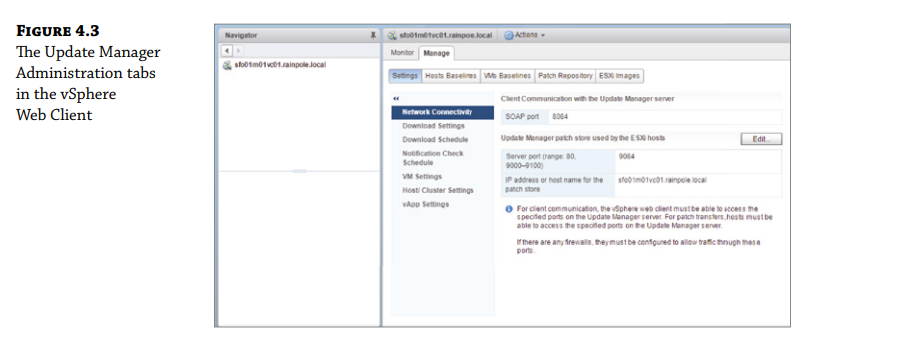
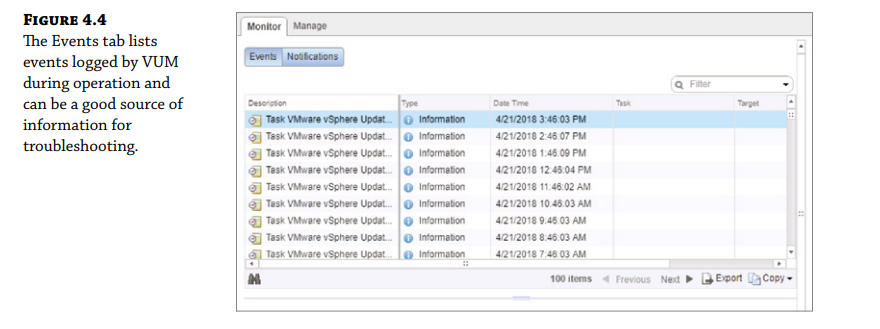
Please note this is not the complete notes; only a few topics have been covered here. For full reference, go through the reference book and the notes you have written during the lecture.

**Unit 02**

**VMware vSphere Update Manager (VUM)**

* **Configuring vSphere Update Manager**

****

****

**Explain each tab shown in the above 2 images in brief**

* **Creating baselines**

VMware provides a few baselines with VUM when it’s installed. The following baselines are present upon installation:

◆ Two dynamic host patch baselines named Critical Host Patches and Non-Critical Host Patches

◆ A dynamic baseline for upgrading VMware Tools to match the host

◆ A dynamic baseline for upgrading VM hardware to match the host

You can also use baseline groups to combine different types of baselines. Each baseline group can include one of each type of upgrade baseline.

You can also use baseline groups to combine different types of baselines. Each baseline group can include one of each type of upgrade baseline.

* **Routine Updates**

VUM uses the term remediation to refer to the process of applying patches and upgrades to a vSphere object that is not in compliance.

By attaching a baseline to a host or VM and performing a scan, VUM can determine whether that object is compliant or noncompliant with the baseline.

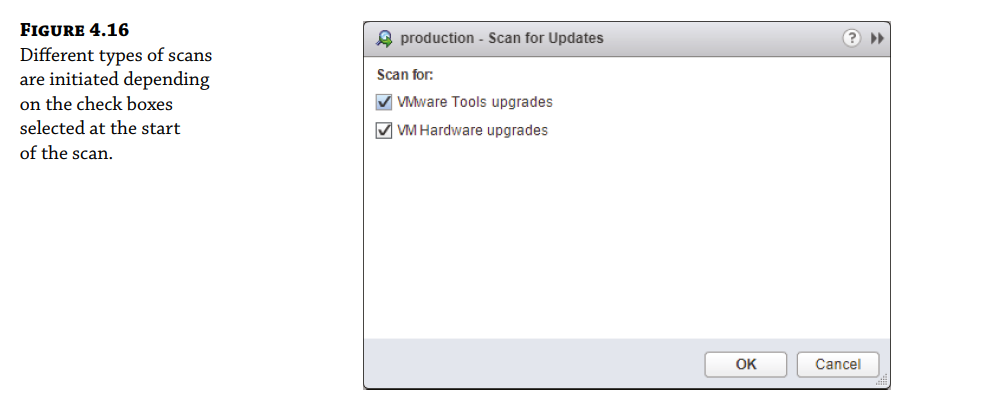
**Compliance with the baseline means that the host or VM has all the patches included in the baseline currently installed and is up-to-date; noncompliance means that one or more patches are missing and the target is not up-to-date compared to the current baseline.**

The first step in this process is creating the baselines that you will attach to your ESXi hosts or VMs.The next step is attaching a baseline to—or detaching a baseline from—ESXi hosts or VMs.

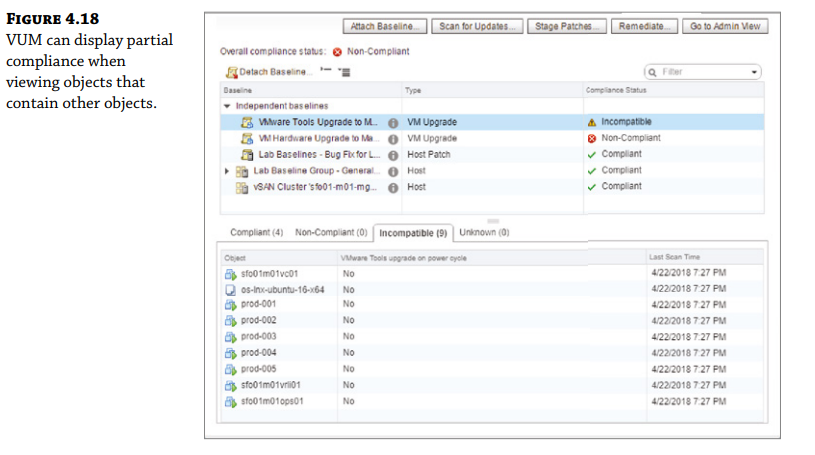
* **Performing a Scan**

The next step after attaching a baseline is to perform a scan. The purpose of a scan is to determine the compliance or noncompliance of an object with the baseline. If the object being scanned matches what’s defined in the baseline, then the object—be it an ESXi host, a VM, or a virtual appliance instance—is compliant. If something is missing from the object, then it’s noncompliant

* **Scanning VMs**

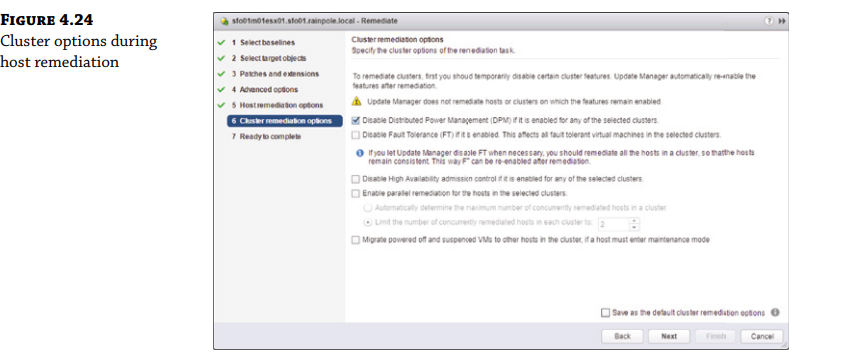
****

**Scanning ESXi Hosts**

****

* **Staging Patches**
* Staging a patch copies the files across to the host to speed up the actual time of remediation/
* Staging host patches is particularly useful for companies whose VUM-connected hosts are spread across slow WAN links. This can substantially reduce the outage required on such sites, especially if the WAN link is particularly slow or the patches themselves are very large.
* Hosts do not need to be in Maintenance mode while patches are being staged, but they do during the remediation phase.
* Staging patches also allows the uploads to be scheduled for a time when heavy WAN utilization is more appropriate, allowing you to remediate the host at a more agreeable time.
* **Remediating Hosts**

After you have attached a baseline to a host, scanned the host for compliance, and optionally staged the updates to the host, you’re ready to remediate, or update, the ESXi host/



* **Upgrading Hosts with vSphere Update Manager**

Perform the following steps to upgrade a host server with VUM 6.0:

1. Import an ESXi image and create a host upgrade baseline.

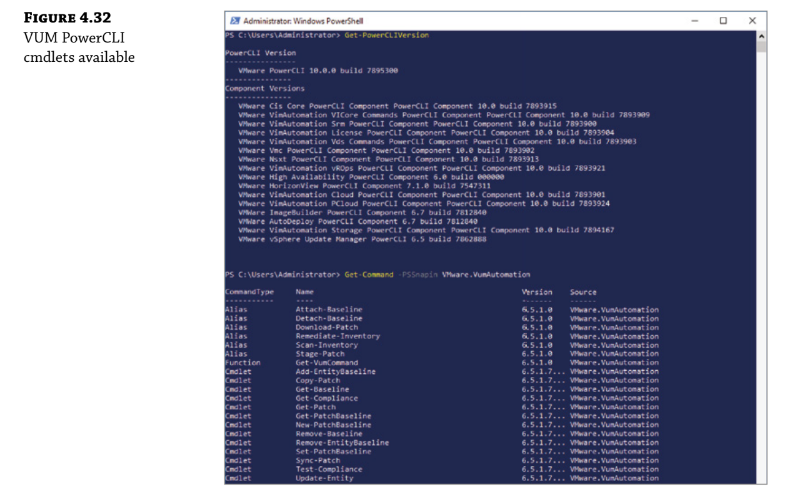
2. Upgrade the host by remediating with the upgrade baseline.

3. Upgrade the VMs’ VMware Tools and hardware.

* **Investigating Alternative Update Options**

**Using vSphere Update Manager PowerCLI**

The VUM PowerCLI cmdlets cover the most common tasks, like working with baselines and scanning, staging, and remediating vSphere objects.



**Upgrading and Patching without vSphere Update Manager**

1. You can run through an interactive install from the ESXi 6.x ISO media, choosing an in-place upgrade, and then perform the same process with the 6.7 media.
2. You can run a kickstart scripted upgrade along with the same ESXi 6.x and 6.7 media to perform an unattended upgrade.
3. Without VUM, upgrading VM hardware can be done via the VMware Host Client or directly in vCenter via the vSphere Web Client. If the hosts are connected to vCenter, then your connected client can manually upgrade the hardware. Even without vCenter, you can still upgrade each VM by connecting via the host client straight at the host.
4. You must shut down the VMs yourself and initialize each upgrade.
5. Similarly, VMware Tools can be upgraded in each guest OS manually from within the VM’s console.

**vSphere Auto Deploy**

The following requirements must be met before you can use Auto Deploy to install ESXi on

your hosts:

◆ PXE and TFTP boot environment

◆ ESXi Image Profiles created with Image Builder

◆ Host Profiles to attach to newly deployed hosts

◆ Deployment Rule to help the hosts get the right image and configuration

**vSphere Network**

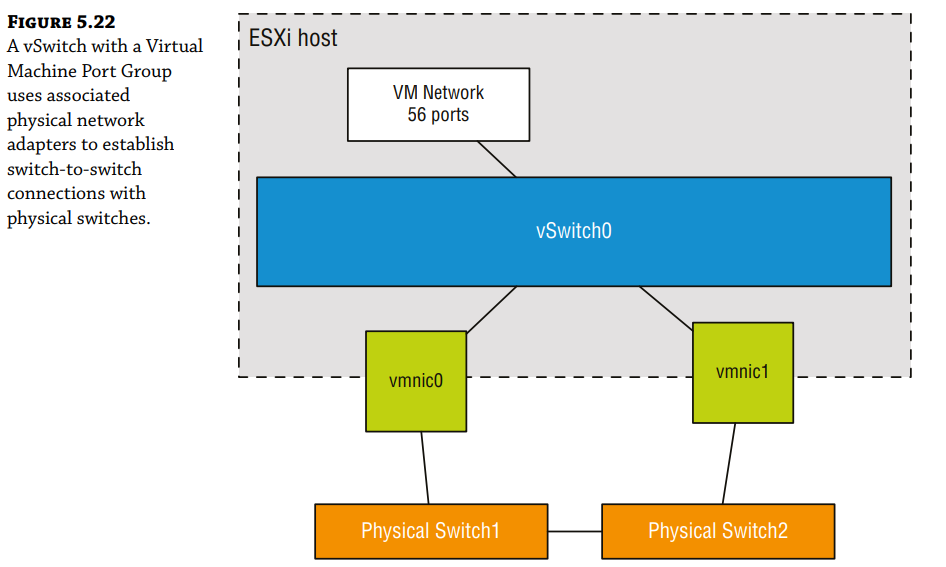
**Port and Port group**

**Uplinks**

**Configuring Virtual Machine Networking**

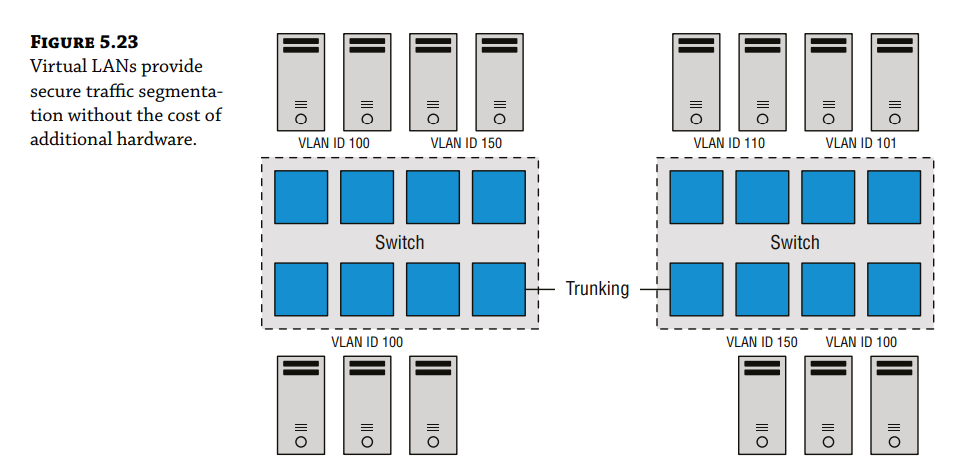
The second type of port group to discuss is the Virtual Machine Port Group, which is responsible for all virtual machine networking. The second type of port group to discuss is the Virtual Machine Port Group, which is responsible for all virtual machine networking.

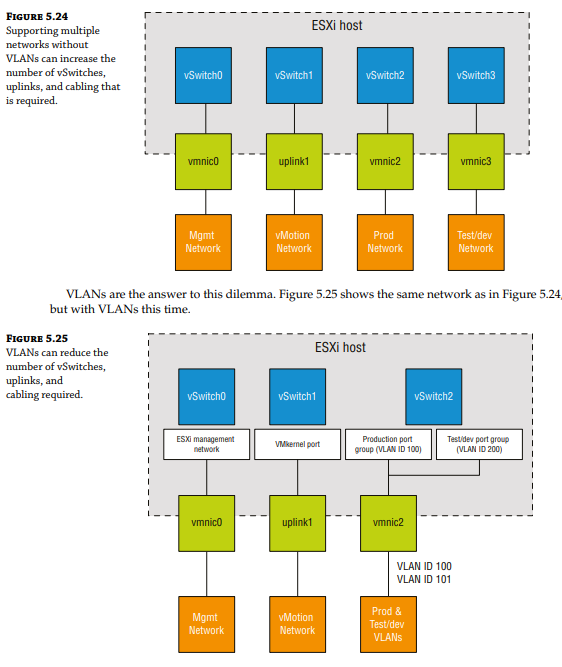
A vSwitch with a Virtual Machine Port Group acts just like an additional unmanaged physical switch. You need only plug in the appropriate uplinks—physical network adapters, in this case—that will connect that vSwitch to the rest of the network.



**Configuring vLANs.**

A virtual LAN (VLAN) is a logical LAN that provides efficient segmentation, security, and broadcast control while allowing traffic to share the same physical LAN segments or same physical switches. Figure 5.23 shows a typical VLAN configuration across physical switches.

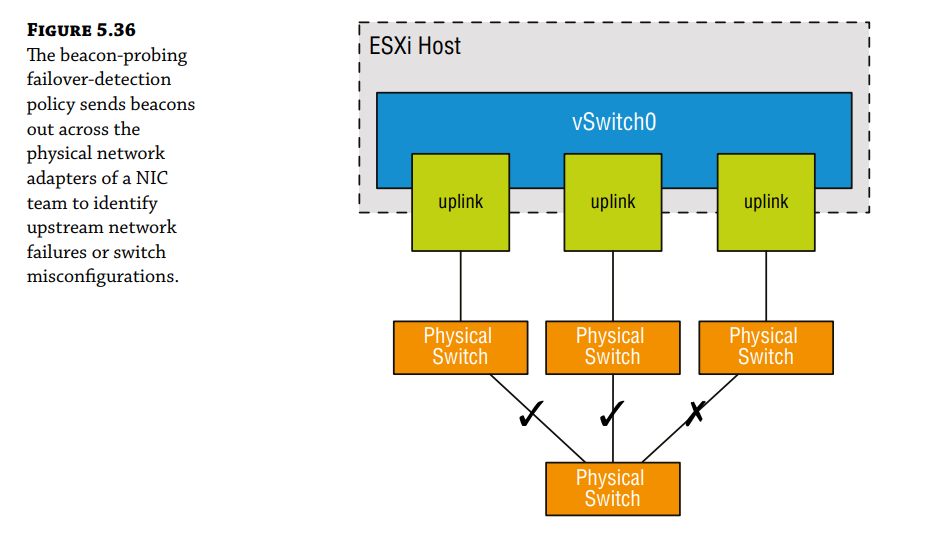


****

**Configuring NIC teaming**

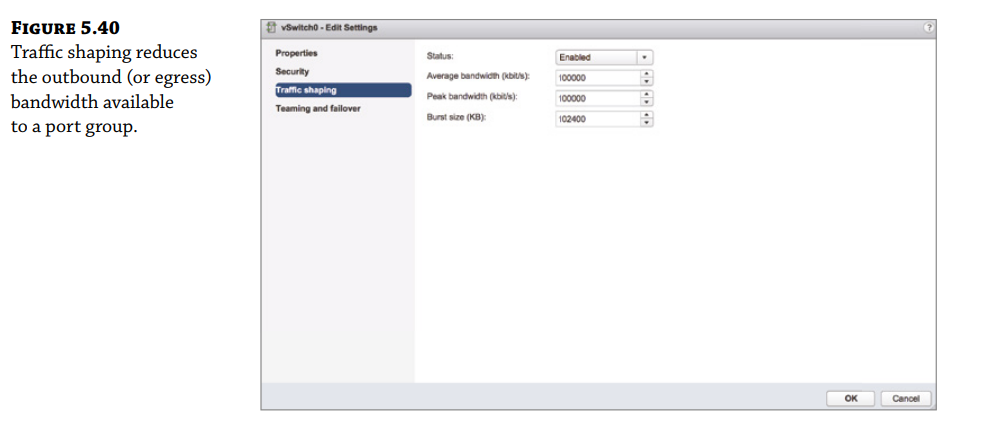
Configuring Failover Detection and Failover Policy

Failover detection with NIC teaming can be configured to use either a link status method or a beacon probing method. The link status failover detection method works just as the name suggests. The link status of the physical network adapter identifies the failure of an uplink. In this case, failure is identified for events like removed cables or power failures on a physical switch. The downside to the setting for link status failover detection is its inability to identify misconfigurations or pulled cables that connect the switch to other networking devices (for example, a cable connecting one switch to an upstream switch). The beacon probing failover detection setting, which includes link status as well, sends Ethernet broadcast frames across all physical network adapters in the NIC team. These broadcast frames allow the vSwitch to detect upstream network connection failures and will force failover when STP blocks ports, when ports are configured with the wrong VLAN, or when a switch-toswitch connection has failed. When a beacon is not returned on a physical network adapter, the vSwitch triggers the failover notice and reroutes the traffic from the failed network adapter through another available network adapter based on the failover policy



Using and Configuring Traffic Shaping

Traffic shaping establishes hard-coded limits for peak bandwidth, average bandwidth, and burst size to reduce a virtual machines outbound bandwidth capability. As shown in Figure 5.40, the Peak Bandwidth value and the Average Bandwidth value are specified in kilobits per second, and the Burst Size value is configured in units of kilobytes. The value entered for Average Bandwidth dictates the data transfer per second across the virtual switch. The Peak Bandwidth value identifies the maximum amount of bandwidth a vSwitch can pass without dropping packets. Finally, the Burst Size value defines the maximum amount of data included in a burst. The burst size is a calculation of bandwidth multiplied by time. During periods of high utilization, if a burst exceeds the configured value, packets are dropped in favor of other traffic; however, if the queue for network traffic processing is not full, the packets are retained for transmission at a later time.



Working with vSphere Distributed Switches

a vSphere Distributed Switch functions as a single virtual switch across all the associated ESXi hosts within a datacenter object.

There are a number of similarities between a vSphere Distributed Switch and a Standard Switch:

◆ A vSphere Distributed Switch provides connectivity for virtual machines and VMkernel

interfaces.

◆ A vSphere Distributed Switch leverages physical network adapters as uplinks to provide

connectivity to the external physical network.

◆ A vSphere Distributed Switch can leverage VLANs for logical network segmentation.

◆ Most of the same load balancing, failback, security, and traffic shaping policies are

available, with a few additions in the vSphere Distributed Switch that increase functionality over the vSphere Standard Switch.

Creating a vSphere Distributed Switch

Removing an ESXi Host from a Distributed Switch

Removing a Distributed Switch

Using NetFlow on vSphere Distributed Switches

NetFlow is a mechanism for efficiently reporting IP-based traffic information as a series of traffic

flows. Traffic flows are defined as the combination of source and destination IP addresses, source and destination TCP or UDP ports, IP, and IP Type of Service (ToS). Network devices that support NetFlow will track and report information on the traffic flows, typically sending this

information to a NetFlow collector.

**Enabling Switch Discovery Protocols**

Previous versions of vSphere supported Cisco Discovery Protocol (CDP), a protocol for exchanging information between network devices. However, it required using the command line to enable and configure CDP.

In vSphere 5.0, VMware added support for Link Layer Discovery Protocol (LLDP), an

industry standard discovery protocol, and provided a location within the vSphere Client where CDP/LLDP support can be configured.

**Enabling Enhanced Multicast Functions**

On top of basic multicast filtering supported by the vSphere Standard Switch, the vSphere Distributed Switch also supports multicast snooping.

In this mode, the distributed switch learns about the membership of a virtual machine

dynamically. This is achieved by monitoring virtual machine traffic and capturing IGMP or

multicast listener discovery (MLD) details when a virtual machine sends a packet containing this information. The distributed switch then creates a record of the destination IP address of the group, and for IGMPv3 it also records the source IP address from which the virtual machine prefers to receive traffic. The distributed switch will remove the entry containing the group details if a virtual machine does not renew its membership within a certain period of time.

**Configuring LACP**

Link Aggregation Control Protocol (LACP) is a standardized protocol for supporting the

aggregation, or joining, of multiple individual network links into a single, logical network link.

Using a version 5.1.0 vSphere Distributed Switch, you must configure the following

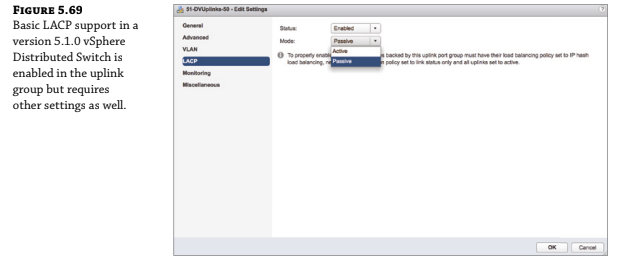
four areas:

◆ Enable LACP in the properties for the distributed switch’s uplink group.

◆ Set the NIC teaming policy for all distributed port groups to Route Based On IP Hash.

◆ Set the network detection policy for all distributed port groups to link status only.

◆ Configure all distributed port groups so that all uplinks are active, not standby or unused.



* When LACP Mode is set to Passive, the ESXi host won’t initiate any communications to the physical switch; the switch must initiate the negotiation.
* When LACP Mode is set to Active, the ESXi host will actively initiate the negotiation of the link aggregation with the physical switch.

**Configuring Virtual Switch Security**

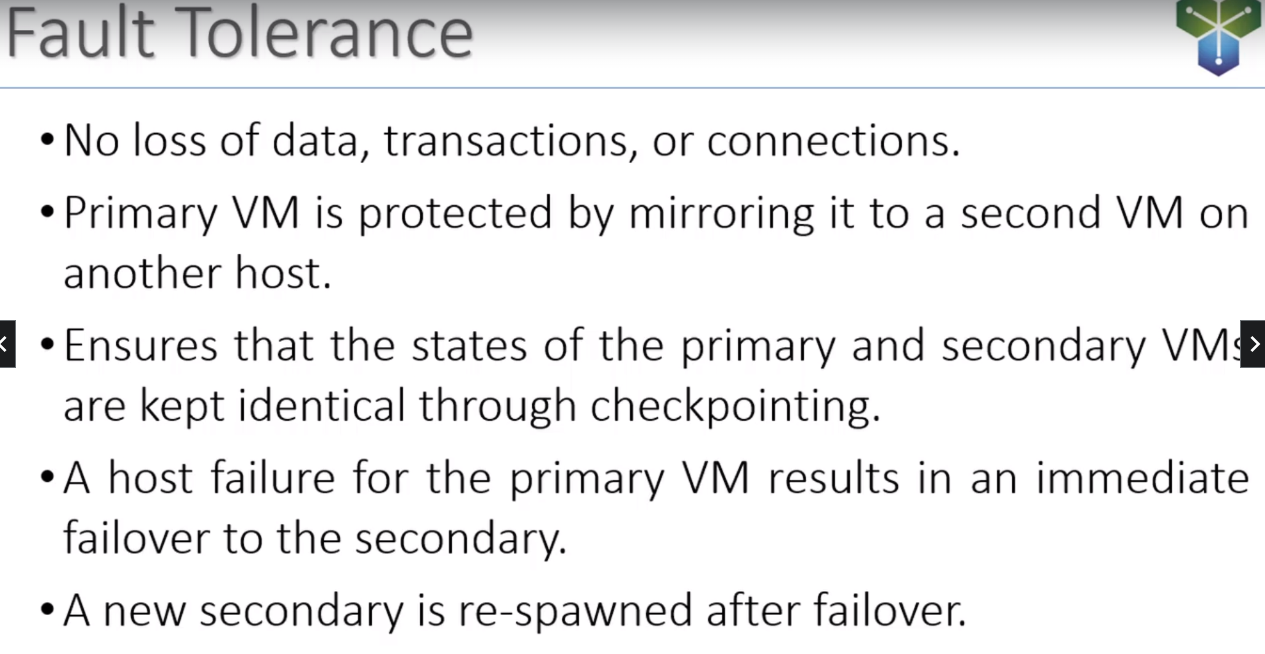
For vSphere Distributed Switches, you apply security policies only at the distributed port group level. The security settings include the following three options:

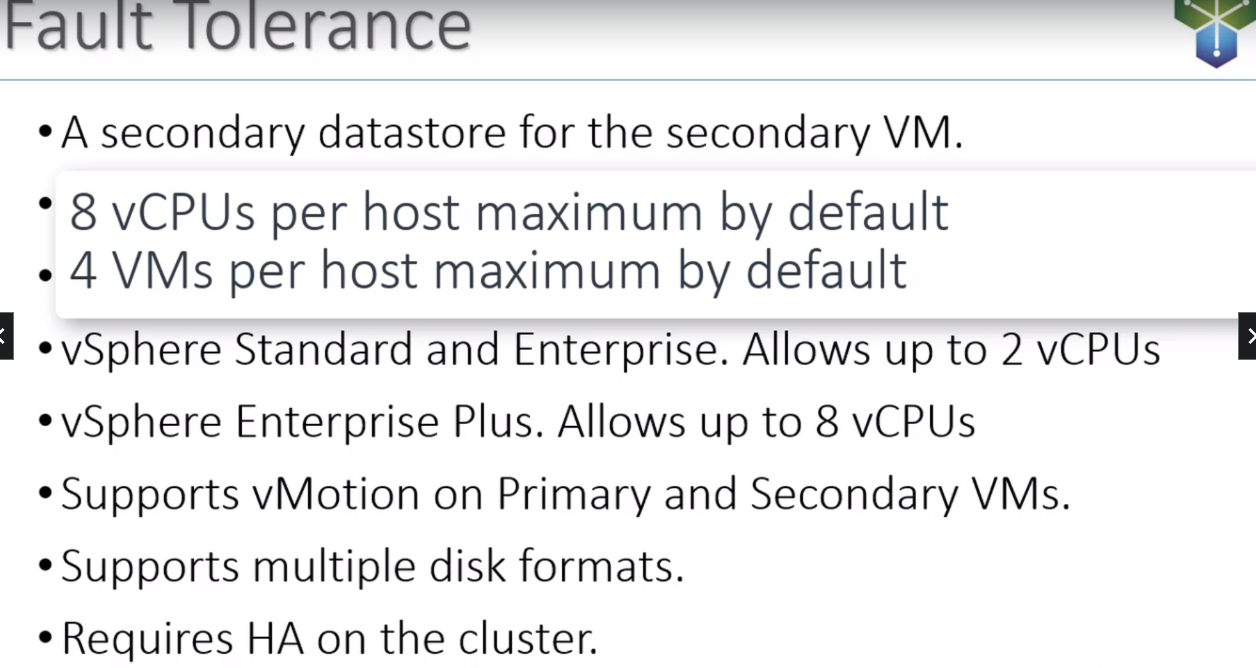
◆ Promiscuous mode

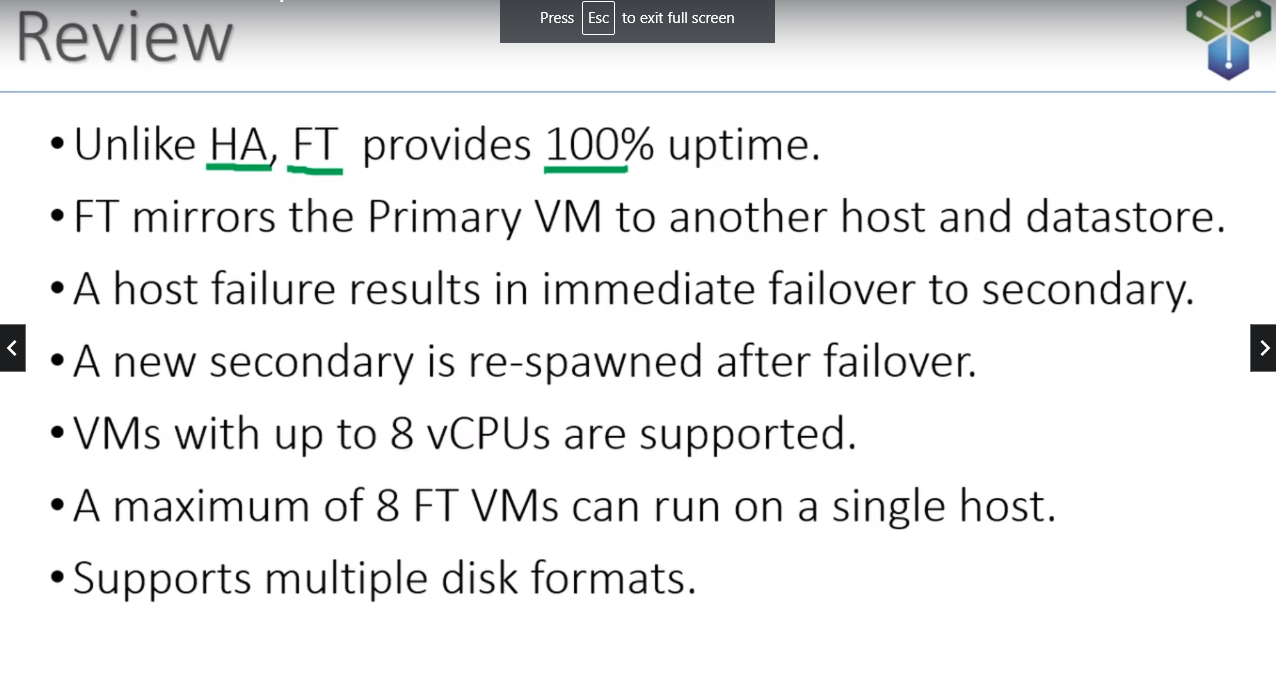
The Promiscuous Mode option is set to Reject by default to prevent virtual network adapters from observing any of the traffic submitted through a vSwitch or distributed switch.

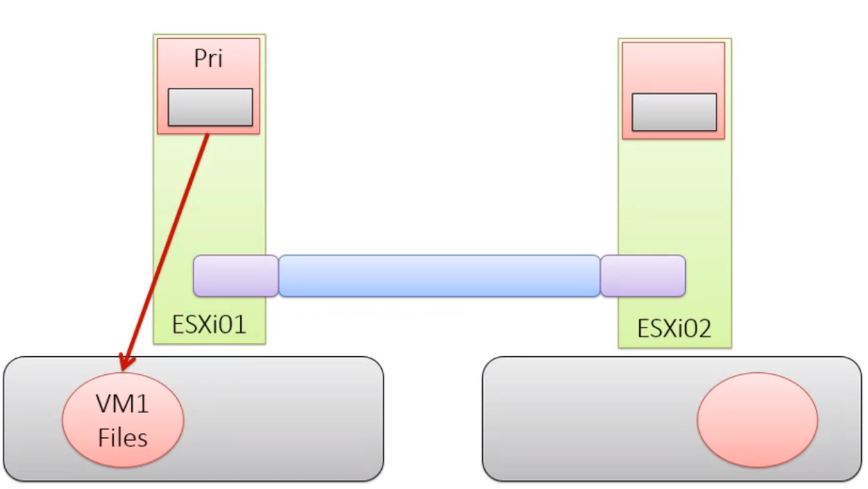
◆ MAC address changes

◆ Forged transmits









FT backup

**Planning for business continuity**

The following two primary types here:

**◆** Protecting against the loss of data due to equipment failure, software malfunction, or

simple user error (such as deleting something by mistake, which most of us have done at one time or another)

◆ Planning for disaster recovery in the event your entire datacenter is rendered unusable or unavailable

**Providing Data Protection**

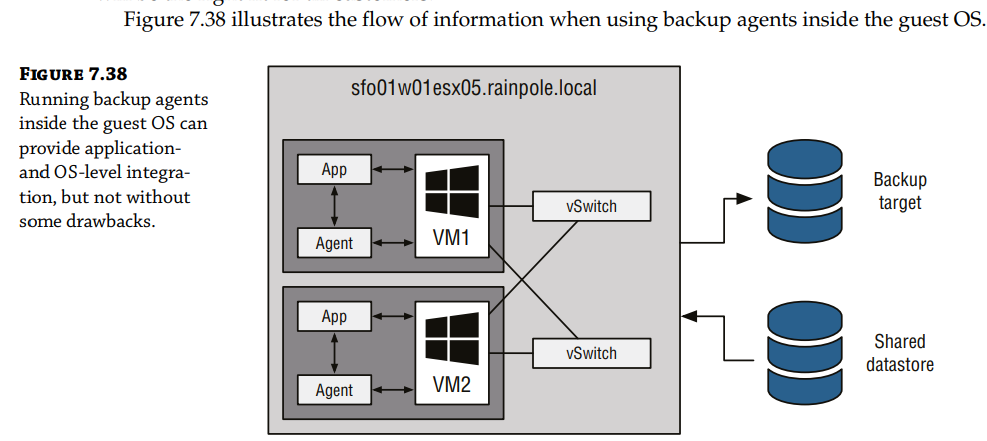
1. **Examining VM Backup Methods**

There are three high-level methods of backing up VMs in a VMware vSphere environment:

◆ Running a backup agent of some sort in the guest OS

◆ Leveraging vSphere snapshots and VADP

◆ Using array-based snapshot integration



However, *running backup agents within the guest OS* has its drawbacks:

◆ The network traffic typically runs across the network, which can create bottlenecks.

This is especially true if the backup traffic runs across the same network as end

user–facing traffic.

◆ To avoid bottlenecks with end user–facing traffic, organizations introduced dedicated

backup networks. This means more NICs in the ESXi hosts, separate vSwitches, separate physical switches, additional vNICs in the VMs, and additional complexity in the guest OS and the solution as a whole. Separate backup networks can also complicate troubleshooting and operations.

◆ The backup agents are individually running in each guest OS instance, so as more and more VMs (and guest OS instances) are consolidated onto physical servers, this creates additional overhead. This can, in some circumstances, translate to longer

backup windows.

◆ Some backup vendors charged a separate license for every installation of the backup

agent, which has decreased the financial benefits of virtualization and consolidation.

*Leverage the snapshot functionality of VMware vSphere to* unlock the VM’s virtual disks and then back up the virtual disks directly. When the backup of the virtual disk is complete, commit the snapshot and you’re finished. The framework for driving this process in an automated fashion—so that backup vendors can make it easier to use—is VADP.

The overall process looks something like this:

1. The backup software requests a snapshot of the virtual disks for the VM to be backed up.

2. VMware vSphere creates a snapshot, and all writes to the virtual disks for that VM now start flowing into the delta disks. The base VMDK files are unlocked.

3. The backup application backs up the base VMDK files.

4. When the backup of the base VMDK files is complete, the backup software requests

vSphere to commit the snapshot.

5. The writes in the delta disk are committed to the base VMDK, and the snapshot

is removed.

6. The process repeats itself for the next VM.

**Snapshots Are Not Backups**

While backup software that utilizes VADP actually does take snapshots as part of its process, taking VM snapshots manually for backups is a really bad idea. It’s not uncommon to see this done within many IT organizations, but there are numerous pitfalls you may come across. For example

◆ No auto committing back to the base VMDK files means you can accidentally have multiple levels of snapshots without knowing (unless you specifically go looking).

◆ Snapshots can inadvertently be rolled back

◆ Snapshots are not stored on secondary storage by default

◆ A snapshot is a change log of the original virtual disk.

◆ VMware recommends a maximum age for a single snapshot be no more than 72 hours

1. **Recovering from Disasters**

High availability makes up only half of the ability to keep your application/systems up in day-to-day operation. The other half is disaster recovery, which is the ability to recover from a catastrophic failure.

A majority of data is kept on the SAN, and the data is replicated to another SAN at your remote disaster recovery colocation site.

To set up SAN replication, a company purchases two SANs to be set up at different locations, and the data is replicated between the two sites.

all replication solutions fall into one of two very broad areas:

◆ Synchronous replication solutions

◆ Asynchronous replication solutions

In synchronous replication solutions, the primary array waits until the secondary array has acknowledged each write before sending a write acknowledgment back to the host, ensuring that the replicated copy of the data is always as current as the primary. In this situation latency comes into play, and it increases significantly with distance. Therefore, you must limit the distance between synchronous replication solutions to keep latency to a minimum.

Asynchronous replication solutions transfer data to the secondary array in chunks and do not wait for a write acknowledgment from the remote array before acknowledging the write to the host.

In a vSphere environment, you can combine SAN- and/or host-based replication—synchronous or asynchronous—with VMware Site Recovery Manager (SRM), a workflow automation tool that helps administrators with the task of orchestrating the startup of all the VMs in a datacenter.

1. **Using vSphere Replication**

SAN-based replication is great, but there may be times when it’s just not feasible. For smaller businesses or remote offices, the size and cost of the infrastructure cannot be justified. Inevitably, DR is still a requirement, and for this reason, VMware has an IP-based replication engine simply called vSphere Replication.

The ability to make a copy of your important data and workloads at a remote location is often one of the top priorities for business management. vSphere Replication was introduced in SRM 5.0 and decoupled as a separate new feature in vSphere version 5.1; it has been continually improved since then. It provides VM-based replication and recovery at the hypervisor level. This means that there are no external requirements to provide the replication, apart from network connectivity between two locations.

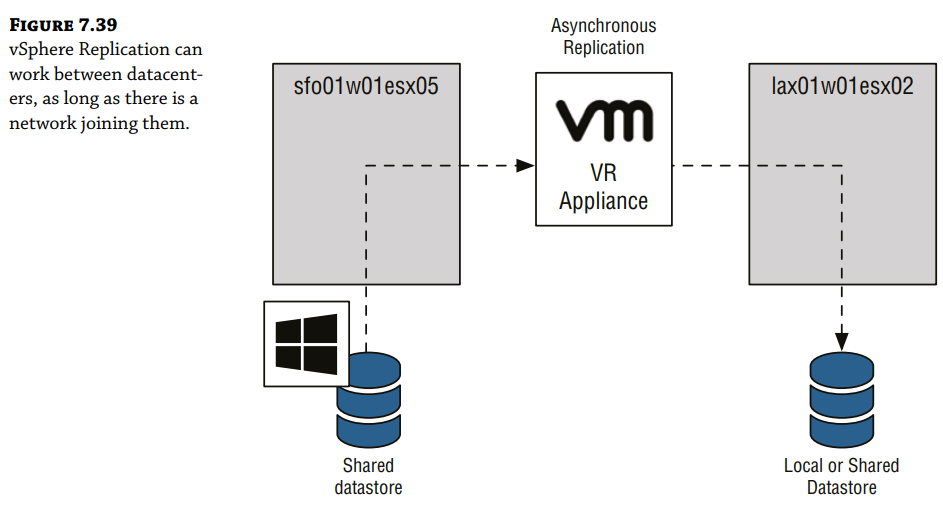
The following constraints affect how vSphere Replication can be configured:

◆ Maximum replication time (RPO): 24 hours

◆ Minimum replication time (RPO): 15 minutes (or 5 minutes if vSAN is used for the source and destination datastores)

◆ Maximum protected VMs: 2000/instance

◆ Maximum instances: 10/vCenter Server



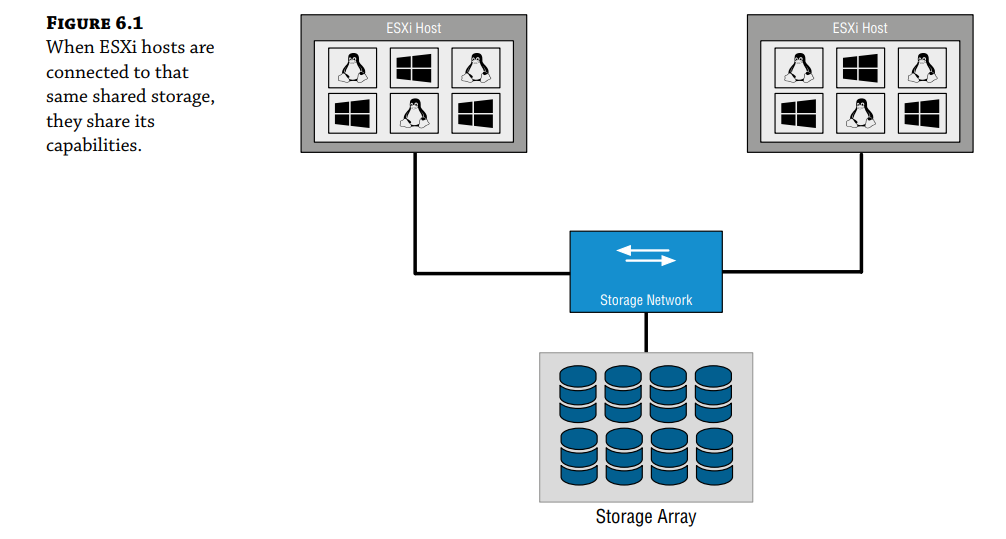
**Chapter 6: Storage devices**

**Importance of Storage Design**

It provides infrastructure as a service(IaaS)

* Advanced Capabilities like vHA, vDRS,vFT
* Performance - consolidation, higher utilization, more flexibility, and higher efficiency
* Availability - The overall availability of your virtualized infrastructure, and by extension, the VMs running on that infrastructure, depend on the shared storage infrastructure.

**Examining Shared Storage Fundamentals**



**Defining Common Storage Array Architectures**

The **elements that make up a shared storage array** consist of external connectivity, storage processors, array software, cache memory, disks, and bandwidth.

* External Connectivity: - The external (physical) connectivity between the traditional storage array and the hosts (in this case, the ESXi hosts) is generally Fibre Channel or Ethernet, though InfiniBand and other rare protocols exist.
* Storage Processors - Different vendors have different names for storage processors, which are considered the brains of the array. They handle the I/O and run the array software.
* Array Software - It is vendor specific. The following list includes some examples of these array capabilities and key functions:
  + Remote storage replication for disaster recovery.
  + Snapshot and clone capabilities for instant point-in-time local copies for test and development and local recovery.
  + Capacity-reduction techniques such as archiving, compression, and deduplication.
  + Automated data movement between performance/cost storage tiers at varying levels of granularity.
  + LUN/file system expansion and mobility.
  + Thin provisioning, which typically involves allocating storage on demand as applications and workloads require it.
  + Storage quality of service (QoS), which means prioritizing I/O to deliver a given MBps, IOPS, or latency.
  + Encryption of data on the fly or at rest by using self-encrypting drives or other means.
* Cache Memory - Every array differs as to how cache memory is implemented, but all have some degree of nonvolatile memory used for various caching functions, delivering lower latency and higher IOPS throughput by buffering I/O using write caches and storing commonly read data to deliver a faster response time using read cache.
* Disks- Arrays differ as to which type of disks (often called spindles) they support and how many they can scale to support.The following list is a quick reference on what to expect under a random read/write workload from a given disk drive:

◆ 7,200 RPM SATA: 80 IOPS

◆ 10K RPM SATA/SAS/Fibre Channel: 120 IOPS

◆ 15K RPM SAS/Fibre Channel: 180 IOPS

◆ Commercial solid-state drives (SSD) based on Multi-Level Cell (MLC) technology: 1,000–100,000s IOPS

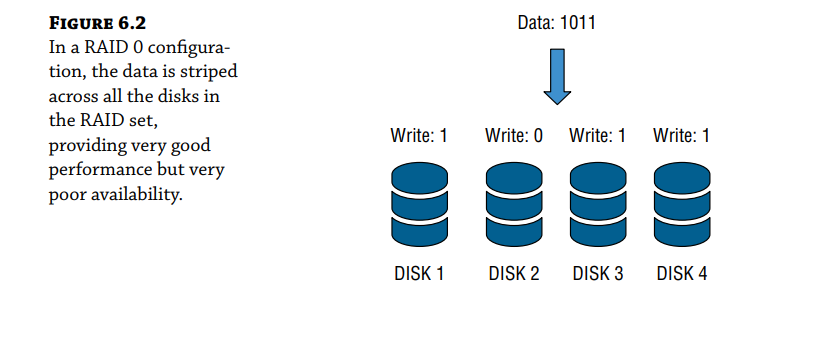
◆ Enterprise flash drives (EFD) based on Single-Level Cell (SLC) technology and much deeper, very high-speed memory buffers: 6,000–100,000s IOPS

**Explaining RAID**

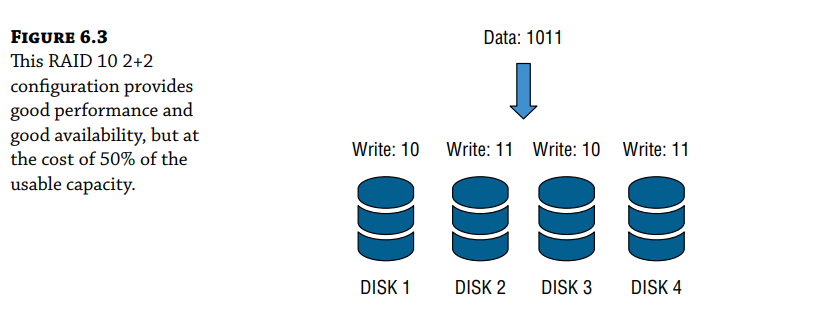
Redundant Array of Inexpensive (sometimes “Independent”) Disks (RAID) is a fundamental and critical method of storing the same data several times. RAID is used to increase data availability (by protecting against the failure of a drive) and to scale performance beyond that of a single drive.

**RAID 0** This RAID level offers no redundancy and no protection against drive failure. Data is spread across all the disks in the RAID group,

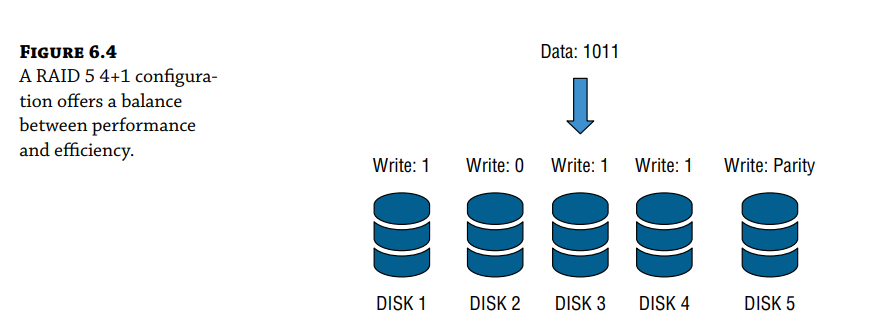
which is often called a stripe.

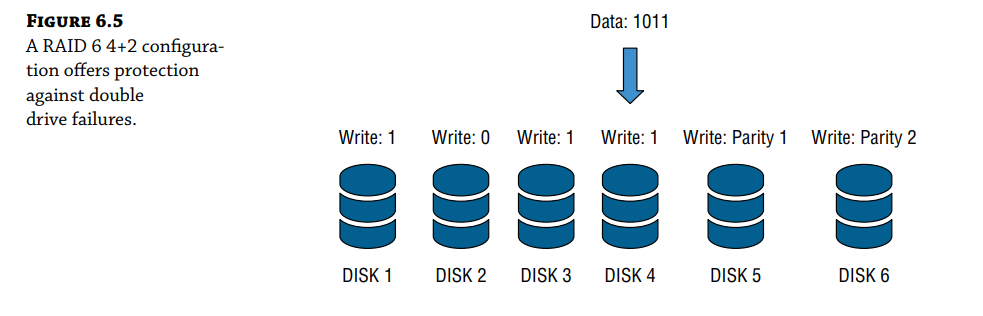


**RAID 1, 1+0, 0+1** These mirrored RAID levels offer high degrees of protection but at the cost of 50% loss of usable capacity. RAID 1 simply writes every I/O to two (or more) drives and can balance reads across all drives (because there are multiple copies). This can be coupled with RAID 0 to form RAID 1+0 (or RAID 10), which mirrors a stripe set, or to form RAID 0+1, which strips data across pairs of mirrors. This has the benefit of being able to withstand multiple drives failing, but only if the drives fail on different elements of a stripe on different mirrors, thus making RAID 1+0 more fault tolerant than RAID 0+1.



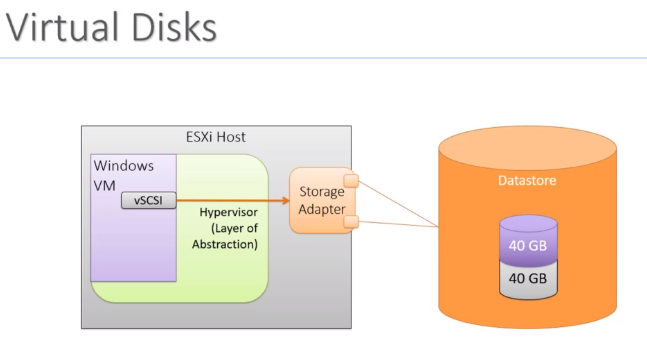
**Parity RAID (RAID 5, RAID 6)** These RAID levels use a mathematical calculation (an XOR parity calculation) to represent the data across several drives.

****

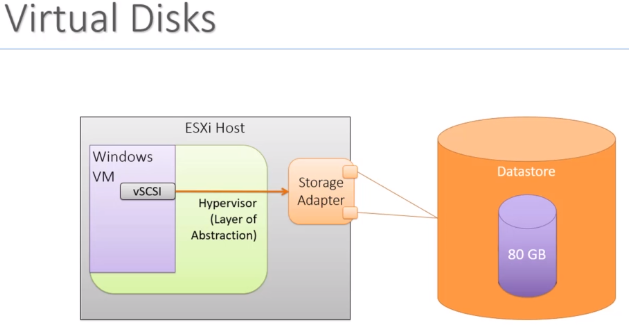
****

**Disk Provisioning**

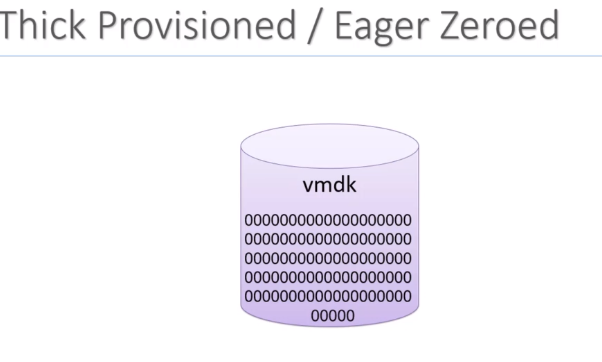
1. Thin - It saves storage space, because out of the full capacity we get half for the VM.



1. Thick -It utilizes full disk space for VM. Here we are sure that we will get space, in case of data expansion.



1. Thick Provisioning - Eager Zeroed - In this mode the disk will take time to create because it will first fill the disk with zeros.

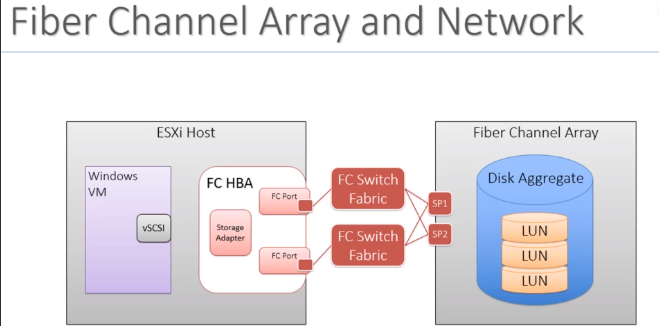


**Choosing a Storage Protocol**

vSphere offers several choices for shared storage protocol, including Fibre Channel, Fibre Channel over Ethernet/FCoE/, iSCSI, and Network File System (NFS), which is a form of NAS.

**Fibre Channel**

SANs are most commonly associated with Fibre Channel storage because Fibre Channel was the first widely adopted protocol used with SANs. However, SAN refers to a network topology, not a connection protocol. In fact, SAN refers to the ability to create block storage access through the use of a network.



Storage processor(SP)

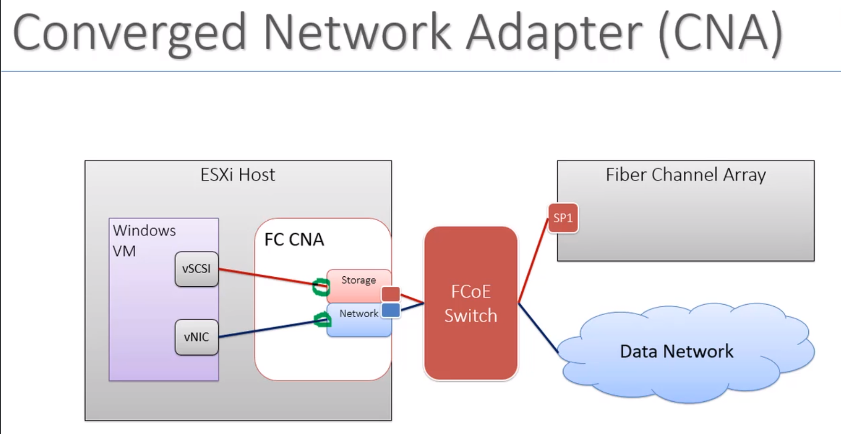
FC HBA - Fibre Channel Host Bus Adapter

LUN - Logical Unit Number

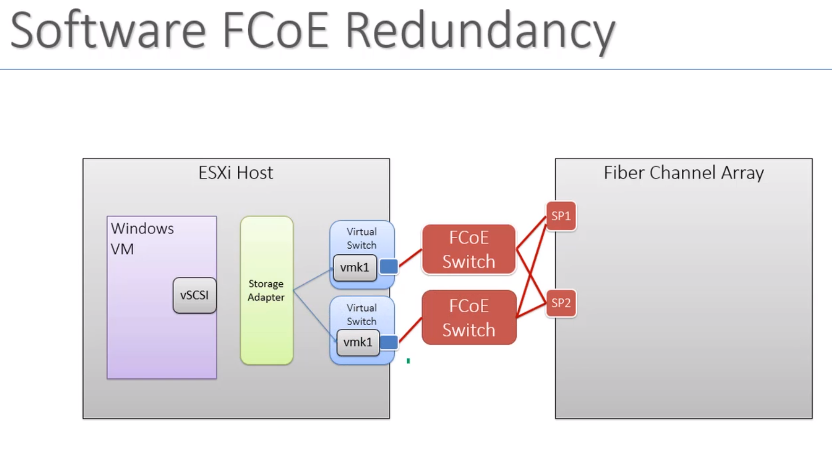
2FC switch and 2SP for redundancy or multipathing.

In the diagram above, the SCSI command will flow from VM to HBA through the Storage Adapter, it will go to port then from FC switch to SP and finally it will reach the specified LUN.

**Fibre Channel over Ethernet**



CNA is a physical adapter

****

For software FCoE you will require Storage Adapter and virtual switch for vmk1 port.

**iSCSI**

**Network File System (NFS)**

**Chapter 8: Securing VMware vSphere**

Securing vSphere involves securing the following components:

◆◆ The ESXi hosts

◆◆ vCenter Server

◆◆ The virtual machines (VMs), including the guest operating systems (guest OSs) running inside the VMs

◆◆ The applications running in the VMs

When you’re considering how to secure the various components involved in a vSphere environment, take into account the following three aspects:

◆◆ Authentication

◆◆ Authorization

◆◆ Accounting

This model—often referred to as the **AAA model**—describes the way in which users must be authenticated (properly identified as who they claim to be), authorized (enabled or permitted to perform a task, which also includes network access controls), and accounted for (all actions are tracked and logged for future reference).

**Securing ESXi Hosts**

Using vCenter Server eliminates the largest part of the need to connect directly to an ESXi host, the need does not go away entirely. There are instances when a task cannot be accomplished through vCenter Server, such as in the following situations:

◆◆ vCenter Server is not available or is down.

◆◆ You are troubleshooting ESXi boot and configuration problems.

You have two basic options: managing users locally on each host or integrating with Active Directory.

**Managing Users Locally**

You need at least two accounts in case one account is unavailable, such as when a user is on vacation or is sick or an accident occurs.

You have two ways of managing users locally: using command-line tools or using the embedded VMware Host Client.

**HW** create, edit and delete a local user account using vSphere Client.

**Enabling Active Directory Integration**

* The centralized security authority for ESXi hosts.
* Before you can join your ESXi hosts into Active Directory, you need to satisfy four prerequisites:
  + The time on your ESXi hosts must be synchronized with the time on the Active Directory domain controllers
  + Your ESXi hosts must be able to resolve the Active Directory domain name and locate the domain controllers via DNS. Typically, this means configuring the ESXi hosts to use the same DNS servers as the Active Directory domain controllers,
  + The fully qualified domain name (FQDN) of the ESXi host must use the same domain suffix as the Active Directory domain.
  + You must create an ESX Admins group in Active Directory. Place the user accounts that should be permitted to authenticate with an ESXi host in this group. You can’t use any other group name; it must be named ESX Admins.
* When the ESXi host is joined to Active Directory, users will be able to authenticate to an ESXi host using their Active Directory credentials.

**Controlling Access to ESXi Hosts**

Authorization, which encompasses access control mechanisms that affect local access or network access.

**Controlling Local Access**

* ESXi offers direct access via the server console through the Direct Console User Interface (DCUI).
* Access to the DCUI on an ESXi host is limited to users who have the Administrator role on that host.
* Controlling local CLI access: ESXi has a CLI environment that is accessible from the server’s physical console. However, by default, this CLI environment—known as the ESXi Shell—is disabled.
* Controlling remote CLI access via SSH: Secure Shell, often referred to as just SSH, is a widely known and widely used encrypted remote console protocol.
* ESXi includes SSH as a method of remote console access. This allows vSphere administrators to use an SSH client—such as PuTTY.exe on Windows or OpenSSH on Linux or Mac OS X—to remotely access the CLI of an ESXi host in order to perform management tasks. However, as with the ESXi Shell, SSH access to an ESXi host is disabled by default.
* As with local CLI access, VMware recommends against using SSH as a means of routinely managing your ESXi hosts

**Controlling Network Access via the ESXi Firewall**

ESXi ships with a firewall that controls network traffic into or out of the host. This firewall gives the vSphere administrator an additional level of control over what types of network traffic are allowed to enter or leave the ESXi hosts.

By default, the ESXi firewall allows only incoming and outgoing connections necessary for managing the VMs and the ESXi host. The following default ports are among those that are open:

◆ TCP 443 and 902: Host Client, vCenter Agent

◆ UDP 53: Domain Name System (DNS) client

◆ TCP and UDP 427: Common Information Model (CIM) Service Location Protocol (SLP)

◆ TCP 8000: vMotion

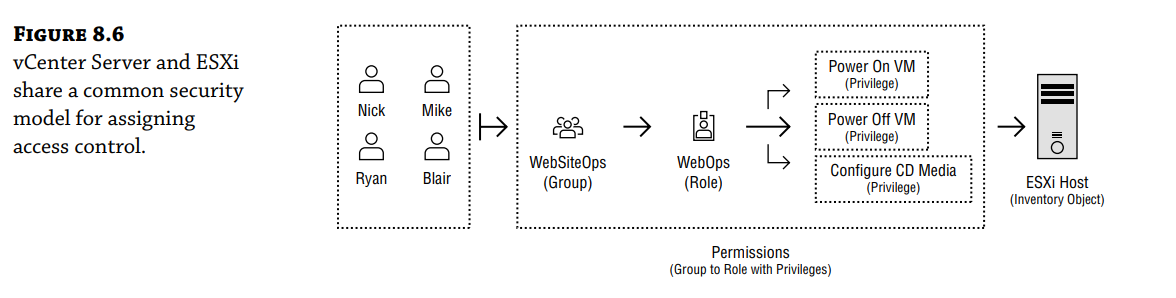
◆ TCP 22: SSH

**Keeping ESXi Hosts Patched**

Another key component in maintaining the security of your vSphere environment is keeping your ESXi hosts fully patched and up-to-date. On an as-needed basis, VMware releases security patches for ESXi. Failing to apply these security patches could expose your vSphere environment to potential security risks.

**Managing ESXi Host Permissions**

Both vCenter Server and ESXi hosts use the same structured security model to allow users to manage portions of the virtual infrastructure. This model consists of users, groups, roles, privileges, and permissions, as shown in fig. below.



From a security standpoint, the items that differ between the non–vCenter Server environment and the vCenter Server environment are predominantly in the following two areas:

◆ The location of the user objects created

◆ The level of granularity of the roles and privileges available in each environment

Four basic components to the vCenter Server/ESXi security model:

**User or Group** A user is an authentication mechanism; a group is a way of collecting users.

In earlier sections of this chapter, you saw hs form a basic building block of the security model.ow to manage users and how ESXi can leverage local users from Active Directory. Users and group

**Privilege** A privilege is an action that you can perform on an inventory object. This would include allocating space in a datastore, powering on a VM, configuring the network, or attaching a virtual CD/DVD to a VM.

**Role** A role is a collection of privileges. Both vCenter and ESXi ship with built-in roles, as you’ll see shortly, and you can also create your own custom roles.

**Permission** A permission allows a user to perform the activities specified by a role assigned to an inventory object. For example, you might assign a role that has all privileges to a particular inventory object. Attaching the role to the inventory object creates a permission.

An ESXi host has the following three default roles:

**No Access** The No Access role works as the name suggests. This role prevents access to an object or objects in the inventory. The No Access role can be used if a user was granted access higher up in the inventory. The No Access role can also be used at lower-level objects to prevent object access. For example, if a user is granted permissions on the ESXi host but should be prevented from accessing a specific VM, you could use the No Access role on that specific VM.

**Read-Only** Read-Only allows a user to see the objects within the vSphere Client inventory. It does not allow the user to interact with any of the visible objects in any way. For example, a user with the Read-Only permission would be able to see a list of VMs in the inventory but could not act on any of them, such as performing a power operation.

**Administrator** The Administrator role has the utmost authority, but it is only a role, and it needs to be assigned using a combination of a user or a group object and an inventory object such as a VM.

**H.W**. how to create, grant and remove roles? How to grant and remove permission to role?

**Configuring ESXi Host Logging**

Capturing information in the system logs is an important aspect of computer and network security. The system logs provide a record, or an accounting, of the actions performed, the events encountered, the errors experienced, and the state of the ESXi host and the VMs on that host.

***Every ESXi host runs a syslog daemon (service) that captures events and logs them for future reference***. Assuming that you’ve installed ESXi onto some local disks, the default location for the logs is a 4 GB scratch partition that the ESXi installer creates. Although this provides long-term storage for the ESXi host logs, there is no centralized location for them, making analysis of the logs more difficult than it should be. You would have to connect to each host individually to review the logs for that host. Further, if you are booting from SAN or if you are using vSphere Auto Deploy, there is no local scratch partition, and logs are stored in memory on the ESXi host—which means they disappear when the ESXi host is restarted. Clearly, this is not an ideal configuration. Not only does it lack centralized access to the logs, but it also lacks long-term storage for the logs. The typical solution to both of these issues is a vSphere integrated or third-party syslog server, a server that runs a syslog daemon and is prepared to accept the log entries from the various ESXi hosts. ***Though the Syslog Collector service in vCenter Server is not long available in vSphere 6.7, to make things easier, VMware introduced Log Insight for vCenter Server, which entitles you to 25 syslog clients (OSIs).***

**Securing vCenter Server**

The following security recommendations are among the more common ones:

◆ **Stay current on all Windows Server patches and updates**. This helps protect you against potential security exploits.

◆ Harden the Windows Server installation using published best practices and guidelines from Microsoft.

In addition to these standard security recommendations, we can offer a few other security recommendations that are specific to vCenter Server and the Platform Services Controller:

◆ Be sure to stay current on vCenter Server patches and updates.

◆ Place the vCenter Server backend database on a separate system (physical or VM), if possible, and follow recommended practices to secure the separate system.

◆ **If you are using Windows authentication with SQL Server, use a dedicated service account for vCenter Server**—don’t allow vCenter Server to share a Windows account with other services or applications.

◆ **Be sure to secure the separate database server and backend database using published security practices from the appropriate vendor.** This includes securing the database server itself (Microsoft SQL Server or Oracle) as well as the underlying OS for that database server (Windows Server, Linux, or other).

◆ Replace the default self-signed SSL certificates with a valid SSL certificate from a trusted root authority for vCenter Server and all of its components.

**Authenticating Users with Single Sign-On**

◆ Configuring Single Sign-On for authentication against Active Directory

◆ Configuring Single Sign-On for authentication against local users

◆ Understanding how vCenter Server authenticates against ESXi with vpxuser

**VPXUSER ACCOUNT**

let’s say Shane, an administrator, wants to log into vCenter Server and create a new

VM. Shane first needs the proper role—perhaps a custom role you created specifically for the purpose of creating new VMs—assigned to the proper inventory object or objects within vCenter Server.

Assuming the correct role has been assigned to the correct inventory objects—let’s say it’s a resource pool—Shane has what he needs to create, modify, and monitor VMs. But Shane’s user account does not have direct access to the ESXi hosts when he’s logged into vCenter Server. In fact, a proxy account is used to communicate Shane’s tasks to the appropriate ESXi host or VM. This account, vpxuser, is the only account that vCenter Server stores and tracks in its backend database.

**Reviewing vCenter Server’s Roles**

1. No Cryptography Administrator - With the No Crypto Administrator role, you can now assign users this role to allow them to access all capabilities of an administrator with the exception of manipulating cryptographic operations such as encrypting VMs, decrypting VMs, or changing vCenter’s Key Management Server. This is practical for lower-level or newer administrators that may be well-versed in vSphere, but you may want to limit their ability to interact with encryption policies.
2. Tagging Admin - With vSphere 6.0 and later, you can now assign users this role to allow them to create, edit, and delete tags—not to mention assign or unassign tags for objects. This is practical for solutions that may want to tag a VM such as an antivirus solution or provisioning engine.
3. Content Library Administrator This role has all of the required permissions to administer the Content Library and the associated contents throughout their respective life cycles. This includes creating a Content Library, adding files and synchronizing across multiple Content Libraries, removing content, and deleting Content Libraries when they are no longer required.
4. AutoUpdateUser This role is mapped to the AutoUpdate@vsphere.local user, which is automatically created in vSphere 6.7. It has an extremely limited amount of permissions, simply allowing the service account to log events, and is primarily used for lifecycle management purposes when integrating with VMware Cloud on AWS.

**Securing Virtual Machines**

**Configuring a Key Management Server for VM and VSAN Encryption** vSphere provides some outstanding data encryption functionality by integrating with an External Key Management Server (KMS) running version 1.1 of the Key Management Interoperability Protocol (KMIP). KMIP is an extensibility protocol that permits interactions with cryptographic keys. Under the context of vCenter Server, the system lets you dynamically create and pull keys into your vSphere environment for the purpose of performing encryption. After trust is established between vCenter Server and your KMS infrastructure, you can set several different security-related policies to help maintain the security of your VMs and datastores.

**Enabling vSAN Encryption**

If you are using vSAN in your environment, then your VSAN clusters will automatically

populate after you set up KMS as your default; however, you still need to enable VSAN encryption on the clusters. Keep in mind that this operation is very I/O-intensive. It’s also destructive because each of the disks and diskgroups need to be reformatted to achieve encryption at the datastore level. So enabling VSAN encryption in the middle of the workweek is ill-advised. Further, it is best practice to deploy your KMS on a separate datastore from the vSAN datastore that you are going to encrypt, thus ensuring that if anything happens to the KMS services, you don’t end up locking down your vSAN datastore. This avoids creating a circular dependency.

**Configuring Network Security Policies**

The key security-related network security policies you can set in the vSphere virtual networking environment are as follows:

◆ Promiscuous mode

◆ MAC address changes

◆ Forged transmits

**Keeping VMs Patched**

As with your ESXi hosts and your vCenter Server computer, it’s imperative to keep the guest OSs in your VMs properly patched. My experience has shown that many security problems could have been avoided with a proactive patching strategy for the guest OSs in the VMs

**Chapter 9 : Creating and Managing Virtual Machines**

VM is an instance of a guest OS running on a hypervisor

**Examining VM from inside**

From the perspective of software running inside a VM, a VM is really just a collection of virtual hardware resources selected for the purpose of running a guest OS instance.

VMware ESXi presents the following fairly generic hardware to the VM:

◆ Phoenix BIOS

◆ Intel 440BX motherboard

◆ Intel PCI AHCI controller

◆ IDE CD-ROM drive

◆ BusLogic parallel SCSI, LSI Logic parallel SCSI, or LSI Logic SAS controller

◆ AMD or Intel CPU, depending on the physical hardware

◆ Intel E1000, Intel E1000e

◆ Standard VGA video adapter

VMware selected this generic hardware to provide the broadest level of compatibility across the entire supported guest OSs. As a result, it’s possible to use commercial off-the-shelf drivers when installing a guest OS into a VM. Figure 9.1 shows a few examples of VMware vSphere providing virtual hardware that looks like standard physical hardware



There is no such physical card as a VMware SVGA 3D display adapter; this is a device that is unique to the virtualized environment. These virtualization-optimized devices, also known as paravirtualized devices, are designed to operate efficiently within the virtualized environment created by the vSphere hypervisor

A VM can include the following types and numbers of virtual hardware devices: ◆ Processors: between 1 and 128 processors with vSphere Virtual SMP (the number of processors depends on your vSphere licenses) ◆ Memory: maximum of 6 TB of RAM ◆ SCSI controller: maximum of 4 SCSI controllers ◆ SATA controller: maximum of 4 SATA controllers ◆ Network adapter: maximum of 10 network adapters ◆ Parallel port: maximum of 3 parallel ports ◆ Serial port: maximum of 32 serial ports ◆ Floppy drive: maximum of 2 floppy disk drives on a single floppy disk controller ◆ A single USB controller with up to 20 USB devices connected ◆ Keyboard, video card, and mouse.

**Examining Virtual Machines from the Outside**

From the perspective of an ESXi host, a VM consists of several types of files stored on a supported storage device. **The two most common files that compose a VM are the configuration file and the virtual hard disk file**. The configuration file—hereafter referred to as the **—is a plain-text file** identified by a .vmx filename extension, and it functions as the virtual resource recipe of the VM. The VMX file defines the virtual hardware that resides in the VM. The number of processors, the amount of RAM, the number of network adapters, the associated MAC addresses, the networks to which the network adapters connect, and the number, names, and locations of all virtual hard drives are stored in the configuration file.

**The virtual hard disk file, identified by a .vmdk filename extension and hereafter referred to as the VMDK file, holds the actual data stored by a VM. Each VMDK file represents a disk device.**

For a VM running Windows, the first VMDK file would typically be the storage location for the C: drive. For a Linux system, it would typically be the storage location for the root, boot, and a few other partitions. Additional VMDK files can be added to provide additional storage locations for the VM, and each VMDK file will appear as a physical hard drive to the VM.

**H.W.** How to create a VM?

Sizing Virtual Machines

Simply sizing your virtual machines to the same specifications used for physical servers can lead to oversizing (or undersizing) virtual machines unnecessarily. Both oversizing and undersizing a virtual machine can lead to performance problems for that virtual machine and for other virtual machines on the same ESXi host.

Instead, a process called capacity planning can help you understand how to size your virtual machines. With capacity planning you learn over time how your current physical servers are utilized and then use that information to size your virtual machines.

Instead, a process called capacity planning can help you understand how to size your virtual machines. **With capacity planning you learn over time how your current physical servers are utilized and then use that information to size your virtual machines you’ll need to support the environment.** For example, if you monitor 70 total physical servers, the tools may tell you that, based on the actual utilization of each server, you need only 7 total ESXi hosts to support those servers as virtual machines. Your results will vary depending on the actual utilization in your environment.

Naming Virtual Machines

**We recommend making the display names of VMs match the hostnames configured in the guest OS being installed**. For example, if you intend to use the name Server1 in the guest OS, the VM display name should match Server1.

Sizing Virtual Machine Hard Disks

The answer to the question—how big to make the hard disks in your VM—is a bit more complicated.

**Working with Installation Media**

VMs have a few ways to access data stored on optical disks. VMs can access optical disks in one of three ways:

1. *Client Device* This option allows an optical drive local to the computer running the vSphere Web Client to be mapped into the VM. For example, if you are using the vSphere Web Client on your corporate-issued laptop, you can simply insert a CD/DVD into your local optical drive and map that into the VM with this option. This is the quick-and-easy method referenced earlier.
2. *Host Device* This option maps the ESXi host’s optical drive into the VM. VMware administrators would have to insert the CD/DVD into the server’s optical drive in order for the VM to have access to the disk.
3. *Datastore or Library ISO File:* These last two options map an ISO image (see the sidebar “ISO Image Basics”) to the VM. Although using an ISO image typically requires an additional step—creating the ISO image from the physical disk—nearly all server software is being distributed as an ISO image that can be leveraged directly from within your vSphere environment.

VMware Tools

* VMware vSphere offers virtualization-optimized (or paravirtualized) devices to VMs in order to improve performance. In many cases, these paravirtualized devices do not have device drivers present in a standard installation of a guest OS.
* The device drivers for these devices are provided by VMware Tools, which is just one more reason why VMware Tools is an essential part of every VM and guest OS installation.
* In other words, installing VMware Tools should be standard practice and not an optional step in the deployment of a VM.
* The VMware Tools package provides the following benefits: ◆ Optimized NIC drivers. ◆ Optimized SCSI drivers. ◆ Enhanced video and mouse drivers. ◆ VM heartbeat. ◆ VSS support to enable guest quiescing for snapshots and backups. Many VMware and third-party applications and tools rely on the VMware Tools VSS integration. ◆ Enhanced memory management. ◆ API access for VMware utilities (such as PowerCLI) to reach into the guest OS.

Note: Quiescing the guest file system is the process of getting the data on a VM into a state suitable for backups. Backup solutions use VM snapshots to copy data from a VM. The operation of quiescing a VM ensures that a snapshot represents a consistent view of the guest file system state at a specific point in time.

**Cloning VMs**

Using vCenter Server to clone a VM—which means also cloning the VM’s virtual disks—keeps you from having to install the guest OS into the cloned VM. By cloning VMs, you eliminate the need to perform a guest OS installation into every new VM.

Creating a Customization Specification

You can create a customization specification in the following two ways:

◆ By creating it during the process of cloning a VM

◆ By using the Customization Specification Manager in vCenter Server

**H.W.** Steps to clone a Virtual Machine

**Introducing vSphere Instant Cloning**

Let’s say your job involves administering a pool of virtual desktops in a virtual desktop infrastructure (VDI) farm, or maybe you are spinning up a large number of VMs for developers to work on an exact replica of the main production environment. VMware has enhanced the traditional cloning capabilities to increase the speed and reduce the resource overhead when there’s a need to create multiple versions of a source VM at the same time.

There are two different ways to instantly clone a virtual machine. The source VM can be in a powered-on and running state or it can be set to a special powered-on and “frozen” state.

**Instant Cloning Running VMs**

The Instant Clone feature will create VMs that run in the exact state their source VM does by leveraging a few technologies: vMotion, Transparent Page Sharing, and delta disks.

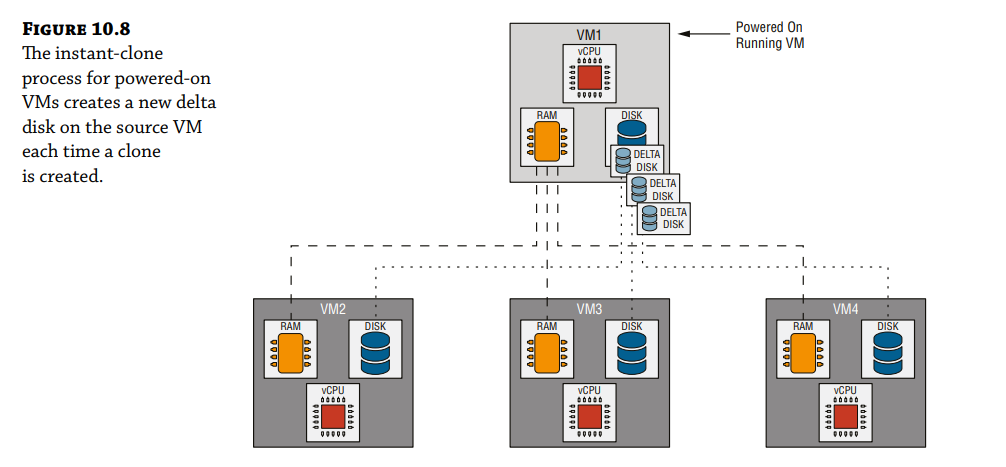
Instant Clone uses a function known as copy-on-write to share both memory and disk data with the source VM, and only making a copy of the changes when the cloned VM needs to modify something.

There are two different ways to instantly clone a virtual machine:

1. instantly cloning a powered-on VM requires no special prerequisites. Simply run the instant clone API against an existing parent VM and a new clone will be created. This may be advantageous if you need an exact copy of a VM at a point in time, but this is not always what you want if the source VM is constantly changing.

Let's say you want to create two instant clones of a source VM, but you initiate two separate clone operations 5 minutes apart. The two resulting VMs are clones of their source at the time of creation, but not necessarily clones of each other if the source VM changed within this 5-minute gap.

When a running VM is instantly cloned, there is a new additional delta disk created on the source VM and the cloned VM’s delta disk is pointed at the new source delta disk in the disk hierarchy. You might recall that there is a limit of 256 snapshots for each VM, which is exactly the same reason there is a limit of 256 instant clones that can be created from a single powered-on VM.



1. Instant Cloning Frozen VMs

You can effectively think of a frozen source VM as a template. This is ideal if you need a large number of identical VMs, a frozen VM only has a single delta disk no matter how many clones are created. Each clone VMs delta disk points back to the same original source VMs delta disk.



A frozen VM is a VM that is powered on but has its CPU halted. It is no longer responding to any OS functions, hardware interrupts, network traffic, or anything else. The ability to freeze a VM requires that VMware Tools is installed in the guest OS and can only be enabled with a command from within the guest OS itself. The only way to unfreeze a frozen VM is to power off or reset the VM.

**Creating Templates and Deploying Virtual Machines**

vCenter Server offers four options for creating templates:

◆ Clone To Template

◆ Clone To Template In Library

◆ Convert To Template

◆ Export OVF Template

**Cloning a Virtual Machine to a Template**(you can refer from pracs)

**Deploying a Virtual Machine from a Template**(you can refer from pracs)

**Using OVF Templates**

1. Open Virtualization Format (formerly called Open Virtual Machine Format) is a Distributed Management Task Force (DMTF) standard format for describing the configuration of a VM. VMware vSphere 6.7 provides OVF support in three ways:

◆ Deploying new VMs from an OVF template (essentially, importing a VM in OVF format) (if required, explain steps in brief)

◆ Exporting a VM as an OVF template (if required,explain steps in brief)

◆ Storing OVF templates within a Content Library (if required,explain steps in brief)

1. Examining OVF Templates

three files make up the OVF template you exported out of vCenter Server:

◆ The name of the manifest file ends in .mf, and the file contains SHA-1 digests of the other two files.

◆The OVF descriptor is an XML document with a filename ending in .ovf and containing information about the OVF template such as product details, virtual hardware, requirements, licensing, a full list of file references, and a description of the contents of the OVF template.

OVF templates can also be distributed as a single file. This single file has a name that ends in .ova and is in TAR format, and the OVF specification has strict requirements about the placement and order of components within the OVA archive

◆The virtual hard disk file has a filename ending in .vmdk.

◆iso file of the VM

◆nvram file

**Using Content Libraries**

Content Libraries are a way of storing VMware templates, OVF templates, ISO/FLP media files, or any file that you may want cataloged separate from your deployed VMs. They can be synchronized between vCenter Servers to allow a “publish once, consume elsewhere” scenario. You can even subscribe to a Content Library that you might not own, such as a public Content Library from your favorite Linux distribution or maybe a private library from your virtual firewall vendor.

**Content Library Data and Storage**

More useful in larger environments with multiple sites, Content Libraries can have any file type uploaded to them for storage and synchronization with other vCenter Servers. When you configure your own Content Library, the storage backing can be either a standard vSphere datastore or it can be configured on a local disk mount point attached to the vCenter Server itself. Most people configure Content Libraries on a datastore for consistency.

**Content Library Synchronization**

The real power behind Content Libraries is the subscription and transfer services that it offers. Content Libraries can be configured in four configurations:

1. Local Content Library:- Local Content Libraries are for individual use only. You cannot subscribe to them or synchronize content between them and other libraries.
2. Local Content Library – Published Externally:- Published Content Libraries are the parent or source library that you can subscribe to. All changes to the content are made to the published library, and the subscribers get those changes based on their individual synchronization settings.
3. Subscribed Content Library – Automatic:- Subscribed Content Libraries that are set to automatically download changes receive all content as soon as it is made available and downloaded from the Published parent library. The content includes all the metadata and the payload binary data.
4. Subscribed Content Library – On Demand:- On Demand Content Libraries only download metadata without any actual payload binary data. When a VM template is requested from this library, synchronization is initiated for that item only.

When you configure Published Content Libraries, they can be password protected, but the credentials are not integrated with vCenter, SSO, or ESXi. This is because Content Libraries are designed to work without vCenter as a boundary.

**H.W.** Create,publish and subscribe to a content library.

**Working with vApps**

With vApps, vSphere administrators can combine multiple VMs into a single unit. For example, a typical multi tier application might have one or more front-end web servers, an application server, and a backend database server. Although each of these servers is a discrete VM and could be managed as such, they are also part of a larger application that is servicing the organization.

**H.W.** how to work with vApps, including creating and editing vApps?

**UNIT 04:** **Managing Resource Allocation**

**Reviewing Virtual Machine Resource Allocation**

A significant advantage of server virtualization is the ability to allocate resources to a virtual machine (VM) based on the actual performance requirements for the guest OS and application or services.

consider the following resource management situations.

◆◆ What does an ESXi host do when it runs out of resources?

◆◆ How does an ESXi host manage VMs that are requesting more resources than the physical server can provide?

◆◆ How can you guarantee that a guest OS and its applications and services get the resources they need without being starved by VMs (e.g., a “noisy neighbor”)?

The fundamental concepts of each mechanism are as follows:

**Reservations:** Reservations act as guarantees for a resource type. Reservations may be used when you want to ensure that, no matter what else is going on, a specific VM or set of VMs is assured access to a set amount of a resource from startup through shutdown.

**Limits:** Limits restrict the amount of a resource type that a VM can consume. By default, a VM has a limit applied based on how it’s constructed. For example, a VM that is configured with a single virtual CPU (vCPU) is limited to using only that single vCPU. This vSphere feature offers you an even greater level of granularity over how the resources are used. Depending on the resource type for which the limit is being applied, the specific behavior of ESXi will change. We will discuss this in detail in this chapter under the sections for each resource type.

**Shares:** Shares establish priority during periods of contention. When VMs compete for

limited resources, an ESXi host prioritizes which VMs move to the front of the queue to gain access to the resources. The feature determines the priority. The VMs with higher share value will be marked for higher priority and therefore will receive prioritized access to an ESXi host’s resources.

**Understanding ESXi Advanced Memory Technologies**

1. Idle Memory Tax(IMT)

Inside each guest OS, VMware Tools should be installed and running where it will use the balloon driver to determine which memory blocks are allocated but idle and, therefore, available to be borrowed.

1. Transparent Page Sharing(TPS)

The next memory-management technology ESXi uses is transparent page sharing (TPS), in which identical memory pages are shared among VMs to reduce the total number of memory pages consumed. **The hypervisor computes hashes of the contents of memory pages to identify pages that contain identical memory.** If it finds a match, TPS compares the matching memory pages to exclude any false positives. Once the pages are confirmed to be identical, the hypervisor will transparently remap the memory pages of the VMs to share the same physical memory page and thereby reduce overall host memory consumption.

1. Ballooning

Ballooning involves the use of a driver—referred to as the balloon driver—installed into the guest OS. The balloon driver is included as part of VMware Tools and is subsequently deployed when the package is installed on a guest OS. **The balloon driver responds to commands from the hypervisor to reclaim memory from the VM’s guest OS. The balloon driver does this by requesting memory from the guest OS—a process called inflating—and then passing that memory back to the hypervisor for use by other VM workloads.**

1. Memory Compression

When an ESXi host reaches the point where hypervisor swapping is necessary, the VMkernel will attempt to compress memory pages and keep them in memory in a compressed memory cache. Pages that can be successfully compressed by at least 50% are placed into the compressed memory cache instead of being written to disk and can then be recovered far faster if the guest OS needs to access the memory page. Memory compression can dramatically reduce the number of pages that must be swapped to disk and improves the performance of an ESXi host that is under strong memory pressure.

1. Swapping

Two forms of swapping are involved in managing memory in ESXi. The first is guest OS swapping, in which the guest OS inside the VM swaps pages out to its own virtual disk according to its memory management algorithms. The second type of swapping is hypervisor swapping. When none of the previously described technologies—transparent page sharing, ballooning, and memory compression—trim guest OS memory usage enough, the ESXi host will invoke hypervisor swapping. Hypervisor swapping means that ESXi will begin swapping memory pages to disk in an effort to reclaim memory that is required for workloads. This swapping takes place without regard to whether the pages are being actively used by the guest OS. Since disk response times are significantly slower than memory response times, virtual-machine guest OS performance will be severely impacted when hypervisor swapping is invoked. For this reason, ESXi will not invoke swapping unless it is necessary. Hypervisor swapping is the last-resort option after all previously discussed memory-management techniques have been exhausted.

**How Reservations, Limits, and Shares Work with Memory**

*Using memory reservation*

When a memory reservation is specified in the virtual-machine resource settings, it is the amount of actual physical-server memory that the ESXi host must provide to this VM for it to power on. This memory reservation guarantees the designated amount of RAM configured in the Reservation setting. In the absence of a reservation, the VMkernel has the option to provide virtual-machine memory from the VMkernel swap. V*Mkernel swap is the hypervisor swapping memory-management technique.* VMkernel swap is implemented as an on-disk file with the .vswp extension and is automatically created when a VM is powered on. Note that the VMX swap files are unrelated to the VMkernel swap file and are overhead for the VMX process running the VM on the ESXi host.

*Using Memory Limits*

It sets the actual limit on how much physical memory may be used by that VM.

To see this behavior in action, we will change the limit on this VM from the default setting of Unlimited to 2,048 MB.

The effective result of the configuration change is as follows:

◆ The VM is configured with 4,096 MB of memory and the guest OS inside the VM has 4,096 MB available to use.

◆ The VM has a reservation of 1,024 MB of memory, which means that the ESXi host must allocate and guarantee 1,024 MB of physical memory to the VM.

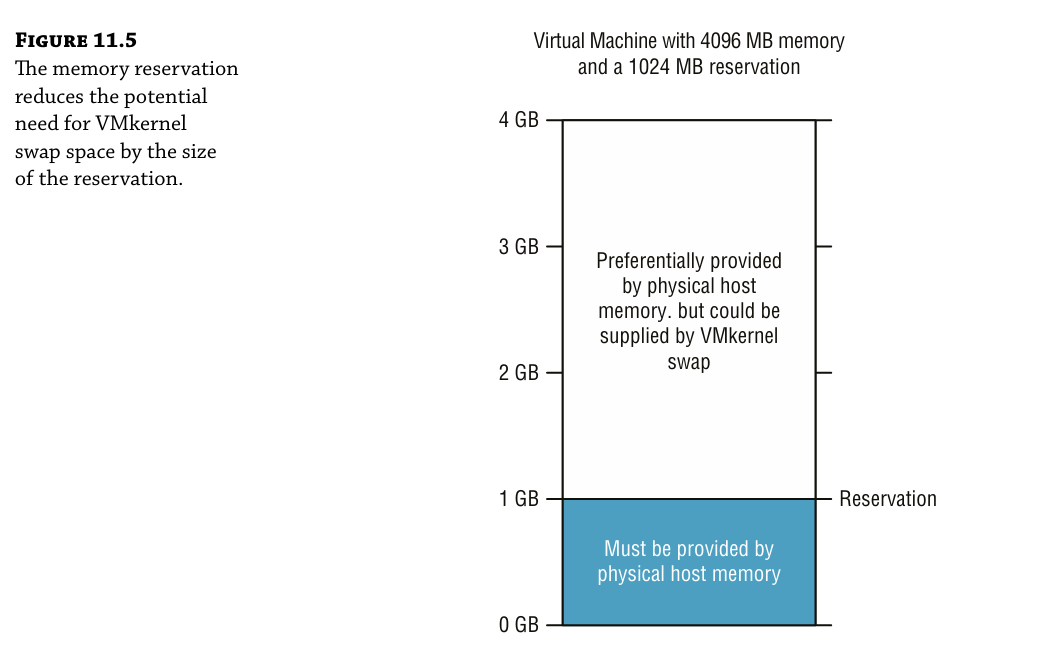
◆ Assuming the ESXi host has enough physical memory available the hypervisor will

allocate memory to the VM, as needed, up to 2,048 MB (the limit). Upon reaching 2,048

MB, the balloon driver inflates to prevent the guest OS from using more memory beyond

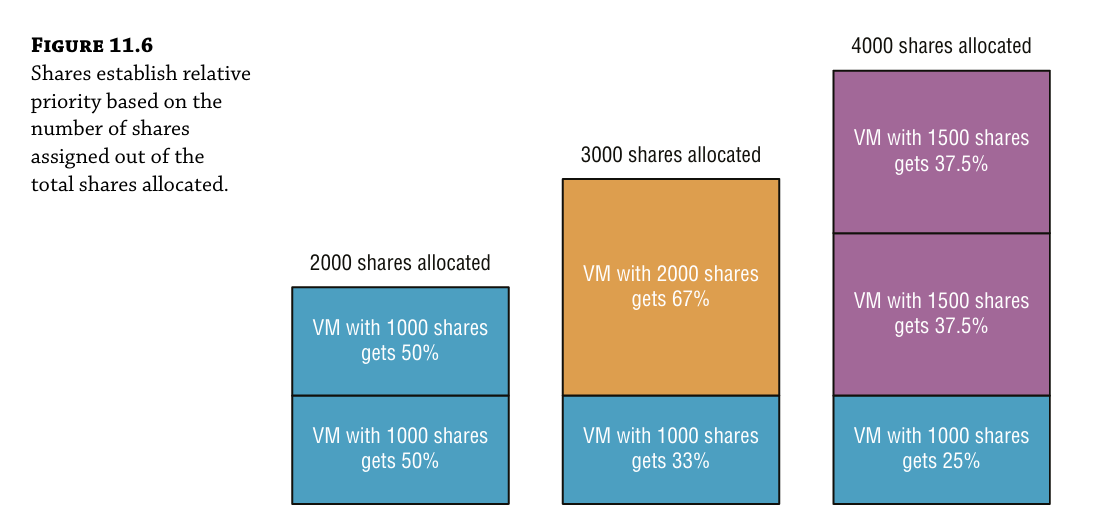
the limit. When the memory demand on the guest OS drops below 2,048 MB, the balloon driver deflates and returns memory to the guest. The effective result of this behavior is that the memory the guest OS uses remains below 2,048 MB (the limit).

◆ The “memory gap” of 1,024 MB between the reservation and the limit may be provided by either physical host memory or the VMkernel swap space. As always, the ESXi hypervisor will allocate physical memory if available.



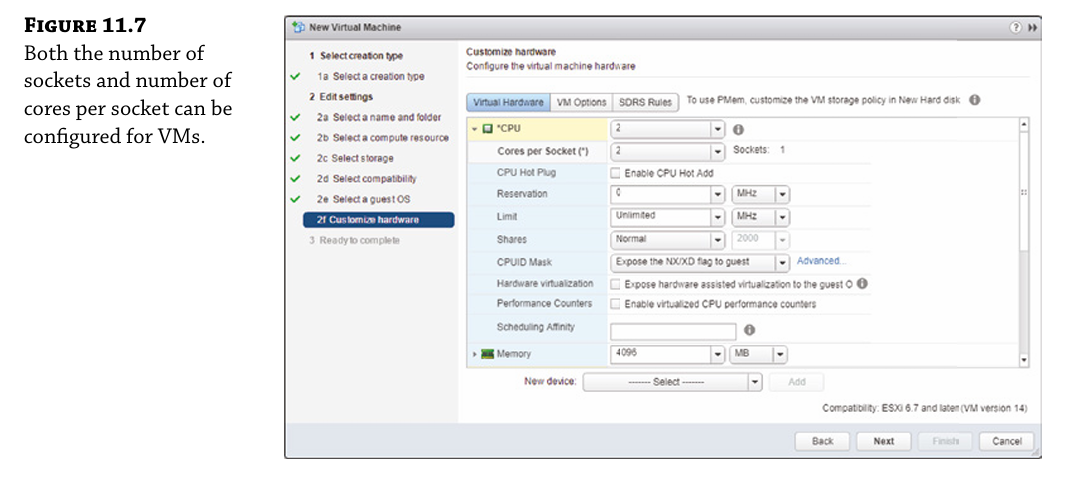
*Using Memory Shares*

Shares are a proportional system that allows you to assign resource prioritization to VMs, but shares are used only when the ESXi host is experiencing physical memory contention—the VMs on a host are requesting more memory than the host can provide. If an ESXi host has plenty of memory available, shares do not play a role. However, when memory resources are scarce and the hypervisor needs to decide which VM should be given access to memory, shares establish a priority setting for a VM requesting memory that is greater than the reservation but less than the limit.

Assume you have two VMs—VM1 and VM2—each with a 1,024 MB reservation and a configured maximum of 4,096 MB, and both are running on an ESXi host with less than 2 GB of memory available to the VMs. If the VMs have an equal number of shares, such as 1,000 each (we will discuss values later), then as each VM requests memory above its reservation value, each will receive an equal quantity of memory from the ESXi host. Furthermore, because the host cannot supply all the memory to both VMs, each will swap equally to disk (VMkernel swap file). This assumes, of course, that ESXi cannot reclaim memory from other running VMs using the balloon driver or other memory-management technologies described earlier. If you change VM1’s Shares setting to 2,000, then VM1 will have twice the shares VM2 has assigned to it. This means that when VM1 and VM2 are requesting the memory above their respective Reservation values, VM1 gets two memory pages for every one memory page that VM2 receives. If VM1 has more shares, VM1 has a higher-priority access to available memory in the host. Because VM1 has 2,000 out of 3,000 shares allocated, it will get 67%; VM2 has 1,000 out of 3,000 shares allocated and therefore gets only 33% percent. This creates the two-to-one behavior described previously. Each VM is allocated memory pages based on the proportion of the total number of shares allocated across all VMs. Figure 11.6 illustrates this behavior. **

**How Reservations, Limits, and Shares Work with CPUs**

When creating a VM using the vSphere Web Client, you must configure two CPU-related fields. First, select how many virtual CPUs you want to allocate to the VM, and then assign the number of cores to those CPUs (see Figure 11.7). These CPU settings allow the VM’s guest OS to use between 1 and 128 virtual CPUs from the ESXi host, depending on the guest OS and the vSphere edition license.

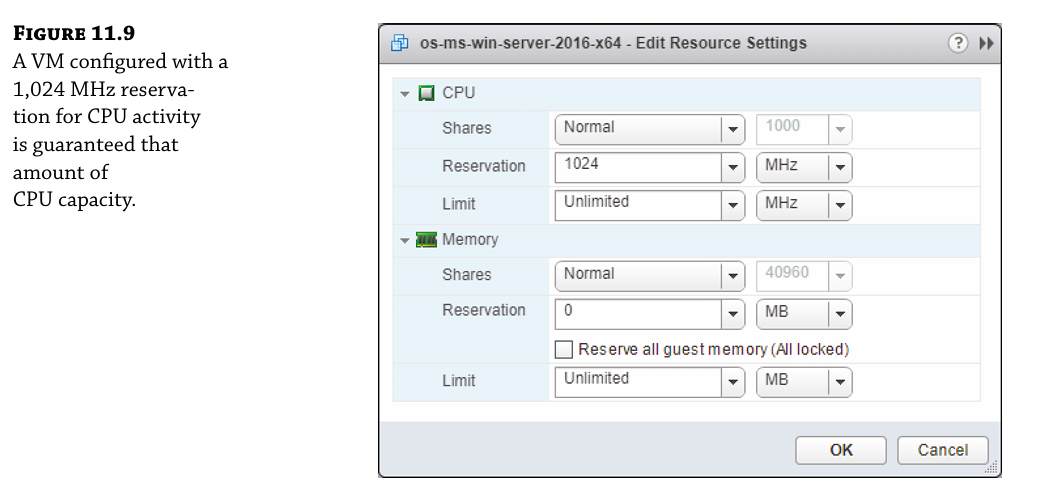


When a VM is created with a single vCPU, the total maximum CPU cycles for the VM are equal to the clock speed of the host system’s core. In other words, if you create a new VM, it can see through the “hole in the system board,”(Think about a virtual system board that has a CPU socket “hole” where the CPU is added— and the guest OS simply looks through the socket and sees one of the cores in the host server.) and it sees whatever the core is in terms of clock cycles per second—an ESXi host with a 2.2 GHz CPU’s 10-core processor in it will allow the VM to see one 2.2 GHz core.

**Setting CPU Affinity:** In addition to reservations, limits, and shares, vSphere offers a fourth option for managing CPU usage: CPU affinity. CPU affinity allows you to statically associate a VM to a specific physical CPU core. CPU affinity is generally not recommended; it has a list of rather significant drawbacks: ◆ CPU affinity prevents vMotion. ◆ The hypervisor is unable to load-balance the VM across all the processing cores in the server. This prevents the hypervisor’s CPU scheduler from making efficient use of the host resources. ◆ Because vMotion is inhibited, you cannot use CPU affinities in a cluster where vSphere DRS isn’t set to Manual operation.

**Using CPU Reservations:**

By default, a VM is not guaranteed any CPU scheduling time by the VMkernel. When the VM has work to be done, it places its CPU request into the CPU queue, and the VMkernel schedules the request in sequence along with all of the other VM requests. On a lightly loaded ESXi host, it is unlikely the VM will wait long for CPU time; however, on a heavily loaded host, the time the VM may wait could be significant.



The ESXi host must satisfy the reservation by providing enough resources to meet the reservations. If each VM you create has a 1,024 MHz reservation and your host has 12,000 MHz of CPU capacity, you can power on no more than 11 VMs (1,024 MHz × 11 = 11,264 MHz), even if each machine is idle.

Suppose you have two VMs, VM1 and VM2, and VM1 has a CPU reservation of 1,024 MHz and VM2 has no reservation. If VM1 is idle and not using its reserved CPU cycles, those cycles can be given to VM2. If VM1 suddenly needs cycles, VM2 doesn’t get them anymore, and they are assigned to VM1.

**Using CPU Limits:**

This effectively limits the ability of a VM to use a maximum number of clock cycles per second, regardless of what the host has available. For instance, you could set a 500 MHz limit on the DHCP server so that when it re-indexes the DHCP database, it will not attempt to take all the 2.2 GHz on the processor core that it can see. The CPU limit allows you to throttle the VM down to use less processing power than is available on the physical core. The Limit setting is a true limit; the VM will not be scheduled to run on a physical CPU core more than the limit specifies, even if there are plenty of CPU cycles available.

**Using CPU Shares:**

Shares of CPU determine how much CPU is provided to a VM in the face of contention with other VMs needing CPU activity. In other words, the VM is granted access to its reservation cycles regardless of what else is happening on the host, but if the VM needs more CPU cycles—and there is competition—then the share values come into play.

Let’s understand the CPU cycle allocation with various scenarios. (refer all the scenarios from reference book pg. 613 to 614)

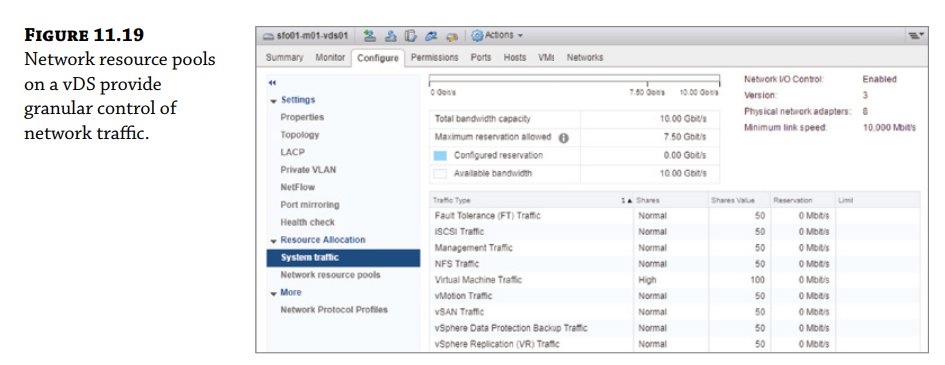
**Using resource pool**

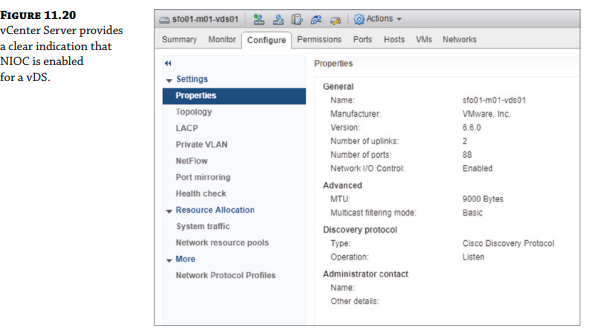
**Regulating Network I/O Utilization**

vSphere offers another type of resource pool, a network resource pool, which allows you to control and prioritize network utilization. Using network resource pools—to which you assign shares and limits—you can control incoming and outgoing network traffic. This feature is referred to as vSphere Network I/O Control or NIOC.

When you enable NIOC on a vSphere Distributed Switch (vDS), vSphere activates the following nine predefined network resource pools:

◆ Fault Tolerance (FT) Traffic ◆ Management Traffic ◆ NFS Traffic ◆ Virtual Machine Traffic ◆ vSAN Traffic ◆ iSCSI Traffic ◆ vMotion Traffic ◆ vSphere Data Protection Backup Traffic ◆ vSphere Replication (VR) Traffic





Two steps are involved in setting up and using NIOC. First, you must enable NIOC on a vDS, and second, you must create and configure the network resource pools as necessary. The first of these steps is already complete if you create a vDS with a version set to 6.0.0 or higher, since NIOC is enabled by default.

Perform the following steps to enable NIOC on an existing vDS:

1. If vSphere Web Client is not already running, open a browser, connect to the vSphere Web Client on your vCenter Server instance, and log on. Because NIOC relies on vDS, and vDS is available only with vCenter, NIOC cannot be used when connected directly to an ESXi host with the ESXi Host Client.

2. Navigate to the Networking view using the navigation bar or the Home screen.

3. Select the vDS for which you want to enable NIOC.

4. Right-click the vDS.

5. Click Edit and then Edit Settings.

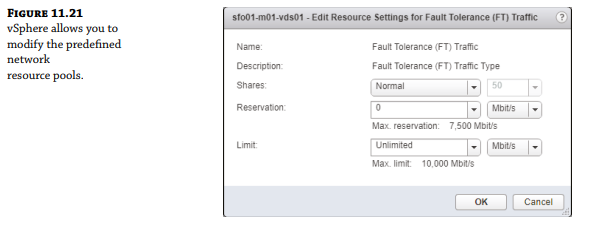
6. Select Enabled in the Network I/O Control drop-down menu, and then click OK.

A network resource pool consists of three settings:

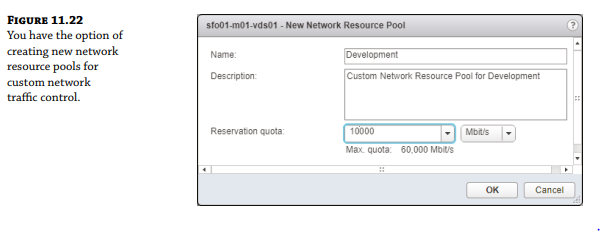
◆ The first value is Shares. Like the shares you used to prioritize access to CPU or memory when there was contention, physical adapter shares in a network resource pool establish priority for access to the physical network adapters when network contention exists. As with other types of shares, this value does not apply when no contention exists. This value can be set to one of three predefined values, or you can set a custom value of up to 100. For the predefined values, Low translates to 25 shares, Normal equates to 50 shares, and High equals 100 shares.

◆ The second value is Reservation. This value guarantees an amount of bandwidth (in Mbps) to the network resource pool.

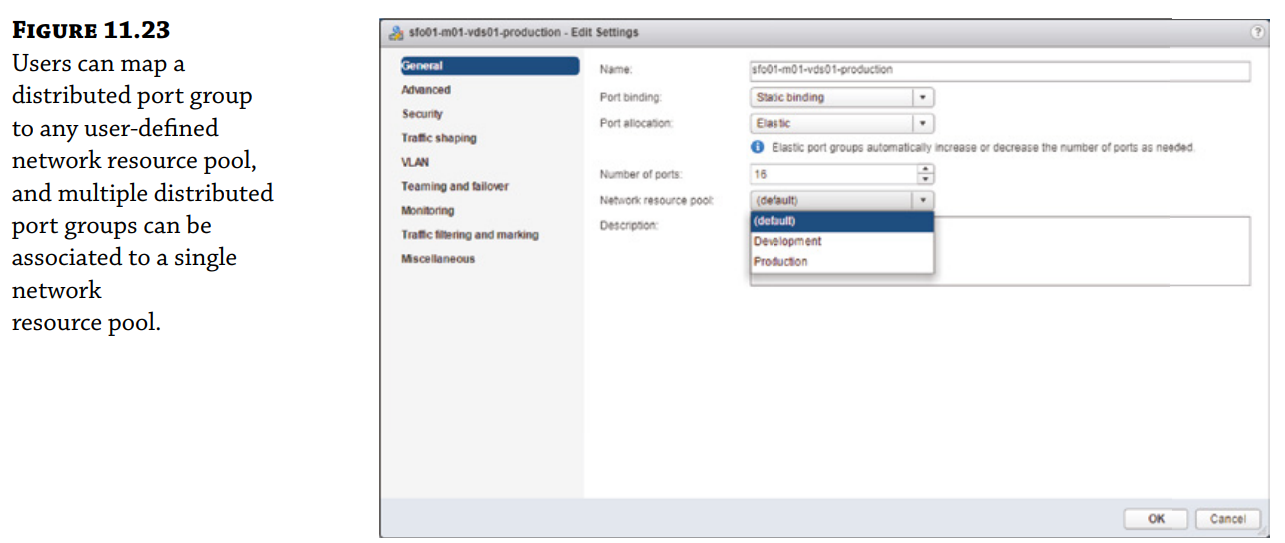
◆ The final value is Limit. This value specifies an upper limit on the amount of network traffic (in Mbps) that the network resource pool may consume. Leaving Unlimited selected means that only the physical adapters can limit the network resource pool. Figure 11.21 shows all three of the values for the predefined Fault Tolerance (FT) Traffic network resource pool.



You might prefer to leave the predefined network resource pools intact and create your own.



NIOC offers a powerful way to help ensure that all the various types of converged network traffic present in a vSphere environment will coexist properly, especially as organizations move to (or beyond) 10 Gbe Ethernet.



**Controlling Storage I/O Utilization**

In many ways, resource allocation and utilization for storage is similar to networking. There are two metrics that vSphere can use to help determine storage utilization. The first is latency, and the second is peak throughput. Using one of these two metrics to detect contention, vSphere can offer shares (to establish priority when contention occurs) as well as limits (to ensure that a VM does not consume more than its fair share of storage resources). The feature that enables this functionality is called Storage I/O Control or SIOC.

Without SIOC, enabling shares on a VM virtual disk(s) is only effective for that specific host; the hosts in a cluster did not exchange information about how many shares each VM was allocated or how many shares were assigned in total.

By using vCenter Server as its central information store, SIOC combines all the assigned shares across all the VMs on all the hosts and allocates storage I/O resources in the proper ratios according to the shares assignment.

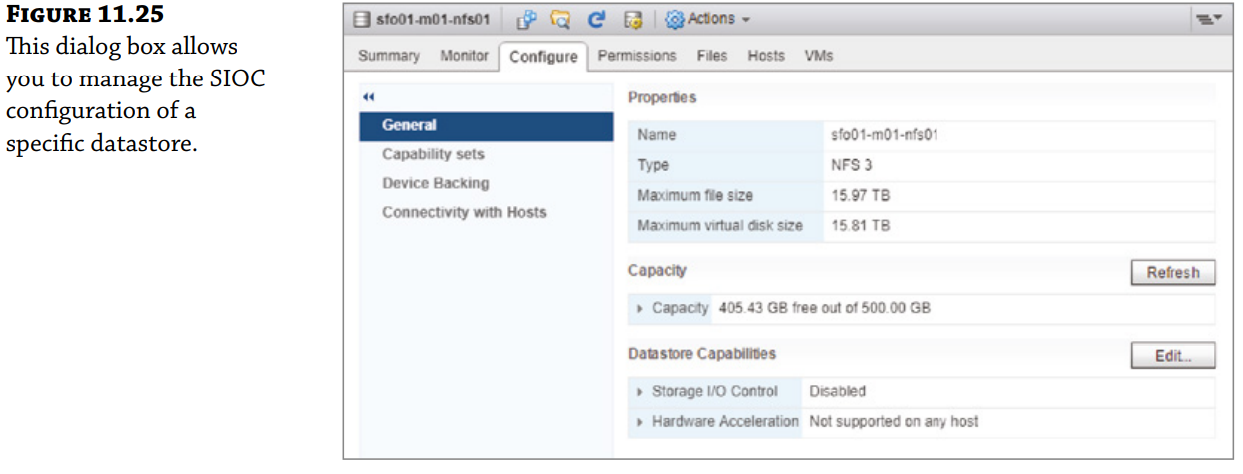
To make this work, SIOC has a few requirements that must be met:

◆ All datastores that are SIOC-enabled must be managed under a single vCenter Server instance.

◆ SIOC is supported on VMFS datastores connected via Fibre Channel (including FCoE) and iSCSI. NFS datastores are also supported. Raw device mappings (RDMs) are not supported.

◆ Datastores must have only a single extent. Datastores with multiple extents are not supported.

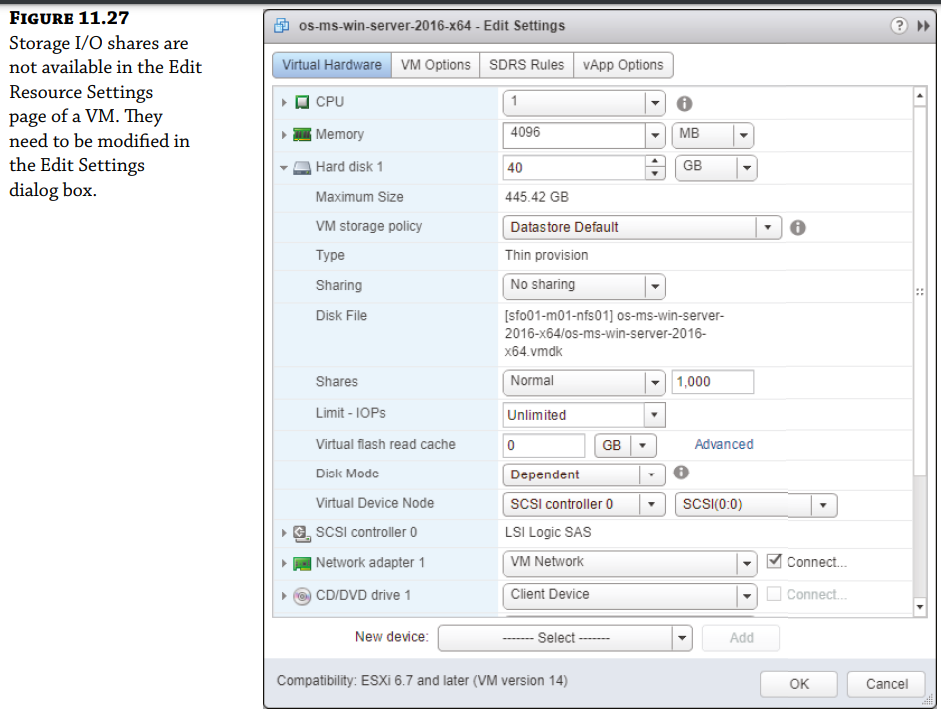
**Enabling Storage I/O Control**



Click on edit to enable SIOC for this datastore.

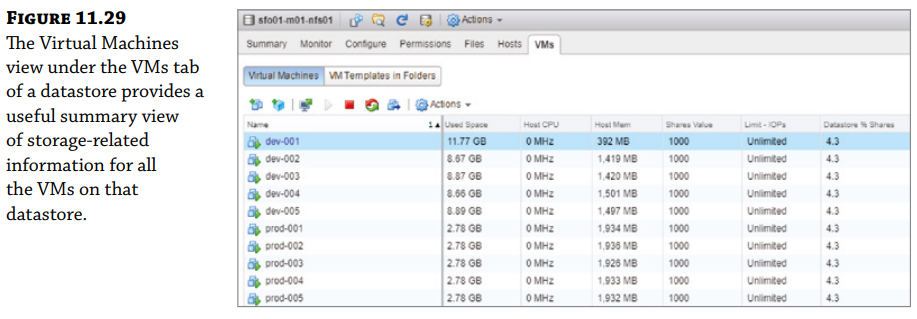
**Configuring Storage Resource Settings for a Virtual Machine**

Assigning Storage I/O Shares



Configuring Storage I/O Limits You can also set a limit on the number of IOPS that a VM may generate.

Viewing Storage I/O Resource Settings for Virtual Machines In the Datastores And Datastore Clusters view, you can view a list of all the datastores managed by a vCenter Server instance.

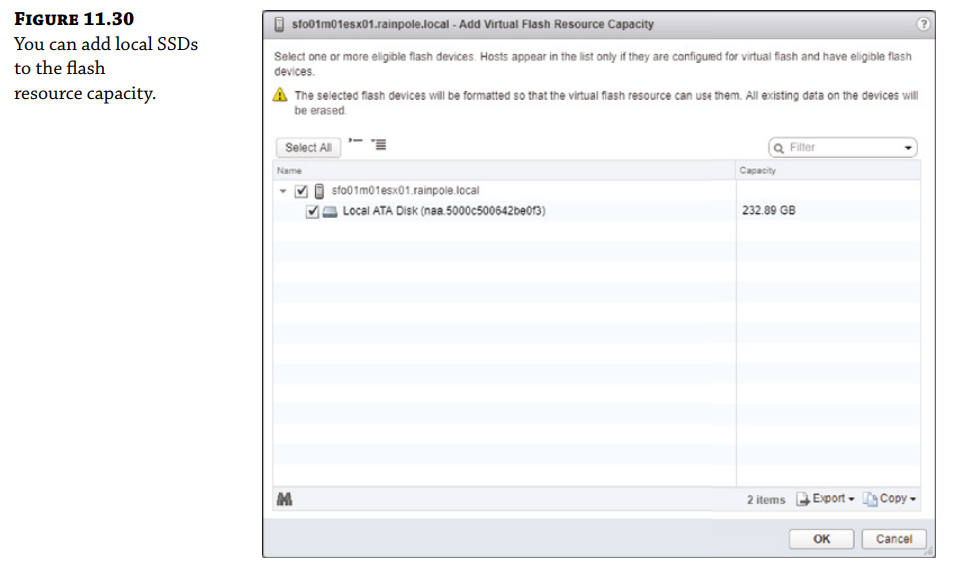


**Using Flash Storage**

As physical-to-virtual ratios increase, and environments become denser, the need for faster storage grows. Flash storage has quickly become an industry standard and many servers and arrays ship with flash-based storage.

vSphere offers two ways of using local flash-based host storage. vSphere Flash Read Cache is a feature that acts as a buffer for I/O on a per-VM basis. The other feature, Swap to Host Cache, is used to allocate local flash disks as swap space. These features are not mutually exclusive; they can be enabled at the same time and can even be backed by the same device, but they work in completely different ways.

**Enabling vSphere Flash Read Cache for Virtual Machines** (refer from reference book pg. No. 640 to 641)



**Configuring Swap to Host Cache**

Swap to Host Cache is a feature that allocates a portion of local flash storage to be the location of a VM swap file. *This feature is beneficial only within environments that have memory contention to the point that the hypervisor must swap unreserved VM memory to disk.*

To configure a host for Swap to Cache, a VMFS datastore that has been identified as being backed by a flash needs to be present. The following steps outline how to enable this feature:

1. If vSphere Web Client is not already running, open a browser, connect the vSphere Web Client on your vCenter Server instance, and then log on.

2. Navigate to the Hosts And Clusters view.

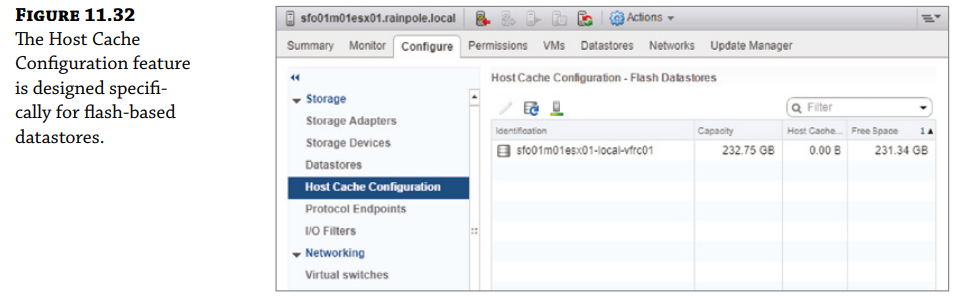
3. Select a host from the Inventory tab.

4. Click the Configure tab.

5. Select Host Cache Configuration

6. Highlight the appropriate flash-backed VMFS datastore and click the Edit pencil icon. 7. Ensure the Allocate Space For Host Cache option is checked. A custom size may be provided if you would like to use this datastore for other purposes.

8. Click OK to complete the configuration.



*Once the Swap to Host Cache feature is enabled, the host proceeds to fill the allocated size on the datastore with .vswp files. These files will be used instead of the .vswp files typically co-located within the same datastore as the other VM files. Keep in mind that this setting must be enabled on every host within a cluster because it is not cluster aware. When a VM is vMotioned to a host that does not have the Swap to Host Cache feature enabled, the normal datastore-based swap files will be used.*

**CHAPTER 12: Balancing Resource Utilization**

VMware vSphere balances the utilization of resources in the following ways:

* vMotion: vMotion, commonly known as live migration, is used to manually balance compute resource utilization between ESXi hosts and clusters.
* Storage vMotion: Storage vMotion is the storage equivalent of vMotion. It is used to manually balance storage utilization between two datastores.
* Cross vCenter vMotion: Cross vCenter vMotion is used to manually migrate workloads between vSphere environments, such as between datacenters.
* vSphere Distributed Resource Scheduler: vSphere Distributed Resource Scheduler (DRS) is used to automatically balance compute resource utilization among two or more hosts in a vSphere cluster.
* Storage DRS: Just as Storage vMotion is the storage equivalent of vMotion, Storage DRS (SDRS) is the storage equivalent of DRS, and it is used to automatically balance storage utilization among two or more datastores within a datastore cluster before and after initial placement of VM files.

**Exploring vMotion**

**vMotion can perform a live migration of a powered-on VM from one host to another without an interruption in service.** This is a zero-downtime operation; network connections are not dropped, and applications or services continue to run uninterrupted. In fact, application owners and users are unaware that the VM has been migrated between the physical hosts. **When vMotion is initiated to migrate a VM between hosts, the resource allocation is also migrated—CPU and memory—from one host to another.** This makes vMotion an extremely effective tool for manually load-balancing VMs across hosts and eliminating “hot spots”—heavily utilized ESXi hosts— within the datacenter.

Fundamentally, vMotion copies the memory content of a powered-on VM from one host to another followed by transferring the control of the VM disk files to the target host. Let’s take a closer look at how vMotion operates in the following sequence:

Refer from fig. 12.1 to 12.5

To enable vMotion, vCenter Server is required, and the hosts and VMs in the process must meet a specific set of requirements. Let’s review these requirements.

◆◆ Each host must be licensed for the vMotion feature.

◆◆ The shared storage for VM files must be accessible by both the source and target host, such as an NFS datastore.

◆◆ A VMkernel port defined and enabled for vMotion traffic is enabled on each host backed by a network interface card (NIC); 1 Gbps or faster is recommended.

**Using Storage vMotion**

vMotion and Storage vMotion are like two sides of the same coin. Traditional vMotion migrates a running VM from one host to another, moving the CPU and memory resource usage between hosts—but leaving the storage of the VM unchanged. vMotion enables you to manually balance the CPU and memory resource by shifting VMs from host to host. Storage vMotion, however, migrates the virtual disks of a powered on VM from one datastore to another, all while the VM continues to execute—and therefore using CPU and memory resources—on the same host. This feature allows you to manually balance the utilization of a datastore by shifting VM storage from datastore to datastore. Like vMotion, Storage vMotion is a live migration; the VM does not incur any downtime when migrating virtual disks from one datastore to another.

The process for Storage vMotion is relatively straightforward:

1. vSphere copies over the nonvolatile files that constitute the VM: the configuration file,

VMkernel swap, log files, and snapshots.

2. vSphere starts a shadow VM on the target datastore. Because this shadow VM does not yet have a virtual disk that has not been copied over, it sits idly waiting.

3. Storage vMotion first creates the target disk. Then a mirror device—a driver that mirrors I/Os between the source and target—is inserted into the data path between the VM and the underlying storage.

4. With the I/O mirroring driver in place, vSphere makes a single-pass copy of the virtual

disk(s) from the source to the target. As changes are made to the source, the I/O mirror

driver ensures that those changes are also reflected to the virtual disks on the target

datastore.

5. When the virtual disk copy is complete, vSphere quickly quiesces(dormant) and resumes the VM to transfer control over to the shadow VM that has been created on the target datastore. As with vMotion, this generally happens so quickly that there is no disruption of service.

The files on the source datastore are deleted.

**Combining vMotion with Storage vMotion**

vMotion and Storage vMotion can be combined into a single operational process to produce what is often called a shared nothing vMotion.

Refer fig. 12.17 to 12.19

**Cross-vCenter vMotion**

Traditionally, vMotion has been an intra-cluster solution for balancing workloads. vSphere 6 pushed vMotion further, allowing you to migrate VMs across clusters, virtual standard switches, virtual distributed switches, and vCenter Servers. In addition, you can migrate between geographically dispersed environments with an RTT of 150 ms or less.

Although these capabilities provide an increase in mobility of workloads, what are the operational impacts? The good news is that the universally unique identifier (UUID) of the VM (not to be confused with the BIOS UUID) is retained during the transfer, so most of the information that you want is transferred along with the VM. vSphere HA configurations and DRS rules, events, tasks, and alarms will all move with the VM, as will any configured shares, reservations, or limits. The only downside is that performance data—which is written to the vCenter Server database—will not be migrated with the VM. (Note, however, that if each of the vCenter Server instances is monitored by the same vRealize Operations instance or cluster, you have another very powerful method to review historical performance and capacity data.)

**H.w. Explain how to perform cross -vCenter vMotion.**

**Exploring vSphere Distributed Resource Scheduler**

DRS is a feature of vCenter Server and has the following two main functions:

◆ To decide which host in a cluster should run a VM when it is powered on, or intelligent placement

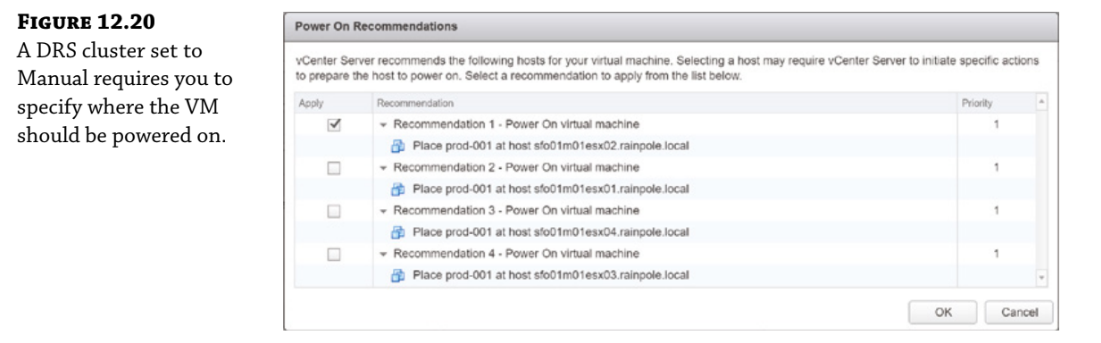
◆ To evaluate the load on the cluster over time and either make recommendations for migrations or use vMotion to automatically migrate VMs and balance the cluster workloads

By default, DRS checks every 5 minutes (or 300 seconds) to see if cluster work-loads are well balanced. Actions within a cluster, such as adding or removing an ESXi host or changing the resource settings of a VM, will invoke DRS. When DRS is invoked, it will calculate the imbalance of the cluster, apply any resource controls (such as reservations, shares, and limits), and, if necessary, generate recommendations for migrations of VMs within the cluster. Depending on the DRS configuration, these recommendations may be applied automatically, meaning that VMs will be automatically migrated between hosts by DRS to maintain cluster balance.

The following are the **three automated behavior of vDRS:**

1. Manual Automated behavior

**When a DRS cluster is set to Manual, each time you power on a VM, the cluster prompts you to select an ESXi host where that VM should be hosted.** The dialog box ranks the available hosts according to suitability at that moment in time: the lower the priority, the better the choice, as shown in Figure 12.20.



The Manual setting will also suggest vMotion migrations when DRS detects an imbalance between the hosts in the cluster.

You find the recommended list of migrations by selecting the cluster in the inventory, selecting the Monitor tab, followed by selecting the vSphere DRS section.

From here, the Run DRS Now button allows you to agree with any pending DRS recommendations and initiate a migration. vMotion handles the migration automatically.

1. Partially Automated behavior

If you select the Partially Automated setting in the DRS Automation settings, DRS will make an automatic decision about which host a VM should run on when it is initially powered on. In this scenario, t**he placement is performed without prompting the user performing the power on task**; **however, it will still prompt for all migrations on the migration recommendations.** Thus, the partially automated configuration automatically performs the initial placement, but migrations are manually invoked.

1. Fully Automated behavior

The third DRS Automation level is Fully Automated. With this setting, DRS **makes the decisions for initial placement and automatic vMotion decisions based on the selected automation level without user intervention.**

Five positions exist for the Fully Automated slider control setting of a DRS cluster. The values of the slider control the range from Conservative to Aggressive. Conservative automatically apply recommendations ranked as priority 1 recommendations. Any other migrations are listed on the DRStab and require administrator approval. If you move the slider bar from the most conservative setting to the next stop to the right, then all priority 1 and priority 2 recommendations are automatically applied; recommendations higher than priority 2 will wait for administrator approval. With the slider all the way over to the Aggressive setting, any imbalance in the cluster that causes a recommendation is automatically approved (even priority 5 recommendations). Be aware that this can cause additional stress in your ESXi host environment because even a slight imbalance will trigger a migration.

**Working with Distributed Resource Scheduler Rules**

vSphere lets you create DRS rules. DRS supports three types of DRS rules:

◆ VM affinity rules—referred to as **Keep Virtual Machines Together** in the vSphere Web Client

◆ VM anti-affinity rules—referred to as **Separate Virtual Machines** in the vSphere Web Client

◆ Host affinity rules—referred to as **Virtual Machines To Hosts** in the vSphere Web Client

**Creating VM Affinity Rules**

VM affinity rules keep VMs together on a host. Consider, for example, a traditional three-tier application where you have a frontend web server, a mid-tier application server, and a backend database server. These components frequently communicate with each other and you would like for the communication to take advantage of the high-speed bus of the hosts rather than going across the network between hosts in the cluster. In this example, you could define an affinity rule—Keep Virtual Machines Together—that would ensure that these three VMs stay together on a host within the cluster.

**Creating VM Anti-Affinity Rules**

Consider, for a moment, our previous example of a traditional three-tier application. But this time, we will slightly modify it. In this example, the traditional three-tier application server has grown. We now have two load-balanced frontend web servers, two load-balanced mid-tier application servers, and an active/active clustered backend database server. In this scenario, you would like to ensure that failure of a host would not have the potential to impact one or more of the tiers completely. In other words, you want to ensure that for each tier, the VMs for that tier are not on the same host in the cluster. In this example, a VM anti-affinity rule can mitigate the concern.

**Working with Host Affinity Rules**

In addition to VM affinity and VM anti-affinity rules, DRS supports a third type of DRS rule: the host affinity rule.

Host affinity rules govern the relationships between VMs and hosts in a cluster, letting you control which hosts in a cluster can run which VMs—sometimes referred to as “sub-cluster capacity.” When combined with VM affinity and VM anti-affinity rules, you can create complex rule sets to model the relationships between applications and workloads in your datacenter.

Before you can start creating a host affinity rule, you must create at least one VM DRS Group and at least one Host DRS Group. There are four host affinity rule behaviors:

◆ Must Run On Hosts In Group

◆ Should Run On Hosts In Group

◆ Must Not Run On Hosts In Group

◆ Should Not Run In Hosts In Group

These rules are, for the most part, self-explanatory. Each rule is either mandatory (a “Must” rule) or preferential (a “Should” rule) plus the affinity (“Run On”) or anti-affinity (“Not Run On”).

**Working with Storage DRS**

**Storage DRS (SDRS) builds on the functionality of Storage I/O Control(SIOC0) and Storage vMotion, providing the ability to perform automated balancing of storage utilization.**

Like DRS, SDRS is built on some closely related concepts and topics:

◆ Just as DRS uses clusters as a collection of hosts on which to act, SDRS uses datastore clusters as a collection of datastores on which it acts.

◆ Just as DRS can perform both initial placement and manual and ongoing balancing, SDRS also performs initial placement of VMDKs and ongoing balancing of VMDKs—as well as related VM files. The initial placement functionality of SDRS is especially appealing because it helps simplify the VM provisioning process for vSphere administrators.

◆ Just as DRS offers affinity and anti-affinity rules to influence placement recommendations, SDRS offers VMDK affinity and anti-affinity functionality.

**SDRS uses the idea of a *datastore cluster*—a group of datastores treated as shared storage resources—to operate.**

Specifically, VMware provides the following guidelines for datastores that are combined into datastore clusters:

◆ Datastores of different sizes and performance characteristics can be combined in a datastore cluster. Generally, this is not a recommended practice unless you have very specific requirements. Additionally, datastores from different arrays and vendors can be combined into a datastore cluster. However, you cannot combine NFS and VMFS datastores into the same datastore cluster.

◆ You cannot combine replicated and non replicated datastores into an SDRS–enabled

datastore cluster.

◆ Datastores shared across multiple datacenters are not supported for SDRS.

Now let’s take a more in-depth look at configuring SDRS to work with the datastore cluster(s)

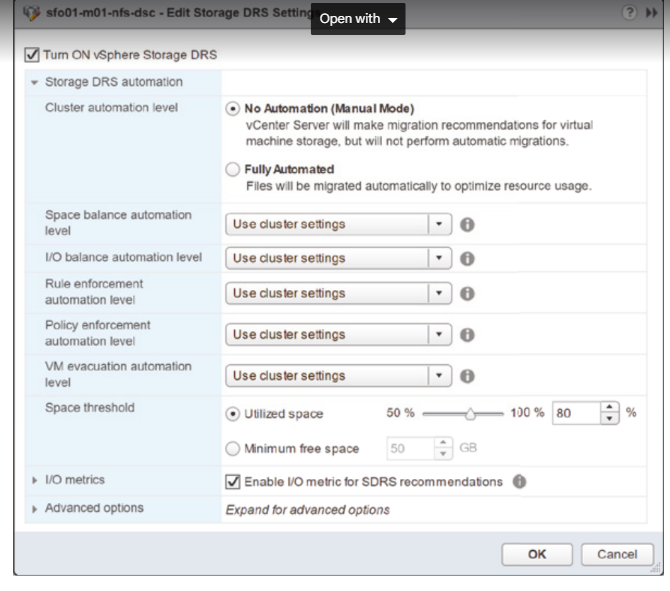
**Configuring Storage DRS**

From the settings pane, you can accomplish the following tasks:

◆ Enable or disable SDRS - From here, you can enable SDRS by selecting Turn ON vSphere Storage DRS. If SDRS is already enabled, you can deselect Turn ON vSphere Storage DRS to disable it.

◆ Configure the SDRS automation level - The SDRS automation levels are No Automation (Manual Mode) and Fully Automated - an “all or nothing” approach to SDRS

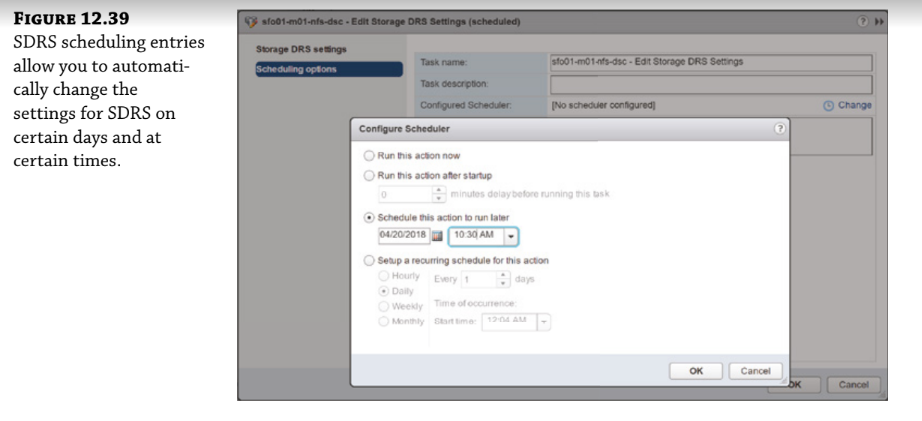
Automation.



◆ Change or modify the SDRS runtime rules - First, if you would like to configure SDRS runtime settings to operate only on the basis of space utilization and not I/O utilization, simply deselect Enable I/O Metric For SDRS Recommendations. This will set SDRS to recommend or perform (depending on the automation level) migrations based strictly on space utilization.

Second, the two Storage DRS Thresholds settings allow you to adjust the thresholds SDRS uses to recommend or perform migrations. By default, the Utilized Space setting is 80%, meaning that SDRS will recommend or perform a migration when a datastore reaches 80% full. The default I/O Latency setting is 15 ms—when latency measurements exceed 15 ms for a given datastore in a datastore cluster and I/O metrics are enabled, then SDRS will recommend or perform a storage migration to another datastore with a lower latency measurement.

◆ Configure or modify custom SDRS schedules - Imagine you normally have SDRS running in Fully Automated mode; however, at night, when backups are running, you do not want SDRS to automatically perform storage migrations. Using a custom SDRS schedule, you can configure SDRS to switch to manual mode during specific times of the day and days of the week. SDRS will then return toFully Automated mode when that time period has passed.



◆ Create SDRS rules to influence SDRS behavior - Just as DRS has affinity and anti-affinity rules, SDRS allows you to create VMDK affinity and anti-affinity rules and VM anti-affinity rules. These rules modify the behavior of SDRS to ensure that specific VMDKs are always kept together (VMDK affinity rule) or separate (VMDK anti-affinity rule) or that all the virtual disks from certain VMs are kept separate (VM anti-affinity rule).

◆ Configure per-VM SDRS settings.

**Chapter 13: Monitoring VMware vSphere performance**

Monitoring performance is a key component of datacenter management. vCenter Server offers a powerful, in-depth tool on the Performance tab that lets you create charts that depict the resource consumption over time for a given ESXi host or VM. The charts provide historical information and can be used for trend analysis. vCenter Server has many objects and counters that allow you to analyze the performance of a single VM or host for a selected interval.

**Using Alarms**

Depending on the object, these alarms can monitor resource consumption or the state of the object and alert you when certain conditions have been met, such as high resource usage or even low resource usage. These alarms can then provide an action that informs you of the condition by email or Simple Network Management Protocol (SNMP) trap. An action can also automatically run a script or offer other means to correct the problem the VM or host is experiencing.

These default alarms are usually generic in nature. Some of the **predefined alarms alert** you if any of the following situations occur:

◆ A host’s storage status, CPU status, voltage, temperature, or power status changes. ◆ A cluster experiences a vSphere High Availability (HA) error.

◆ A datastore runs low on free disk space.

◆ A VM’s CPU usage, memory usage, disk latency, or even fault tolerance status changes.

**Scope of Alarm**

When you define an alarm on an object, that **alarm applies to all objects beneath that object** in the vCenter Server hierarchy. The default set of alarms is defined at the vCenter Server object and therefore applies to all objects—data centers, hosts, clusters, datastores, networks, and VMs— managed by that instance of vCenter Server. If you were to create an alarm on a resource pool, the alarm would apply only to VMs found in that resource pool. Similarly, if you were to create an alarm on a specific VM, that alarm would apply only to that specific VM. Alarms are also associated with specific types of objects. For example, some alarms apply only to VMs, whereas other alarms apply only to ESXi hosts.

**Creating Alarms**

You can create many different types of alarms —such as how much CPU time a VM is consuming or how much RAM an ESXi host has allocated —or these alarms could monitor for specific events, such as when a specific distributed virtual port group is modified.

First you have to set up a Mail server or SNMP trap to receive alarm notifications.

To create a alarm,

Log in to web client → go to navigation tab → select any vSphere object → go to configure tab → scroll down to alarm definition → Click on add → Give name to alarm, select target type(image 1) → click on next to set up the alarm condition(image 2)

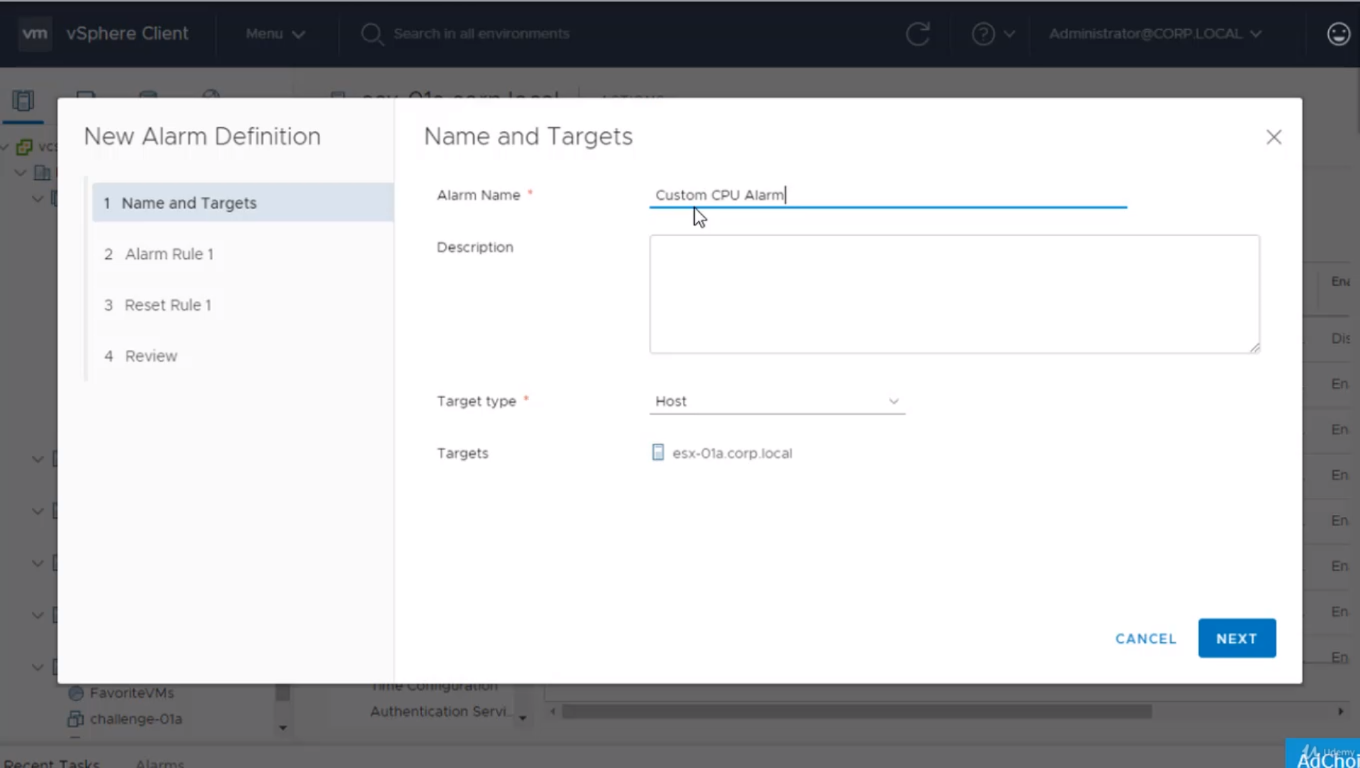


Image 1

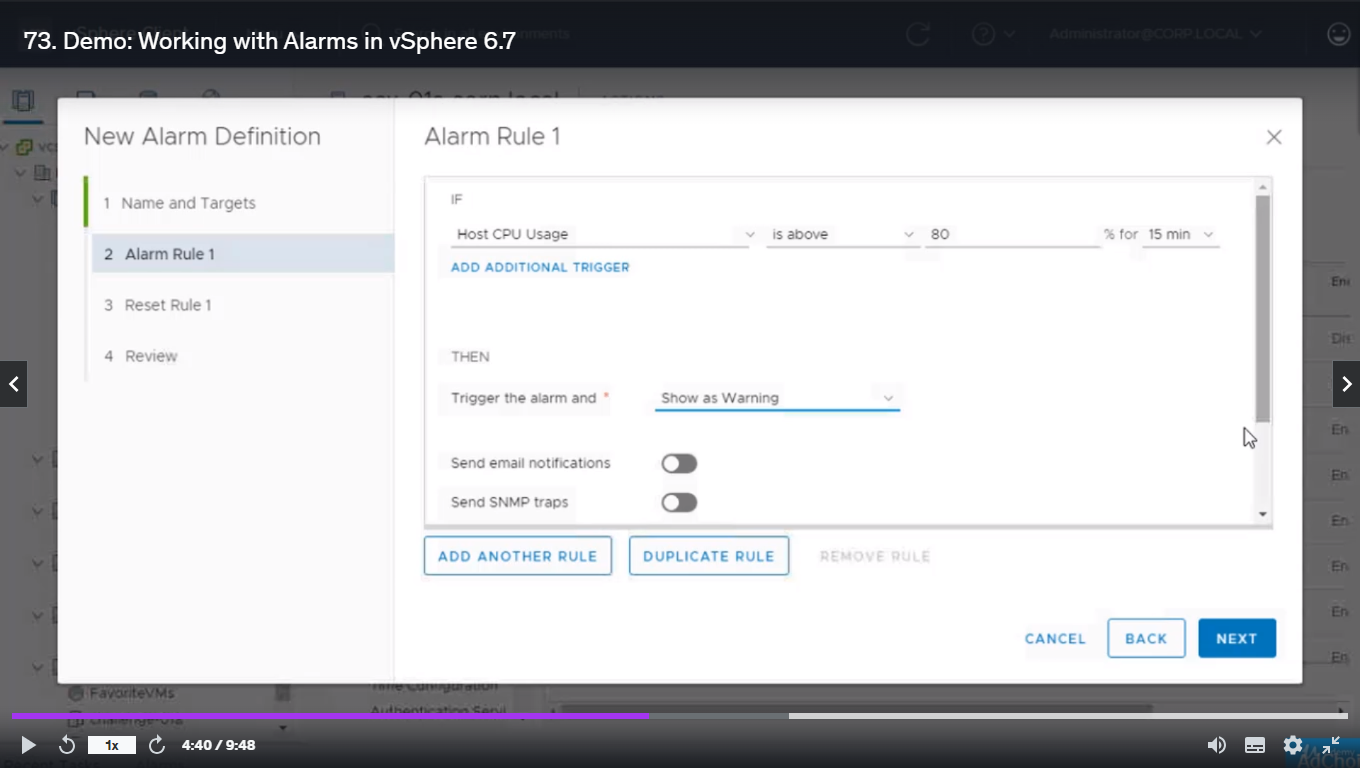
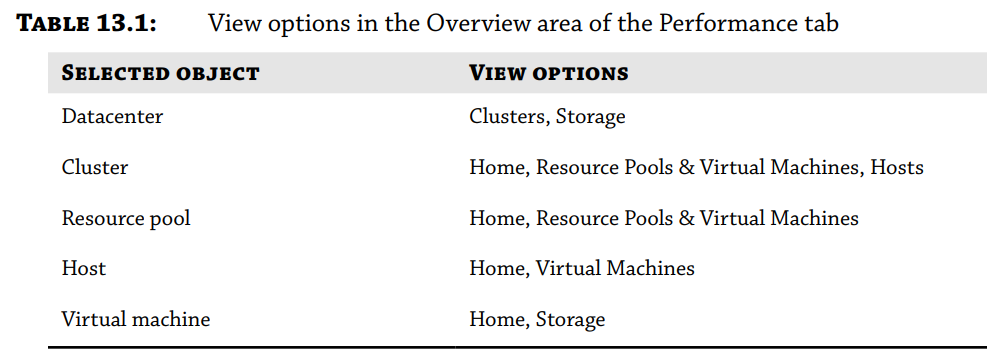


Image 2

**Working with performance charts**

Alarms are a great tool for alerting you of specific conditions or events, but they don’t provide the **detailed information** that you sometimes need, such as a resource being used that is still under a warning or critical state. This is where vCenter Server’s performance charts come in. By clicking the Monitor ➢ Performance tab for a datacenter, cluster, host, or VM, you can learn a wealth of information.

**The Overview layout is the default view when you access the Monitor Performance tab.** At the top of the Overview layout are options to change the view and the time range. The contents of the View drop-down list change depending on the object you select in the vSphere Web Client. Table 13.1 lists the options available for each object.



Advanced layout provides you a single chart option, but it provides extensive controls for viewing performance data.

Once you open the advanced layout on the left corner you can view chart metrics and counters.

All the available chart metrics are listed here:

◆ CPU ◆ Cluster Services ◆ Datastore ◆ Disk ◆ Memory ◆ Network ◆ Power ◆ Storage Adapter ◆ Storage Path ◆ System ◆ Virtual Flash ◆ Virtual Disk ◆ Virtual Machine Operations ◆ vSphere Replication

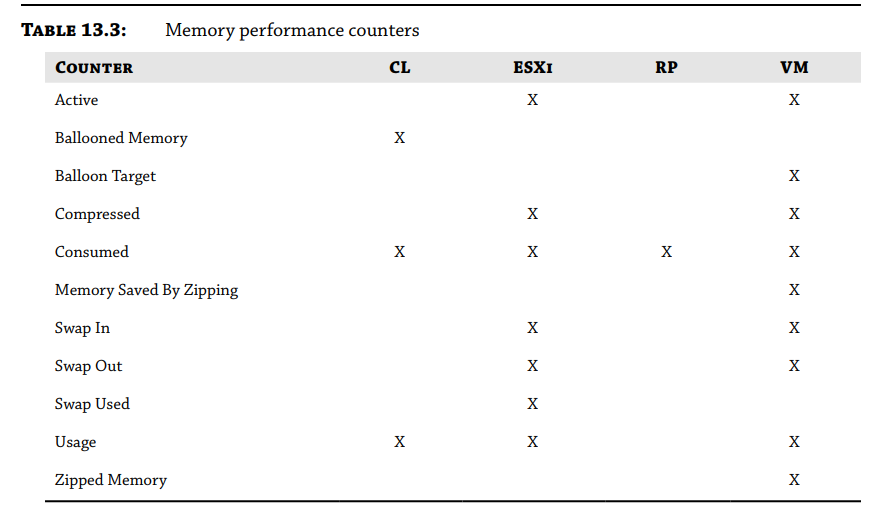
**Viewing CPU Performance Information**

If you select the CPU resource type in the Chart Options dialog box, you can choose which objects and counters you’d like to see in the performance chart. Note that the CPU resource type is not available when viewing the Performance tab of a datacenter object (DC). It is available for clusters (CL), ESXi hosts (ESXi), resource pools (RP), and individual virtual machines (VM). (Write only 2 to 4 points from the below table in your answer)



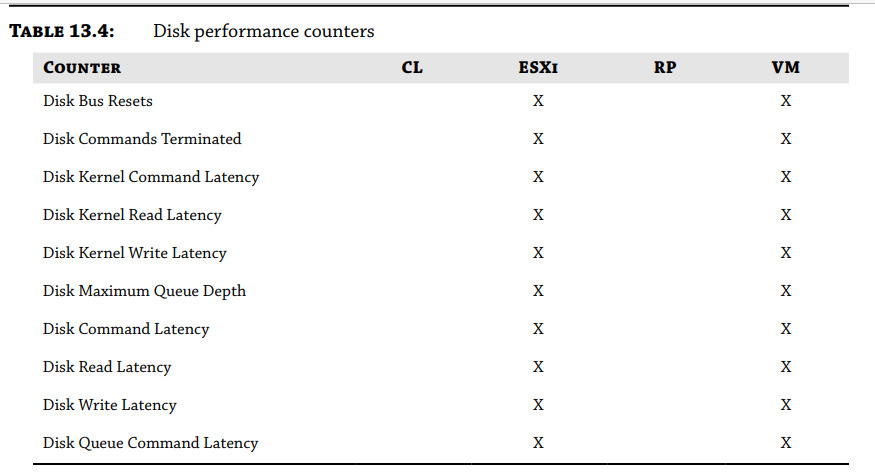
**Viewing Memory Performance Information**

If you select the Memory resource type in the Chart Options section of the Chart Options dialog box, you can display various objects and counters. The Memory resource type is not available when viewing the Performance tab of a datacenter object. It is available for clusters, ESXi hosts, resource pools, and individual VMs



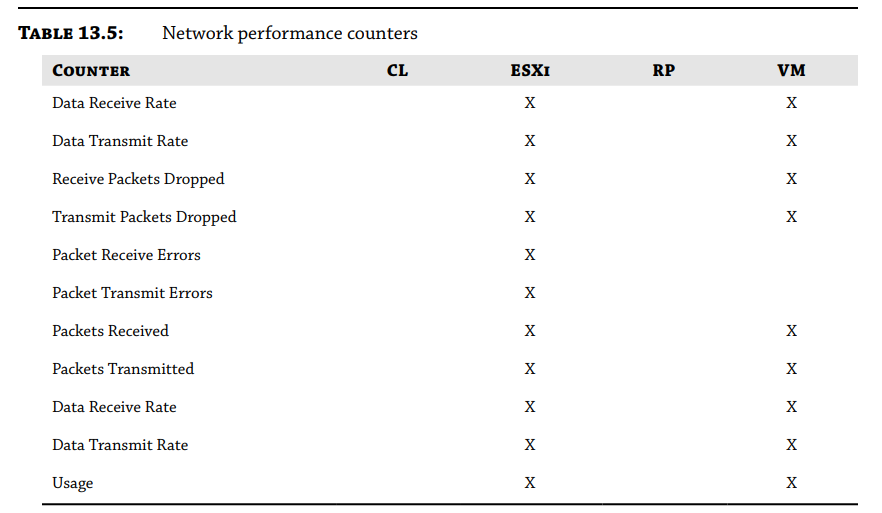
**Viewing Disk Performance Information**

Note that these counters aren’t supported for data centers, clusters, and resource pools, but they are supported for ESXi hosts and VMs. Not all counters are visible in all display intervals.



**Viewing Network Performance Information**

Network performance counters are available only for ESXi hosts and VMs; they are not available for datacenter objects, clusters, or resource pools.

****

**Viewing System Performance Information**

ESXi hosts and VMs also offer some performance counters in the System resource type. Data Centers, clusters, and resource pools do not support any system performance counters.

Metrics like resource memory shared, memory swapped, CPU usage can be viewed for ESXi host whereas Usage can be viewed for VM as well.

**DIY: (from reference book)** Viewing Datastore Performance Information, Viewing Storage Path Performance Information

**Chapter 14: Automating VMware vSphere**

Automation can help you complete the tasks - creating multiple virtual machines (VMs) from a template, changing the network configuration on a group of VMs or hosts, migrating VMs to new datastores with Storage vMotion, and gathering information about an environment for internal and external audits - much more quickly, provide greater consistency, and ultimately save your organization money by reducing the risk of errors and unplanned outages.

**Why Use Automation?**

The simple, but obvious, answer is that you are only one person. An individual can perform only a finite amount of work manually in any given hour, day, or week. With automation, modern datacenter administrators can increase their efficiency, accuracy, and capacity.

**Efficiency:** Automation enables you to complete repeated tasks with less effort. Scripts and workflows do not get distracted by a colleague or a meeting, causing it to miss a step or not complete its work in a timely manner.

**Accuracy:** Automation delivers consistency across repetitive and mundane tasks. Configuration changes, reports, and process workflows can be automated with high confidence that errors will not occur.

**Capacity:** Automation can often increase productivity. Tasks that would often take hours

manually can now be completed in minutes or seconds with automation. These benefits apply to any modern data center environment.

**vSphere Automation Options**

The most widely adopted automation tool to manage and automate vSphere is **PowerCLI.** Built on PowerShell, PowerCLI is a multi-platform, command-line and scripting tool that appeals to Windows and Linux users alike.

The **vSphere Automation SDKs** include the libraries for programmatically accessing features available via the API, such as virtual machine (VM) and vCenter Server appliance management. In addition, the SDKs contain samples for interoperating with vSphere APIs.

**vRealize Orchestrator** is an orchestration engine that is included with vSphere licensing entitlements. It provides the ability to use the out-of-the-box workflows or design and deploy custom, scalable workflows to automate vSphere processes.

**Automating with PowerCLI**

PowerCLI provides the ability to easily manage nearly all components in a vSphere environment.

Individuals who are comfortable with console administration will find PowerCLI easy to pick up and use.

**PowerShell and PowerCLI**

Based on the .NET Framework, PowerShell is a command-line and scripting language developed by Microsoft. The following are the various **components of PowerShell:**

1. CMDLETS - compiled .NET classes that perform a single action on an object. PowerShell cmdlets follow a <verb>-<singular noun> syntax followed by a set of parameters. For example, to get today’s date, you would execute **Get-Date** in PowerShell, or to get all VMs in a vCenter Server instance, you would simply execute **Get-VM** in PowerCLI.
2. OBJECTS - PowerShell is built on the Microsoft .NET Framework, as such, an object is a container of properties and methods.

When you execute the PowerCLI cmdlet Get-VM while connected to a vCenter Server instance, the information that is returned is a collection of VM objects. Each VM can be treated as its own object that contains unique information for that VM.

If you have a large inventory and want things to run faster while you learn, you can use the Select-Object cmdlet to select only the first few returned objects. In the following example, the command will return only the first three VM objects in your inventory:

*Get-VM | Select -First 3*

1. VARIABLES - In PowerShell, variables begin with a $ followed by alphanumeric characters. There are both global and user-defined variables, and we will touch on examples of each. Think of a variable as an empty container where you can store objects within PowerShell.

Imagine that you want to store a list of all VM objects in a variable for later use:

*$vm = Get-VM | Select-Object Name, PowerState*

*$vm | Select -First 3*

1. PIPELINE - PowerShell took the traditional pipeline and kicked it up a notch. In the past, pipelines were a means to pass text from one command to another, simplifying the syntax of a given operation. In PowerShell, pipelines are a means to pass entire objects from one cmdlet to another. This ability increased the power of the pipeline while simplifying its capacity to accomplish almost any administrative action.

*Get-Cluster <cluster name> | Get-VMHost Where-Object{$\_.ConnectionState -eq "Maintenance"}*

In the next example, return only the hosts in the cluster that are running ESXi 6.7.0:

*Get-Cluster <cluster name> | Get-VMHost |ˋWhere-Object{$\_.Version -eq "6.7.0"}*

**H.W.** How to install and configure PowerCLI on windows, linux and macOS? Explain the steps.

**Getting Started with PowerCLI**

The first and most important thing to remember about PowerCLI is that the PowerShell’s

<verb>-<noun> nomenclature for cmdlets makes it easy to read and easy to find the right cmdlets for the job. In the following sections, we will discuss the basic starting points of PowerCLI.

**Finding Cmdlets**

Using the following command in PowerCLI will return a list of all available log cmdlets:

*Get-VICommand \*log\**

Alternatively, you could use the Get-Command cmdlet and narrow your search criteria to only the <noun> portion of the cmdlet name:

*Get-Command -Noun \*log\**

You will notice that the Get-Command cmdlet returns more results than Get-VICommand. Why is this? The Get-Command cmdlet will return all PowerShell cmdlets that contain log in the <noun> position of cmdlet name, whereas Get-VICommand will return only PowerCLI cmdlets that contain log in the <noun> position of the cmdlet name.

**Getting Help**

The Get-Help cmdlet can provide helpful information about cmdlets or topics within PowerShell.

Once you identify the cmdlet you need, you can use Get-Help to find out more about it, like the parameters or parameter sets, a description of how it is used, and examples of how to use it. For example, try the following:

*Get-Help Get-command*

The command will return a brief synopsis of Get-Log cmdlet capabilities, the syntax for usage, a more detailed description, and links associated with the cmdlet.

*Get-Help Get-Log -examples*

*Get-Help Get-Log -online*

**Connecting to vCenter Server and ESXi Hosts**

The first cmdlet that any PowerCLI user must know is Connect-VIServer. This cmdlet has many features that make connecting to your vCenter Server or ESXi hosts easy.

Connecting to a vCenter Server is the most common use of Connect-VIServer. To connect, you must provide, at a minimum, the vCenter Server name, a username, and a password:

*Connect-VIServer -Server <vCenter Server FQDN or IP> -User <username> -password <password>*

Many organizations follow the practice of separate administrative accounts and, as such, it is important to demonstrate ways to protect credentials. When you call Get-Credential in a script, it will prompt for a username and password. It stores them in the PSCredential object, which can then be used by Connect-VIServer as follows:

*$credential = Get-Credential*

*Connect-VIServer -Server <vCenter Server FQDN or IP> -Credential $credential*

In the following example, we’re connecting to two vCenter Server instances:

*Connect-VIServer -Server sfo01m01vc01.rainpole.local,sfo01m01vc02.rainpole.local -Credential $credential*

In the next example, we’re connecting directly to two ESXi hosts:

*Connect-VIServer -Server* *sfo01m01esx01.rainpole.local,sfo01m01esx02.rainpole.local -Credential $credential*

In the following example, we specify the vCenter Server instances that will be disconnected in the session:

*Disconnect-VIServer -Server sfo01m01vc01.rainpole.local,sfo01m01vc02.rainpole.local -Confirm:$false -d* [*www.google.com*](http://www.google.com) *-p 8080*

Use the -Confirm parameter to prevent being prompted to disconnect.

And in the next example, we use a wildcard (\*) to disconnect all vCenter Server instances in the session:

*Disconnect-VIServer \* -Confirm:$false*

**Your First One-Liner: Reporting**

One of the most common uses of PowerCLI is reporting. You can gather tremendous amounts of data about an environment in a short period of time. One-liners are scripts that can be written out and executed in a single line using the pipeline to pass information.

Using PowerCLI, you can return the information in moments:

Get-VMHost | Get-VMHostSyslogServer | Export-CSV C:\Mastering vSphere\Reports\Syslog.csv -NoTypeInformation

This one-liner gathers a collection of all ESXi hosts in inventory and then collects the syslog server settings for each. The final pipeline takes that syslog server information and exports it to a CSV file.

**Your First One-Liner: Configuration Change**

You can also modify the environment, assuming you have the appropriate privileges. You have identified systems that did not have the correct syslog settings. To update these hosts, you need to identify the supporting cmdlet. In situations like this, where a Get -verb is used in a cmdlet, it is common that a Set -verb is also available. In this instance, you will want to use the Set-VMHostSyslogServer cmdlet. Running *Get-Help Set-VMHostSyslogServer -full* returns the syntax and tells you that you will need to specify the -SysLogServer parameter, and perhaps the -SysLogServerPort parameter if the syslog collector endpoint is listening on a specific port. Assume in this scenario that you are sending your logs to a vRealize Log Insight cluster with hostname sfo01vrli01.rainpole.local. Go ahead and specify port 514:

*Get-VMHost | Set-VMHostSyslogServer -SysLogServer sfo01vrli01.rainpole.local -SysLogServerPort 514*

**H.W.** Write a PowerCLI command to disconnect VM and hosts.