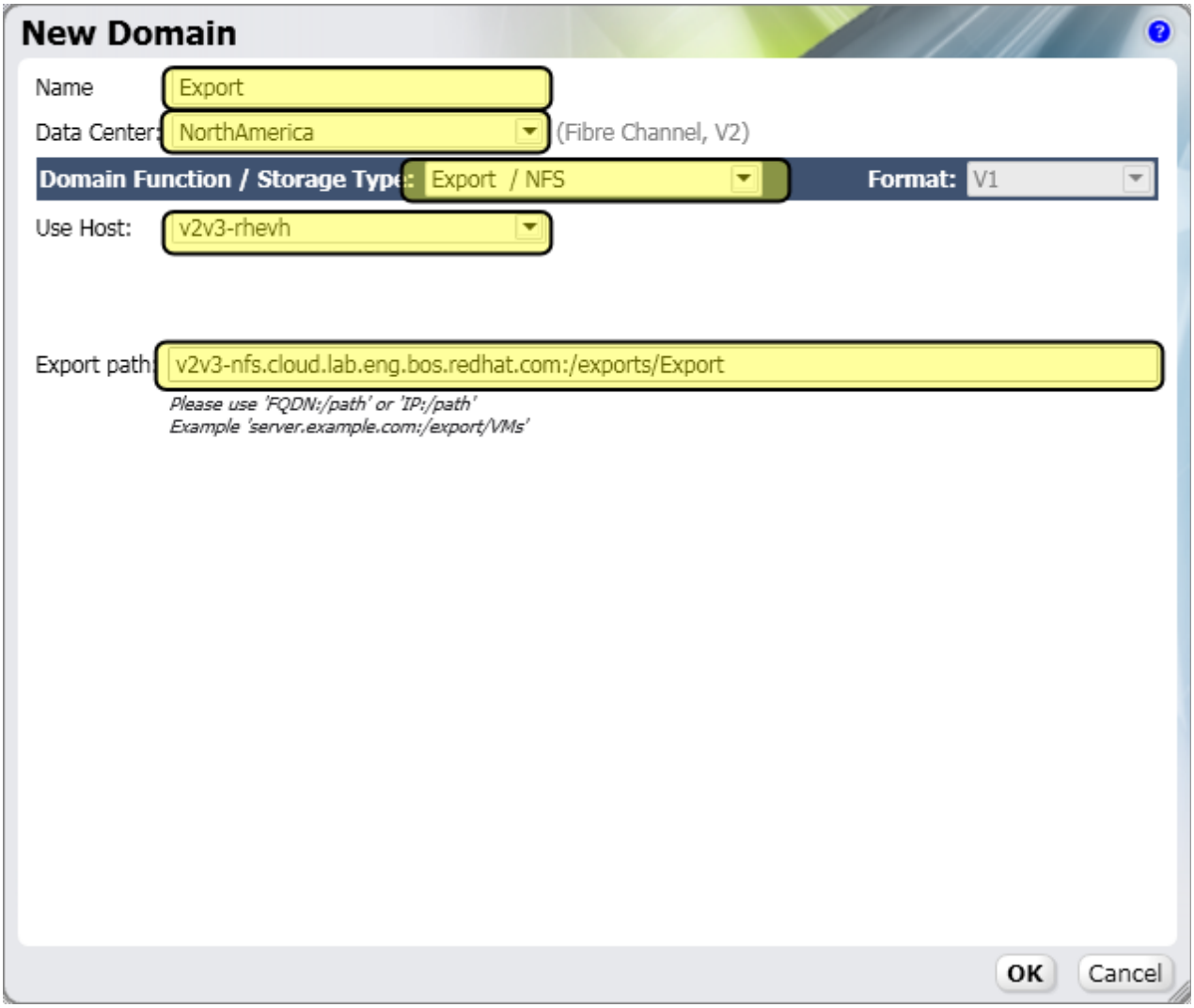


Figure 5.3.1: New Domain

The new export domain must be activated. Select the **Data Centers** tab (1) then select the **NorthAmerica** data center (2). Make sure the **Storage** tab (3) is selected under the **Details** pane. Select on the **Activate** button (4). The status (5) of the export domain changes from **Inactive** to **Locked.** After a few moments, the status changes to **Active**.

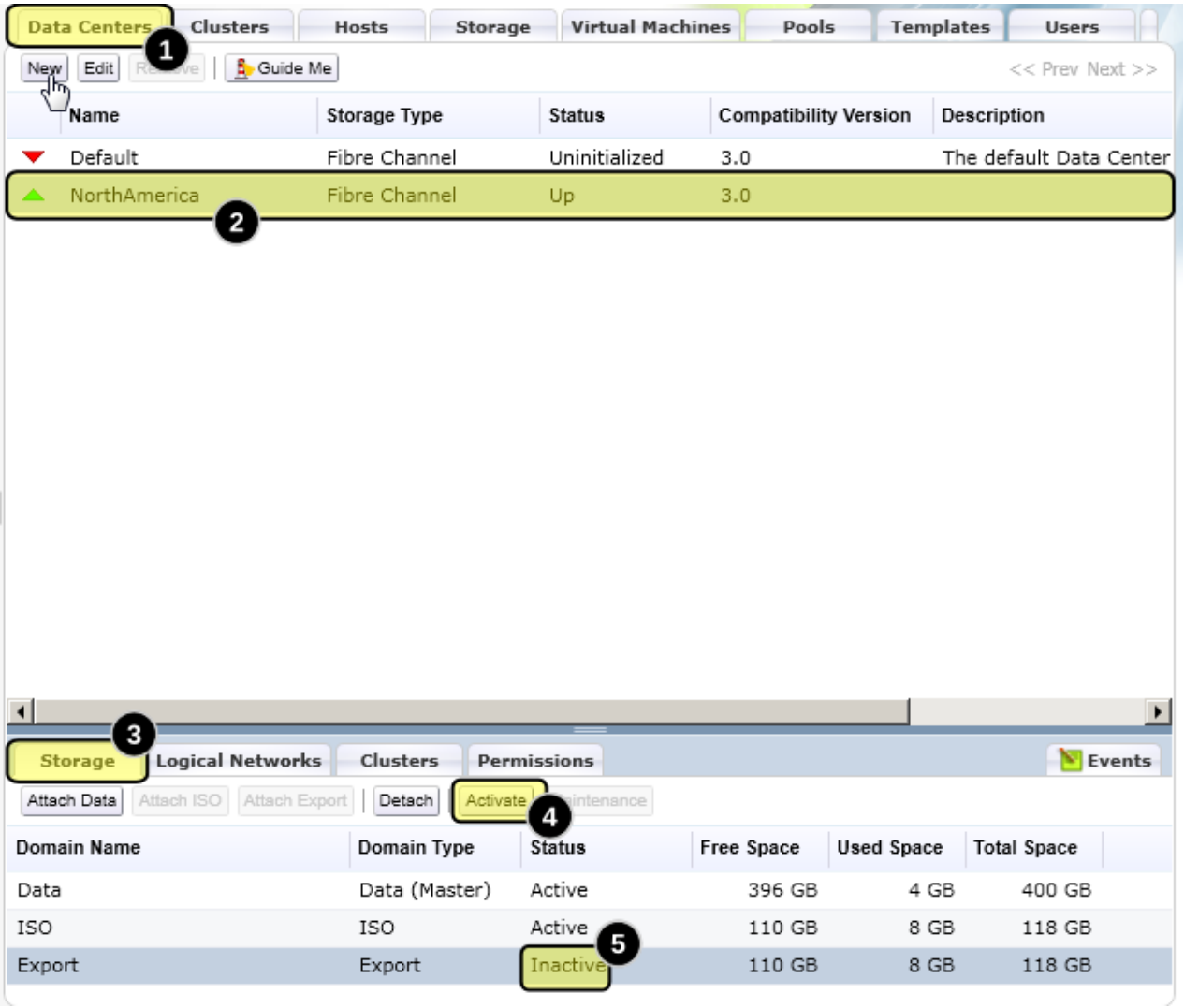


Figure 5.3.2: Activating the Export Domain

5.4 Upload Required ISOs to the Red Hat Enterprise Virtualization Environment

During the migration process, the `virt-v2v` command or the Red Hat Enterprise Virtualization Manager may install new packages or drivers into the virtual machines. What is installed varies depending on the operating system on the virtual machine.

The packages and drivers that are installed for a particular operating system are configured in the */etc/virt-v2v.conf* file and the */var/lib/virt-v2v/virt-v2v.db* file. The */var/lib/virt-v2v/virt-v2v.db* file is the default configuration database and should not be modified. New entries should be created in the */etc/virt-v2v.conf* file. Entries in the */etc/virt-v2v.conf* file have precedence over the */var/lib/virt-v2v/virt-v2v.db* file. The packages and drivers are provided by the **`Red Hat Network`**, the **`virtio-win`** package, or the **`Guest Tools ISO`**.

The following is a partial */var/lib/virt-v2v/virt-v2v.db* file.

```
                    [ ... output abbreviated ... ]

<!-- RHEL 6 has always supported virtio -->
<capability os='linux' distro='rhel' major='6' name='virtio'/>

<capability os='linux' distro='rhel' major='5' name='virtio'>
  <dep name='kernel' minversion='2.6.18-128.el5'/>
  <dep name='lvm2' minversion='2.02.40-6.el5'/>
  <dep name='selinux-policy-targeted' minversion='2.4.6-203.el5'
       ifinstalled='yes'/>
</capability>

                    [ ... output abbreviated ... ]
```

In the above example, *CAPABILITY* tags define the operating systems and the minimum package versions required. These packages are installed using the update agent within the guest's operating system. The version of the required packages are defined using the *DEP* tags. The configuration shows that Red Hat Enterprise Linux 6 has no dependencies, this is because all versions of Red Hat Enterprise Linux 6 support *VirtIO*. However, Red Hat Enterprise Linux 5 shows three dependencies are required and the dependencies minimum version required. In the above file, if the **`kernel`**, **`lvm2`**, or **`selinux-policy-targeted`** packages do not meet their respective minimum versions, they are updated by the `virt-v2v` command.

The *APP* tags also define the operating systems and minimum package versions required. The packages listed under the *APP* tags will be installed from the local directories located on the system running the `virt-v2v` command.

The *PATH* tag defines the location of the file on the system. If the file location is not specified using its absolute path, the *PATH-ROOT* tag must exist and the file is located under the directory it specifies.

```
[ ... output abbreviated ... ]

  <!-- RHEL 5
       All of these RPMS are from RHEL 5.3, which was the first version of
       RHEL 5 to support VirtIO -->

  <app os='linux' distro='rhel' major='5' arch='i686' name='kernel'>
    <path>rhel/5/kernel-2.6.18-128.el5.i686.rpm</path>
    <dep>ecryptfs-utils</dep>
  </app>

  <app os='linux' distro='rhel' major='5' arch='x86_64'
       name='ecryptfs-utils'>
    <path>rhel/5/ecryptfs-utils-56-8.el5.x86_64.rpm</path>
  </app>

[ ... output abbreviated ... ]

  <path-root>/var/lib/virt-v2v/software</path-root>

[ ... output abbreviated ... ]
```

The above example shows the kernel package for Red Hat Enterprise Linux 5 is */var/lib/virt-v2v/software/rhel/5/kernel-2.6.18-128.el2.i686.rpm*. The *DEP* tag shows that Red Hat Enterprise Linux 5 also requires the **`ecryptfs-utils`** package installed. Another set of *APP* and *PATH* tags show the location of the required **`ecryptfs-utils`** package.

5.4.1 Red Hat Enterprise Linux Based Virtual Machines

The `virt-v2v` command may install a new kernel and drivers in the virtual machine. This happens if the existing kernel does not run the VirtIO drivers. This behavior can be controlled by installing a kernel that is capable of supporting the VirtIO drivers prior to executing the `virt-v2v` command. A Red Hat Enterprise Linux 5.2 kernel does not support the VirtIO drivers, in this case the `virt-v2v` command installs a later kernel as indicated in the *virt-v2v.conf* file.

The kernel packages must be available during the execution of the `virt-v2v` command. The packages will be installed using the virtual machines update agent. If the packages cannot be installed using the update agent, the packages must be manually downloaded and placed in the appropriate location as defined in the */etc/virt-v2v.conf* and */var/lib/virt-v2v/virt-v2v.db* configuration files.

The man page for the *virt-v2v.conf* file and the file itself can be referred to when configuring any of the above.

5.4.2 Windows Based Virtual Machines

The `virt-v2v` command installs new drivers that allow the virtual machine to boot. The **`virtio-win`** and **`libguestfs-winsupport`** packages must be installed to provide the necessary files. Once the new drivers are installed, the virtual machine can boot in the new virtualization environment. However, the virtual machine does not have all the drivers required to operate correctly in the new environment.

The remaining required drivers are provided by the **`Guest Tools ISO`** when the virtual machine is imported into the Red Hat Enterprise Virtualization environment.

5.4.3 Guest Tools ISO

The rhev-iso-uploader command on the Red Hat Enterprise Virtualization Manager uploads files to the ISO storage domains defined in the environment. The **`--iso-domain`** option specifies the ISO storage domain for the uploaded files. See the man page for the rhev-iso-uploader command for more information about the command and its options.

Upload the Guest Tools ISO to the storage domain called ISO using the following command. Supply the admin password when asked.

```
# rhevm-iso-uploader --iso-domain ISO upload /usr/share/rhev-guest-tools-
iso/rhev-tools-setup.iso
Please provide the REST API password for the admin@internal RHEV-M user
(CTRL+D to abort):
```

The **`Guest Tools ISO`** is now uploaded and available to be used in the environment.

5.5 Gather Virtual Machine Information

New storage drivers, network drivers, and video drivers are installed during the migration process. This can affect the configuration of the virtual machines in the new environment. Adjustment of the video resolution might be required or the network adapters might change names after the migrated system boots.

Some operating systems (e.g. Windows) do not migrate the network configuration over to the new virtual adapter. This is because the adapter is viewed as a new adapter in the guest operating system instead of a replacement adapter. Because of this, it is a good idea to make note of any important network information. The following table contains information about the virtual machines used in this paper. The table records some of the more common configuration parameters.

Virtual Machine	Configuration	
v2v3-W2K8-dc	Interface Name	Local Area Connection
	IP Address and Network Mask	10.16.136.68/21
	Default Gateway:	10.16.143.254
	DNS Servers:	10.16.143.247
		10.16.142.248
	Static Routes:	None defined
	Notes:	Mission critical system Primary Domain Controller
v2v3-W7	Interface Name	Local Area Connection
	IP Address and Network Mask	10.16.136.69/21
	Default Gateway:	10.16.143.254
	DNS Servers:	10.16.136.68
		10.16.143.247
		10.16.143.248
	Static Routes:	None defined
	Notes:	none
v2v3-rhel	Interface Name	eth0
	IP Address and Network Mask	10.16.136.70/21 via dhcp
	Default Gateway:	10.16.143.254
	DNS Servers:	10.16.136.68
		10.16.143.247
		10.16.143.248
	Static Routes:	None defined
	Notes:	none

Table 5.5-1: Virtual Machine Notes

6 Migrating the Virtual Machines

The migration of the virtual machines can be a slow process depending on the size of the virtual disks and how many virtual machines are being migrated. Before the virtual machines can be migrated, all the snapshots must be committed, VMware Tools must be uninstalled, and the virtual machine must be powered off. Once these tasks are done, the migration process can begin.

6.1 Committing Snapshots

Virtual machines running on the ESXi server may have snapshots of the disk images. These snapshots must be committed to the original disk image before the virtual machine is exported using the **virt-v2v** command. The **virt-v2v** command only migrates the original disk image files and not the snapshot deltas.

To commit snapshots, use the right mouse button to select the virtual machine name and choose **Snapshot | Snapshot Manager**. As seen in the figure below.

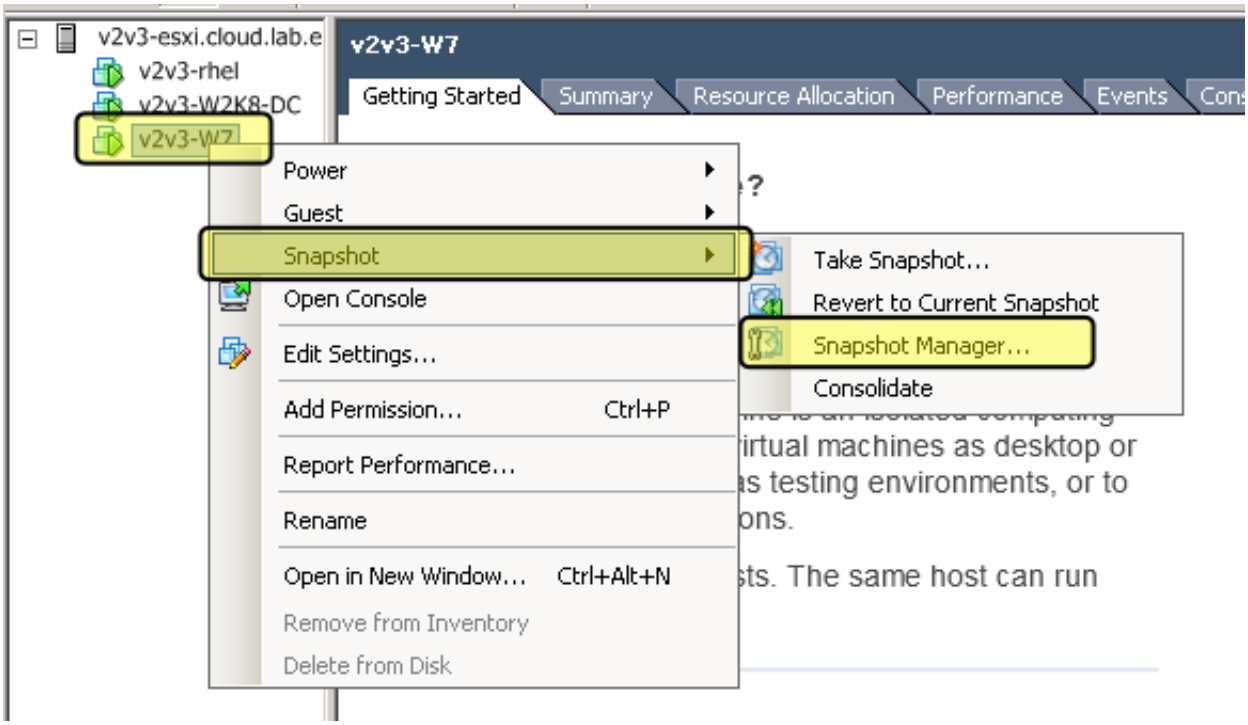


Figure 6.1.1: Snapshots

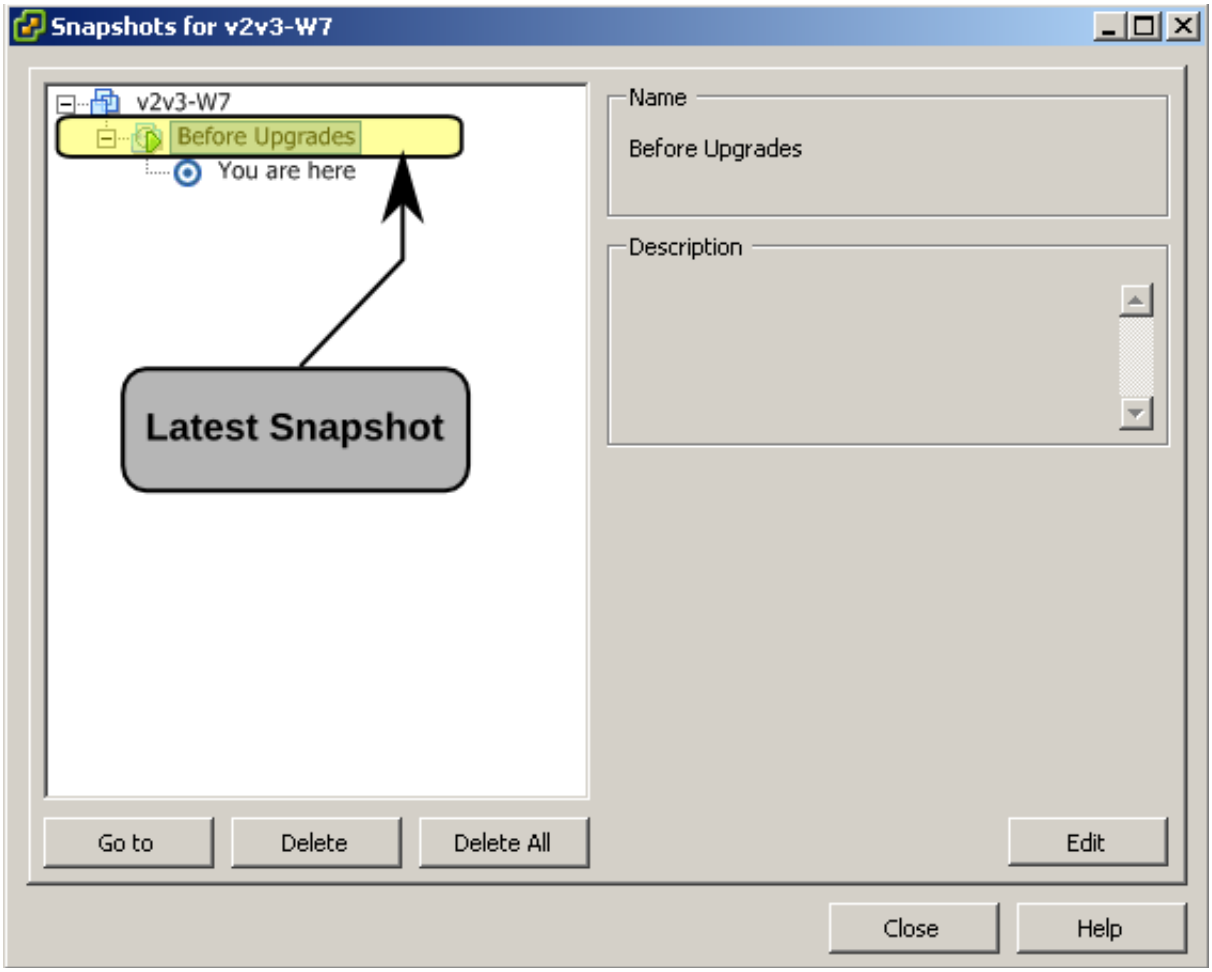


Figure 6.1.2: Snapshot Manager

A window opens up that lists the snapshots that have been made on a virtual machine. Choose the latest snapshot and select the **Delete All** button. This commits all the snapshots.

A confirmation window pops up asking if consolidating and removing all the snapshots is intended to be done. Select the **Yes** button to commit and remove the snapshots.

6.2 Uninstalling VMware Tools

VMware Tools must not be installed in the virtual machines when the export process is started. If a virtual machine is exported while **VMware Tools** is installed, it may not function correctly in the new environment[3]. If **VMware Tools** are uninstalled, it is a good idea to reboot the virtual machine to make sure the virtual machine still functions correctly before the export process is started.

Uninstall **VMware Tools** from the Windows 2008 Domain Controller virtual machine by selecting the **Start Button | Control Panel | Programs | Programs and Features**. Select **VMware Tools** and an **Uninstall** button appears in the bar above the software selections. Select the **Uninstall** button. The **VMware Tools** software is uninstalled.

The virtual machine is rebooted by typing the <**Alt-F4**> key combination at the virtual machines console and selecting **Restart** from the drop down menu that is titled **What do you want the computer to do?**. Selecting the **Comment** box allows specifying a reason for the reboot, this is required. Enter ***"Making sure the system functions correctly after VMware Tools are uninstalled."*** in the Comment box. Select the **OK** button start the reboot process.

Uninstall **VMware Tools** from the Windows 7 virtual machine using the same steps as those to uninstall them from the Windows 2008 Domain Controller virtual machine. The Windows 7 virtual machine is rebooted in a similar fashion as well.

Uninstall the **VMware Tools** from the Red Hat Enterprise Linux virtual machine by executing the `vmware-uninstall-tools.pl` command.

```
# vmware-uninstall-tools.pl
Uninstalling the tar installation of VMware Tools.

Stopping services for VMware Tools

vmware-tools stop/waiting

[... output abbreviated ...]

The removal of VMware Tools 8.6.5 build-621624 for Linux completed
successfully.  Thank you for having tried this software.
```

Reboot the Red Hat Enterprise Linux virtual machine by issuing the `shutdown -r now` command.

3 https://bugzilla.redhat.com/show_bug.cgi?id=678232

6.3 Shutting Down the Virtual Machines

A virtual machine must be powered off before it can be exported using the `virt-v2v` command. When an attempt is made to export a virtual machine that is powered on, the `virt-v2v` command displays an error message. Below is an example of this error. The command and its options are explained in **Section 6.4.1**.

```
# virt-v2v -i libvirt -ic esx://v2v3-esxi/?no_verify=1 \
  -o rhev -os v2v3-nfs:/exports/Export                \
  --network rhevm v2v3-rhel

virt-v2v: Guest v2v3-rhel is currently running. It must be shut down first.
```

Enabling `ssh` on the ESXi server allows access to the `vim-cmd` command on the ESXi server. This command provides a means to query and control the power state of the virtual machines.

The power state of the virtual machines can be displayed using the `ssh` client and the ESXi systems `vim-cmd` command. In order to query the virtual machines power state, the Virtual Machines Vmid is required. The `vim-cmd` displays this as well. The first command in the following example displays the virtual machine names and their respective Vmids. The second command displays the power state for one of the virtual machines using its Vmid.

```
# ssh v2v3-esxi vim-cmd vmsvc/getallvms |awk '{print $1 " " $2}'

Vmid Name
1 v2v3-W2K8-DC
2 v2v3-W7
3 v2v3-rhel

# ssh v2v3-esxi vim-cmd vmsvc/power.getstate 3

Retrieved runtime info
Powered on
```

Red Hat Enterprise Linux based virtual machines can be powered off remotely using the `ssh` command line client and Windows based virtual machines can be powered off remotely using the Linux `net` command that is provided by the **`samba-common`** package. The virtual machines may also be powered off using the `vim-cmd` command on the ESXi server. If the `vim-cmd` command is used, the virtual machines on the ESXi server may not perform a graceful shutdown since **`VMware Tools`** are not installed. Use the `vim-cmd` method with caution since a non-graceful shutdown of a virtual machine is not recommended.

Shutdown the virtual machines remotely using the appropriate commands. The following output shows all the virtual machines shutting down successfully except the Windows 7 desktop virtual machine.

```
# ssh v2v3-rhel "shutdown -h now"
root@v2v3-rhel's password: [PASSWORD]

# net rpc SHUTDOWN -f -I v2v3-w2k8 -U administrator
Enter administrator's password: [PASSWORD]

Shutdown of remote machine succeeded

# net rpc SHUTDOWN -f -I v2v3-w7 -U refarch
Enter refarch's password: [PASSWORD]

Could not connect to server 10.16.136.69
Connection failed: NT_STATUS_UNSUCCESSFUL
```

The Windows 7 virtual machine is shutdown by typing the <**Alt-F4**> key combination at the virtual machines console and selecting **Shut down** from the drop down menu that is titled **What do you want the computer to do?**. Select the **OK** button to start a graceful shutdown.

After waiting a few minutes to allow the virtual machines to shutdown properly, the `vim-cmd` command is used to check the power state of the Virtual Machines.

```
# ssh v2v3-esxi vim-cmd vmsvc/getallvms | awk '{print $1 "  " $2}'
Vmid Name
1 v2v3-W2K8-DC
2 v2v3-W7
3 v2v3-rhel

# ssh v2v3-esxi vim-cmd vmsvc/power.getstate 1

Retrieved runtime info
Powered off

# ssh v2v3-esxi vim-cmd vmsvc/power.getstate 2

Retrieved runtime info
Powered off

# ssh v2v3-esxi vim-cmd vmsvc/power.getstate 3

Retrieved runtime info
Powered off
```

6.4 Exporting the Virtual Machines

Exporting the virtual machines is done by executing the `virt-v2v` command from a Red Hat Enterprise Linux 6 system that contains the **`virt-v2v`**, **`virtio-win`**, and **`libguestfs`** packages installed. These packages are available under the **`Red Hat Enterprise Linux Server Supplementary`** and **`Red Hat Enterprise Linux Server V2V Tools for Windows`** channels of the **`Red Hat Network`**.

```
# rhn-channel -u admin -p [Password] -a -c rhel-x86_64-server-supplementary-
6 -c rhel-x86_64-server-v2vwin-6

# rhn-channel -l
rhel-x86_64-server-6
rhel-x86_64-server-supplementary-6
rhel-x86_64-server-v2vwin-6
```

This system must also have access to the NFS export domain. The `virt-v2v` command is executed on a Red Hat Enterprise Linux 6.2 server for this paper. The following section builds the command line that is used to export the virtual machines.

6.4.1 virt-v2v Command

6.4.1.1 Netrc

Before the `virt-v2v` command can export virtual machines from an ESXi server, a *.netrc* file must be created in the home directory of the user executing the `virt-v2v` command. This file contains the login authentication information for the ESXi server. The *NETRC(5)* man page contains information about the syntax of the *.netrc* file. The machine name in the file must be identical to the one used on the command line. If an internet protocol (IP) address or a fully qualified domain name are used on the command line, then they must appear in the *.netrc* file. Because the command line in this paper is using the short DNS name, that is the name used in the file. The root user is used to execute the `virt-v2v` command in this paper, so the complete path to the *.netrc* file used in this paper is */root/.netrc*. The *.netrc* file used for this paper is listed below.

```
machine v2v3-esxi
  login root
  password password
```

6.4.1.2 Options

The `virt-v2v` command requires a few options in order to export the virtual machine from the ESXi server. The first option required specifies virtual machine information. The option used to do this is the *-i* option. The information can be gathered directly from the hypervisor (*-i libvirt*) or by using an XML file (*-i libvirtxml*) that defines the virtual machine. To export a virtual machine from an ESXi server, the hypervisor method must be used. This method is also the default method and is not required to be specified on the command line. However, it is specified for the writing of this paper in order to help readability.

The location of the hypervisor must be specified as well. This is done using the *-ic* option. The location is specified using the URI format. To connect to an ESXi server, the URI specification is in the format of *esx://server_ip_or_name/*.

Some ESXi servers may contain invalid SSL security certificates. This can occur if the host name of the ESXi server has changed or if the certificate has not been signed by a Certificate Authority. This prevents the `virt-v2v` command from connecting to the server. This is the case for the ESXi server used in this paper. Adding *?no_verify=1* to the end of the URI disables certificate checking.

The incomplete command built so far looks like the following:

```
# virt-v2v -i libvirt -ic esx://v2v3-esxi/?no_verify=1
```

The method in which to store the exported image must be specified as well. This is done using the *-o* option. The `virt-v2v` command can store the images as either a libvirt guest (*-o libvirt*) or as an image to be imported into a Red Hat Enterprise Virtualization environment (*-o rhev*). Since the virtual machines are being imported into a Red Hat Enterprise Virtualization environment, the *-o rhev* option is used. This option also requires the *-os* option to be specified. The *-os* option defines the location of the Export Domain that is defined in the Red Hat Enterprise Virtualization environment.

The command line is almost complete and now looks like:

```
# virt-v2v -i libvirt -ic esx://v2v3-esxi/?no_verify=1 \
  -o rhev -os v2v3-nfs:/exports/Export
```

The `virt-v2v` command allows a default network or bridge to be defined if a network cannot be mapped between the virtualization environments. The *-b* option specifies a default bridge and the *-n* option specifies a default network. Only one of these options can be specified at a time and both take a single argument that identifies the bridge or network.

Although the networks are mapped in the */etc/virt-v2v.conf* file, it is a good idea to specify a default bridge or network in case a mapping was missed.

The default network on a Red Hat Enterprise Virtualization environment is called *rhevm*. Using this as a default network helps prevent virtual machines that have networks that were not mapped to continue to have some form of network access.

The final argument to the `virt-v2v` command is the virtual machine name. The virtual machine names are displayed in the vSphere client interface or they can be listed from the ESXi command line. The command in the following example displays the Virtual Machines Identifier (Vmid) and the Virtual Machines name using the ESXi command line.

```
# ssh v2v3-esxi vim-cmd vmsvc/getallvms | awk '{print $1 "  " $2}'

Vmid  Name
1  v2v3-W2K8-DC
2  v2v3-W7
3  v2v3-rhel
```

The following command line could be used to migrate the virtual machines in this Reference Architecture, however this Reference Architecture uses *PROFILES* created in the */etc/virt-v2v.conf* file to shorten the command line.

```
# virt-v2v -i libvirt -ic esx://v2v3-esxi/?no_verify=1 \
  -o rhev -os v2v3-nfs:/exports/Export \
  --network rhevm [VM_NAME]
```

6.4.1.3 Profiles

The `virt-v2v` command allows the use of profiles to specify the method to use to get the virtual machine information, network mappings, and output storage locations. Profiles are specified in the */etc/virt-v2v.conf* file. The following is the contents of the configuration file used for this Reference Architecture.

```
<virt-v2v>
  <profile name="v2v3-rhev">
    <method>rhev</method>
    <storage>
      v2v3-nfs:/exports/Export
    </storage>

    <network type='bridge' name='VM Network'>
      <network type='network' name='infrastructure'/>
    </network>

    <network type="default">
      <network type="network" name="rhevm"/>
    </network>
  </profile>
</virt-v2v>
```

The <method> tag specifies the output method.

The <storage> tag specifies the storage location and can also specify the format of the destination disk file and if the disk should be a sparse or preallocated type. The destination disk files will have the same format and allocation as the original files since this configuration file does not specify the format and allocation type.

Two networks are configured in the file. The first network definition maps the bridged network called VM Network on the ESXi server to the infrastructure network in the Red Hat Enterprise Virtualization environment. The second definition maps all other networks on the ESXi server to the rhevm network in the Red Hat Enterprise Virtualization environment.

With the use of the above profile, the command line used to migrate the virtual machines during the writing of this paper is:

```
# virt-v2v -p v2v3-rhev -i libvirt \
  -ic esx://v2v3-esxi/?no_verify=1 [VM_NAME]
```

6.4.2 Export

With the **`VMware Tools`** uninstalled from the virtual machines and the virtual machines shut down, the virtual machines can be exported from the ESXi server.

The virtual machines are exported using the `virt-v2v` command.

```
# virt-v2v -p v2v3-rhev -i libvirt -ic esx://v2v3-esxi/?no_verify=1 v2v3-
rhel
v2v3-rhel_v2v3-rhel: 100% [=========================================]D
0h07m48s
virt-v2v: v2v3-rhel configured with virtio drivers.

# virt-v2v -p v2v3-rhev -i libvirt -ic esx://v2v3-esxi/?no_verify=1 v2v3-
W2K8-DC
v2v3-W2K8-DC_v2v3-W2K8-DC: 100% [===================================]D
0h26m19s
virt-v2v: v2v3-W2K8-DC configured with virtio drivers.

# virt-v2v -p v2v3-rhev -i libvirt -ic esx://v2v3-esxi/?no_verify=1 v2v3-W7
v2v3-W7_v2v3-W7: 100% [=============================================]D
0h26m14s
virt-v2v: v2v3-W7 configured with virtio drivers.
```

6.5 Importing the Virtual Machines

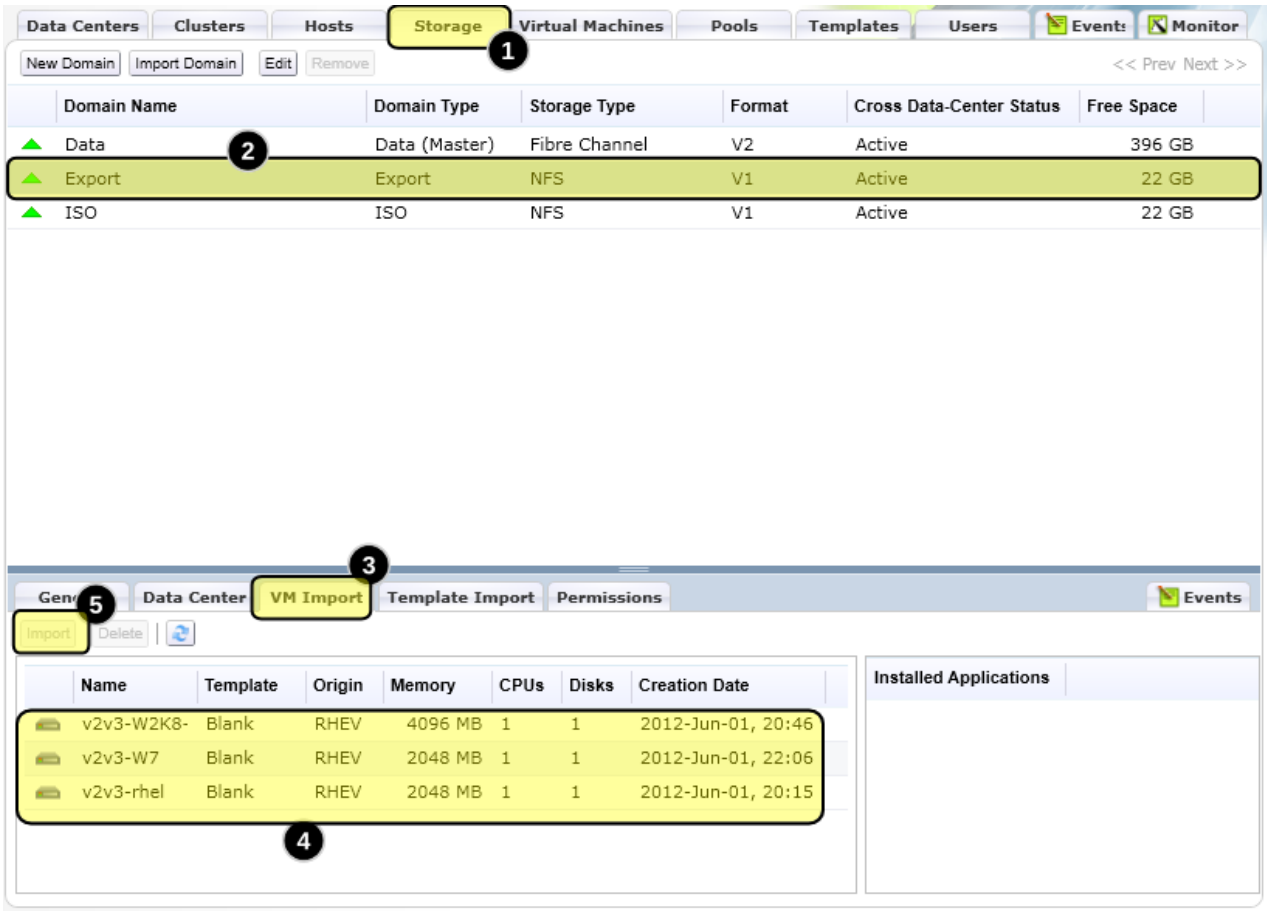
The virtual machines can be imported using either the Red Hat Enterprise Virtualization Manager graphical user interface or the REST API interface. Both methods are explained below.

6.5.1 Graphical User Interface

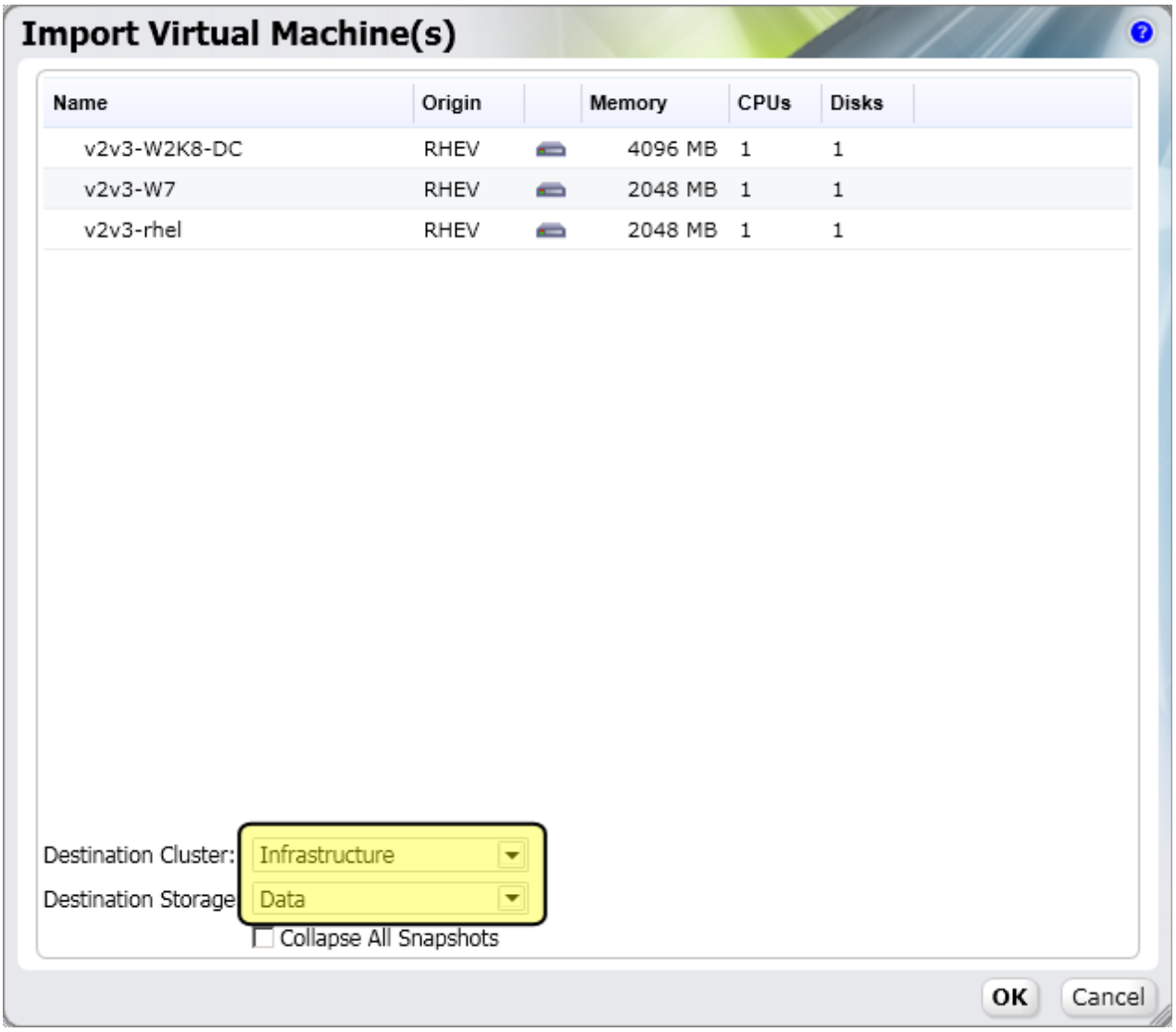
Importing virtual machines using the graphical interface is simple to do and allows importing multiple virtual machines at the same time. Open the management console on the Red Hat Enterprise Virtualization Manager and use **Figure 6.5.1.1: Import Selection** as a reference to choose the virtual machines to import.

Select on the **Storage** tab (1) to view the storage domains. Choose the **Export** export domain in the domain list (2) and a **Details** pane opens. Select on the **VM Import** tab (3) in the **Details** pane and a list of virtual machines is displayed (4). These virtual machines are ready to be imported. Select all the virtual machines by holding down the <**ctrl**> key while selecting each virtual machine.

The **Import** button (5) becomes visible. Select the **Import** button to open up the **Import VM** window, as seen in **Figure 6.5.1.3: Import Notification**.



The **Import Virtual Machine(s)** window lists the virtual machines that are selected for import. This is seen in the following figure. Choose **Infrastructure** as the **Destination Cluster** and **Data** as the **Destination Storage**. These are highlighted in the figure below. Select the **OK** button to start the import process.



A window appears indicating the virtual machines are being imported. This window is shown below. Select the **Close** button to close the window.

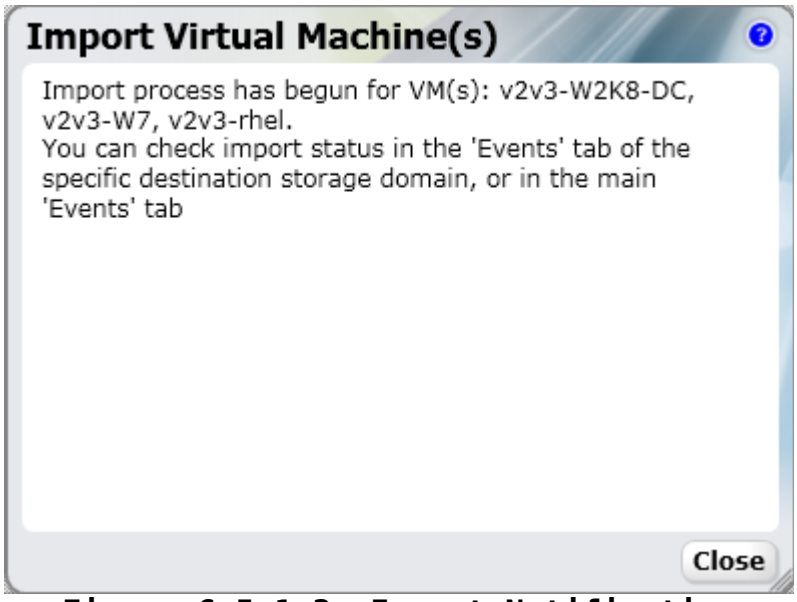
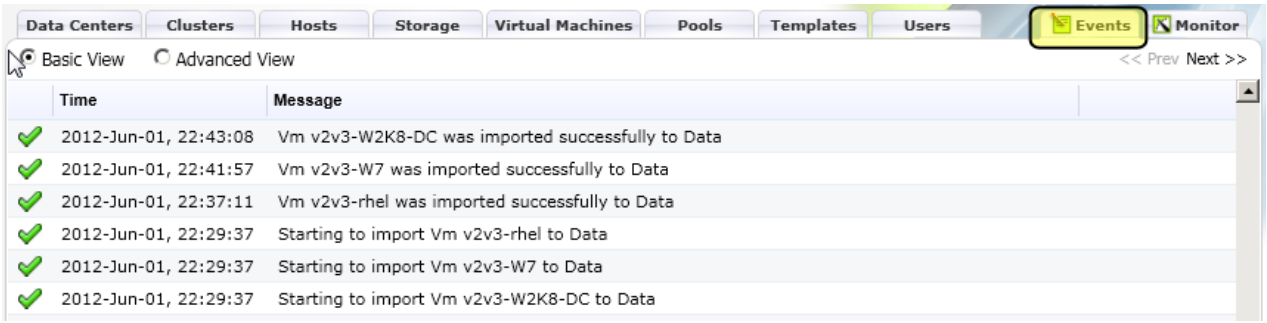
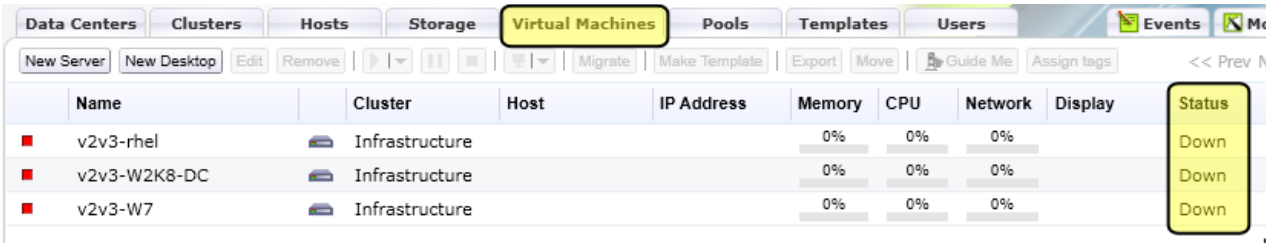


Figure 6.5.1.3: Import Notification

The status of the import process is viewed by using the **Events** Tab as shown in **Figure 6.5.1.5: Virtual Machine Status**. When the process is finished, the events log indicates the import was successful.



During the import process, the **Status** of the virtual machines is set to **Locked**. Once the process is complete, the **Status** changes to **Down** as shown in **Figure 6.5.1.5: Virtual Machine Status**.



6.5.2 REST API

The Red Hat Enterprise Virtualization Manager provides a *REST API* interface that allows access to a broad set of features within the Red Hat Enterprise Virtualization Manager. Accessing the *REST API* can be done using the **`curl`** command.

This paper discusses the REST API only briefly in an attempt to provide the information required to understand its use within this paper. An in-depth discussion of the REST API and how to access it is out of the scope of this paper. Read the *Red Hat Enterprise Virtualization 3.0 – REST API Guide*[4] for more information on the *REST API*.

Port 8443 on the Red Hat Enterprise Virtualization Manager is a secure connection. The certificate file for the server must be downloaded and used with the `curl` command.

The `curl` command can be used to download the certificate.

```
# curl -o ca.crt http://v2v3-rhevm:8080/ca.crt

 % Total  % Received % Xferd  Average Speed   Time    Time     Time  Current
                              Dload  Upload   Total   Spent    Left  Speed
102 3571 102  3571    0     0  1357k     0 --:--:-- --:--:-- --:--:-- 3487k
```

4 http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Virtualization/3.0/html/REST_API_Guide/index.html

The REST API is accessed using a URI. The following syntax is used for the URI:

https://[RHEVM_FQDN]:8443/api/[ENTRY_POINT].

Where:

[RHEVM_FQDN] is the fully qualified domain name of the Red Hat Enterprise Virtualization Manager

[ENTRY_POINT] is a reference to the information in the *REST API*.

The curl command to access the *REST API* is in the following format:

curl --silent --cacert ***[CERT]*** \

--header “Content-Type: application/xml” \

--user “admin@internal:***[PASSWORD]***” \

--request “***[TYPE]***” \

--data “***[XML]***” \

URI

Where:

[CERT] is the name of the certificate file downloaded from the Red Hat Enterprise Virtualization Manager.

[PASSWORD] is the password used for the internal admin account.

[TYPE] is the http method type of *GET*, *PUT*, or *POST*. To retrieve information from the REST API, a *GET* call method is made. Modifying information in the *REST API* is done using either the *POST* or *PUT* methods.

[XML] is the XML data used to submit information to the *REST API*. The --data option is not used with *GET* methods.

[URI] is the URI of the Red Hat Enterprise Virtualization Manager, port, and entry point.

Resources within the environment are assigned a UUID. The UUID of the storage domains and virtual machine files are needed to import the virtual machines. The following command retrieves the UUIDs of the storage domains.

```
# curl --silent --cacert ca.crt --header "Content-Type: application/xml" \
> --user "admin@internal:[Password]" \
> --request "GET" \
> https://v2v3-rhevm.cloud.lab.eng.bos.redhat.com:8443/api/storagedomains \
> | grep -E "domain id|name"

    <storage_domain id="61855efc-3453-4c41-b414-21e61dca358f"
href="/api/storagedomains/61855efc-3453-4c41-b414-21e61dca358f">
        <name>Data</name>
    <storage_domain id="2a88e486-b1e2-4b96-852e-36b7f44a1d5e"
href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e">
        <name>Export</name>
    <storage_domain id="63940dbe-38c2-4462-94b1-68403ca431e6"
href="/api/storagedomains/63940dbe-38c2-4462-94b1-68403ca431e6">
        <name>ISO</name>
```

Using the UUID of the Export domain, the UUIDs of the virtual machines ready to be imported are retrieved.

```
# curl --silent --cacert ca.crt --header "Content-Type: application/xml" \
> --user "admin@internal:[Password]" \
> --request "GET" \
>
https://v2v3-rhevm.cloud.lab.eng.bos.redhat.com:8443/api/storagedomains/2a88
e486-b1e2-4b96-852e-36b7f44a1d5e/vms \
> | grep -E "vm id|name"

    <vm id="21f48683-7ce8-4375-933d-d2e34b180fd6"
href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/21f48683-
7ce8-4375-933d-d2e34b180fd6">
        <name>v2v3-W2K8-DC</name>
    <vm id="da3de408-edea-488a-8e34-85d14282bdba"
href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/da3de408-
edea-488a-8e34-85d14282bdba">
        <name>v2v3-W7</name>
    <vm id="f73748bb-02af-4012-af19-70cf995cb90b"
href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/f73748bb-
02af-4012-af19-70cf995cb90b">
        <name>v2v3-rhel</name>
```

The following commands start the import of the virtual machines. The output returned from the *REST API* indicates a state of **pending**.

```
# curl --silent --cacert ca.crt \
> --header "Content-Type: application/xml" \
> --user "admin@internal:[Password]" \
> --request "POST" \
> --data "
> <action>
>   <storage_domain>
>     <name>Data</name>
>   </storage_domain>
>
>   <cluster>
>     <name>Infrastructure</name>
>   </cluster>
> </action>
> " \
>
https://v2v3-rhevm.cloud.lab.eng.bos.redhat.com:8443/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/21f48683-7ce8-4375-933d-d2e34b180fd6/import

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<action id="6d505ce8-16cb-4abf-806a-df5ff8dd072d"
href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/21f48683-7ce8-4375-933d-d2e34b180fd6/import/6d505ce8-16cb-4abf-806a-df5ff8dd072d">
    <link rel="parent" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/21f48683-7ce8-4375-933d-d2e34b180fd6"/>
    <link rel="replay" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/21f48683-7ce8-4375-933d-d2e34b180fd6/import"/>
    <async>true</async>
    <storage_domain>
        <name>Data</name>
    </storage_domain>
    <cluster>
        <name>Infrastructure</name>
    </cluster>
    <status>
        <state>pending</state>
    </status>
</action>

# curl --silent --cacert ca.crt \
> --header "Content-Type: application/xml" \
> --user "admin@internal:[Password]" \
> --request "POST" \
> --data "
> <action>
>   <storage_domain>
>     <name>Data</name>
>   </storage_domain>
>
>   <cluster>
```

```
>     <name>Infrastructure</name>
>   </cluster>
> </action>
> " \
>
https://v2v3-rhevm.cloud.lab.eng.bos.redhat.com:8443/api/storagedomains/2a88
e486-b1e2-4b96-852e-36b7f44a1d5e/vms/da3de408-edea-488a-8e34-
85d14282bdba/import

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<action id="6e58715a-95c7-4d65-b58f-0d1690f06409"
href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/da3de408-
edea-488a-8e34-85d14282bdba/import/6e58715a-95c7-4d65-b58f-0d1690f06409">
    <link rel="parent" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-
36b7f44a1d5e/vms/da3de408-edea-488a-8e34-85d14282bdba"/>
    <link rel="replay" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-
36b7f44a1d5e/vms/da3de408-edea-488a-8e34-85d14282bdba/import"/>
    <async>true</async>
    <storage_domain>
        <name>Data</name>
    </storage_domain>
    <cluster>
        <name>Infrastructure</name>
    </cluster>
    <status>
        <state>pending</state>
    </status>
</action>

# curl --silent --cacert ca.crt \
> --header "Content-Type: application/xml" \
> --user "admin@internal:[Password]" \
> --request "POST" \
> --data "
> <action>
>   <storage_domain>
>     <name>Data</name>
>   </storage_domain>
>
>   <cluster>
>     <name>Infrastructure</name>
>   </cluster>
> </action>
> " \
>
https://v2v3-rhevm.cloud.lab.eng.bos.redhat.com:8443/api/storagedomains/2a88
e486-b1e2-4b96-852e-36b7f44a1d5e/vms/f73748bb-02af-4012-af19-
70cf995cb90b/import

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<action id="fe84dcbb-ba48-48a8-b587-6925c9bc0b6e"
href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/f73748bb-
02af-4012-af19-70cf995cb90b/import/fe84dcbb-ba48-48a8-b587-6925c9bc0b6e">
    <link rel="parent" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-
36b7f44a1d5e/vms/f73748bb-02af-4012-af19-70cf995cb90b"/>
```

```
    <link rel="replay" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-
36b7f44a1d5e/vms/f73748bb-02af-4012-af19-70cf995cb90b/import"/>
    <async>true</async>
    <storage_domain>
        <name>Data</name>
    </storage_domain>
    <cluster>
        <name>Infrastructure</name>
    </cluster>
    <status>
        <state>pending</state>
    </status>
</action>
```

After a few moments, the state of the virtual machines change to **image_locked** and eventually to a state of **down**. The **image_locked** state indicates the import process has started. Upon completion of the import process, the state changes to **down**.

The status of the import process is listed as the status of the virtual machine. The virtual machine UUID is used to query the status of the import process.

```
# curl --silent --cacert ca.crt --header "Content-Type: application/xml" \
> --user "admin@internal:[Password]" \
> --request "GET" \
>https://v2v3-rhevm.cloud.lab.eng.bos.redhat.com:8443/api/vms/f73748bb-02af-
4012-af19-70cf995cb90b | grep -v link

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vm id="f73748bb-02af-4012-af19-70cf995cb90b" href="/api/vms/f73748bb-02af-
4012-af19-70cf995cb90b">
    <name>v2v3-rhel</name>
    <description>Imported with virt-v2v</description>
    <actions>
    </actions>
    <type>server</type>
    <status>
        <state>image_locked</state>
    </status>
    <memory>2147483648</memory>
    <cpu>
        <topology cores="1" sockets="1"/>
    </cpu>
    <os type="other_linux">
        <boot dev="hd"/>
    </os>
    <high_availability>
        <enabled>false</enabled>
        <priority>0</priority>
    </high_availability>
    <display>
        <type>vnc</type>
        <monitors>1</monitors>
    </display>
    <cluster id="e312e4fa-a15f-11e1-8663-525400f758d2"
href="/api/clusters/e312e4fa-a15f-11e1-8663-525400f758d2"/>
    <template id="00000000-0000-0000-0000-000000000000"
href="/api/templates/00000000-0000-0000-0000-000000000000"/>
    <creation_time>2012-06-18T19:57:44.635-05:00</creation_time>
    <origin>rhev</origin>
    <stateless>false</stateless>
    <placement_policy>
        <affinity>migratable</affinity>
    </placement_policy>
    <memory_policy>
        <guaranteed>2147483648</guaranteed>
    </memory_policy>
    <usb>
        <enabled>false</enabled>
    </usb>
</vm>
```

6.6 Performing Post Migration Tasks

A few more tasks are required to be performed after the virtual machines are imported into the Red Hat Enterprise Virtualization environment. These include booting the virtual machines the first time in the new environment, testing the functionality of the virtual machines, and cleaning up any temporary files that were created and removing the virtual machines form the old environment.

6.6.1 First Boot

After the import process is complete and the virtual machines are powered on for the first time, the virtual machine operating system may perform additional tasks during its first boot in the new environment.

Windows based virtual machines may require a reboot immediately after the first post-migration boot. This is due to the *VirtIO* drivers being installed. The guest operating system indicates if a reboot is required. Network configuration may require reconfiguration since Windows operating systems treat the new network drivers as a new network adapter and do not move the network configuration over from the previous adapter. It is also possible that the virtual machine must be re-activated with Microsoft.

6.6.2 Windows Virtual Machines

In the **Virtual Machines** tab on the Red Hat Enterprise Virtualization Manager, use the right mouse button to select the **v2v3-W2K8-DC** virtual machine and choose **Run**. After a few moments, use the right mouse button to select the **v2v3-W2K8-DC** virtual machine again and choose **Console**.

Let the virtual machine power up and then log into it using the administrator account. The virtual machine indicates it is performing some configuration. After a few moments, it indicates it requires a restart. Restart the system now.

After reboot, the network configuration requires reconfiguration since the adapter changed.

Select on the **Start** button | **Control Panel** | **Network and Internet** | **Network Sharing Center**. Select on the **Change adapter settings** link. A new adapter is displayed called **Local Area Connection 2**. Use the right mouse button to select the new adapter and choose **Properties** from the drop down menu.

Using the information gathered in **Table 5.5-1: Virtual Machine Notes**, verify and fix the network configuration. A window may appear indicating another adapter is configured to use this IP address. The same window asks if the configuration for the other adapter should be removed. If this occurs, select the **Yes** button to remove the other adapter configuration and to continue. Close the windows and reboot the system if any network changes were made or if the system indicates a reboot is necessary.

The networking should now be operational. Use the `ping` command from a command prompt to test the basic networking and `nslookup` to verify DNS.

```
C:\> ping /n 3 10.16.143.254

Pinging 10.16.143.254 with 32 bytes of data:
Reply from 10.16.143.254: bytes=32 time=1ms TTL=255
Reply from 10.16.143.254: bytes=32 time=<1ms TTL=255
Reply from 10.16.143.254: bytes=32 time=<1ms TTL=255

Ping statistics for 10.16.143.254:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\> nslookup www.redhat.com
Server:  v2v3-w2k8-dc.v2v3.cloud.lab.eng.bos.redhat.com
Address:  10.16.136.68

Non-authoritative answer:
Name:    e1890.b.akamaiedge.net
Address:  96.6.5.214
Aliases:  www.redhat.com
```

Since Windows based virtual machines may require reactivation after the import process, select the **Start** button and use the right mouse button to select **Computer**. Choose **Properties** from the drop down menu. The properties window has a section titled **Windows Activation**. This indicates whether the virtual machine requires reactivation and how long until the current activation expires. Select the **Activate Windows** link and follow the instructions to reactivate the virtual machine.

The date and time as reported by the virtual machine is verified to be correct. This is necessary for proper functionality within the domain.

The above steps are also used to power on and configure the **v2v3-W7** virtual machine the first time. For brevity, these steps are not repeated.

6.6.3 Red Hat Enterprise Linux Virtual Machines

Red Hat Enterprise Linux based virtual machines perform an SELinux relabel of the file systems if SELinux is enabled. The network information should not require re-entry. Red Hat Enterprise Linux should move the network configuration over to the new adapter.

Start the `v2v3-rhel` virtual machine and open its console. Upon boot, the Red Hat Enterprise Linux virtual machine initiates an SELinux relabeling of the file systems. This is normal and must be allowed to finish.

The networking in the Red Hat Enterprise Linux virtual machine does not require reconfiguration. This is verified using the `ping` and `nslookup` commands.

```
# ping -c 3 10.16.143.254
PING 10.16.143.254 (10.16.143.254) 56(84) bytes of data.
64 bytes from 10.16.143.254: icmp_seq=1 ttl=255 time=0.517 ms
64 bytes from 10.16.143.254: icmp_seq=2 ttl=255 time=0.624 ms
64 bytes from 10.16.143.254: icmp_seq=3 ttl=255 time=0.581 ms

--- 10.16.143.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.517/0.574/0.624/0.043 ms

# nslookup www.redhat.com
Server:    10.16.143.68
Address:   10.16.143.68#53

Non-authoritative answer:
www.redhat.com   canonical name = wildcard.redhat.com.edgekey.net.
Name: e1890.b.akamaiedge.net
Address: 69.192.39.214
```

The date is verified in order to ensure functionality within the domain. The `date` command in the output below shows the date and time to be correct.

```
# date
Fri Jun 15 15:24:44 CDT 2012
```

6.6.4 Testing

The virtual machines should be tested to make sure they are still functioning as desired.

The v2v3-W2K8-DC is a domain controller and is providing a *DNS* service, a *KERBEROS* service, and an *ACTIVE DIRECTORY* authentication service.

The DNS service is tested by using the `dig` command while logged into the v2v3-rhel server. The `dig` command takes an argument that begins with an *asperand (@)* and is followed by an IP address of a DNS server. This tells the `dig` command which DNS server to query when asking for name resolution. The `dig` command is used twice to check the DNS service. The first time the `dig` command is issued, it queries the DNS server for the local entry on the domain controller for the *v2v3-W2K8-DC.v2v3.cloud.lab.eng.bos.redhat.com* server. This checks to see that the local DNS resolution is working. The second time the `dig` command is issued, it checks to see if the DNS service correctly forwards unknown request to the next hop DNS server.

```
# dig @10.16.136.68 v2v3-W2K8-DC.v2v3.cloud.lab.eng.bos.redhat.com

; <<>> DiG 9.7.3-P3-RedHat-9.7.3-8.P3.el6_2.3 <<>> @10.16.136.68 v2v3-W2K8-
DC.v2v3.cloud.lab.eng.bos.redhat.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56476
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;v2v3-W2K8-DC.v2v3.cloud.lab.eng.bos.redhat.com.   IN A

;; ANSWER SECTION:
v2v3-W2K8-DC.v2v3.cloud.lab.eng.bos.redhat.com.    3600 IN    A 10.16.136.68

;; Query time: 0 msec
;; SERVER: 10.16.136.68#53(10.16.136.68)
;; WHEN: Fri Jun 15 15:30:58 2012
;; MSG SIZE  rcvd: 80


# dig @10.16.136.68 www.redhat.com

; <<>> DiG 9.7.3-P3-RedHat-9.7.3-8.P3.el6_2.3 <<>> @10.16.136.68
www.redhat.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18917
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.redhat.com.            IN    A

;; ANSWER SECTION:
www.redhat.com.        600   IN    CNAME wildcard.redhat.com.edgekey.net.
```

```
wildcard.redhat.com.edgekey.net. 15883 IN CNAME
      wildcard.redhat.com.edgekey.net.globalredir.akadns.net.
wildcard.redhat.com.edgekey.net.globalredir.akadns.net.  337 IN CNAME
e1890.b.akamaiedge.net.
e1890.b.akamaiedge.net.     20    IN    A     23.15.23.214

;; Query time: 87 msec
;; SERVER: 10.16.136.68#53(10.16.136.68)
;; WHEN: Fri Jun 15 15:28:41 2012
;; MSG SIZE  rcvd: 191
```

The Active Directory service is checked by using a domain user account to log into the v2v3-W7 virtual machine. This also tests if the v2v3-W7 virtual machine is still functioning as a domain member.

Login to the v2v3-W7 using the domain user account **refarch**. The user account **refarch** does not exist on the v2v3-W7 virtual machine. A successful login indicates the domain controller is functioning and the v2v3-W7 system is acting as a member of the domain.

The v2v3-rhel virtual machine uses the v2v3-W2K8-DC domain controller for local account authentication. This is verified by issuing the `ssh` command on the v2v3-v2v migration server to connect to the v2v3-rhel virtual machine using a domain account.

```
# ssh refarch@v2v3-rhel.cloud.lab.eng.bos.redhat.com
refarch@v2v3-rhel.cloud.lab.eng.bos.redhat.com's password: [PASSWORD]
Last login: Fri Jun 15 15:45:12 2012 from v2v3-v2v
Kickstarted on 2012-06-14

$ whoami
refarch

$ pwd
/home/V2V3/refarch
```

The successful login indicates that local account authentication is using the domain controller.

6.6.5 Cleaning Up

The migration process leaves virtual machine configuration files and disk images on the export domain and leaves the virtual machines in the VMware environment. Delete the images from the export domain using the **Delete** button in the **Import** tab on the Red Hat Enterprise Virtualization Manager. See **Figure 6.5.1.1: Import Selection**.

After the virtual machines are verified to function as desired in the Red Hat Enterprise Virtualization environment, they can be deleted from the VMware environment and the Export Domain.

7 Automating the Process

Automating the migration process can be done using simple bash scripts, having `ssh` access to the ESXi server, and using the *REST API*.

Prior to starting the migration, the virtual machines must have **`Vmware Tools`** uninstalled and they must be shut down. After these are done, the `vim-cmd` is used on the ESXi server via `ssh` to make sure the virtual machines are powered off.

The functions access an associative array called **config** that contains the values of the options passed to the script on the command line.

The following `bash` function retrieves the list of virtual machines and their respective virtual machine ids from the ESXi server and assigns the values to an associative array called **esxVM**. The function then loops through the list of virtual machine names and check the power state. If the virtual machine is powered on, the counter **vms_on** is incremented. If the force option was passed to the function, then the function will try to power off the virtual machine.

The function then returns the number of virtual machines that were powered on when the function was called.

```
declare -A config esxVM

esxi_vms() {
  state=$1

  echo "Checking power state of virtual machines."
  # Get virtual machine names and vmids
  while read vmid vmname extra
  do
    esxVM[${vmname}]=${vmid}
  done < <(
    ssh ${config[esx_srv]} vim-cmd vmsvc/getallvms 2>/dev/null | grep -v
Vmid
    )

  vms_on=0

  # Check power state of virtual machine
  for i in ${!esxVM[@]}
  do
    echo -e "  -- ${i}: \c"

    ssh ${config[esx_srv]} vim-cmd vmsvc/power.getstate ${esxVM[${i}]}
2>/dev/null | grep -q off
    RC=$?
    if [ $RC -ne 0 ]
    then
      ((vms_on++))

      if [ "${state}" == "force" ]
      then
```

```
        ssh ${config[esx_srv]} vim-cmd vmsvc/power.off ${esxVM[${i}]}
2>/dev/null
      else
        echo "on"
      fi
    else
      echo "off"
    fi
  done

  return ${vms_on}
  }
```

The following function loops through the virtual machine names and executes the `virt-v2v` command for each virtual machine.

```
v2v_vms() {
  echo "Executing virt-v2v against the virtual machines"
  for i in ${!esxVM[@]}
  do
    virt-v2v -p ${config[profile]} -i libvirt -ic esx://${config[esx_srv]}/?
no_verify=1 ${i}
  done
  }
```

The final function imports the virtual machines into the Red Hat Enterprise Virtualization environment. The function assumes the certificate file for the Red Hat Enterprise Virtualization Manager has been downloaded. The function queries the *REST API* to get the UUID of the Export storage domain. The function queries the *REST API* for the UUIDs of the virtual machines in the Export domain. The function loops through the UUIDs and imports the virtual machines into the Red Hat Enterprise Virtualization environment.

```
rhev_import() {

  echo "Importing the virtual machines."
  domainid=$( curl --silent --cacert ${config[cacert]}  \
    --header "Content-Type: application/xml" \
    --user "${config[rhevm_user]}:${config[rhevm_password]}"         \
    --request "GET"                                \
    https://${config[rhevm_srv]}:8443/api/storagedomains?search=name
%3dExport \
    | xpath /storage_domains/storage_domain/@id 2>/dev/null \
    | sed -e 's/id="//' -e 's/"//' -e 's/ //'
    )

    uuids=$( curl --silent --cacert ${config[cacert]}  \
    --header "Content-Type: application/xml" \
    --user "${config[rhevm_user]}:${config[rhevm_password]}"         \
    --request "GET"                                \
    https://${config[rhevm_srv]}:8443/api/storagedomains/${domainid}/vms \
    | xpath /vms/vm/@id 2>/dev/null \
    | sed -e 's/id="//g' -e 's/"//g'
    )
```

```
  for i in ${uuids}
  do
    echo "Importing VM ${i}"
    curl --silent --cacert ca.crt \
    --header "Content-Type: application/xml" \
    --user "${config[rhevm_user]}:${config[rhevm_password]}" \
    --request "POST" \
    --data "
      <action>
        <storage_domain>
          <name>Data</name>
        </storage_domain>
        <cluster>
          <name>Infrastructure</name>
        </cluster>
      </action>"  \
    https://${config[rhevm_srv]}:8443/api/storagedomains/${domainid}/vms/$
{i}/import
   done
  }
```

Each function is executed in order within a bash script. The complete bash script is listed in **Appendix C: Scripts** and is available on the Red Hat Customer Portal[5]. The output generated to migrate the three virtual machines used in this Reference Architecture is displayed below.

```
# ./migrate_v2v3.sh -e v2v3-esxi -o v2v3-rhev -c ca.crt -r v2v3-
rhevm.cloud.lab.eng.bos.redhat.com -u "admin@internal" -p [PASSWORD] -f

Checking power state of virtual machines.
  -- v2v3-rhel: Powering off VM:
  -- v2v3-W7: Powering off VM:
  -- v2v3-W2K8-DC: Powering off VM:
Executing virt-v2v against the virtual machines
v2v3-rhel_v2v3-rhel: 100% [=========================================]D
0h08m26s
virt-v2v: v2v3-rhel configured with virtio drivers.
v2v3-W7_v2v3-W7: 100% [=========================================]D
0h27m20s
virt-v2v: v2v3-W7 configured with virtio drivers.
v2v3-W2K8-DC_v2v3-W2K8-DC: 100% [=================================]D
0h26m49s
virt-v2v: v2v3-W2K8-DC configured with virtio drivers.
Importing the virtual machines.
Importing VM 0672e777-f046-4b5c-b495-531aeb6967cd
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<action id="191a41e7-5b37-4abc-ba02-935307fb1901"
href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/0672e777-
f046-4b5c-b495-531aeb6967cd/import/191a41e7-5b37-4abc-ba02-935307fb1901">
    <link rel="parent" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-
36b7f44a1d5e/vms/0672e777-f046-4b5c-b495-531aeb6967cd"/>
```

5 https://access.redhat.com/sites/default/files/migrate_esx2rhev.tgz

```
    <link rel="replay" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-
36b7f44a1d5e/vms/0672e777-f046-4b5c-b495-531aeb6967cd/import"/>
    <async>true</async>
    <storage_domain>
        <name>Data</name>
    </storage_domain>
    <cluster>
        <name>Infrastructure</name>
    </cluster>
    <status>
        <state>pending</state>
    </status>
</action>
Importing VM 2882f3a2-72dc-4e6e-aea3-6a6a007efa43
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<action id="88b51ea6-3ffe-4aac-a179-4df85d870b2f"
href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/2882f3a2-
72dc-4e6e-aea3-6a6a007efa43/import/88b51ea6-3ffe-4aac-a179-4df85d870b2f">
    <link rel="parent" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-
36b7f44a1d5e/vms/2882f3a2-72dc-4e6e-aea3-6a6a007efa43"/>
    <link rel="replay" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-
36b7f44a1d5e/vms/2882f3a2-72dc-4e6e-aea3-6a6a007efa43/import"/>
    <async>true</async>
    <storage_domain>
        <name>Data</name>
    </storage_domain>
    <cluster>
        <name>Infrastructure</name>
    </cluster>
    <status>
        <state>pending</state>
    </status>
</action>
Importing VM 88b485a0-39eb-4779-a431-cd1e52c3c6a5
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<action id="02d82835-078d-4f77-89be-145ea26b1801"
href="/api/storagedomains/2a88e486-b1e2-4b96-852e-36b7f44a1d5e/vms/88b485a0-
39eb-4779-a431-cd1e52c3c6a5/import/02d82835-078d-4f77-89be-145ea26b1801">
    <link rel="parent" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-
36b7f44a1d5e/vms/88b485a0-39eb-4779-a431-cd1e52c3c6a5"/>
    <link rel="replay" href="/api/storagedomains/2a88e486-b1e2-4b96-852e-
36b7f44a1d5e/vms/88b485a0-39eb-4779-a431-cd1e52c3c6a5/import"/>
    <async>true</async>
    <storage_domain>
        <name>Data</name>
    </storage_domain>
    <cluster>
        <name>Infrastructure</name>
    </cluster>
    <status>
        <state>pending</state>
    </status>
</action>
```

The functions listed above and the complete script are samples only and are used to show how simply the migration process can be automated. Scripts in a production environment should perform more error checking and logging.

8 Conclusion

This paper demonstrated the migration of three virtual machines configured to use active directory authentication with each other. The migrations were performed from a VMware ESXi server to a Red Hat Enterprise Virtualization environment using the `virt-v2v` utility and the Red Hat Enterprise Virtualization Manager.

This paper demonstrated the steps required before starting the migration such as mapping and creating networks, creating export domains, and gathering information about the ESXi environment and virtual machines.

Committing snapshots and uninstalling VMware Tools was also demonstrated before the `virt-v2v` utility was used to export the virtual machines from the ESXi server.

Importing the virtual machines into the Red Hat Enterprise Virtualization environment and configuring after first boot in the new environment was demonstrated. The virtual machines were tested for correct functionality.

Sample scripts were also used to show how the migration process can easily be automated using bash, the VMware vim-cmd command, and the REST API.

As seen from this paper, virtual machines can easily be migrated from a VMware ESXi server to a Red Hat Enterprise Virtualization solution with the correct tools and careful planning. By planning the migration process, the impact to the existing production systems can be limited.

Appendix A: Securing NFS using IPTables

The NFS version 3 service uses multiple inbound ports to function properly. Some of these ports are static and others are random by default. The random ports make it difficult to create firewall rules on systems running the NFS service. However, the NFS service can be configured to use static ports instead of random ports.

This is accomplished by editing the */etc/sysconfig/nfs* configuration file and changing four configuration options that cause the NFS service to use static ports instead of random ports. The following table lists the ports and configuration options used by the NFS service.

Daemon	Port	Protocol	Configuration Option
portmap	111	UDP, TCP	Not needed
rpc.nfsd	2049	UDP, TCP	Not needed
rpc.lockd	Random	UDP, TCP	LOCKD_TCPPORT LOCKD_UDPPORT
rpc.mountd	Random	UDP, TCP	MOUNTD_PORT
rpc.statd	Random	UDP, TCP	STATD_PORT

Table 8-1: NFS Service Ports

Three static ports must be identified to use for the random ports listed in the above table. The netstat command can be used to list the ports already in use by the system. The options used are listed below.

Command options:

--tcp	Show tcp sockets.
--udp	Show udp sockets.
--listening	Show listening sockets.
--numeric	Show numerical addresses and ports instead of mapping information to host names and service names.
--program	Show the process id and program name that owns the socket.

The output below shows the ports that are currently in use. Randomly chosen ports 50000, 50001, and 50002 are free for both UDP and TCP traffic. They are used for the static ports required to help secure nfs.

```
# netstat --tcp --udp --listening --numeric --program
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State   PID/Program name
tcp        0      0 0.0.0.0:22    0.0.0.0:*       LISTEN  1350/sshd
tcp        0      0 127.0.0.1:25  0.0.0.0:*       LISTEN  1426/master
tcp        0      0 0.0.0.0:2049  0.0.0.0:*       LISTEN  -
tcp        0      0 0.0.0.0:111   0.0.0.0:*       LISTEN  1199/rpcbind
tcp        0      0 :::22         :::*            LISTEN  1350/sshd
tcp        0      0 ::1:25        :::*            LISTEN  1426/master
tcp        0      0 :::111        :::*            LISTEN  1199/rpcbind
udp        0      0 0.0.0.0:111   0.0.0.0:*               1199/rpcbind
udp        0      0 0.0.0.0:950   0.0.0.0:*               1199/rpcbind
udp        0      0 0.0.0.0:68    0.0.0.0:*               1117/dhclient
udp        0      0 0.0.0.0:976   0.0.0.0:*               1224/rpc.statd
udp        0      0 :::111        :::*                    1199/rpcbind
udp        0      0 :::950        :::*                    1199/rpcbind
```

Edit the */etc/sysconfig/nfs* file and make sure the following configuration lines exist.

```
# Port rpc.mountd should listen on.
MOUNTD_PORT=50000

# Port rpc.statd should listen on.
STATD_PORT=50001

# TCP port rpc.lockd should listen on.
LOCKD_TCPPORT=50002

# UDP port rpc.lockd should listen on.
LOCKD_UDPPORT=50002
```

The **nfs** and **nfslock** services must be restarted and configured to start at boot.

```
# service nfslock restart
Stopping NFS locking:                                      [  OK  ]
Stopping NFS statd:                                        [  OK  ]
Starting NFS statd:                                        [  OK  ]

# service nfs restart
Shutting down NFS mountd:                                  [  OK  ]
Shutting down NFS daemon:                                  [  OK  ]
Shutting down NFS services:                                [  OK  ]
Starting NFS services:                                     [  OK  ]
Starting NFS daemon:                                       [  OK  ]
Starting NFS mountd:                                       [  OK  ]

# chkconfig nfslock on

# chkconfig nfs on
```

The `netstat` command now shows the NFS services are running on the static ports.

```
# netstat --tcp --udp --listening --numeric --program
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State   PID/Program name
tcp        0      0 0.0.0.0:22    0.0.0.0:*       LISTEN  1350/sshd
tcp        0      0 127.0.0.1:25  0.0.0.0:*       LISTEN  1426/master
tcp        0      0 0.0.0.0:2049  0.0.0.0:*       LISTEN  -
tcp        0      0 0.0.0.0:111   0.0.0.0:*       LISTEN  1199/rpcbind
tcp        0      0 0.0.0.0:50000 0.0.0.0:*       LISTEN  1324/rpc.mountd
tcp        0      0 0.0.0.0:50001 0.0.0.0:*       LISTEN  1224/rpc.statd
tcp        0      0 0.0.0.0:50002 0.0.0.0:*       LISTEN  -
tcp        0      0 :::22         :::*            LISTEN  1350/sshd
tcp        0      0 ::1:25        :::*            LISTEN  1426/master
tcp        0      0 :::2049       :::*            LISTEN  -
tcp        0      0 :::111        :::*            LISTEN  1199/rpcbind
tcp        0      0 :::50000      :::*            LISTEN  1324/rpc.mountd
tcp        0      0 :::50001      :::*            LISTEN  1224/rpc.statd
tcp        0      0 :::50002      :::*            LISTEN  -
udp        0      0 0.0.0.0:111   0.0.0.0:*               1199/rpcbind
udp        0      0 0.0.0.0:2049  0.0.0.0:*               -
udp        0      0 0.0.0.0:950   0.0.0.0:*               1199/rpcbind
udp        0      0 0.0.0.0:68    0.0.0.0:*               1117/dhclient
udp        0      0 0.0.0.0:50000 0.0.0.0:*               1324/rpc.mountd
udp        0      0 0.0.0.0:976   0.0.0.0:*               1224/rpc.statd
udp        0      0 0.0.0.0:50001 0.0.0.0:*               1224/rpc.statd
udp        0      0 0.0.0.0:50002 0.0.0.0:*               -
udp        0      0 :::111        :::*                    1199/rpcbind
udp        0      0 :::2049       :::*                    -
udp        0      0 :::950        :::*                    1199/rpcbind
udp        0      0 :::50000      :::*                    1324/rpc.mountd
udp        0      0 :::50001      :::*                    1224/rpc.statd
udp        0      0 :::50002      :::*                    -
```

Now that the NFS services are using static ports, the IPTables firewall can be configured to allow traffic to pass.

The following shows the iptables INPUT chain. This chain is not allowing traffic on the ports used by NFS.

```
# iptables --numeric --list INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source     destination
ACCEPT     all  --  0.0.0.0/0  0.0.0.0/0    state RELATED,ESTABLISHED
ACCEPT     icmp --  0.0.0.0/0  0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0  0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0  0.0.0.0/0    state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0  0.0.0.0/0    reject-with icmp-host-prohibited
```

The following bash loop adds entries to the INPUT chain of iptables that allow the NFS connections to function.

```
# for i in 50000 50001 50002 2049 111
> do
> iptables -I INPUT -m udp -p udp --dport $i -s 10.16.136.0/21 -j ACCEPT
> iptables -I INPUT -m tcp -p tcp --dport $i -s 10.16.136.0/21 -j ACCEPT
> done
```

The iptables INPUT chain is now configured to accept connection on the ports used by NFS.

```
# iptables –numeric –list INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  10.16.136.0/21       0.0.0.0/0           tcp dpt:111
ACCEPT     udp  --  10.16.136.0/21       0.0.0.0/0           udp dpt:111
ACCEPT     tcp  --  10.16.136.0/21       0.0.0.0/0           tcp dpt:2049
ACCEPT     udp  --  10.16.136.0/21       0.0.0.0/0           udp dpt:2049
ACCEPT     tcp  --  10.16.136.0/21       0.0.0.0/0           tcp dpt:50002
ACCEPT     udp  --  10.16.136.0/21       0.0.0.0/0           udp dpt:50002
ACCEPT     tcp  --  10.16.136.0/21       0.0.0.0/0           tcp dpt:50001
ACCEPT     udp  --  10.16.136.0/21       0.0.0.0/0           udp dpt:50001
ACCEPT     tcp  --  10.16.136.0/21       0.0.0.0/0           tcp dpt:50000
ACCEPT     udp  --  10.16.136.0/21       0.0.0.0/0           udp dpt:50000
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state
RELATED,ESTABLISHED
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp
dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with
icmp-host-prohibited
```

After the iptables rules are verified, they are saved using the `service iptables save` command.

```
# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[  OK  ]
```

Appendix B: Enabling ssh Access to the ESXi Server

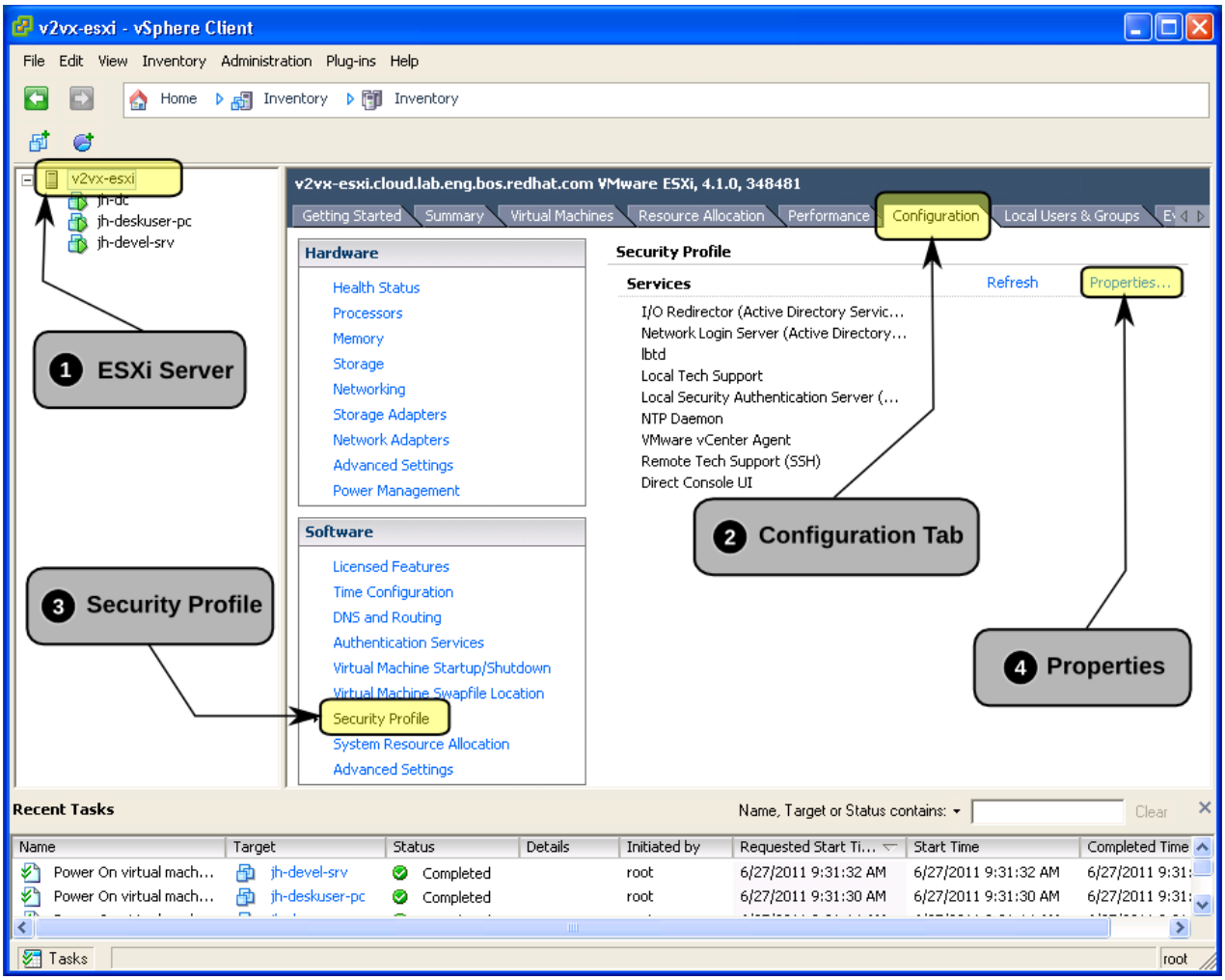
The ability to access the ESXi server using `ssh` can be enabled on the ESXi server. This allows information about the virtual machines and the ESXi environment to be easily accessed and the virtual machine environment to be reconfigured. This is not necessary for the export process to work correctly.

The **`ssh`** access can be enabled by using the root credentials and logging into the ESXi server using the **`vSphere client`**. Connect to the ESXi hypervisor using the vSphere client, using the login credentials for the root account.

After the vSphere Client opens, a window layout similar to **Figure 6.1.1: Snapshots** is displayed. The server named *v2vx-esxi* that requires ssh access enabled is selected (❶). After the server is selected, the configuration settings are accessed by selecting on the **Configuration** tab (❷).

The **Security Profile** link (❸) under the **Software** section lists the services on the ESXi server. By selecting the **Properties** link (❹), a **Service Properties** window opens and the settings for the services can be adjusted.

The **SSH** service controls the **ssh** daemon. Selecting this service and then selecting the **Options** button opens the SSH **(TSM-SSH) Options** window. Select **Start and stop with host** and select the Start button. After the service starts, select the **OK** button to close the window and return to the main vSphere window.



Testing the newly enabled **ssh** connection is done by connecting to the ESXi server using the `ssh` command.

```
# ssh root@v2v3-esxi
Warning: Permanently added 'v2v3-
esxi.cloud.lab.eng.bos.redhat.com,10.16.136.60' (RSA) to the list of known
hosts.
Password: [PASSWORD]
The time and date of this login have been sent to the system logs.

VMware offers supported, powerful system administration tools.  Please
see www.vmware.com/go/sysadmintools for details.

The ESXi Shell can be disabled by an administrative user. See the
vSphere Security documentation for more information.
~ #
```

The password for the root user account on the ESXi server must be entered each time the `ssh` command is used to connect to the server. This behavior can be stopped by creating and using a private/public key pair with `ssh`.

A key pair is generated and then the public portion of the key is placed in on the ESXi server. A key length of 2048 bits is created using the `ssh-keygen` commmand and specifying the `-b` option with an argument of 2048. The `ssh-copy-id` command is used to copy the public key to the ESXi system. The `-i` option is used with the `ssh-copy-id` command to specify the public key to copy to the ESXi system.

```
# ssh-keygen -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): [ENTER]
Enter passphrase (empty for no passphrase): [ENTER]
Enter same passphrase again: [ENTER]
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
d0:df:0f:81:e2:5c:60:83:3e:59:96:83:3b:ef:87:2d root@v2v3-
v2v.cloud.lab.eng.bos.redhat.com
The key's randomart image is:
+--[ RSA 2048]----+
|       o+.       |
|      oo=o .     |
|     ..=o.o .    |
|      *+ + . .   |
|        .    o   |
|       . o    .  |
|        E o      |
|         o       |
+-----------------+

# cat .ssh/id_rsa.pub | ssh root@v2v3-esxi "cat - >> /etc/ssh/keys-
root/authorized_keys"
Warning: Permanently added 'v2v3-esxi,10.16.136.60' (RSA) to the list of
known hosts.
Password: [PASSWORD]
```

Appendix C: Scripts

The following script can be downloaded from the Red Hat Customer Portal.[6]

```
#! /bin/bash

# This script migrates virtual machines from an ESXi server i
# to a Red Hat Enterprise Virtualization environment.o
#
# The perl-XML-XPath package is required as is ssh access to the ESXi
server.
# A netrc file must be configured for access to the ESXi server.
# The Red Hat Enterprise Virtualization Managers certificate must be
# downloaded for use with the REST API.
#
# This script appears in the
# Migrating to Red Hat Enterprise Virtualization 3 - VMware ESXi
# Reference Architecture.


declare -A config esxVM
count=0

esxi_vms() {
  state=$1

  echo "Checking power state of virtual machines."
  # Get virtual machine names and vmids
  while read vmid vmname extra
  do
    esxVM[${vmname}]=${vmid}
  done < <(
    ssh ${config[esx_srv]} vim-cmd vmsvc/getallvms 2>/dev/null | grep -v
Vmid
    )

  vms_on=0

  # Check power state of virtual machine
  for i in ${!esxVM[@]}
  do
    echo -e "  -- ${i}: \c"

    ssh ${config[esx_srv]} vim-cmd vmsvc/power.getstate ${esxVM[${i}]}
2>/dev/null | grep -q off
    RC=$?
    if [ $RC -ne 0 ]
    then
      ((vms_on++))

      if [ "${state}" == "force" ]
      then
        ssh ${config[esx_srv]} vim-cmd vmsvc/power.off ${esxVM[${i}]}
```

6 https://access.redhat.com/sites/default/files/migrate_esx2rhev.tgz

```
2>/dev/null
      else
        echo "on"
      fi
    else
      echo "off"
    fi
  done

  return ${vms_on}
  }


v2v_vms() {
  echo "Executing virt-v2v against the virtual machines"
  for i in ${!esxVM[@]}
  do
    virt-v2v -p ${config[profile]} -i libvirt -ic esx://${config[esx_srv]}/?
no_verify=1 ${i}
  done
  }

rhev_import() {

  echo "Importing the virtual machines."
  domainid=$( curl --silent --cacert ${config[cacert]}  \
    --header "Content-Type: application/xml" \
    --user "${config[rhevm_user]}:${config[rhevm_password]}"         \
    --request "GET"                                  \
    https://${config[rhevm_srv]}:8443/api/storagedomains?search=name
%3dExport \
    | xpath /storage_domains/storage_domain/@id 2>/dev/null \
    | sed -e 's/id="//' -e 's/"//' -e 's/ //'
    )

    uuids=$( curl --silent --cacert ${config[cacert]}  \
    --header "Content-Type: application/xml" \
    --user "${config[rhevm_user]}:${config[rhevm_password]}"         \
    --request "GET"                                  \
    https://${config[rhevm_srv]}:8443/api/storagedomains/${domainid}/vms \
    | xpath /vms/vm/@id 2>/dev/null \
    | sed -e 's/id="//g' -e 's/"//g'
    )

  for i in ${uuids}
  do
    echo "Importing VM ${i}"
    curl --silent --cacert ca.crt \
    --header "Content-Type: application/xml" \
    --user "${config[rhevm_user]}:${config[rhevm_password]}" \
    --request "POST" \
    --data "
      <action>
        <storage_domain>
          <name>Data</name>
```

```
        </storage_domain>
        <cluster>
          <name>Infrastructure</name>
        </cluster>
      </action>"  \
    https://${config[rhevm_srv]}:8443/api/storagedomains/${domainid}/vms/$
{i}/import
   done
  }


usage() {
  echo "usage: $0 options"
  echo -e "\nWhere:"
  echo "  -e esx_srv            FQDN of ESXi server"
  echo "  -o profile            Profile from /etc/virt-v2v.conf"
  echo "  -c cacert             Certificate file from RHEV Manager"
  echo "  -r rhevm_srv          FQDN of RHEV Manager Server"
  echo "  -u rhevm_user         User for RHEV Manager"
  echo "  -p rhevm_password     Password for RHEV Manager"
  echo "  -f                    Force shutdown virtual machines (Optional)"

  exit
}


## Main
#

while getopts ":e:o:c:r:u:p:f" opt
do
  case ${opt} in
      e) config[esx_srv]=${OPTARG}
         ((count+=1))
         ;;
      o) config[profile]=${OPTARG}
         ((count+=2))
         ;;
      c) config[cacert]=${OPTARG}
         ((count+=4))
         ;;
      r) config[rhevm_srv]=${OPTARG}
         ((count+=8))
         ;;
      u) config[rhevm_user]=${OPTARG}
         ((count+=16))
         ;;
      p) config[rhevm_password]=${OPTARG}
         ((count+=32))
         ;;
      f) config[force]="force"
         ;;
      *) usage
         ;;
  esac
```

```
done

if [ ${count} -ne 63 ]
then
  usage
fi


esxi_vms ${config[force]}
sleep 5

v2v_vms

rhev_import
```

Appendix D: References

The following are sources used for reference during the writing of this paper.

1. **Deploying Red Hat Enterprise Virtualization(RHEV) for Servers**
   https://inquiries.redhat.com/go/redhat/rhev-for-servers
2. **Red Hat Enterprise Virtualization 3.0 Administration Guide**
   http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Virtualization/3.0/html/Administration_Guide/index.html
3. **Red Hat Enterprise Virtualization 3.0 REST API Guide**
   http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Virtualization/3.0/html/REST_API_Guide/index.html
4. **Red Hat Enterprise Linux 6 Security Guide**
   http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/
5. **Red Hat Enterprise Virtualization Product Page**
   http://www.redhat.com/virtualization/rhev/
6. **Red Hat Enterprise Linux 6 V2V Guide**
   http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/V2V_Guide/index.html
7. **RHEV 3.0 Backup & Recovery using Acronis**
   https://access.redhat.com/knowledge/refarch/2012-rhev-30-backup-recovery-using-acronis
8. **Bug 678232 - virtv2v convert windows guest with vmware-tools from esx,it will show the vmware-tools error msg**
   https://bugzilla.redhat.com/show_bug.cgi?id=678232

Appendix E: Revision History

Revision 1.0	Monday June 18, 2012	John Herr
Initial Release		