



---

# Veeam Unofficial VMware VCP-DCV 2020 Study Guide

---

**Shane Williford**

VMware vExpert  
2011-2020, VCP-DCV,  
VCAP-DCA, VCP-Cloud,  
VCP-DT certified

---

**Paul Wilk**

VMware vExpert  
2018-2020, VCP-DCV,  
VCP-NV, VCP-CMA certified



# Contents

## Exam Sections . . . . . 2

### SECTION 1: VMware vSphere architectures & technologies . . . . . 3

- Objective 1.1: Identify prerequisites & components for vSphere implementation . . . . . 4
- Objective 1.2: Identify vCenter Server High Availability (HA) requirements. . . . . 7
- Objective 1.3: Identify storage types for vSphere . . . . . 8
- Objective 1.4: Differentiate between NIOC and SIOC . . . . . 18
- Objective 1.5: Manage vCenter inventory efficiently . . . . . 19
- Objective 1.6: Describe and differentiate among vSphere HA, DRS & SDRS functionality . . . . . 23
- Objective 1.7: Describe and identify resource pools and use cases . . . . . 26
- Objective 1.8: Differentiate vDS and vSS . . . . . 28
- Objective 1.9: Describe the purpose of cluster and the features it provides . . . . . 32
- Objective 1.10: Describe Virtual Machine file structure. . . . . 38
- Objective 1.11: Describe vMotion and Storage vMotion technology . . . . . 39

### SECTION 2: VMware product and solutions. . . . . 42

- Objective 2.1: Describe vSphere integration with other VMware products . . . . . 43
- Objective 2.2: Describe HA solutions for vSphere . . . . . 48
- Objective 2.3: Describe the options for securing a vSphere environment. . . . . 53

### SECTION 4: Installing, configuring and setting up a VMware vSphere solution . . . . . 58

- Objective 4.1: Understand basic log output from vSphere products. . . . . 59
- Objective 4.2: Create and configure vSphere objects. . . . . 61
- Objective 4.3: Set up a content library . . . . . 66
- Objective 4.4: Set up ESXi hosts . . . . . 69
- Objective 4.5: Configure virtual networking. . . . . 73
- Objective 4.6: Deploy and configure VMware vCenter Server Appliance (VCSA) . . . . . 76

- Objective 4.7: Set up identity sources. . . . . 78
- Objective 4.8: Configure an SSO domain. . . . . 80

### SECTION 5: Performance-tuning and optimizing a VMware vSphere solution . . . . . 82

- Objective 5.1: Determine effective snapshot use cases . . . . . 83
- Objective 5.2: Monitor resources of VCSA in a vSphere environment. . . . . 84
- Objective 5.3: Identify impacts of VM configurations . . . . . 85

### SECTION 7: Administrative and operational tasks in a VMware vSphere solution . . . . . 88

- Objective 7.1: Manage virtual networking . . . . . 89
- Objective 7.2: Manage datastores. . . . . 93
- Objective 7.3: Configure a storage policy . . . . . 97
- Objective 7.4: Configure host security . . . . . 99
- Objective 7.5: Configure role-based user management . . . . . 101
- Objective 7.6: Configure and use vSphere compute and storage cluster options . . . . . 104
- Objective 7.7: Perform different types of migrations. . . . . 108
- Objective 7.8: Manage resources of a vSphere environment . . . . . 110
- Objective 7.9: Create and manage VMs using different methods . . . . . 111
- Objective 7.10: Create and manage templates. . . . . 113
- Objective 7.11: Manage different VMware vCenter Server objects . . . . . 114
- Objective 7.12: Set up permissions on datastores, clusters, vCenter and hosts . . . . . 116
- Objective 7.13: Identify and interpret affinity/anti-affinity rules . . . . . 119
- Objective 7.14: Understand use cases for alarms. . . . . 123
- Objective 7.15: Utilize VMware vSphere Update Manager (VUM) . . . . . 124
- Objective 7.16: Configure and manage host profiles . . . . . 127

### About the Authors . . . . . 131



## Navigation

You can use a page header for navigation. Click on "Contents" or "Section" returns you to the relevant page.

## Exam Sections

The Professional vSphere 6.7 Exam validates that an individual can implement, manage, and troubleshoot a vSphere v6.7 infrastructure, using best practices to provide a powerful, flexible, and secure foundation for business agility that can accelerate the transformation to cloud computing.



### SECTION 1: VMware vSphere architectures & technologies



### SECTION 2: VMware product and solutions

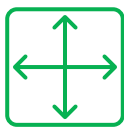


### SECTION 3: Planning and Designing

There are no testable objectives for this section.



### SECTION 4: Installing, configuring and setting up a VMware vSphere solution

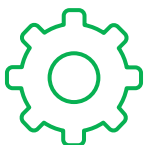


### SECTION 5: Performance-tuning and optimizing a VMware vSphere solution

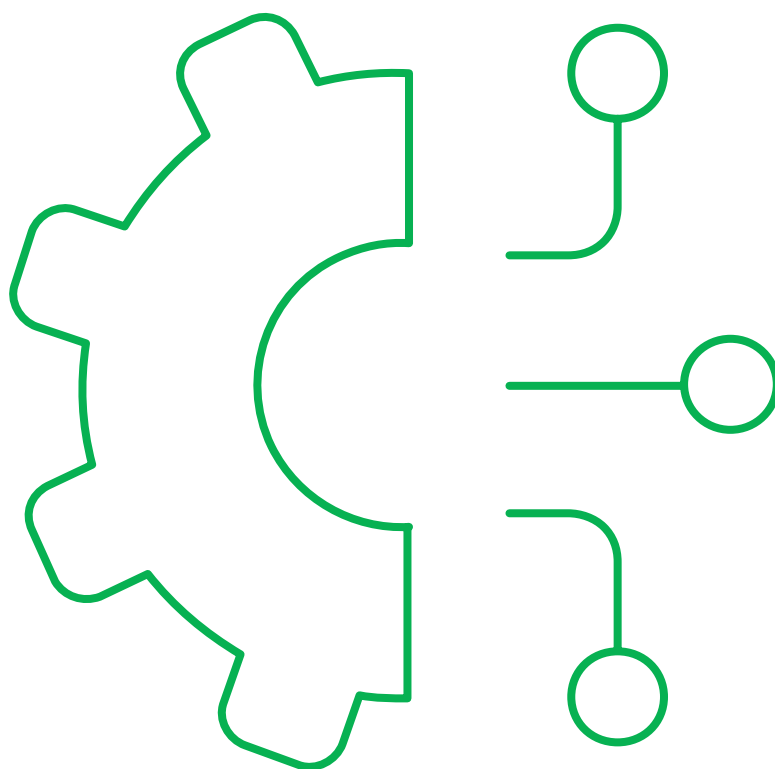


### SECTION 6: Troubleshooting and Repairing

There are no testable objectives for this section.



### SECTION 7: Administrative and operational tasks in a VMware vSphere solution



## SECTION 1: VMware vSphere architectures & technologies

Objective 1.1: Identify prerequisites & components for vSphere implementation . . . . .	4	Objective 1.7: Describe and identify resource pools and use cases . . . . .	26
Objective 1.2: Identify vCenter Server High Availability (HA) requirements. . . . .	7	Objective 1.8: Differentiate vDS and vSS . . . . .	28
Objective 1.3: Identify storage types for vSphere . . . . .	8	Objective 1.9: Describe the purpose of cluster and the features it provides . . . . .	32
Objective 1.4: Differentiate between NIOC and SIOC . . . . .	18	Objective 1.10: Describe Virtual Machine file structure. . . . .	38
Objective 1.5: Manage vCenter inventory efficiently . . . . .	19	Objective 1.11: Describe vMotion and Storage vMotion technology . . . . .	39
Objective 1.6: Describe and differentiate among vSphere HA, DRS & SDRS functionality . . . . .	23		



## Objective 1.1: Identify prerequisites & components for vSphere implementation

The two core components of vSphere are ESXi and vCenter Server. But the Platform Services Controller (PSC) also plays a major role in proper vSphere function. In this objective I will cover all the prerequisites and components making up the three vSphere components, as well as provide several common component maximums you should be familiar with. Installation steps and more detailed component feature explanations will be discussed in other objectives throughout this guide.

### vSphere (ESXi) prerequisites and components

ESXi requirements include:

- Verify hardware support on VMware’s Hardware Compatibility List (HCL)
- Enable the following BIOS settings:
  - Intel-VD or AMD-RVI hardware virtualization to run 64-bit VMs
  - The NX/XD bit; optionally enable hyperthreading
- Minimum two CPU cores/four GB RAM
- 64-bit processors released after September 2006
- Supported gigabit ethernet controller
- ESXi boot storage requirements:
  - Boot storage of at least one GB
  - If booting from local, SAN or iSCSI LUNs, minimum of 5.2 GB
- ESXi boot requirements:
  - UEFI supports boot from hard drives, CD drives and USB
  - Changing from legacy to UEFI after ESXi install is not supported, as this will cause an error: Not a VMware boot bank
- Firewall ports – review on pages 14-16
- ESXi host client is supported by the following browser versions and higher: Chrome 50, Firefox 45, MS Internet Explorer 11/Edge 38, Safari 9

The following are several vSphere maximums per ESXi host. To see a full list, refer to the following website: <https://configmax.vmware.com>

Host CPU	768	VMFS/FC/iSCSI volume, LUN size	64 TB	Network switch ports (vSS and vDS)	4,096
Host memory	16 TB	VMFS volumes	1,024	VMDirectPaths	8
VM vCPU	4,096	FC or iSCSI LUNs	1,024	Hosts per cluster	64
Virtual machines	1,024	Physical NICs	32 – 1 GbE; 16 – 10 GbE	Resource pools	1,600 (tree depth = 8)
Fault tolerance VMs	4	Number of vDS	16		



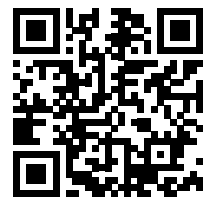
### Resource:



[ESXi 6.7 Installation and Setup Guide, Update 2 \(Apr 2019\)](#), pages 10-17



[vCenter Server 6.7 Installation and Setup Guide, Update 2 \(March 2020\)](#), pages 27-37 (VCSA); 86-89 (Windows)





## vCenter Server prerequisites and components

### VCSA appliance

- VCSA composed of: Photon OS 1.0, vCenter services, PSC services (if embedded), PostgreSQL database, VMware Update Manager, and hardware version 10
- vCenter Server contains the following services:
  - vCenter Server
  - vSphere Client
  - vSphere Web Client
  - vSphere ESXi Dump Collector
  - vSphere Update Manager Extension
  - VMware Syslog Collector (Windows only)
- Minimum install on vCenter 5.5 or ESXi 5.5
- Hardware requirements are the same, whether it's an Embedded or External PSC

	Number of vCPUs	Memory
Tiny environment (up to 10 hosts or 100 virtual machines)	2	10 GB
Small environment (up to 100 hosts or 1,000 virtual machines)	4	16 GB
Medium environment (up to 400 hosts or 4,000 virtual machine)	8	24 GB
Large environment (up to 1,000 hosts or 10,000 virtual machines)	16	32 GB
X-Large environment (up to 2,000 hosts or 35,000 virtual machines)	24	48 GB

- Storage requirements are the same, whether it's an Embedded or External PSC

	Default Storage Size	Large Storage Size	X-Large Storage Size
Tiny environment (up to 10 hosts or 100 virtual machines)	250 GB	775 GB	1650 GB
Small environment (up to 100 hosts or 1,000 virtual machines)	290 GB	820 GB	1700 GB
Medium environment (up to 400 hosts or 4,000 virtual machine)	425 GB	925 GB	1805 GB
Large environment (up to 1,000 hosts or 10,000 virtual machines)	640 GB	990 GB	1870 GB
X-Large environment (up to 2,000 hosts or 35,000 virtual machines)	980 GB	1030 GB	1910 GB



- Firewall ports – review on pages 28-33
- Client machine used to install the VCSA
  - Windows 7 desktop or higher; Windows 2016 Server or higher, with at least two CPUs of four cores, 2.3 GHz, four GB RAM, 32 GB storage
  - Visual C++ if using Windows 7
  - Linux OS support: SUSE 12 or Ubuntu 14.04 with at least one CPU with two cores of 2.4 GHz, eight GB RAM, and 150 GB storage

#### **vCenter Server on Windows**

- Minimum 64-bit Windows 2008 SP2, physical or virtual
- Non-domain controller
- Database 64-bit DSN
- PostgreSQL for up to 10 ESXi hosts/200 VMs, or external SQL or Oracle for larger environments
- Verify time is synchronized between PSC and vCenter Windows machines
- If not using LOCAL SYSTEM to run vCenter Server service, the account used for the service needs the following permissions:
  - Member of the Administrators group
  - Log on as a service right
  - Act as part of the operating system (if user is domain user) right
- If using Active Directory as a vCenter identity source, the machine must be a domain member computer, otherwise it's not required
- Storage: `Program Files` – 6 GB, `ProgramData` – 8 GB, `System folder` – 3 GB

---

Hosts	2,000
Powered-on and registered VMs	25,000/35,000
Linked vCenters	15
Concurrent web client sessions	180
vMotion operations per host	4 – 1 GbE / 8 – 10 GbE
Storage vMotions per host	2
vMotion/Storage vMotion per datastore	128/8
Datastore clusters	256
Datastores per datastore cluster	64
Number of vDS	128
Network switch ports (vSS and vDS)	4,096

---



## Platform Services Controller (PSC) prerequisites and components

### PSC appliance

- Embedded – Requirements are the same as VCSA with Embedded PSC above
- External – Requirement is two vCPU/four GB RAM with 60 GB of storage
- PSC contains the following services:
  - vCenter Single Sign-On (SSO)
  - vSphere License Service
  - VMware Certificate Authority (VMCA)
- Lookup Services

### PSC on Windows

- Embedded – Requirements are the same as VCSA with Embedded PSC above
- External – Requirements are the same as VCSA with Embedded PSC above
- Storage: `Program Files` – 1 GB, `ProgramData` – 2 GB, `System folder` – 1 GB

## Objective 1.2: Identify vCenter Server High Availability (HA) requirements

### vCenter HA requirements

vCenter HA must meet the following requirements:

- vCenter Server Standard license
- VCSA minimum four vCPU/16 GB RAM, requiring a small vCenter deployment size (see Section 1.1 above) or larger
- Protected vCenter Server Appliance (VCSA) needs to be version 6.5 and higher
- Datastore support for VMFS, vSAN, and NFS
- ESXi 5.5 or higher and strongly recommended to use three hosts. If vSphere DRS is used, three hosts are required
- If a management vCenter Server is used, version 5.5 or higher
- Network requires a separate subnet for vCenter HA with less than 10 ms latency



Resource:



[vSphere 6.7 Availability Guide, Update 2 \(April 2019\)](#), page 71





## Objective 1.3: Identify storage types for vSphere



### Resource:

### Local storage

ESXi host local storage refers to internal hard disks or external storage systems connected to the host via SAS or SATA protocols. Supported local devices include:

- SCSI
- SAS
- USB
- IDE
- SATA
- Flash and NVMe

USB and IDE cannot be used to store VMs.

Since this storage type is local to the ESXi host, you cannot share storage with the other hosts. As such, the use of vSphere features such as High Availability (HA) and vMotion are not available.

### Networked storage

These external storage systems are accessed by ESXi hosts over a high-speed storage network and are shared among all hosts configured to access the system. Storage devices are represented as Logical Units (LUNs) and have a distinct UUID name, World Wide Port Name (WWPN) for FC and an iSCSI Qualified Name (IQN) for iSCSI.

- Shared LUNs must be presented to all ESXi hosts with the same UUID
- You cannot access LUNs with different transport protocols (i.e. FC and iSCSI)
- LUNs can contain only one VMware File System (VMFS) datastore

VMware supports the following types of networked storage:

#### Fibre Channel (FC)

Uses a Storage Area Network (SAN) to transfer data from ESXi hosts to the storage system using the FC protocol. This protocol packages SCSI commands into FC frames. To connect to the FC SAN, the host uses a FC Host Bus Adapter (HBA).

FC supports three Storage Array types:

- Active-active – Supports access to LUNs simultaneously through all available storage ports. All paths are active unless a path fails.
- Active-passive – One storage processor is active, providing access to a given LUN, while other storage processors act as a backup (standby) for the LUN, yet can provide active access to other LUNs. If an active port fails, the backup/standby path can be activated by servers accessing it.
- Asymmetric – Asymmetric Logical Unit Access (ALUA) provides different levels of access per port.

Access to a FC LUN can be configured in one of two ways:

- LUN Zoning defines which hosts HBA devices can connect to which Storage Processors (SPs) or targets. Using single-initiator-single-target is the preferred zoning method.
- LUN Masking is a permission-based process making a LUN available to some ESXi hosts and unavailable to other hosts.



[vSphere 6.7 Storage Guide, Update 2 \(June 2019\)](#)



[vSAN 6.7 Planning and Deployment Guide, Update 3 \(August 2019\)](#)

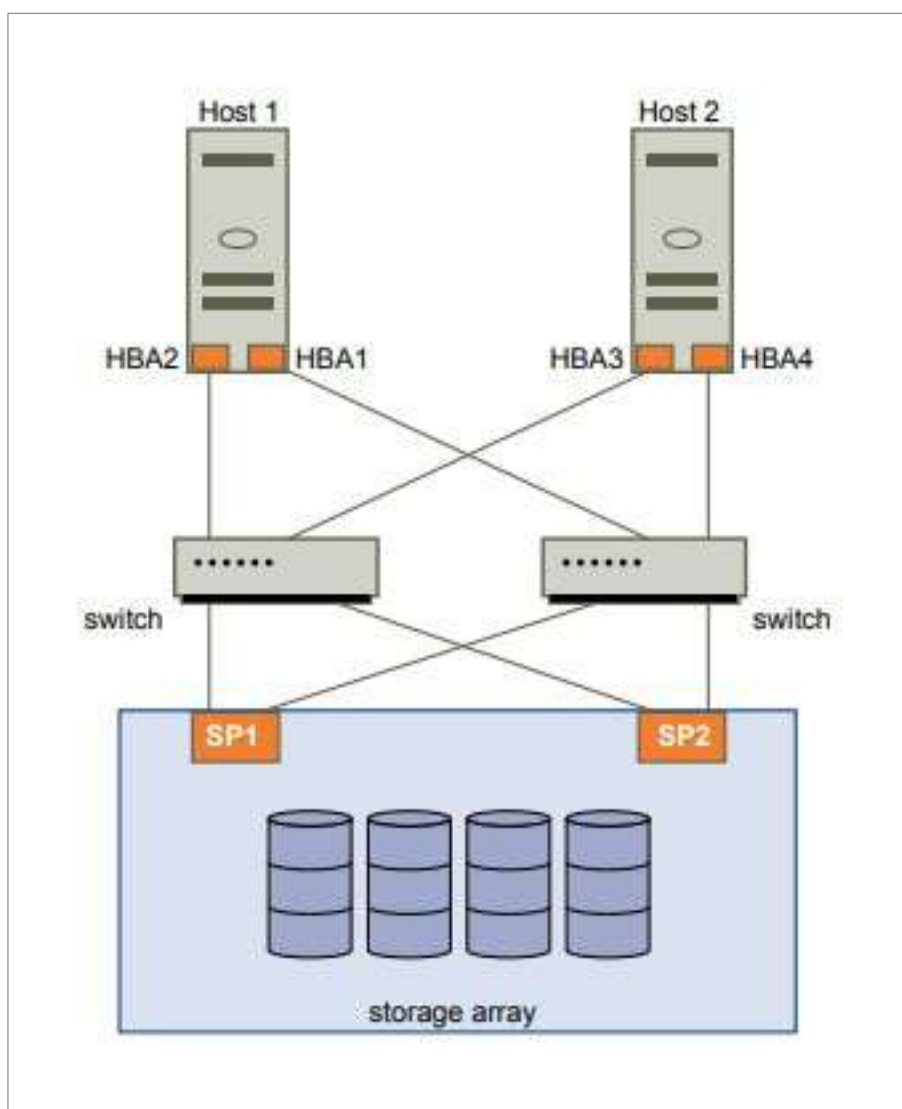
Increase the SCSI TimeoutValue in Windows VMs registry to 60 so the VM can tolerate delayed I/O from a path failover.

N-Port ID Virtualization can be used by VMs for RDM access. A WWPN and WWNN are paired in a VM .vmx file, and the VMkernel instantiates a virtual port (VPort) on the host HBA used for LUN access.

FC HBA considerations:

- Do not mix HBA vendors
- Ensure the same firmware level on each HBA
- Set the timeout value for detecting failover as discussed above
- ESXi supports 16 GB end-to-end FC connectivity

FC Multipathing is a method to ensure a remote storage system is highly available to an ESXi host. To achieve a highly available connection to the storage system, multiple FC HBAs must be used. View VMware's representation below:

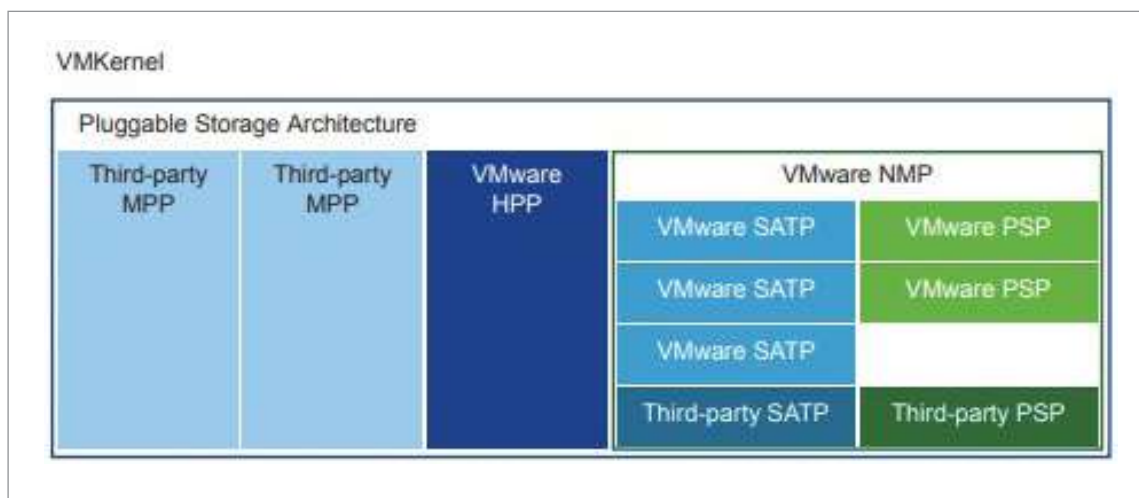




Multipathing is possible by way of the Pluggable Storage Architecture (PSA), a VMkernel layer within ESXi that's an open and modular framework coordinating various software modules responsible for multipathing. The following components make up the PSA:

- Native Multipathing Plugin (NMP) is VMware's VMkernel multipathing plugin ESXi uses by default, providing a default path selection algorithm based on storage array type. NMP manages two sub-plugins:
  - **Path Selection Plugin (PSP)** is responsible for selecting the physical path for I/O requests and uses one of three policies:
    1. `VMW_PSP_MRU` – The most recently used policy selects the first working path discovered at system boot. When it becomes unavailable, an alternate path is chosen and if it becomes available again I/O is not reverted to it. It does not use the preferred path setting but it can use path ranking. This is default for most active-passive arrays.
    2. `VMW_PSP_FIXED` – The fixed policy uses a designated preferred path, and if it doesn't get assigned, the first working path discovered at boot is selected. When it becomes unavailable, an alternate path is chosen and if it becomes available again, I/O is reverted to it. This policy is default for most active-active storage systems.
    3. `VMW_PSP_RR` – The round-robin policy uses an automatic path selection algorithm rotating through configured paths, selecting optimal paths based on I/O bandwidth and latency. Active-passive arrays use all active paths, while active-active arrays use all available paths.
  - **Storage Array Type Plugin (SATP)** is responsible for array-specific operations, determine the state of a path, monitors path health, performs path activation, and detects path errors. Rules are searched first by drivers, then vendor or model, or lastly by transport. One of the following generic SATP policies is used:
    4. `VMW_SATP_LOCAL` – is used for local devices
    5. `VMW_SATP_DEFAULT_AA` – is used for active-active arrays
    6. `VMW_SATP_DEFAULT_AP` – is used for active-passive arrays
    7. `VMW_SATP_DEFAULT_ALUA` - is used for ALUA-compliant arrays
    8. The default SATP if none found is "\_AA" and its PSP is `VMW_PSP_FIXED`.
    9. If SATP "\_ALUA" policy is used, the default PSP is `VMW_PSP_MRU`.
- Multipathing Plugins (MPP) are third-party plugins created by using various VMkernel APIs. These third-party plugins offer multipathing and failover functions for a particular storage array.
- VMware High Performance Plugin (HPP) is like the NMP but is used for high-speed storage devices, such as NVMe PCIe flash.
- Claim Rules determine whether an MPP or NMP owns a path to a storage device. NMP claim rules match the device with a SATP and PSP, whereas MPP rules are ordered with lower numbers having precedence over higher ones.

Below is VMware's visual representation of the PSA and plug-ins architecture:



### Fibre Channel over Ethernet (FCoE)

Hardware FCoE Adapters are Converged Network Adapters (CNAs) containing both FC and network capabilities.

- Appear as both a vmnic and FCoE adapter in the vSphere Client
- No configuration is necessary

Software FCoE Adapters use the native FCoE protocol stack in the VMkernel to perform FCoE processing with a one of two compatible network interface card (NIC) categories:

- With Partial FCoE, Offload NIC offers Data Center Bridging and I/O capabilities
- Without FCoE, Offload NIC offers Data Center Bridging and 10 Gbps speed
- Network switch configuration for FCoE
  - Turn on Priority-Based Flow Control (PFC) and set to AUTO
  - Set MTU size to 2,500 or more
  - Switch firmware version must be compatible with FCoE
- The below FCoE Adapter best practices should be followed:
  - Ensure latest microcode on the FCoE Adapter
  - For Adapter with multiple ports, add each port to a different vSwitch to avoid an all paths down (APD) condition during a disruptive event, such as an MTU change
  - If you move an adapter port with active traffic from one vSwitch to another, reboot ESXi host
  - If changing the vSwitch for an adapter port causes a failure, move the port back to the original vSwitch to resolve the issue



### Internet SCSI (iSCSI)

Connect to storage systems using ethernet connections between ESXi hosts and high-performance storage systems using iSCSI HBAs or NICs.

iSCSI names are generated by ESXi and are represented as IQNs or EUIs to identify an iSCSI node.

- IQN example – `iqn.1998-01.com.vmware.iscsi:hostname1`
- EUI example – `eui.0123456789ABCDEF`

iSCSI supports four storage types:

- Active-active – same as discussed in FC above
- Active-passive – same as discussed in FC above
- Asymmetric – same as discussed in FC above
- Virtual port – supports access to all LUNs through a single virtual port, and handles connection balancing and failovers transparent

Discovery sessions return targets to the ESXi hosts and occur in one of two methods:

- Dynamic Discovery or SendTargets obtains a list of accessible targets
- Static Discovery can only access a particular target by name and address

Authentication is used by a name and key pair. ESXi uses the Challenge Handshake Authentication Protocol (CHAP).

Access control is set up on the iSCSI storage system by Initiator Name, IP Address or CHAP.

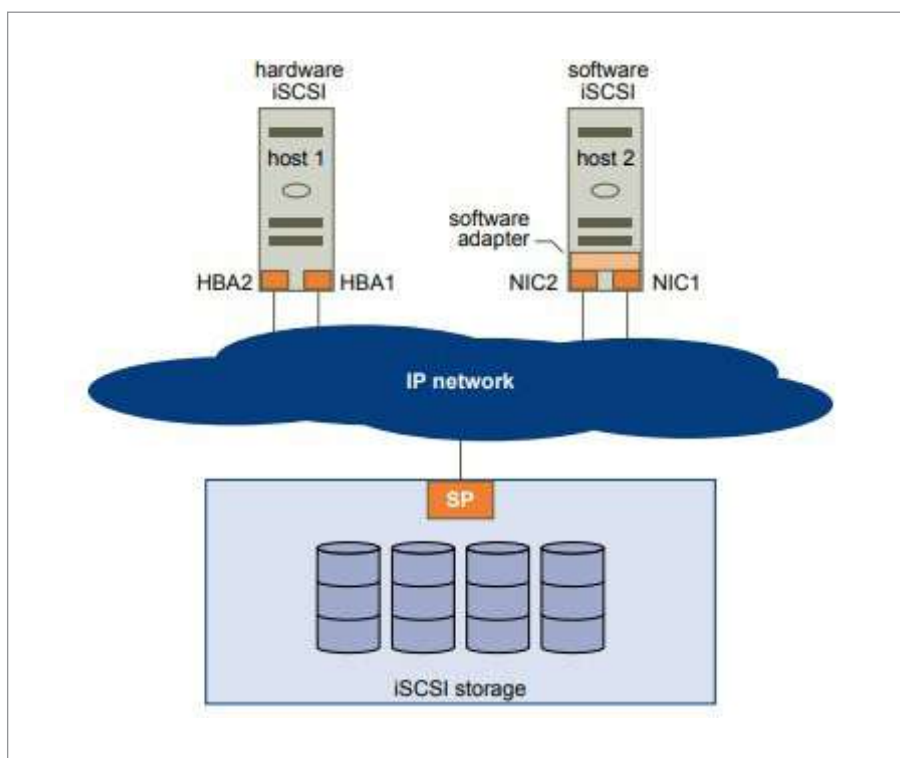
Jumbo Frames of MTU size greater than 1500 (generally 9000) is required. Jumbo frame configuration is needed on vSwitch, Software iSCSI Adapter, and VMkernel port.

There are two types of iSCSI initiators to connect to iSCSI targets:

- Software iSCSI uses built-in VMware code in the VMkernel to connect to iSCSI storage through standard NICs
- Hardware iSCSI uses third-party adapters to offload iSCSI processing from the ESXi host, using one of two categories:
  - **Dependent Hardware** relies on VMware networking, iSCSI configuration and management interfaces (ex. Broadcom 5709)
  - **Independent Hardware** uses its own networking and iSCSI configuration and management interfaces (ex. QLogic GLA4052)

iSCSI Port binding is used for multipathing to ensure a highly available network storage configuration. What this entails is assigning an ESXi host physical NIC to a separate VMkernel port, then binding each VMkernel port to the Software iSCSI Adapter. For hardware multipathing, two physical HBAs are required and are utilized similar to FC multipathing.

Hardware and software multipathing are visually represented by VMware as in the image below:



### Networked-Attached Storage (NAS)

Uses the TCP/IP network to access remote file servers through the NFS client in the VMkernel using standard network adapters.

ESXi uses one of two NFS protocols, version 3 and 4.1.

NFS 3 and NFS 4.1 have the following differences:

	NFS 3	NFS 4.1
Security	AUTH_SYS	AUTH_SYS and Kerberos (krb5)
Encryption	Not supported	AES 128 or AES 256
Multipathing	Not supported	Through session trunking
Locking mechanism	Client-side locking	Server-side locking
vSphere features (HA, DRS, FT, etc.)	All supported	Does not support: SIOC, Site Recovery Manager and SDRS

To upgrade NFS 3 datastores to NFS 4.1, you must storage vMotion VMs off the NFS 3 datastore and unmount it, then remount it back as NFS 4.1 and storage vMotion VMs back.

NFS 3 firewall rule behavior:

- **When adding or mounting an NFS 3 datastore**, if the `nfsClient` rule set is disabled, ESXi enables it and disables the Allow All IP Address policy by setting



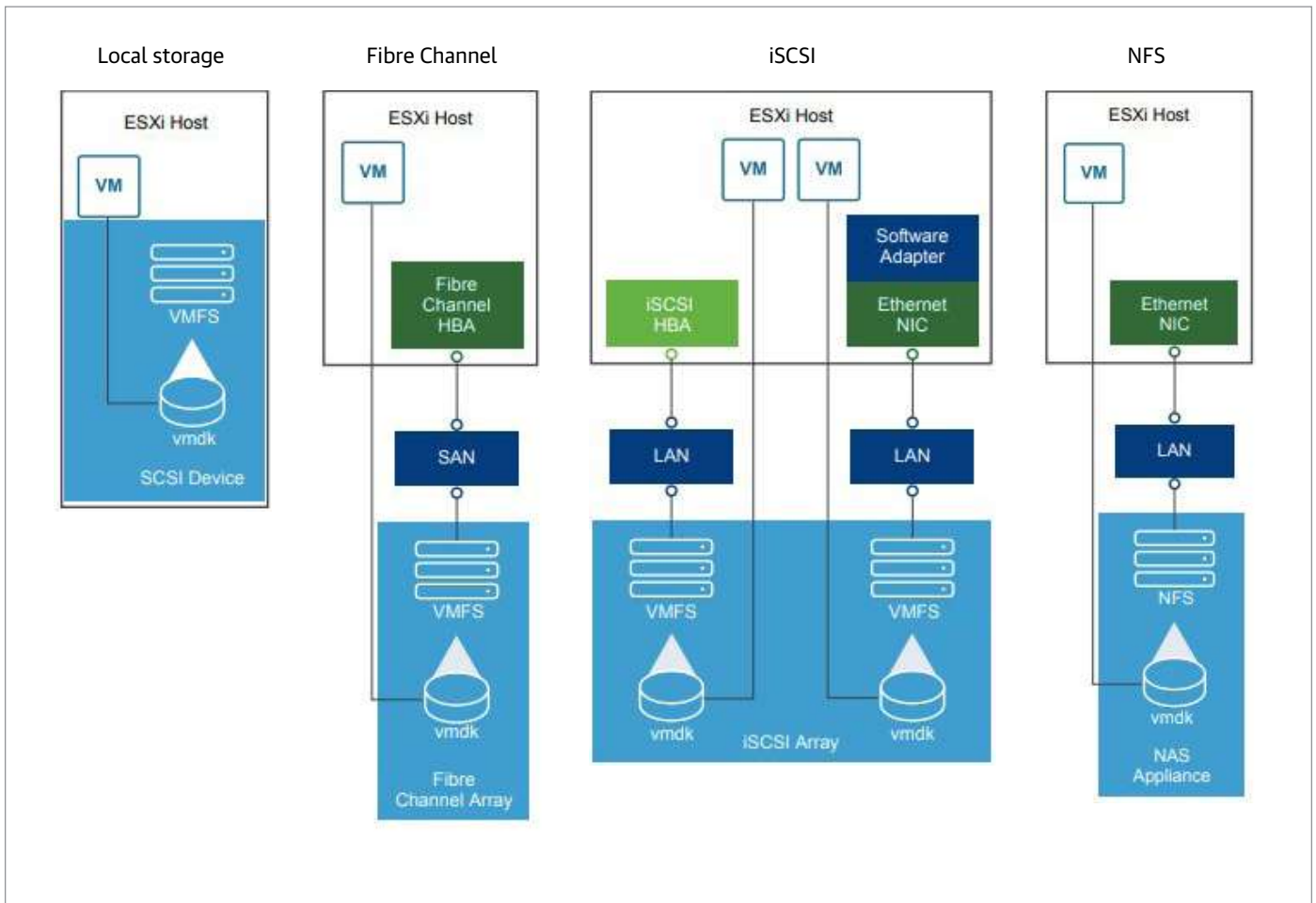
the `allowedAll` flag to `FALSE`. The NFS server IP is added to the allowed list of outgoing IP addresses.

- If the `nfsClient` rule set is enabled, its state and the Allow All IP Address policy is not changed. The NFS server IP is added to the allowed list of outgoing IP addresses.
- **When removing or unmounting an NFS 3 datastore**, if no remaining NFS datastores are mounted from the NFS server IP of the datastore being removed, ESXi removes the NFS server IP.
- If there are no other mounted NFS 3 datastores, ESXi disables the `nfsClient` rule set.

NFS 4 firewall rule behavior:

- **When mounting the first NFS 4.1 datastore**, ESXi enables the `nfsClient` rule set, opens port 2049 for all IP addresses and sets the `allowedAll` flag to `TRUE`.
- **Unmounting an NFS 4.1 datastore does not affect the firewall state.** Port 2049 remains open until closed explicitly.

Below is a visual representation of the major storage types used in vSphere:





## Software Defined Storage (vSAN)

VMware has its own proprietary version of software defined storage called Virtual SAN (vSAN). vSAN aggregates all ESXi host local storage into a single datastore shared by all ESXi hosts in the vSAN Cluster. To implement vSAN, the following minimum requirements must be met:

- Three ESXi hosts or two ESXi hosts and a Witness for Stretched Clusters
- ESXi 5.5 Update 1
- Eight GB RAM
- One Gb ethernet adapter for hybrid clusters or 10 Gb ethernet for All Flash
- One flash drive for cache and one hard disk drive for capacity storage
- Storage controllers configured for passthrough (recommended) or RAID-0
- Read cache and advanced features disabled or enable read cache 100%
- Set queue depth greater than or equal to 256
- Have uniform vSphere and vCenter versions and similar hardware vendor
- ESXi hosts configured for same on-disk format, version 2 or version 3
- A valid vSAN Standard license (Enterprise, for encryption or stretched clusters)
- Must turn vSphere HA off to enable vSAN
- vSAN datastores cannot be used for vSphere HA heart beating
- RTT no greater than 1 ms for Standard vSAN, 5 ms for Stretched, and 200 ms from the main site to the witness host

vSAN consists of the following components:

- vSphere, minimum 5.5 Update 1
- vSphere Cluster
- Disk groups, consisting of at least one flash device for cache and up to seven hard disk drives for capacity storage; maximum of five groups per ESXi host
- Witness is the component containing metadata that serves as a tiebreaker when a decision must be made regarding availability of surviving datastore components
- Storage Policy-Based Management (SPBM) for VM storage requirements
- Ruby vSphere Console (RVC) is a CLI console for managing and troubleshooting a vSAN Cluster
- vSAN Observer runs on RVC and is used to analyze performance and monitor a vSAN Cluster
- Fault domains
- vSAN storage objects

A vSAN datastore has the following components:

- A VM Home Namespace is the home directory where all VM configuration files are stored, such as .vmx, .vmdk, .log and snapshot descriptor files
- VMDK is the VM disk that stores a VM hard disk contents





- A VM Swap object is created when a VM is powered on
- Snapshot Delta VMDKs are created when a snapshot is taken
- A memory object is created when the snapshot memory is selected when taking a VM snapshot

vSAN does not support vSphere features: SIOC, DPM, RDMS, or VMFS.

VMs are in one of two compliance states, Compliant and Noncompliant, viewable on the Virtual Disks page > Physical Disk Placement tab.

vSAN components can be in one of the two following states:

- **Absent** is when vSAN detects a temporary component failure, for example a host is restarted
- **Degraded** is when vSAN detects a permanent component failure, for example a component is on a failed device

vSAN objects are in one of the two following states:

- **Healthy** is when at least one full RAID-1 mirror is available or the minimum required number of data segments are available
- **Unhealthy** is when there is no full mirror available or the minimum required number of data segments are unavailable for RAID-5 or RAID-6 objects

vSAN Cluster can be one of two types:

- Hybrid clusters have a mix of flash storage, used for cache; and hard disk storage, used for capacity
- All-flash clusters use flash for both cache and capacity

vSAN Clusters can be deployed in one of the following ways:

- **Standard vSAN Cluster** deployment consists of a minimum of three ESXi hosts
- **Two-Node vSAN Cluster** deployment is generally used for ROBO locations with an optional witness host at a separate remote location
- **Stretched vSAN Cluster** provides resiliency against the loss of an entire site by distributed ESXi hosts in a cluster across two sites with latency less than 5 ms between the sites and a witness host at a third location

## Virtual Volumes (VVols)

VVOLs, identified by a unique GUID, is an encapsulation of VM files, VMDKs, and their derivatives stored natively on a storage system. VVOLs are created automatically when performing a VM operation: creation, cloning, snapshotting.

The following are VVol types that make up core VM elements:

- **Data-VVols** correspond directly to each VM .vmdk file, either thick- or thin- provisioned
- **Config-VVol** is a home directory that represents a small directory containing VM metadata, such as .vmx, .log, etc.; is thin-provisioned
- **Swap-VVol** holds a VM's memory pages and is created when a VM powers on and is thick-provisioned by default



- **Snapshot-VVol** holds content of VM memory for a snapshot and is thick-provisioned
- Other is used for a special feature, such as Content-Based Read Cache (CBAC)

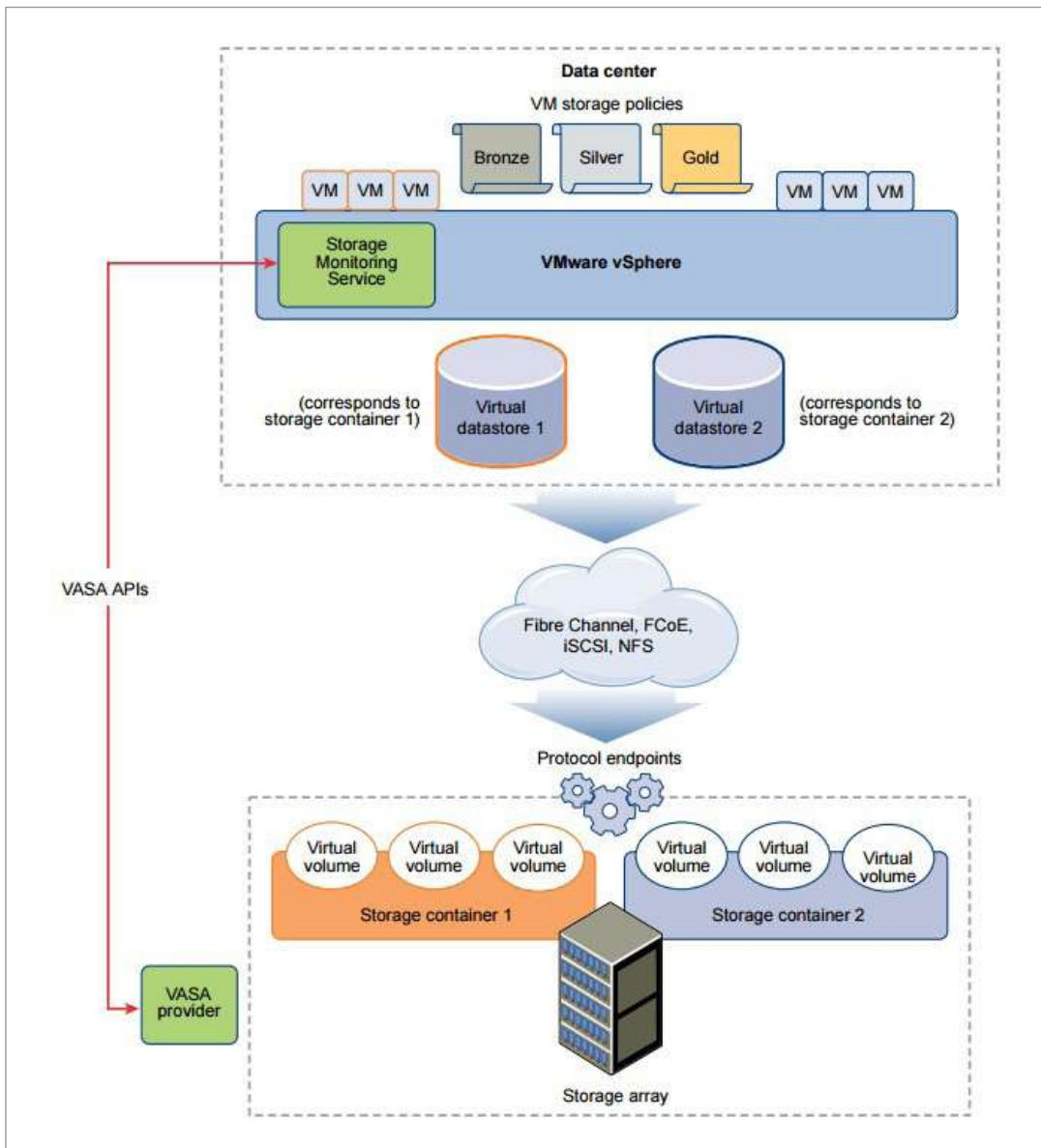
**NOTE: VM at minimum creates a Data-VVol, Config-VVol and Swap-VVol.**

Storage Providers, or VASA Provider, is a software component acting as a storage awareness service for vSphere, mediating out-of-band communication between vCenter Server and ESXi hosts on one side and the storage system on the other.

Storage Containers are a pool of raw storage capacity or an aggregation of storage capabilities a storage system can provide to virtual volumes.

Protocol Endpoints are logical I/O proxy for ESXi hosts to communicate with virtual volumes and virtual disk files virtual volumes encapsulate.

Below is VMware's visual representation of the VVol architecture:





## Persistent Memory (Non-Volatile Memory or NVM)

These devices combine performance and speed for VM workloads such as acceleration databases and analytics, which require high bandwidth, low latency and persistence.

## Objective 1.4: Differentiate between NIOC and SIOC



Resource:

### Network I/O Control (NIOC)

NIOC, first introduced in Virtual Distributed Switch (vDS) version 6, is used to allocate network bandwidth to business-critical applications and resolve situations where several types of traffic compete for network resources. NIOC allocates bandwidth to VMs via shares, reservations and limits, concepts of which will be discussed in more detail in Objective 1.7.

NIOC has the following requirements:

- NIOC requires a vSphere Enterprise *Plus* license
- NIOC version 2 requires ESXi 5.1/5.5/6.0 and vDS 5.1/5.5
- NIOC version 3 requires ESXi 6.0 and higher and vDS 6.0 and higher. VMs cannot use SR-IOV if NIOC v3 is enabled

Bandwidth can be allocated for the following network traffic types:

- Management
- Fault Tolerance (FT)
- NFS
- vSAN
- vMotion
- vSphere Replication (vRep)
- vSphere Data Protection Backup (VDPB)
- VM

NIOC can be implemented in one of two allocation models for VMs:

- Allocation across an entire vDS
- Allocation on the physical adapter that carries VM traffic

vSphere guarantees enough bandwidth by utilizing Admission Control at ESXi host and cluster levels based on bandwidth reservation and teaming policy.

- Admission control on the vDS verifies if an ESXi host physical adapter can supply minimum bandwidth to the VM according to the teaming policy and bandwidth reservation, and the VM network adapter reservation is less than the quota of the network resource pool
- Admission control with vSphere DRS ensures when powering on a VM, the VM is placed on an ESXi host to guarantee bandwidth for the VM according to the teaming policy
- Admission control with vSphere HA ensures a VM is restarted on an ESXi host according to the bandwidth reservation and teaming policy



[vSphere 6.7 Networking Guide, Update 2 \(April 2018\)](#), pages 176-184



[vSphere 6.7 Resource Management Guide, Update 2 \(April 2019\)](#), pages 53-59



## Storage I/O Control (SIOC)

SIOC allows cluster-wide storage I/O prioritization, allowing better workload consolidation, and helps reduce cost due to over provisioning, using shares and limits constructs per-VM, to control I/O allocated to VMs during congestion.

SIOC has the following minimum requirements and is configured per datastore:

- vCenter Server and ESXi 4.1 or higher
- Datastore with SIOC can be managed by only one vCenter Server
- Datastore with SIOC cannot have RDMS; but iSCSI, FC, NFS are supported
- Datastore with SIOC cannot have multiple extents

SIOC integrates with Storage DRS based on SPBM rules. The SDRS `EnforceStorageProfile` advanced option has one of three values:

- 0 (default value) means SPBM policy is not enforced on the SDRS cluster
- 1 means SPBM policy is soft enforced on the SDRS cluster
- 2 means SPBM policy is hard enforced on the SDRS cluster

## Objective 1.5: Manage vCenter inventory efficiently

VMware provides several options to manage the vCenter Server inventory. To do so efficiently, it's recommended to automate and use tags wherever possible to free your time up when needing to perform recurring tasks, and to make searching, reporting and even backup more powerful. For this objective, I will highlight only a few tools that I feel VMware allows for efficient vCenter inventory management.

### vCenter inventory objects hierarchy

- Data centers
- Clusters
- ESXi hosts
- Resource pools
- Virtual machines and vApps
- Inventory folders

### vCenter inventory efficiency tools

#### Tags

Tags and attributes allow users to attach metadata to vCenter inventory objects for easier sort and search capability.

- Categories allow grouping of related tags together, for example: `operating_system`
- Tags are labels attached to an inventory object, for example: `windows`, `linux`, `mac_os`

Categories and tags replicate to other vCenter Servers when using Enhanced Linked Mode. Categories and tags are also maintained across linked domains when using Hybrid Linked Mode.



### Resource:



[vCenter Server and ESXi Host 6.7 Management Guide, Update 2 \(April 2019\)](#), pages 80-89; 122



You can configure category cardinality to either allow assigning only one tag per category or multiple tags in a category.

To configure tagging, click the menu drop-down and select Tags and Custom Attributes. From here you can create categories and tags and assign them to vCenter objects.

<input type="checkbox"/>	Category Name ▾	Description ▾	Multiple Cardinality ▾
<input type="checkbox"/>	Veeam_Backups		true

Once categories and tags are created, you can easily assign tags to your vCenter inventory objects in the vSphere Client (see below image) by right-clicking on object > Tags and Custom Attributes, then select Assign tag; or by using CLI.

- Export System Logs...
- Edit Settings...
- Move to folder...
- Rename...
- Edit Notes...
- Tags & Custom Attributes ▶
  - Assign Tag...
  - Remove Tag
  - Edit Custom Attributes...
- Add Permission...
- Alarms ▶



## Automation

Another way to efficiently manage vCenter inventory is by using automation tools such as PowerCLI, esxcli, and vSphere CLI tools. I will share some common commands using `esxcli`, `vim-cmd` and `PowerCLI`:

- `esxcli`
  - **Enable Maintenance Mode:**  
`esxcli system maintenanceMode set --enable yes`
  - **Check SNMP settings:**  
`esxcli system snmp get`
  - **List multipathing modules:**  
`esxcli storage core plugin list --plugin-class=MP`
  - **List devices controlled by NMP:**  
`esxcli storage nmp devices list`
  - **List all host SATPs:**  
`esxcli storage satp list`
  - **List host IPv4 addresses:**  
`esxcli network ip interface ipv4 get` or `vicfg-nic list`
  - **Check host vmnics link state and MTU:**  
`esxcli network nic list`
  - **Bring up a vmnic:**  
`esxcli network nic up -n vmnic#`
  - **List all VMs on host**  
`esxcli vm process list`
- `vim-cmd`
  - **Start (Stop, Restart) a VM**  
`vim-cmd vmsvc/power.on (.off, .reboot) vmID`
  - **Unregister a VM**  
`vim-cmd vmsvc/unregister vmID`
  - **Register a VM**  
`vim-cmd solo/registervm pathtoVMXfile`
- `PowerCLI`
  - **List all VMs on a host**  
`Get-VMhost nameofHost | Get-VM`
  - **List VM snapshots**  
`Get-VMhost nameofHost | Get-VM | Get-Snapshot`



### Scheduled tasks

Scheduled tasks are a great way to automate recurring tasks within your vCenter Server inventory hierarchy. With scheduled tasks, you can:

- Change a VM power state
- Clone/create/deploy a VM
- Make a VM snapshot
- Enable/disable DPM
- Scan for updates

The caveat to scheduled tasks is that you cannot run them against multiple objects, but rather on one object at a time. To create a scheduled task, go to the inventory object in the vSphere Web Client > **Monitor**, select **Tasks and Events** tab, select **Scheduled Tasks** from the list and then **Schedule a New Task, Check** (host profile) **Compliance**. For example:

The screenshot shows the vSphere Web Client interface. At the top, there are tabs: Summary, Monitor (selected), Configure, Permissions, Snapshots, Datastores, and Network. Below these are sub-tabs: Issues, Performance, Tasks & Events (selected), Policies, and Utilization. On the left, there is a sidebar with a back arrow and three items: Tasks, Events, and Scheduled Tasks (highlighted in blue). On the right, there is a text box with a clock icon: "To create a scheduled task, select an action from the Actions menu for an object in the Inventory Lists, click the Actions icon, and then select the action you want to schedule. The Actions icon indicates the actions that you can schedule on the object." Below this is a button labeled "Schedule a New Task" with a dropdown arrow. At the bottom right, there is a table with two columns: "Task" and "Schedule".

Task	Schedule



## Objective 1.6: Describe and differentiate among vSphere HA, DRS & SDRS functionality

After reviewing this objective, you may conclude a lot of function and feature set information for vSphere HA, DRS and Storage DRS has been neglected, and you would be correct. The reason for this is because further discussion of vSphere clusters is provided in Objective 1.9. Also keep in mind not every detail of every vSphere function and feature can be included in this study guide. We've only provided many of the high points we feel are needed to help you in your VCP studies. If any remaining details or explanations are needed to further enhance your understanding of vSphere concepts and features discussed, we ask you to take the time to read and study VMware's well-written user guides.

### vSphere HA

vSphere HA provides High Availability to virtual machines by combining VMs and the hosts they reside on into vSphere Clusters. ESXi hosts are monitored and, in the event of a host failure, VMs are restarted on other available ESXi hosts in the cluster.

vSphere HA works in the following way: When vSphere HA is enabled, vCenter Server installs the HA Agent (.fdm) on each ESXi host. The ESXi host with the most datastores connected to it has the advantage of becoming the master, as determined by an election process, and the remaining hosts are subordinates. The master host monitors the subordinate hosts every second through network heartbeats, which is the management network, or if vSAN is used it's the vSAN network. If heartbeats are not received through this method, before the master determines a subordinate host has failed, the master detects the liveness of the subordinates by seeing if the subordinate host is exchanging datastore heartbeats. The default number of datastores used in heartbeating is two and a maximum of five can be configured using the vSphere HA advanced option: `das.heartbeatDsPerHost`. When vSphere HA is enabled, a root directory is created on datastores used for heartbeating, named `.vSphere-HA`. vSAN datastores cannot be used for heartbeating.

The master is responsible for detecting the following types of host failures:

- A **host failure** is when a host stops functioning
- A **host network partition** is when a host loses connection to the master
- A **host network isolation** is when a host is running but cannot observe traffic from other host HA agents or is unable to ping the network isolation address. When a host is running but isolated, you can configure one of three responses:
  - Disabled means host isolation responses are suspended. Also, if an individual VM has its isolation response disabled, then no isolation response is made for that VM.
  - Power off and restart VMs hard powers VMs and restarts them on another host.
  - Shutdown and restart VMs gracefully shut down VMs if VMware Tools is installed on the VM and restarts them on another host.



### Resource:

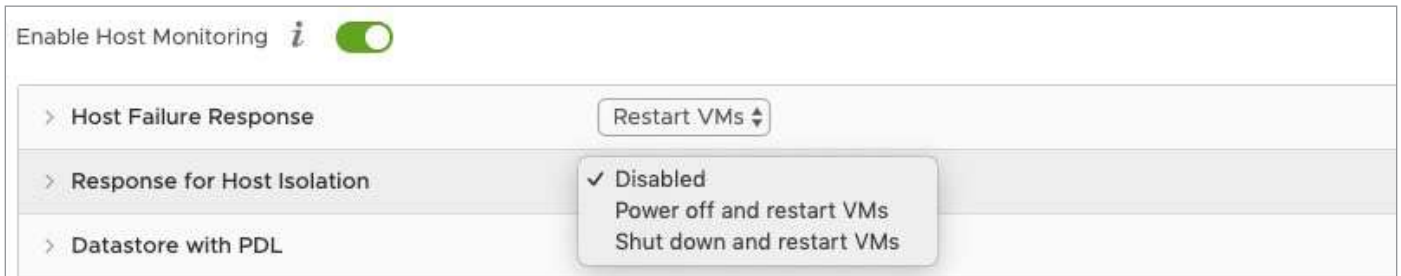


[vSphere 6.7 Availability Guide, Update 2 \(April 2019\)](#), pages 11-47



[vSphere 6.7 Resource Management Guide, Update 2 \(April 2019\)](#), pages 70-101; 103-119





- A **Proactive HA failure** occurs when there is a failure in a host component resulting in loss of redundancy, for example a power supply failure. For these failures, remediation can be handled via automation in the vSphere Availability section of the vSphere Client.

The vSphere HA master hosts use several factors when determining which available Cluster hosts to restart VMs on:

- File accessibility of a VM's files
- VM host compatibility with affinity rules
- Available resources on a host to meet VM reservations
- ESXi host limits, for example VMs per host or vCPUs per host maximums
- Feature constraints as such that vSphere HA does not violate anti-affinity rules or fault tolerance limits

### vSphere Distributed Resource Scheduler (DRS)

Simply put, DRS is a collection of ESXi hosts and VMs with shared resources and management interface. When VMs are powered-on in the cluster, DRS automatically places VMs on hosts in such a way to optimize resource utilization in the cluster. DRS performs admission control during this Initial Placement phase by verifying if there are enough cluster resources to power VMs on. If there isn't, the operation is not allowed. If there are available resources, DRS displays a recommendation on the **Power On Recommendations** section as to where to place the VM(s). If the DRS Automation Level is set to Fully Automated, this recommendation is performed automatically. For group VM power-on, initial placement recommendations are given at the cluster level. If any VM in a group power-on has placement-level settings set to manual, then all VM power-ons will be manual.

Additionally, when a DRS cluster becomes resource-unbalanced, DRS migrates (or recommends migration of) VMs to other hosts in the cluster using VMware vMotion. If the Automation Level is manual (which is the default) or partially automated, a recommendation is given and manual intervention is required. If needed, you can specify per-VM migration settings.

Migration recommendations can be one of the following:

- Balance average CPU load
- Balance average memory load
- Satisfy resource pool reservations
- Meet affinity rules
- Host is entering maintenance mode



DRS also allows you flexibility in controlling VM placement using affinity and anti-affinity rules, allowing you to either keep VMs together on the same host or place them on separate hosts. You can create two types of rules:

- **VM-Host** affinity or anti-affinity rules are specified against a group of VMs and group of ESXi hosts. Good use case is for licensing requirements
  - Affinity rule means group of VMs should or must run on a group of hosts
  - Anti-affinity rule means VMs in a group cannot run on a group of hosts
- **VM-VM** affinity or anti-affinity rules are used between individual VMs
  - Affinity rule means DRS tries to keep a pair of VMs on the same host
  - Anti-affinity means DRS tries to keep a pair of VMs on different hosts

Take note if two rules are in conflict of each other, the older rule takes precedence and the newer rule is disabled. Also DRS gives higher precedence to anti-affinity rules than affinity rules. Any violations of affinity rules can be found under the Cluster > Monitor tab > vSphere DRS section, then click Faults.

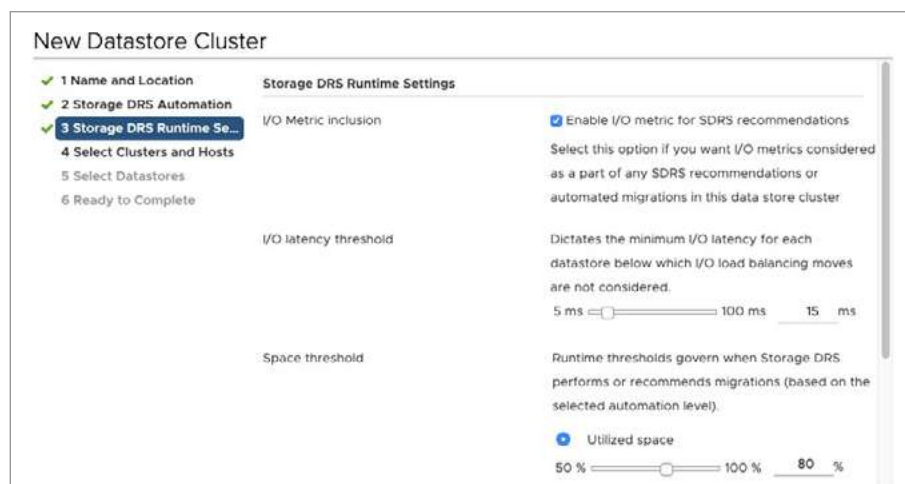
### vSphere Storage DRS (SDRS)

SDRS is the result of creating a Datastore Cluster. A Datastore Cluster is a collection of datastores to manage storage resources and support resource allocation policies created at the cluster level. Like compute DRS, you are prompted for initial placement of VM virtual disks if SDRS is configured in manual mode. In automatic mode, VM disks are moved for you.

You can use the following resource management capabilities to trigger SDRS recommendations:

- **Space utilization load balancing** is used to trigger recommendations based off a set threshold on a datastore. If the threshold is exceeded, a migration is recommended.
- **I/O latency load balancing** can be based off a set I/O latency threshold, below which migrations are not allowed.
- **Anti-affinity rules** can be used to have VM disks on different datastores.

Each setting above can be configured to: use cluster settings, no automation (manual mode), or fully automated.





## Objective 1.7: Describe and identify resource pools and use cases



### Resource:



[vSphere 6.7 Resource Management Guide, Update 2 \(April 2019\)](#), pages 60-65

### Resource Pools (RPs) characteristics

Resource Pools are logical abstractions for flexible management, grouped in hierarchies, of CPU and memory resources.

Each standalone ESXi host and vSphere Cluster has an invisible root resource pool. When adding an ESXi host to a cluster, any resource pool configured on the host will be lost and the root resource pool will be a part of the cluster.

RPs utilize share, reservation, and limit constructs, each explained below:

- **Shares** represent the relative importance of CPU and memory resources of sibling powered-on VMs and resource pools, with respect to a parent resource pool's total resources, when resources are under contention. Because shares are relative and values change based on the number of VMs in a resource pool, you may have to change a VM's share value (see below) when moving in or out of a resource pool:
  - High represents four times that of low
  - Normal represents two times that of low
  - Low represents the value of one
- **Reservation** represents a guaranteed amount of resource to a VM or resource pool:
  - Expandable reservation is a setting allowing child resource pools to get more compute resources from its direct parent RP when the child resource pool is short of resources; the setting is enabled by default
  - Expandable reservation is recursive, meaning if the direct parent RP doesn't have enough resources available, it can ask the next higher parent resource pool for resources
  - If there are not enough resources available, a VM cannot power on
- **Limits** are upper bounds of a resource; default value is unlimited

Resource pools behave in the following ways:

- A VM's reservation and limit settings do not change when moving into a new resource pool.
- A VM's percentage shares adjust to reflect the total number of shares in use in a new resource pool, if a VM is configured with default high, normal, or low values. If share values are custom, the percentage share is maintained.
- Resource pools use Admission Control to determine if a VM can be powered on or not. Admission Control determines the unreserved resource capacity (CPU or memory) available in a resource pool.

A root resource pool can be segregated further with child resource pools, which owns some of the parent resource pool resources.

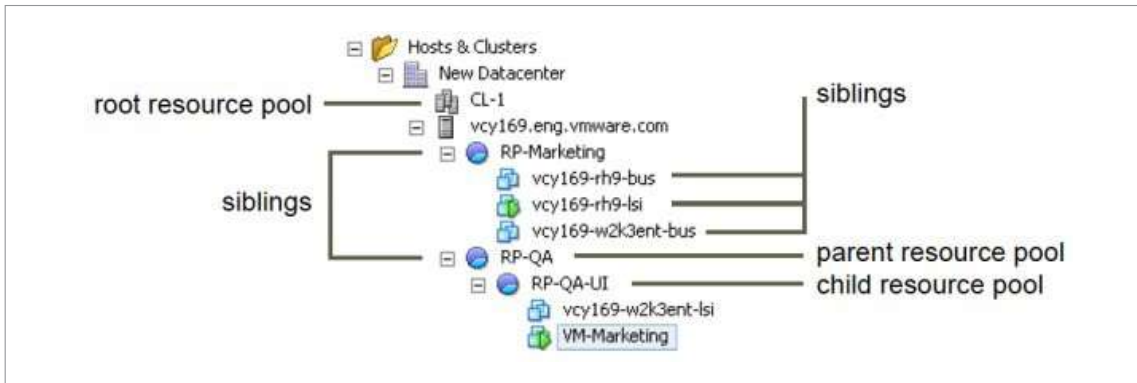
A resource pool can contain child resource pools, VMs, or both.



Resource pools have the following hierarchical structure:

- Root resource pool is on a standalone ESXi host or cluster level
- Parent resource pool is at a higher level than sub-resource pools
- Sibling is a VM or resource pool at the same level
- Child resource pool is below a parent resource pool

Below is VMware's visual representation of a resource pool hierarchical structure:



## RP use cases and share concepts

Resource pools have the following use cases:

- Compartmentalize cluster resources for individual departments
- Isolation between resource pools and sharing within resource pools
- Access-control delegation to department administrators
- Provides separation of resources from hardware for management and maintenance
- Provides ability to manage VMs with multitier services

When figuring out share values needed for VM workloads, it's best to work from the VM-level up to resource pools, instead of cluster resource pool down.

Below a resource pool assignment example to further explain shares:

- A cluster has 100 CPUs and 100 GB RAM
- Calculate high shares allocation in a RP:
  - $100 \text{ CPU} \times 4 = 400$  Shares of CPU for "high"
  - Divide the number of shares by total number of shares for each value (high, normal, low) to determine the percentage of RP shares needed for high value VMs (57%, 29%, 14%)
  - Multiply the percentage by the total CPU resources for the resource pool, then divide the number by how many VMs are high allocated
  - Do the same procedure for memory
- Calculate normal and low shares allocation in a RP following the same steps above

- To further explain the shares calculation, review the following example:
  - Cluster = 12 GHz CPU
  - RP1 is allocated 8 GHz, configured for high, which is 67% of cluster CPU
  - RP2 is allocated 4 GHz, configured for normal, which is 33% of CPU
  - VM1 is in RP1
  - VM2 and VM3 are in RP2
  - Since VM1 is the only VM in RP1, it gets 67% of cluster CPU resources
  - Because there are two VMs in RP2, each VM will get 16% of Cluster CPU, which is determined by 33% RP2 Cluster allocation divided by the number of VMs in RP2

## Objective 1.8: Differentiate vDS and vSS

 Resource:



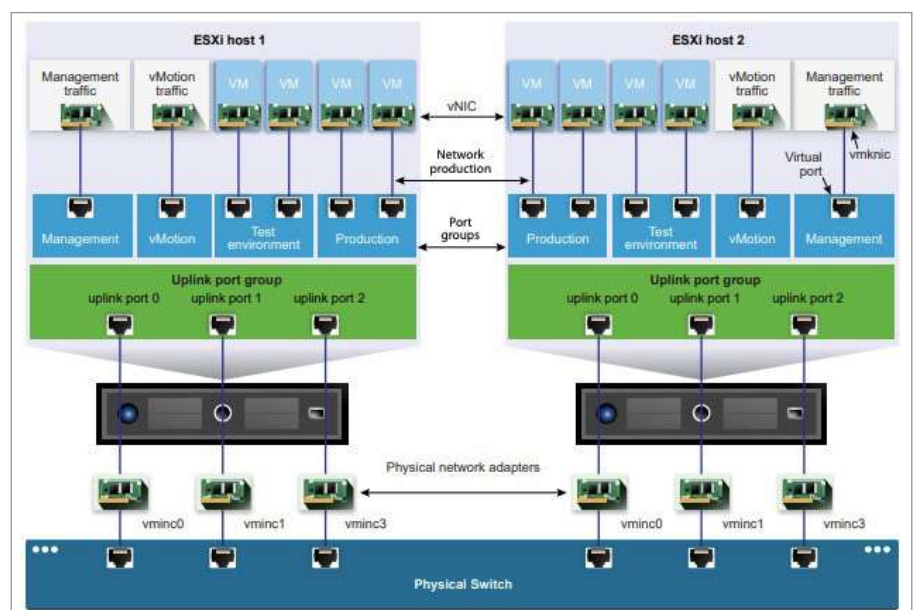
[vSphere 6.7 Networking Guide, Update 2 \(April 2018\)](#)

### vSphere Standard Switch (vSS)

To provide network connectivity to ESXi hosts and VMs, you connect host physical NICs (pNics) to standard switch uplink ports to provide external network connectivity. In turn, VMs have virtual NICs (vNICs) you connect to vSS port groups. And each vSS port group can use one or more host pNics to handle network traffic. When connecting two or more pNics to a standard switch, they are transparently teamed. If no pNic is connected to a port group, VMs connected to the same port group can communicate with each other in what's referred to as an isolated environment because there is no communication with the external network. To ensure efficient use of host resources, standard switch ports are dynamically scaled up and down, up to the host maximum supported.

Each standard switch port group is identified by a network label, unique to the host. You can use network labels to make network configuration of VMs portable across hosts. This can be achieved by creating the same network label name on port groups in a data center that use pNics connected to one broadcast domain on the physical network. For example, you can create Production and Test port groups on hosts sharing the same broadcast domain.

VMware provides a nice visual representation of standard switches, as shown in the picture:





When creating a standard switch, you have the option to create it in one of three ways – VMkernel network adapter, physical network adapter, or the VM port group for a standard switch.

**1 Select connection type**

**2 Select target device**

**3 Port properties**

**4 IPv4 settings**

**5 Ready to complete**

**Select connection type**  
Select a connection type to create.

- VMkernel Network Adapter**  
The VMkernel TCP/IP stack handles traffic for ESXi services such as vSphere vMotion, iSCSI, NFS, FCoE, Fault Tolerance, vSAN and host management.
- Virtual Machine Port Group for a Standard Switch**  
A port group handles the virtual machine traffic on standard switch.
- Physical Network Adapter**  
A physical network adapter handles the network traffic to other hosts on the network.

CANCEL BACK NEXT

- **VMkernel Network Adapter** is used to handle traffic for ESXi host management, vMotion, NFS, iSCSI, FCoE, FT, or vSAN services. Keep in mind when configuring virtual networking, if creating a vMotion network, all ESXi hosts for which you want to migrate VMs to must be in the same broadcast domain – Layer 2 subnet.
- **Virtual Machine Port Group** handles traffic for VMs on the standard switch.
- **Physical Network Adapter** provides external network connectivity.

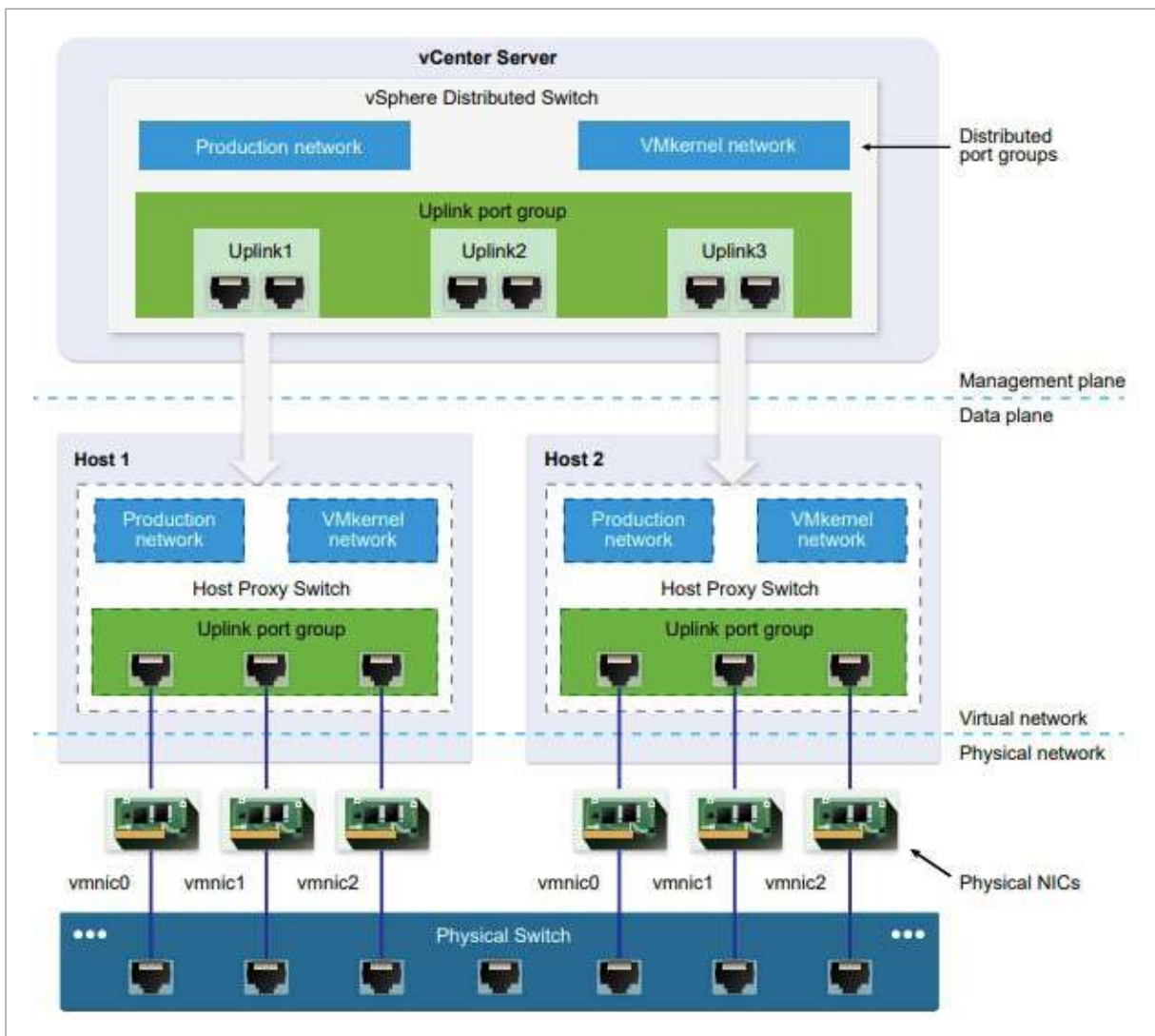
As mentioned in the beginning, virtual networking management, such as creating a standard switch, adding port groups, configuring VLAN tagging such as EST/VST/VGT, and MTU size will be discussed in subsequent objectives in this guide.





## vSphere Distributed Switch (vDS)

A Distributed Switch provides centralized management and monitoring of vSphere networking configuration among all ESXi hosts associated with it. A vDS is setup on vCenter and all configured settings are propagated to hosts connected to it. This is also very relevant information to keep in mind in distinguishing between a standard and distributed switch – standard switches are local to ESXi hosts, whereas distributed switches are a vCenter construct and shared among hosts connected to it.



A distributed switch consists of two logical sections – the data plane (also called “host proxy switch”), is used for package switching, filtering, tagging, etc.; and a management plane, used for configuring data plane functionality. The management plane resides on vCenter, while the data plane remains local to each ESXi host. vDS versions have evolved over the past few VMware releases. When creating a distributed switch, you can view when certain feature sets were added to which version.

Keep in mind, if you are needing to upgrade your vDS to a newer version, ESXi hosts, and VMs connected to the vDS will experience a brief downtime.

A distributed switch has two abstractions used to create consistent networking configuration for hosts, VMs, and VMkernel services:

- **Uplink port group**, or dvUplink, is defined at vDS creation and can have one or more uplinks. Uplinks are a template used to configure physical connections to hosts, as well as failover and load balancing policies. As with a standard switch, you assign pNics to vDS uplinks, configure appropriate policies, and those settings are propagated down to the host proxy switches.
- **Distributed port group** provides network connectivity to VMs and accommodate VMkernel traffic, each identified using a network label unique to the data center. Like uplinks, you configure load balancing and teaming policies, but additionally configure VLAN, security policy, traffic shaping and others, which are also propagated down to the host proxy switches.

Since this objective states to "differentiate" between the two virtual switches, I'll summarize some major differences:

- vSS is host-based, whereas vDS is centralized in vCenter
- vSS comes with ESXi installed by default, whereas vDS requires Enterprise *Plus* licensing
- Since vSphere 5.1 both vSS and vDS have the capability to rollback/recover the management network via the host DCUI
- vDS has two component 'planes' that make up its construct – management and data planes
- vDS offers more feature sets, including:
  - NIOC capability
  - Load balance policy based off physical NIC load
  - Netflow traffic monitoring
  - Ingress traffic shaping, in addition to egress (vSS is egress only)
- vDS offers networking vMotion

Lastly, I'll leave you with a few best practices to keep in mind when setting up your vSphere virtual networking environment. For a full list of VMware recommendations, review pg. 249.

- Isolate your vSphere network services to improve security and performance, i.e. management, vMotion, FT, NFS, etc.
- Dedicate a second pNic to a group of VMs if possible or use NIOC and traffic shaping to guarantee bandwidth to VMs
- For highest security and traffic service isolation, create separate standard and/or distributed switches and dedicate pNics to them
- Keep vMotion traffic on a separate network as guest OS memory content is transmitted over the network
- Configure the same MTU size on all VMkernel network adapters, otherwise you might experience network connectivity issues
- Use vmxnet3 vNics on VMs for best performance

## New features and enhancements

### Distributed switch: 6.6.0

- MAC Learning

### Distributed switch: 6.5.0

- Port Mirroring Enhancements

### Distributed switch: 6.0.0

- Network I/O Control version 3
- IGMP/MLD snooping





## Objective 1.9: Describe the purpose of cluster and the features it provides



### Resource:



[vSphere 6.7 Resource Management Guide, Update 2 \(April 2019\)](#), pages 70-102

### vSphere Cluster

As discussed in Objective 1.6, a vSphere DRS Cluster is used to pool ESXi host CPU and memory resources to more efficiently distribute those resources among VMs associated with those hosts in the cluster. Clusters are not solely used to pool host resources, but also provide High Availability to VMs in the event of host failure. In the previous objective, I just explained the high-level concepts behind vSphere HA, DRS, and Storage DRS. In this objective, I will go more in-depth about each cluster service.

### vSphere HA

To be able to use vSphere HA, the following requirements must be met:

- vSphere Standard license
- Shared storage
- vSphere Cluster with minimum of two ESXi hosts, each configured with a management network, used for HA network heartbeating
- ESXi host access to all VM networks
- Depending on vSphere HA features needing, VMware Tools installed

After the above requirements have been met, you then need to create a vSphere Cluster at the data center level in vCenter. Right-click the data center object and select New Cluster:

Name	Value
Name	New Cluster
Location	DC1-SITE
vSphere DRS	<input type="checkbox"/>
vSphere HA	<input type="checkbox"/>
vSAN	<input type="checkbox"/>

These services will have default settings - these can be changed later in the Cluster Quickstart workflow.

CANCEL OK



After vSphere HA has been enabled, you can then configure features and options within the cluster. First, vSphere HA not only helps make VMs highly available, but can also protect against potential VM "split-brain" scenarios in which a VM may be running on an isolated ESXi host, but the master cannot detect the VM because the host is isolated. As such, the master will try and restart the VM. The restart succeeds if the isolated host no longer has access to the datastore on which the VM resides. This causes a VM split-brain because now two instances of the same VM are running. The vSphere HA setting used to mitigate this scenario is called VM Component Protection (VMCP) and is a combination of two vSphere HA settings:

- **Datastore with PDL** (permanent device loss), refers to unrecoverable accessibility to a storage device, mitigated only by powering off VMs. When enabled, you can only select Disabled, Issue Events, or Power off and restart VMs
- **Datastore with APD** (all paths down), refers to a transient or unknown loss to a storage device. APD setting options are more granular than PDL

▼ Datastore with APD

All Paths Down (APD) Failure Response      Allows you to configure the cluster to respond to APD Datastore failures

Disabled  
No action will be taken on the affected VMs.

Issue events  
No action will be taken on the affected VMs. Events will be generated.

Power off and restart VMs - Conservative restart policy  
A VM will be powered off, if HA determines the VM can be restarted on a different host.

Power off and restart VMs - Aggressive restart policy  
A VM will be powered off, if HA determines the VM can be restarted on a different host, or if HA cannot detect the resources on other hosts because of network connectivity loss (network

CANCEL    OK

**IMPORTANT:** If Host Monitoring or VM Restart Priority settings are disabled, VMCP cannot be used to perform VM restarts. Also, VMCP requires minimum of ESXi 6.0 and is not supported for FT VMs, vSAN, RDM, or VVols.

One of the most important HA configurations you need to decide upon that best suits your environment is the concept of Admission Control. vSphere HA Admission Control ensures a HA-enabled cluster has enough reserved resources available for VM restarts in the event of one or more ESXi host failures.



Any action on a cluster resource that potentially violates constraints imposed by Admission Control are not permitted, such as:

- Powering on VMs
- Migrating VMs
- Increasing CPU or memory reservations of VMs

Admission Control can be configured for one of three options – cluster resource percentage, slot policy, or dedicated failover hosts. I discuss each below:

- **Cluster resource percentage** is a policy where vSphere HA ensures a percentage of CPU and memory are reserved for failover. The policy is enforced by:
  - Calculating current powered-on VM CPU and memory usage
  - Calculating total cluster CPU and memory resources
  - Calculating current failover capacity, by taking total cluster resource – VM requirements/total cluster resource, for both CPU and memory
  - Determines if current capacity of either CPU or memory is more or less than the configured failover capacity. If current capacity is less, VM operations listed above are not permitted
- **Slot policy** is where a specified number of ESXi hosts can fail and enough resources will still be available to restart all VMs on failed hosts. This policy is determined by:
  - Calculating slot size, which is a logical representation of CPU and memory sized to satisfy requirements of powered-on VMs. Slot size is calculated by VM CPU reservation (or 32 MHz if none configured) and memory reservation then selecting the largest value of each.
  - Determines how many slots each cluster host holds by dividing the largest of each resource, CPU and memory, by the total resource of each host. The host with the largest number of slots is "removed" and the remaining host slots are added.
  - Determines current failover capacity, which is the number of ESXi hosts that can fail and still have available slots for powered-on VMs.
  - Determines if current capacity of either CPU or memory is more or less than the configured failover capacity. If current capacity (or slots) is less, VM operations listed above are not permitted.
- **Dedicated failover hosts** is the easiest policy of all the admission control policies. If this policy is selected, one or more ESXi hosts cannot be used in a cluster and are solely used in the event of a host failure. If a failover host cannot be used, for example there aren't enough host resources available, then other available cluster hosts are used. Note that VM-VM affinity rules are not enforced on failover hosts.



Regardless of the Admission Control policy used, be aware of one other setting you can configure – **Percentage degradation VMs tolerate**. View the image below for explanation.

**NOTE: DRS must be enabled for this option to be available.**

Performance degradation VMs tolerate	<u>100</u> %
Percentage of performance degradation the VMs in the cluster are allowed to tolerate during a failure. 0% - Raises a warning if there is insufficient failover capacity to guarantee the same performance after VMs restart. 100% - Warning is disabled.	

Lastly, some vSphere HA options can be modified to fit your organizational needs. To do so, you can add advanced configurations in the vSphere Availability Advanced options section. I will share a few of the more common settings, but you can review the full list on pages 41-43.

- `das.isolationaddressX` is the ping IP address used to determine if a host is network isolated. You can have multiple addresses (0-9). The address used by default is the management network gateway address
- `das.heartbeatDsPerHost` is number of heartbeat datastores to use
- `das.slotcpuinmhz` defines maximum bound CPU slot size
- `das.slotmeminmb` defines the maximum bound memory slot size
- `das.isolationshutdowntimeout` specifies time a VM waits to power down (default is 300 secs)
- `das.ignoreRedundantNetWarning` ignores the 'no HA redundant network' warning message

## DRS

To get started using DRS in your cluster, you must ensure you meet the following requirements:

- vSphere Enterprise *Plus* license
- vCenter Server
- Shared storage
- vMotion network configured on each cluster host

Once enabled, you can select the automation level you'd like to use in your environment – fully automated, partially automated, or manual. For environments configured to be fully automated, VM migrations occur based on the configured Migration Threshold setting, which has five priority levels, each of which is discussed below, from more conservative to more aggressive:

- Priority level 1 means DRS will apply recommendations that must take place to satisfy cluster constraints, such as affinity rules or host maintenance
- Priority level 2 means DRS provides recommendations only when workloads are extremely unbalanced



- Priority level 3 is the default level and means DRS provides recommendations when workloads are moderately unbalanced
- Priority level 4 means DRS provides recommendations when workloads are fairly unbalanced. This setting is recommended for bursting environments
- Priority level 5 means DRS provides recommendations when workloads are even slightly unbalanced. This setting may generate frequent vMotions

Additional DRS configuration options:

- The **VM Distribution** setting allows you to distribute VMs evenly among all hosts in the cluster
- **Memory Metric for Load Balance** allows you to balance workload based on VM consumed memory instead of active memory
- **Predictive DRS** is a feature where vCenter works with vRealize Operations to proactively balance VMs based on predictive patterns

The screenshot shows the 'Edit Cluster Settings' dialog for 'Cluster-CO'. At the top, 'vSphere DRS' is enabled with a green toggle. Below it are four tabs: 'Automation', 'Additional Options' (selected), 'Power Management', and 'Advanced Options'. Under 'Additional Options', there are three settings:

- VM Distribution**:  For availability, distribute a more even number of virtual machines across hosts.
- Memory Metric for Load Balancing**:  Load balance based on consumed memory of virtual machines rather than active memory. This setting is only recommended for clusters where host memory is not over-committed.
- CPU Over-Commitment**:  Enable. Over-commitment ratio: 0 :1 (vCPU:pCPU)

At the bottom right, there are 'CANCEL' and 'OK' buttons.

DRS Clusters can be in one of three states: valid (green), overcommitted (yellow), and invalid (red).

Keep in mind the following behaviors when using DRS, affinity rules, and resource pools:

- When DRS is disabled, any configured affinity rules are lost
- When DRS is disabled, all resource pools are removed and only recoverable if you save a snapshot of them
- When adding a standalone host to a DRS Cluster or removing a host from the cluster, then that host's resource pools are removed



## Datastore Cluster

Most precursor Storage DRS information was shared in Objective 1.6. In this objective, I will cover some of the configurations and features SDRS offers.

Before creating Datastore Clusters, you must meet certain requirements. Since SDRS uses storage vMotion to move VM disks among cluster datastores based off set policies, you must first meet **storage vMotion requirements**:

- vSphere Standard license to support storage vMotion
- ESXi 5.0 or later
- Enough free memory on the ESXi host for storage vMotion
- Enough space on destination datastores
- Destination datastore is not in maintenance mode

To create Datastore Clusters, you must then meet the following requirements:

- Cannot use different datastore types (i.e. NFS with VMFS)
- Hosts attached to the datastore must be ESXi 5.0 or later
- The datastore cannot be in more than one vCenter data center
- Datastores must have homogenous hardware acceleration

Adding a datastore to a Datastore Cluster has similar requirements:

- The host attached to the datastore must be ESXi 5.0 or later
- The datastore cannot be in more than one vCenter data center

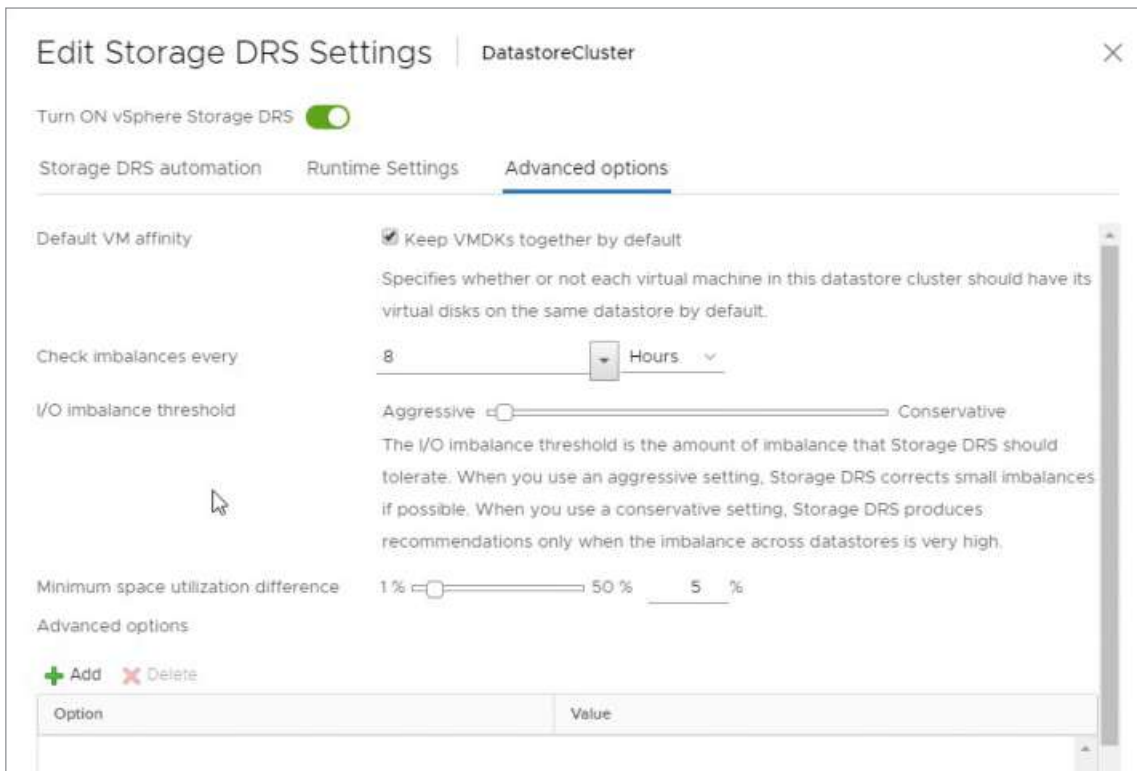
To enable SDRS, you create a new Datastore Cluster at the vCenter data center object level, selecting Actions > Storage > New Datastore Cluster, then configure each setting that meets your organizational needs and requirements.

If needing to do maintenance on Cluster datastores, keep in mind the following when needing to place a datastore in maintenance mode:

- SDRS must be enabled on the cluster the datastore is in
- No CD image files can be stored on the datastore
- There are at least two datastores remaining in the SDRS Cluster

Lastly, I'll discuss just a few advanced options you can set, depicted by the accompanied screenshot:

- VM disk affinity allows you to keep all disks associated with a VM together or on separate datastores
- Interval at which to check datastore imbalance
- Minimum space utilization difference refers to the % difference between source and destination datastore. For example, if the current space usage on source is 82% and destination is 79% and you configure "5", the migration is not performed



## Objective 1.10: Describe Virtual Machine file structure



## Resource:

### Virtual Machine file types

Virtual Machines are comprised of the following file types:

- **.vmx** – file containing parameters for all of a VM configuration's settings
- **.vmxf** – additional configuration file
- **.nvram** – stores the VM BIOS or EFI information
- **.log** – VM log file
- **.vswp** – memory swap file only available when a VM is powered on
- **.vmss** – a file that stores the VM state when in suspended mode
- **.vmsd** – a snapshot file
- **.vmsn** – the snapshot data file
- **.vmdk** – descriptor file containing characteristics for each VM virtual disk
- **.vmtx** – file annotating a VM is a template; replaces the .vmx
- **flat.vmdk** – a data disk file
- **delta.vmdk** – a snapshot data disk file
- **.ctk** – file used for changed blocked tracking for backup solutions



[vSphere 6.7 Virtual Machine Administration Guide, Update 2 \(April 2019\)](#), pg. 13



## Objective 1.11: Describe vMotion and Storage vMotion technology



### Resource:



[vCenter Server and ESXi Host 6.7 Management Guide, Update 2 \(April 2019\)](#), pages 134-146; 147-149

### vMotion

Before I get into the details of vMotion, I'll first start with cold migration. Cold migration simply means movement of a VM between ESXi hosts across clusters, data centers, or even vCenter Server instances while a VM is either in the powered-off or suspended state. If your application can afford a brief downtime and to have the destination host checked for fewer compatibility requirements than using vMotion, this may be a valid option for you. When a powered-off VM is migrated, if the VM guest OS is 64-bit, the destination host is checked for guest OS compatibility only. No other CPU-compatibility checks are performed. If migrating a suspended VM, the destination does need to meet CPU compatibility checks.

When performing cold migration:

- If performing cold storage migration, the VM `.vmx`, `.nvram`, `.log.`, and `.vmss` (suspended) files are moved to the target datastore, as well as VM disks. Something to be aware of when performing a cold migration – VM migration data (called provisioning traffic) uses the management network for cold migrations. If you need to perform multiple VM cold migrations, and often, it's recommended to isolate this traffic using a VMkernel network adapter for provisioning traffic and isolate the traffic on a dedicated VLAN or subnet
- The VM is then registered on the target host
- Once migrated, the source VM (and disks, if performing with storage migration) are deleted

**vMotion** refers to the process of migrating a VM from a source to destination ESXi host while powered-on, migrating the entire state of the VM. Like cold migration, you can choose one of two vMotion types – compute migration, storage migration, or both. The state information transferred refers to the VM memory content and all information identifying and defining the VM like BIOS, device, CPU, MAC address, chipset state information, etc. Unlike cold migration where compatibility checks for a destination host may be optional, the checks for vMotion are required. During vMotion, the following stages occur:

- vCenter verifies the VM is in a stable state with the source host
- The VM state (contents shared above) is moved to the destination host
- The VM resumes normal operation on the destination host

Before using vMotion, make sure to meet vMotion requirements:

- vSphere Standard license, Enterprise *Plus* for cross-vCenter vMotion
- Shared storage among hosts
- vMotion network configured on each host
  - Ensure each host is configured for a VMkernel adapter for vMotion
  - If using standard switch (vSS), make sure network labels match across hosts
  - Ensure each host uses a dedicated network for vMotion traffic





- Ensure the network used for vMotion has at least 250 Mbps bandwidth
- Best practice to use one pNic as failover
- Best practice to configure jumbo frames for best performance

You can perform several types of vMotion migrations, as listed below:

- **vMotion** for migrating VM compute to a destination host
- **Long-distance vMotion** for long distances if latency < 150ms
- **vMotion to another data center**
- **vMotion to another vCenter Server** starting from vSphere 6.0 and later; vCenter Servers must be in Enhanced-Linked Mode; following compatibility:
  - MAC addresses must be compatible between vCenters
  - Cannot migrate from a vDS to vSS
  - Cannot migrate from differing vDS versions
  - Cannot migrate from internal network, i.e. no pNic
  - Cannot migrate if vDS not working properly
- **Storage vMotion** VM disks must be in persistent mode or virtual RDMs

Starting with vSphere 6.5, vMotion always uses encryption for migrations. Cross-vCenter vMotion has the following supportability:

- Using vMotion encryption with unencrypted VMs to another vCenter is supported
- Using vMotion encryption with encrypted VMs to another vCenter is not supported because one vCenter cannot verify the other vCenter KMS
- For local vMotion, encrypted VMs always use encrypted vMotion and this function cannot be disabled
- For local vMotion, an unencrypted VM can use one of three states:
  - Disabled: do not use encrypted vMotion, even if available.
  - Opportunistic: use vMotion if source and destination hosts support it, fall back to unencrypted vMotion otherwise. This is the default option.
  - Required: allow only encrypted vMotion. If the source or destination host does not support vMotion encryption, do not allow the vMotion to occur.
- To configure encryption, go to VM settings > VM options tab, Encryption section and select the option best suited for your organization

Be aware of some vMotion limitations and considerations (full list on pg. 138):

- Source and destination host IPs must be same family, i.e. all IPv4 or IPv6
- You can vMotion VMs with host-connected USBs if enabled for vMotion
- You cannot vMotion VMs using source host devices not accessible to the destination host, i.e. physical CD drive
- You cannot vMotion with VMs connected to client devices

To improve vMotion compatibility between source and destination hosts, you can mask certain CPU feature sets to the VM by enabling **Enhanced vMotion Compatibility (EVC)**. If there is a mismatch between source and destination host



user-level, or kernel-level CPU features, migration will not be allowed. User-level features include SSE3, SSSE3, AES, etc. and kernel-level features include AMD-NX or Intel-XD security. When using EVC, you configure all cluster hosts processors to present the feature set of a baseline processor, called EVC mode, to mask certain CPU features so hosts can present the feature set of an earlier processor. EVC only masks CPU feature sets affecting vMotion compatibility. To enable EVC, hosts must meet the following requirements:

- Check CPU compatibility on the HCL
- ESXi 6.0 or later
- vCenter Server
- A single processor vendor used, Intel, or AMD
- Configure hosts for vMotion
- Enable BIOS features if available:
  - Intel-V or AMD-VT hardware virtualization
  - Interl-XD or AMD-NV
- Power off VMs or migrate out of the cluster. Be aware when changing EVC in a cluster, as a VM does not use the new feature set until it is powered-off then back on (not restarted)

In Objective 1.1 I provided vCenter vMotion limits but will provide a quick review here. If using one GbE pNICs, you can have four simultaneous vMotions; if using a 10 GbE pNIC, you can have eight; a total of 128 simultaneous vMotions can occur on each datastore.

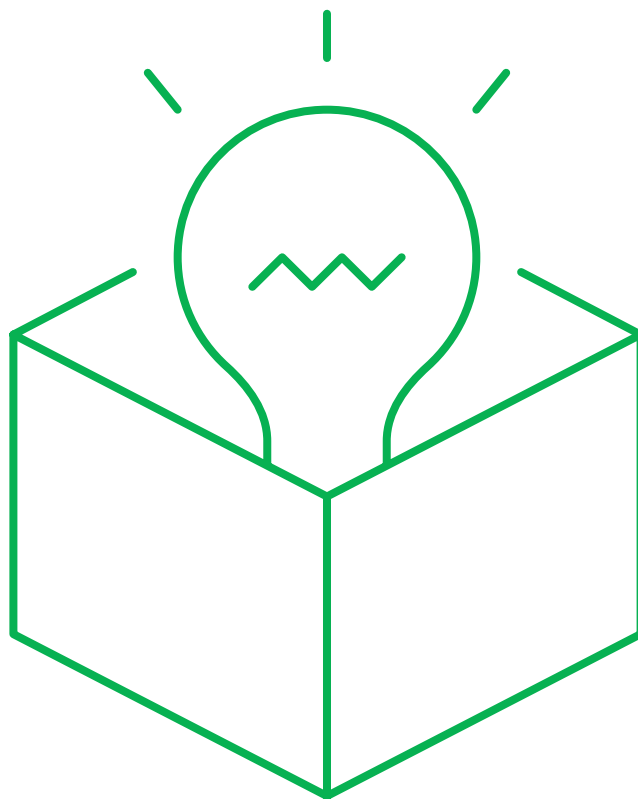
### Storage vMotion

Storage vMotion and its requirements were discussed in the Datastore Cluster section of Objective 1.9, so I will not cover them again here. Return to Objective 1.9 for a quick review if needed. I will just add what was not shared in that objective.

To use encryption with storage vMotion, be aware of the following supportability: If disks are encrypted, data transmitted is encrypted.

As mentioned in the vMotion section, I covered storage vMotion limits in Objective 1.1, but will also provide a quick review here. You can have only two simultaneous storage vMotion migrations per host and eight total per datastore.

Operation	ESXi Version	Derived Limit Per Host
vMotion	5.0, 5.1, 5.5, 6.0	8
Storage vMotion	5.0, 5.1, 5.5, 6.0	2
vMotion Without Shared Storage	5.1, 5.5, 6.0	2
Other provisioning operations	5.0, 5.1, 5.5, 6.0	8



## SECTION 2: VMware product and solutions

Objective 2.1: Describe vSphere integration with other VMware products . . . . .	43
Objective 2.2: Describe HA solutions for vSphere . . . . .	48
Objective 2.3: Describe the options for securing a vSphere environment. . . . .	53



## Objective 2.1: Describe vSphere integration with other VMware products

I believe what VMware is asking here is how the main components of vSphere (ESXi and vCenter Server) integrate with its other products, such as NSX, vRealize, Site Recovery Manager, VMware Horizon Suite, and vSphere Replication. I will certainly cover those from a high-level, but I feel it's also important to understand how vSphere components behave with certain vSphere services. Interoperability, if you will. So, I will start this objective covering how vSphere "integrates" with certain features and services before discussing other VMware products.

### vSphere interoperability (HA, DRS, vSAN, Fault Tolerance)

#### When using Affinity required ("must run on" or "must not run on") rules:

- **vSphere HA** does not perform: VM failovers
- **DRS** does perform:
  - VM migration when a host is entering maintenance mode
  - DRS does not power-on VMs (initial placement) or migrate (balance cluster load)
- **DPM** does not perform: Power optimization management
- **FT**: It is recommended to use Affinity rules with Fault Tolerance to keep the primary VM and the secondary VM on separate hosts

#### When using DRS

FT is supported when using vSphere 6.7. If you run FT VMs on 6.0 or 6.5 ESXi hosts, they must be managed by vCenter 6.7.

#### When using vSAN

FT is supported but recommended to use different networks (for vSAN network and FT logging). Also, keep the primary VM and secondary VM in separate fault domains.

### VMware Enterprise PKS

VMware Enterprise PKS is a Kubernetes-based container solution with advanced networking, a private container registry, and life-cycle management simplifying deployment and operation of Kubernetes clusters and allowing businesses to provision, operate, and manage Enterprise-grade Kubernetes clusters on private or public clouds at scale using BOSH and Pivotal Ops Manager.

### vSphere integration with NSX

Before delving into the two NSX types, let me first briefly talk about what NSX is in general. NSX is the result of VMware's acquisition of Nicira in 2012 and its software-defined networking solution, the Network Virtualization Platform (NVP). VMware NSX is a software-defined networking solution allowing businesses to solve complex problems surrounding security, automation, and agility in today's hybrid cloud data centers. With NSX, you have the functional equivalent of a network hypervisor, allowing virtual provisioning of network services layers L2-L7, including switching, routing, access control, firewall, QoS, and others.



## Resource:



[vSphere NSX-T Data Center Installation Guide v3.0 \(May 2020\)](#)

vRealize Suite Installation Guides (Automation Install, Orchestrator Install, Operations Manager Install, and Log Insight User Guide)

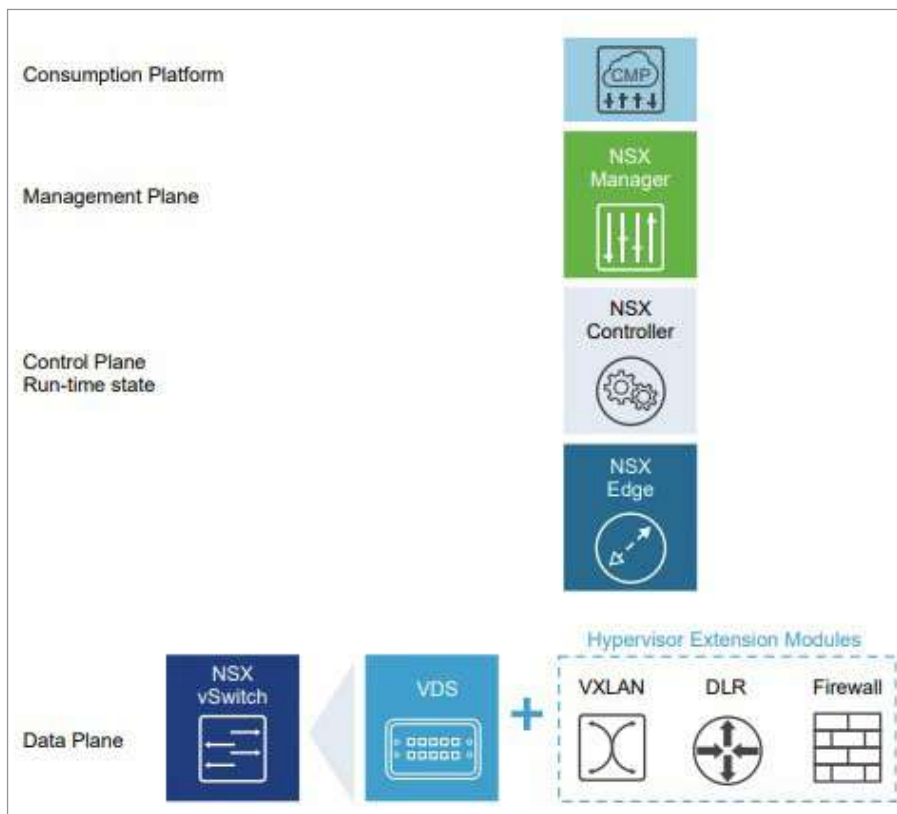


NSX provides four benefits to businesses who are looking to modernize their networking environment:

- **Micro-segmentation** helps businesses implement a "zero-trust" model of allowing only known traffic sources and disallowing everything else and enforcing network security policies consistently on any workload in their environment.
- **Multi-cloud networking** allows businesses to better meet their needs by spanning cloud providers, as well as private cloud infrastructure, to help with scale, agility, and flexibility.
- **Network automation** allows businesses to streamline IT network operations of provisioning, configuration, and monitoring of their NSX environment.
- **Cloud-native apps** is simply a term to describe a new way organizations are delivering their business applications via containers. NSX allows organizations to apply consistent network policies and rules, no matter what platform their applications run on.

NSX operates within three different planes, each of which I describe below:

- **Management plane** provides an API entry point to NSX for configuring, querying, performing operational tasks, and is built by the NSX Manager, the centralized network managing component of NSX Data Center





- **Control plane** runs the NSX Controller cluster, and the advanced distributed state management system providing functions for logical switching and routing. It also maintains information about all hosts, logical VXLAN switches, and distributed logical routers
- **Data plane** consists of the NSX Virtual Switch, based on the vDS, to enable services such as kernel modules, user space agents, configuration files, etc. packaged as VIBs to run within the hypervisor kernel to provide logical firewall and distributing routing capabilities

Lastly, the difference in the VMware NSX types (V and T) are: NSX-V is VMware's original NSX solution, based on their native hypervisor, vSphere. It has since been rebranded as NSX Data Center for vSphere. NSX-T is VMware's newest version of NSX and is geared towards the multi-hypervisor, multi-cloud business model. It has been rebranded as NSX-T Data Center. You now are not restricted to just VMware's product line to use this powerful software-defined network solution.

### vSphere integration with vRealize Suite

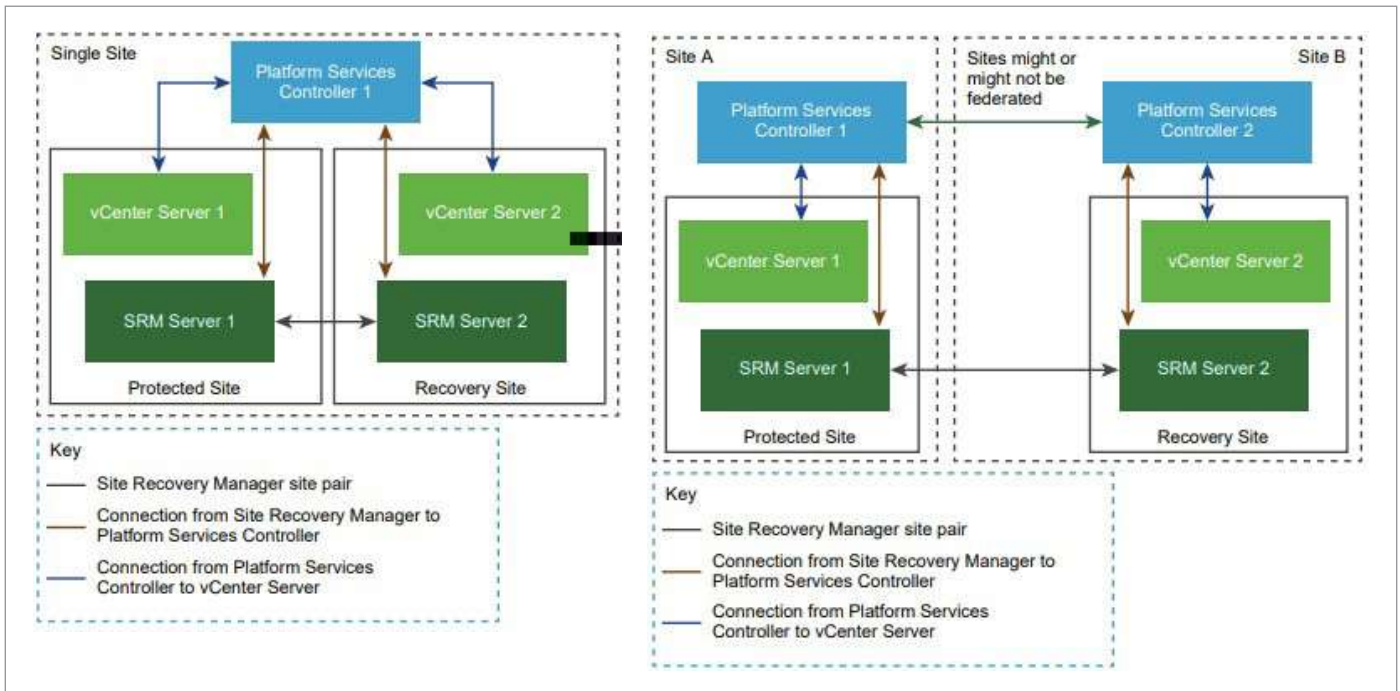
vRealize Suite is VMware's product line for provisioning and on-going management of infrastructure and applications across vSphere, as well as other hypervisors and platforms (i.e. multi-hypervisor and multi-cloud).

- vRealize Automation (vRA) enables administrators to automate delivery and lifecycle management of infrastructure and application services. It is a way to provide a secure portal where authorized users can request IT services, allowing businesses to manage their cloud and IT services while ensuring compliance with business policies.
- vRealize Orchestrator (vRO) is a development and process-automation platform providing a library of extensible workflows, allowing businesses to create and run automated, configurable processes to manage vSphere and other third-party technologies. vRO automates management and operational tasks such as service desks, change management systems, and IT asset management systems.
- vRealize Operations Manager (vROM) is a VMware solution allowing businesses to monitor and respond to alerts and issues in their environment, as well as address performance issues.
- vRealize Log Insight is a scalable log aggregation and indexing solution that collects, imports, and analyzes logs to help provide answers to system, service, and application problems within an environment.

### vSphere integration with Site Recovery Manager (SRM)

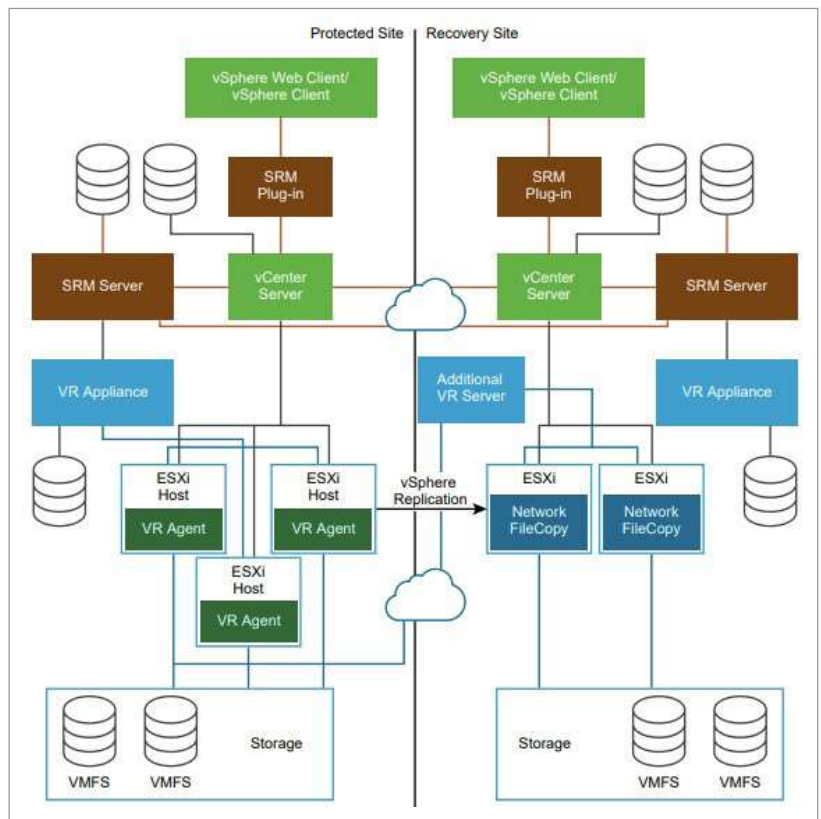
SRM is VMware's backup, disaster recovery, and business continuity tool. With it, you can plan, test, and run recovery of VMs between a protected vCenter Server site and a recovery vCenter Server site. SRM allows the use of either an appliance or Windows-based central management, deployed at both sites. You can utilize one of a few deployment models – a single-site or two-site deployment model, as shown on the next page, respectively.

You can use this solution in conjunction with VMware's vSphere Replication, as discussed next.



### vSphere integration with vSphere Replication

vSphere Replication is an extension of vCenter Server that provides a hypervisor-based VM replication and recovery. You can use this solution in place of storage replication to replicate VMs to a remote site to protect them against partial or complete site failure. You can configure point-in-time (PiT) snapshots of VMs, a maximum of 24, to best fit your organizational needs. You can combine this solution to use with SRM for a wholistic DR solution. For implementation, vSphere Replication requires an appliance deployment in both the source and recovery sites. After deployment, and requirements have been met, you can begin configuring replication protection for your VM workloads. Not only can you configure replication to a recovery site, but you can also perform replication back to the source site for bi-directional replication. On the right is a depiction of using vSphere Replication with SRM:



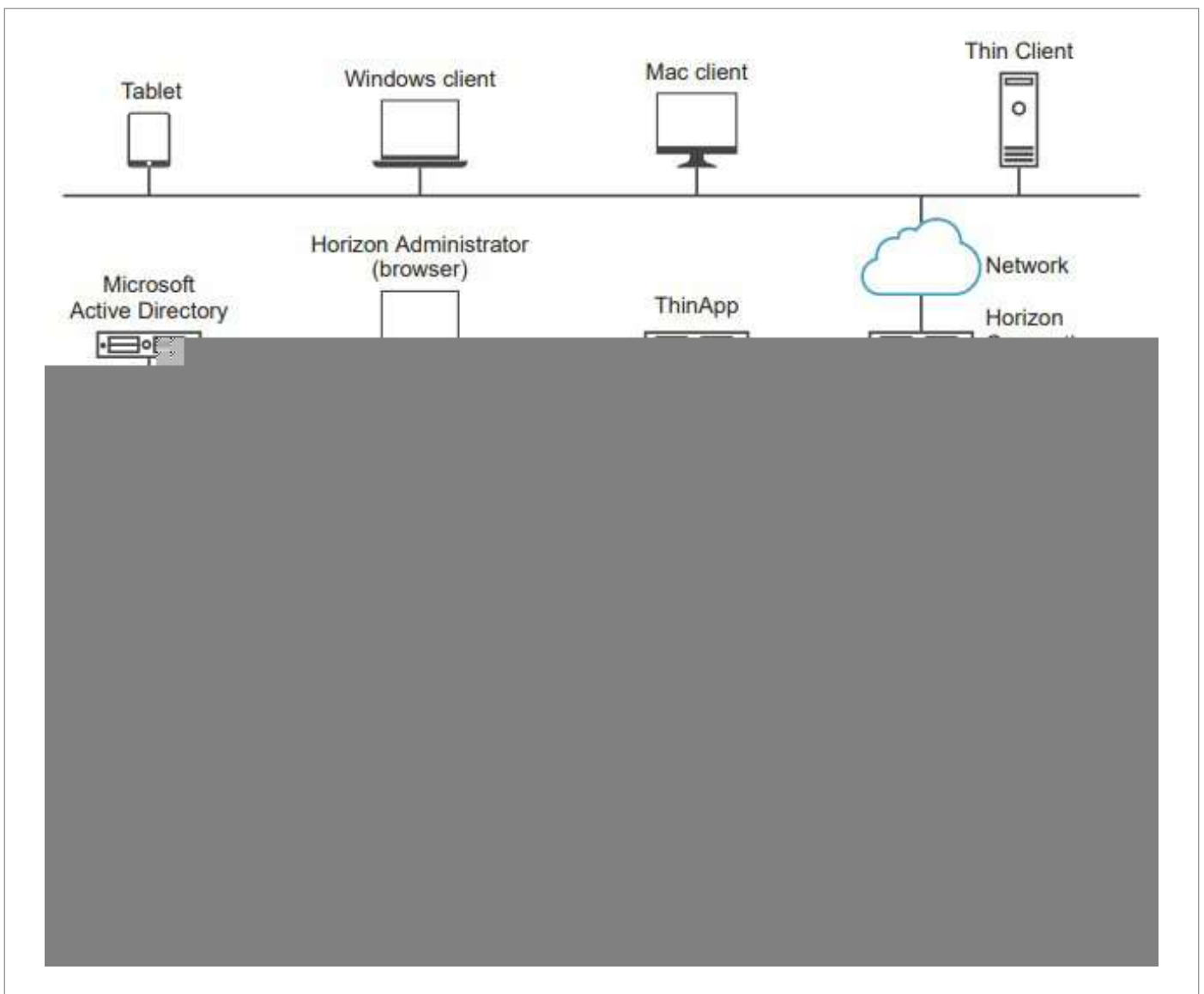




## vSphere integration with VMware Horizon

VMware Horizon allows IT departments to run remote desktop and applications in the data center and deliver them to users as a managed service. Users get a personalized, familiar environment they can access from several devices from within the organization or at home, and IT administrators have centralized management and control, efficiency, and security having those desktop workloads residing in the data center. Some advantages of using Horizon include:

- Reliability and security by granting access where access is needed and restricting access where it's not; AD-integration, RADIUS support
- Convenience by having a unified management console built to scale
- Manageability by providing centralized control of multiple solutions
- Hardware independence – run desktops/apps from virtually any device
- Below shows how all of Horizon's components fit together:







## Objective 2.2: Describe HA solutions for vSphere



### Resource:



[vSphere 6.7 Availability Guide, Update 2 \(April 2019\)](#)

### vSphere High Availability (HA)

In Objective 1.6, I discussed in great detail how vSphere HA works. I will provide a quick review here. When vSphere HA is enabled on a vSphere Cluster in vCenter, the HA Agent (.fdm) is installed on each ESXi host. From there, an election process is undergone to determine which ESXi host becomes the master. The host with the most datastores connected to it has the advantage of becoming the master and then all other hosts are subordinates. The master host has several responsibilities:

- Monitoring the state of subordinate hosts
- Monitor power-state of protected cluster VMs
- Manage a list of hosts and protected VMs
- Reports cluster health state and acts as vCenter management interface to the cluster

The master monitors the subordinate hosts every second through network heartbeats, which occurs through the management network, or the vSAN network if vSAN is used. If heartbeats are not received through this method, before the master determines a subordinate host has failed, the master detects the liveness of the subordinates by seeing if the subordinate host is exchanging datastore heartbeats. If there is no liveness detected, the subordinate host is determined to have failed in one of three states:

- Failed – Host stops functioning; master cannot communicate with subordinate HA agent and subordinate not responding to pings
- Isolation – Host is still running but unable to communicate through the agent on the management (HA) network. If the host is also unable to ping a designated isolation address (management gateway by default), it is then determined to be isolated
- Partition – Host loses network connectivity to master

### vCenter Server High Availability (VCSA HA)

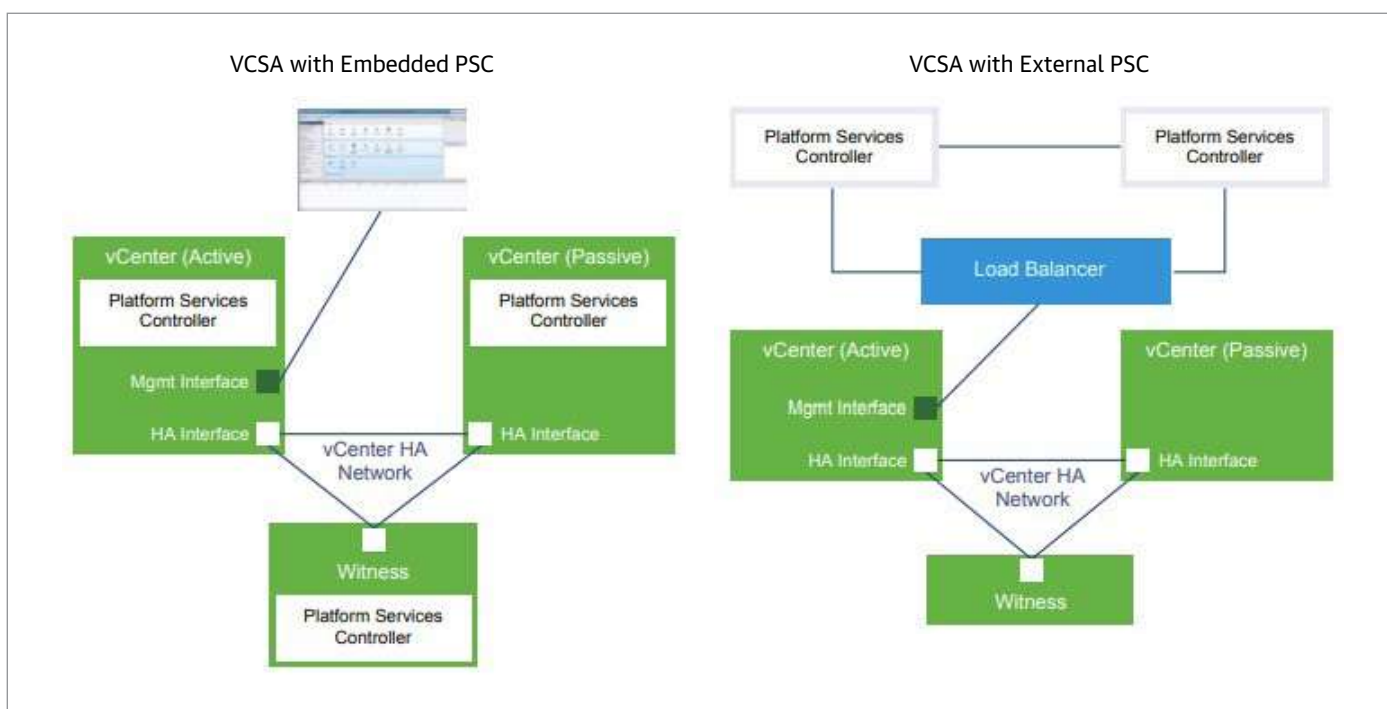
In Objective 1.2 I provided requirements needed before enabling vCenter HA. In this objective I will cover the concept behind vCenter HA in a bit more detail. Simply put, vCenter HA protects the vCenter Server Appliance (VCSA) against host and hardware failure, as well as reduce downtime due to VCSA patching.

After configuring your network for vCenter HA, you implement a three-node cluster comprised of Active, Passive, and Witness nodes.

- The **Active node**:
  - Uses the active vCenter
  - Contains the public IP
  - Uses the vCenter HA network to replicate data to the Passive node
  - Uses the vCenter HA network to communicate with the Witness node
- The **Passive node**:
  - Is initially a clone of the Active node

- Receives updates from and synchronizes state with the Active node
- Automatically takes over Active node role in the event of a failure
- The Witness node is solely a lightweight clone of the Active node protecting against a split-brain scenario

Your vCenter HA deployment can be slightly different when using external PSC vs running VCSA with embedded PSC. When using external PSC, you need to implement a load balancer. View the architectural differences below:



You can implement VCSA HA in one of two ways:

- **Basic configuration** requires one of two environment VCSA setups to implement: the VCSA is managing its own ESXi host and VM (i.e. self-managed vCenter) or the VCSA is managed by another vCenter Server (management vCenter) and both are in the same SSO domain (i.e. v6.5). The workflow for Basic is as follows:
  - User deploys the VCSA that becomes the Active node
  - User adds a second network port group used for vCenter HA on each host
  - User starts vCenter HA process, selecting Basic and supplying IP address, target host or cluster, and datastore for each clone
  - The system automatically clones the Active node as a Passive node, with the same network and hostname settings and adds a second vNic to each node
  - The system clones the Active node as a Witness node
  - The system sets up the vCenter HA network the nodes communicate on

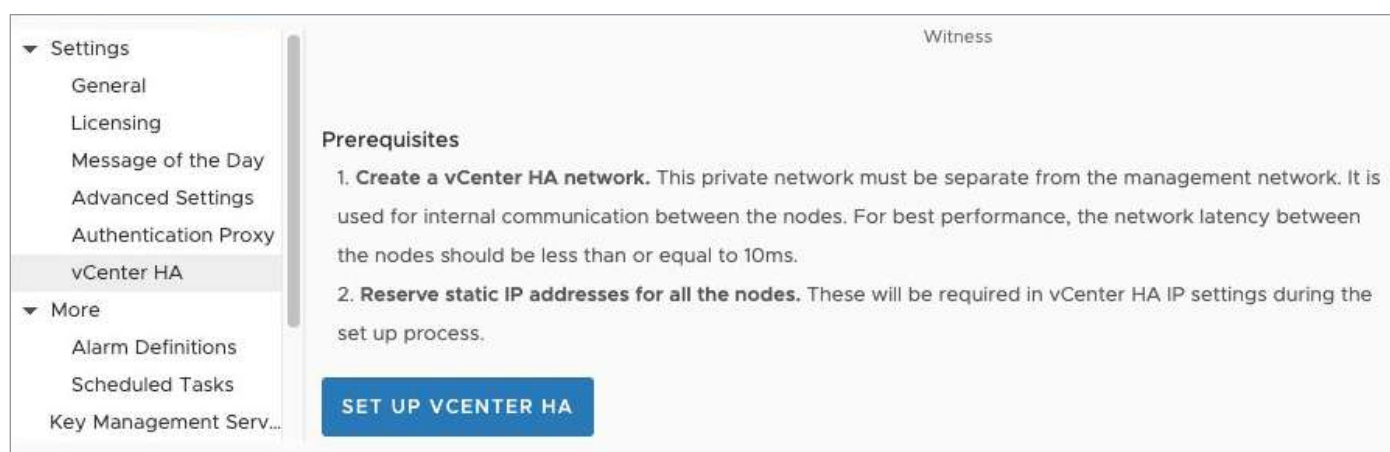


- **Advanced configuration** allows more control over the deployment. With this option, you must clone the Active node manually. And if you remove vCenter HA, you must delete all nodes you created. The Advanced workflow is as follows:
  - User deploys the VCSA that becomes the Active node
  - User adds a second network port group used for vCenter HA on each host
  - User adds a second vNic to the Active node
  - User logs in to the VCSA with vSphere Client
  - User starts vCenter HA process, selecting Advanced, and supplying IP address and subnet information for the Passive and Witness nodes
  - User logs into the management vCenter and creates two clones of the Active node
  - User returns to the vCenter HA configuration to finish the process
  - The system sets up the vCenter HA network the nodes communicate on

vCenter HA uses two networks on the VCSA:

- **Management network** configured with a static IP, serves client requests
- **vCenter HA network** connects the three nodes and replicates the appliance state, as well as monitors heartbeats. This network
  - Must have a static IP
  - Must be on a different subnet than the management network
  - Must have network latency < 10ms between all three nodes
  - Not add a default gateway for the cluster network

To configure vCenter HA, click on the **vCenter Server** in the vSphere Client > **Configure** tab, then select **vCenter HA** under the Settings section. Choose the desired deployment type, Basic or Advanced, by deselecting the option to **Automatically create clones for Passive and Witness nodes**, then follow the workflow described above:



After setting up your vCenter HA environment, you can initiate a failover to verify the Passive node takes over as Active. From the vCenter HA screen, click to Initiate Failover to begin the process.



You can also place the vCenter HA Cluster in maintenance mode or even disable it. In vCenter HA settings, select Edit then choose any of the shown options – Enable, Maintenance Mode, Disable, or even Remove vCenter HA cluster.

Option	Result
Enable vCenter HA	Enables replication between the Active and Passive nodes. If the cluster is in a healthy state, your Active node is protected by automatic failover from the Passive node.
Maintenance Mode	In maintenance mode, replication still occurs between the Active and Passive nodes. However, automatic failover is disabled.
Disable vCenter HA	Disables replication and failover. Keeps the configuration of the cluster. You can later enable vCenter HA again.
Remove vCenter HA cluster	Removes the cluster. Replication and failover no longer are provided. The Active node continues to operate as a standalone vCenter Server Appliance. See <a href="#">Remove a vCenter HA Configuration</a> for details.

If you need to shut down the cluster nodes for any reason, they must be done in certain order:

- Passive node
- Active node
- Witness

They can then be started up in any order.

Lastly, you can back up the Active node for added protection against failure. Keep in mind that before restoring the Active node, if secondary nodes are still present, you need to remove them, as not doing so may result in unpredictable behavior.

## vSphere Fault Tolerance (FT)

Virtual machine Fault Tolerance can be used to ensure continuity and higher levels of availability for mission critical VMs, built on ESXi to provide HA using an identical copy of a VM on a separate host. The protected VM is called the primary and the duplicate is called the secondary. The primary continuously replicates its state so the secondary can take over in the event of a host failure. It is worth noting again that this type of High Availability is only for when an ESXi host on which the primary VM fails. This is not a backup solution since all data written to the primary VM gets replicated to the secondary. So, if the primary VM gets corrupted, the secondary will inherit the same corruption by way of the FT replication that occurs between the two.

Some use cases for FT can be when users need to maintain their session state to critical, highly used VMs. In the event of a host failure the primary VM resides on, failover to the second happens instantaneously and is transparent to the users. Another use case can be for applications that have no other means to do clustering or for applications that do provide a clustering solution but are too complex to deploy. Lastly, you can use FT on-demand when applications have critical use



(i.e. seasonal) periods.

Before implementing FT, verify that the following requirements are met:

- vSphere Standard license supports two vCPUs; Enterprise *Plus* license supports eight vCPUs
- Host CPUs must be compatible with vMotion
- Host CPUs to support hardware MMU, Intel Sandy Bridge or later, or AMD Bulldozer or later
- 10 Gb logging network for FT
- No more than four FT VMs and no more than eight vCPUs per host are allowed
- vSphere HA cluster created
- vMotion network configured on each host

When preparing to enable FT for a VM, the following validation checks are performed:

- SSL certification checking enabled in vCenter Server settings
- The ESXi host the VM is on is in a HA or HA/DRS cluster
- ESXi 6.0 or later
- The VM does not have snapshots
- The VM is not a template
- The VM does not have vSphere HA disabled
- The VM does not have a 3D video device enabled

To enable FT, right-click the VM > Fault Tolerance > Turn on Fault Tolerance. If the option is dimmed, the following reasons are why this may be the case: no license, host is in maintenance mode, VM is disconnected or orphaned, or no permissions to enable it.

When using Fault Tolerance, the following vSphere features are not supported:

- Snapshots
- Storage vMotion
- VVols
- Linked clones
- SPBM
- I/O filters
- Disk encryption

The following features and devices are not supported:

- VMDK greater than two TB
- Physical RDM
- CD-ROM or floppy backed by remote physical device
- USB
- NPIV
- 3D video devices



## Objective 2.3: Describe the options for securing a vSphere environment



### Resource:



[vSphere 6.7 Security Guide, Update 2 \(April 2020\)](#)

### Securing ESXi

Below are some best practices, recommendations, and security features to be aware of with respect to securing ESXi hypervisors:

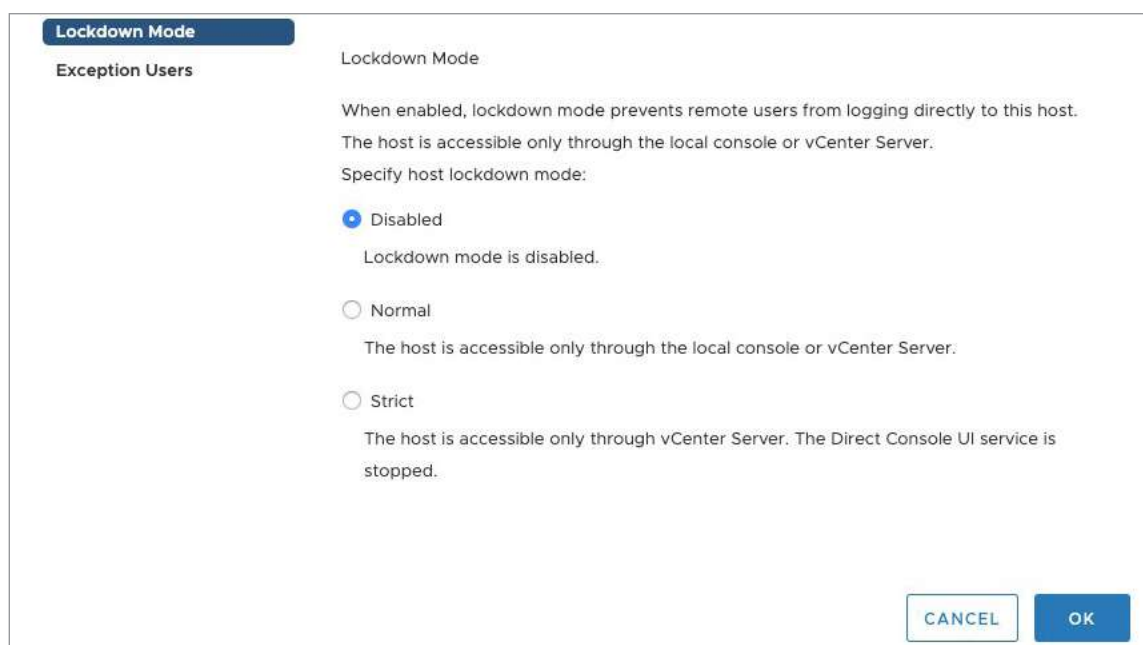
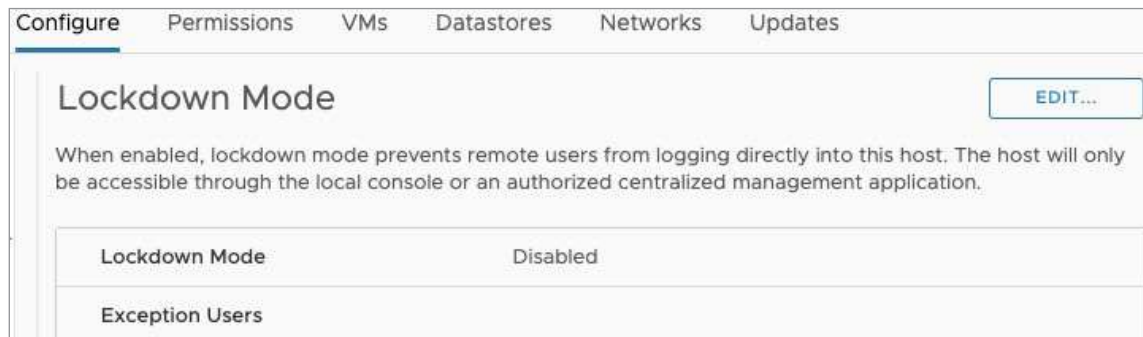
- User accounts and least privileges – do not let administrators log in to hosts using root. Instead create user accounts and grant permissions needed (least privilege). For ease of user management, you can connect hosts to a directory service, like Active Directory, by selecting your **Host > Configure** tab, **System** section > then **Authentication Services** and choose **Join Domain**
- Configure a user lockout policy that best fits your IT department's needs. By default, users are locked out after 10 failed login attempts and reset after two minutes
- ESXi passwords require a minimum of four character classes – lowercase, uppercase, numbers, special characters; the password length must be more than seven (i.e. eight) and less than 40; and cannot contain dictionary words
- A minimum amount of firewall ports are open after ESXi installation and ESXi runs only services essential for it to function. To disable unneeded ports and services, select your **Host > Configure** tab, **System** section > then choose either Firewall or Services

Service name ↑	TCP ports	UDP ports	Allowed IP addresses
Connection: Incoming			
CIM Secure Server	5988	--	All
CIM Server	5988	--	All
CIM SLP	427	427	All
DHCP Client	--	68	All
DHCPv6	546	546	All
DVSSync	--	8301, 8302	All
Fault Tolerance	8300	--	All
lofiltervp	9080	--	All
NFC	902	--	All
SNMP Server	--	161	All
SSH Server	22	--	All

- Automate host management where possible. This also ensures a consistent environment and eases troubleshooting knowing a configuration change is made from a single location (the script or host profile)



- Enable lockdown mode if possible but be careful with this feature, as you could lose access to your host if not utilized correctly. To enable, select your **Host** > **Configure** tab, **System** section > **Security Profile**



Lockdown mode behavior when enabled:

- **Normal lockdown mode** – Privileged users can access vCenter through the vSphere Web Client or SDK; users who are on the Exception Users list who have administrative access can log in to the DCUI or any user in the `DCUI . Access` ESXi host Advanced setting can access the DCUI
- **Strict lockdown mode** – As with normal mode, privileged users can access vCenter through the Web Client or SDK; DCUI access is disabled for all users
- For SSH or Shell access, when in lockdown mode, users with administrative permissions in the Exception Users list or who are in the `DCUI . Access` Advanced setting can use these services. To add users to this group, you first create a local host user or you can use an AD user, then from the **Host** > **Configure** tab > **System** section, select **Advanced System Settings** search for `DCUI . Access`, and then type in the user





- Manage host certificates using VMCA, using one of three modes – VMCA (default), Custom for use with third party or Enterprise CAs, or Thumbprint as a fallback for ESXi 5.5 hosts
- SSH and Secure Shell services are disabled by default; SSH v1 is not supported. By default, root and any users assigned an administrator role can access the ESXi Shell, as well as AD users in the ESX Admins group; but only root can run system commands
- Use UEFI secure boot, if enabled in the host hardware BIOS, to boot only if the OS bootloader is cryptographically signed. This feature was first released in vSphere 6.5

## Securing vCenter Server

Some security features of vCenter Server are similar to ESXi host security. For example, configure user permissions with least privileges and do not use administrator accounts. Instead, create an account and assign only needed permissions. To control user account management, it is recommended to use a Directory Service like Active Directory. In addition, implement the following vCenter Server security measures:

- Minimize datastore access of the Datastore.browser datastore privilege
- Minimize guest OS VM access of the Guest Operations privilege
- Change the vpxuser password occurrence, if needed; default is 30 days. If needing to change, go into vCenter Advanced settings and set the VirtualCenter.VimPasswordExpirationInDays parameter
- Perform standard Windows OS patching, A/V installation, and RDP encryption for vCenter on Windows deployments

Below is a list of common task permissions you should be aware of:

Task	Privileges	Minimum default user
Create VM	Destination folder or data center: (several privileges) Destination host/cluster/VP: <b>resource.AssignVMtoResourcePool</b> Destination DS: <b>datastore.AllocateSpace</b> Assigning network to VM: <b>network.AssignNetwork</b>	Folder/DC: Administrator On host, etc.: RP administrator Destination DS: Datastore consumer to assign network: network administrator
Deploy VM from template	Several privileges for destination folder or DC, on template, on destination host/cluster/VP, destination DS, and network	Administrator (except datastore and network)
Take snapshot	Source VM/folder: <b>virtualMachine.SnapshotMgmt.CreateSnap</b> Destination DS or DS folder: <b>datastore.AllocateSpace</b>	On VM: VM power user Destination DS: Datastore consumer
Move VM into resource pool	Source VM/folder: <b>resource.AssignVMtoResourcePool</b> <b>VirtualMachine.Inventory.move</b> Destination VP: <b>resource.AssignVMtoResourcePool</b>	On VM: Administrator Destination VP: Administrator
Install guest OS	Source VM: (several privileges) <b>Datastore with ISO: datastore.BrowseDatastore</b>	On VM: VM power user On datastore with ISO: VM power user
Migrate VM with vMotion	Source VM or folder: <b>resource.migratePoweredOnVM</b> Destination host/cluster/VP (if different than source): <b>resource.AssignVMtoResourcePool</b>	On VM: Resource pool administrator Destination: Resource pool administrator





Task	Privileges	Minimum default user
Cold migrate VM	Source VM or folder: <b>resource.migratePoweredOffVM</b> Destination host/cluster/RP (if different than source): <b>resource.AssignVMtoResourcePool</b> Destination datastore (if different than source): <b>datastore.AllocateSpace</b>	On VM: Resource pool administrator Destination: Resource pool administrator Destination DS: Datastore consumer
Migrate VM with svMotion	Source VM or folder: <b>resource.migratePoweredOnVM</b> Destination datastore: <b>datastore.AllocateSpace</b>	On VM: Resource pool administrator Destination: Datastore consumer
Move host into cluster	Source host: <b>host.inventory.addHostToCluster</b> Destination host: <b>host.inventory.addHostToCluster</b>	On host: Administrator Destination cluster: Administrator

## Securing Virtual Machines

Use the following best practices when securing your VM workloads:

- Perform proper guest OS patching and security measures
- For VM creation, use templates for consistency and security
- Minimize VM console access
- Modify vCenter password requirements if needed; default requirements = length of eight, at least one lowercase, one number, one special character
- Limit VMware tools **VM.Interaction.VMware Tools install** privilege
- Limit the **VM.Interaction** and **VM.Configuration** VM-to-device communication privilege
- Monitor use of vSphere client plugins
- Remove unnecessary VM devices like serial, parallel, or floppy devices
- Configure VM encryption
- Avoid the use of independent, non-persistent virtual disks
- Use **Virtual Trusted Platform (vTPM)** to VMs to perform cryptographic co-processor capabilities in software, allowing a guest OS to store and create private keys. The keys are not exposed to the guest OS, so the attack surface is reduced. You can add vTPM to a new or existing VM. When backing up a vTPM VM, make sure to include the VM `.nvram` file or you cannot restore it.
- **Requirements for vTPM:**
  - vCenter 6.7
  - VM EFI and virtual hardware 14
  - VM encryption
  - KMS
  - Windows Server 2016 or Windows 7, 64-bit
- Some VM advanced security options to note:
  - Disable copy/paste in console: `isolation.tools.copy.disable`; set the value to `true` (add another line replacing `copy` with `paste`)



- Prevent sending configuration information to ESXi hosts:  
`tools.isolation.tools.setInfo.disable` set to `true`
- Set VM log count: `vmx.log.keepOld` set to desired number
- Set .vmx file size (default = 1MB): `tools.setInfo.sizeLimit=2048`
- Prevent virtual disk shrinking: `isolation.tools.diskWiper.disable` set to `true` and `isolation.tools.diskShrink.disable` set to `true`
- Prevent use of guest OS file system: `isolation.tools.hgfsServerSet.disable` ; `true`

## Securing vSphere networking

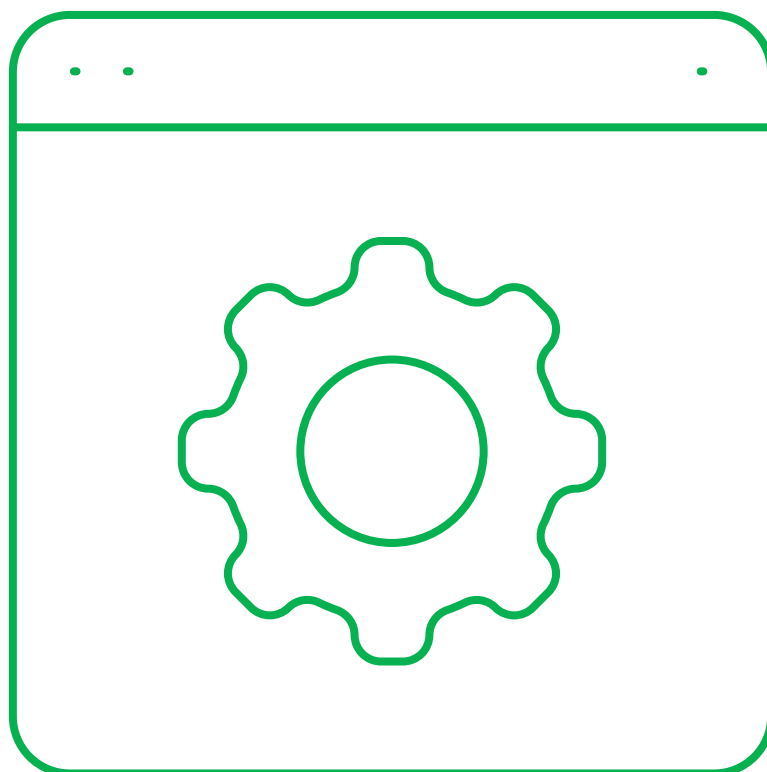
To secure your networking environment, implement segmentation, and isolation, either via the use of VLANs, multiple virtual switches, or dedicated physical NICs to different traffic types, subnets, or a combination of these. It also goes without saying, make sure access to your physical network environment is restricted, not just user access, but access to the physical devices too.

You can secure your virtual network environment further using vSS or vDS security policies, configured on the virtual switches or you can override the policies if needed for granular configuration on virtual switch port groups.

- Configure virtual LANs (VLANs) using one of three modes:
  - External Switch Tagging (EST)
  - Virtual Switch Tagging (VST) – use VLAN IDs 1-4095
  - Virtual Guest Tagging (VGT) – use VLAN ID 4095
- **Promiscuous mode** eliminates reception the VM adapter performs, allowing the guest OS to receive all traffic observed on the wire; is set to reject by default and recommended to keep it at this setting
- **Forged transmits** affects egress VM traffic, whereas the ESXi host verifies the source MAC of the guest with the effective MAC of the VM network adapter. Set to reject to protect against MAC impersonation
- **MAC address changes** to reject to protect against MAC impersonation. The only use case to have this enabled is if using Microsoft Network Load Balance (NLB)

Other networking security practices recommended are:

- If STP is used, configure it with Portfast
- Ensure Netflow traffic for use with a vDS is sent to an authorized collector IP
- Do not configure virtual switch port groups to use the native VLAN



## SECTION 4: Installing, configuring and setting up a VMware vSphere solution

Objective 4.1: Understand basic log output from vSphere products. . . . .	59	Objective 4.5: Configure virtual networking. . . . .	73
Objective 4.2: Create and configure vSphere objects. . . . .	61	Objective 4.6: Deploy and configure VMware vCenter Server Appliance (VCSA) . . . . .	76
Objective 4.3: Set up a content library . . . . .	66	Objective 4.7: Set up identity sources. . . . .	78
Objective 4.4: Set up ESXi hosts . . . . .	69	Objective 4.8: Configure an SSO domain. . . . .	80



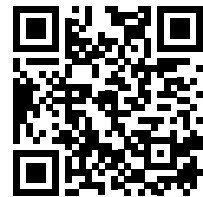
## Objective 4.1: Understand basic log output from vSphere products



### Resource:



[vSphere 6.7 Security Guide, Update 2 \(April 2020\)](#), pages 114-116



[Location of vCenter Server log files \(1021804\)](#)

### vSphere (ESXi) logs

To increase ESXi host security, it is recommended to redirect host logs to a datastore to persist upon reboots, as logs are by default stored on the in-memory file system and only 24 hours of log data is kept. It is also recommended, if available, to direct logs to a central host – a log management server – for more in-depth analysis, search capability, and troubleshooting.

To configure directing logs to a remote server, modify each ESXi Host Advanced settings parameter, selecting **Host > Configure** tab > **System** section and select **Advanced System Settings** and search for `Syslog.global.logHost`. Enter the URL in this format: `ssl://<hostName>:1514`. ESXi hosts store all its logs in the `/var/log` directory. Each log has the following functions:

- VMkernel – `vmkernel.log`; logs related to ESXi and VMs
  - Warnings – `vmkwarning.log`; logs VM-related activities
  - Summary – `vmksummary.log`; ESXi uptime and availability
- ESXi host agent – `hostd.log`; information about VM and ESXi configuration and management services
- vCenter agent – `vpaa.log`; information about vCenter agent
- Shell – `shell.log`; records Shell commands and events
- vSphere HA agent – `fdm.log`
- Authentication – `auth.log`; contains local authentication info
- System messages – `syslog.log`; general log information used for troubleshooting; was formerly **messages.log**

Besides the local CLI-based log locations, you can log into the host DCUI (direct-console UI) to view several logs – Syslog, VMkernel, Config, vCenter (vpaa), Management (hostd) or Observation (vobd). Lastly, if for some reason your vCenter Server is not accessible, you can view ESXi host logs directly using the Host Client, select **Monitor**, then choose **Logs** tab.

Log	Description
<code>/var/log/vmauthd.log</code>	vMotion authentication daemon log
<code>/var/log/syslog.log</code>	General system log
<code>/var/log/sysboot.log</code>	System boot log
<code>/var/log/shell.log</code>	ESXi shell activity log
<code>/var/loa/hostd.log</code>	Host agent log

15 items

```

3187 2020-04-02T15:00:12Z snmpd: load_ifmib: using VSS module portcfg
3188 2020-04-02T15:00:37Z snmpd: load_ipmi_sel: loaded 0 IPMI SEL entries in 0 seconds
3189 2020-04-02T15:00:37Z snmpd: send_env_notifications: sent 0 of 27 SEL entries as notifications, 8 already sent
3190 2020-04-02T15:01:00Z crond[2099040]: USER root pid 4687928 cmd /sbin/auto-backup.sh
    
```



## vCenter Server logs

The location of the logs for vCenter Server depends on the vCenter Server deployment type in your environment, Appliance (VCSA) or Windows.

**VCSA** – Logs are located in `/var/log/vmware`

- `vpxd.log` – main vCenter log consisting of all vSphere Client and WebServices connections, internal tasks, and events, and communication with vCenter agent (vpxa) on ESXi hosts
- `vpxd-profiler.log` – profiled metrics performed in vCenter
- `vpxd-alert.log` – non-fatal info about vpxd processes
- `cim-diag.log` – (or `vws.log`) CIM monitoring information
- `ls.log` – health report for Licensing Service extension
- `eam.log` – health report for ESXi Agent Monitor extension
- `vimtool.log` – dump of string during vCenter installation of DNS, username and JDBC creation output
- `stats.log` – performance data history from ESXi hosts
- `sms.log` – health report for Storage Monitor Service
- `vmdir.log` – logging for Directory Services
- `drmdump\` – actions taken by DRS

**vCenter on Windows** – log location is:

`C:\ProgramData\Vmware\vCenterServer\Log`s

You can also view logs in the vSphere WEB Client by selecting **vCenter** > **Monitor** tab > **System Logs** tab. From the drop-down menu, you can switch the log type if needed.

The screenshot shows the vSphere Web Client interface. The 'Monitor' tab is selected, and the 'System Logs' sub-tab is active. A dropdown menu is open, showing a list of log files including 'vCenter Server log [vpxd-837.log]', 'vCenter Server log [vpxd-profiler-254.log]', 'vCenter Server log [vpxd-profiler.log]', 'vCenter Server log [vpxd.log]', and 'vpxd-profiler [vpxd-profiler-254.log]'. The 'vCenter Server log [vpxd-837.log]' is selected. Below the dropdown, a preview of log content is visible, showing entries with timestamps and details such as 'Originator@6876 sub=vpxLro opID=21721c83 [VpxLRO] -- FINISH Iro-28059429' and 'BEGIN session[5259491a-a1c1-fd7f-68d5-88024149b796]521cbc47-16ca-2fac-8a53-20c21b5de09c -- CatalogSyncManager --'. At the bottom, there are controls for 'Showing 2000 of 127045 lines', 'Show line numbers', 'Show Next 2000 Lines', and 'Show All Lines'.



## Virtual Machine logs

Log locations for VMs are, by default, stored with the VM's configuration file (.vmx), and named `vmware-#.log`. From an ESXi host, location is: `/vmfs/volumes/<datastore-id>/<vm-name>`

## Objective 4.2: Create and configure vSphere objects



Resource:

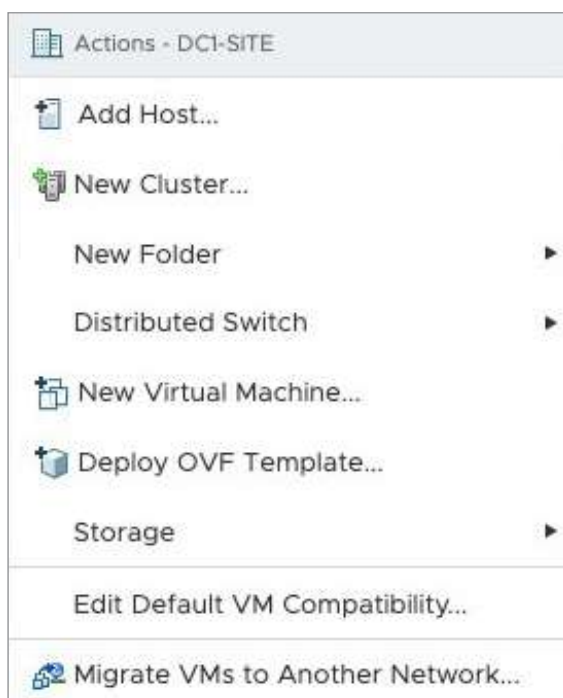
### Create and configure vSphere data centers

A vCenter Server data center object is the building block of your vSphere hierarchy. From this object, more powerful objects in the hierarchy are created. Though important to begin laying out your vSphere (vCenter) hierarchy, there is nothing to configure. Data centers are simply a logical container from which to create the remaining objects in your inventory. To create a data center object, log in to vCenter Server, right-click on the **vCenter Server** > **New data center**.

### Create and configure Virtual Machines

I'll only briefly cover a simple VM creation since this topic will be covered in more detail in Objective 7.9.

You can create a new VM in several ways, but I'll share just one here – right-click a vCenter **cluster** object or a **standalone host** > **New Virtual Machine**.



After you create your VM, you can modify its settings as needed, anywhere from CPU to memory, add hard disks, or other virtual devices. In the VM Options tab (second screenshot), you can select the guest OS version, enable encryption, as well as BOOT options and many other items.



[vCenter Server and ESXi Host 6.7 Management Guide, Update 2 \(April 2019\)](#)



[vSphere 6.7 Virtual Machine Administration Guide, Update 2 \(April 2019\)](#)



[vSphere 6.7 Resource Management Guide, Update 2 \(April 2019\)](#)



### Edit Settings | admin01

Virtual Hardware | VM Options

ADD NEW DEVICE

> CPU	4		
> Memory	8	GB	
> Hard disk 1	100	GB	
> SCSI controller 0	LSI Logic SAS		
> Network adapter 1	VM-510		<input checked="" type="checkbox"/> Connect...
> Network adapter 2	VM-510		<input checked="" type="checkbox"/> Connect...
> Network adapter 3	VM-510		<input checked="" type="checkbox"/> Connect...
> CD/DVD drive 1	Client Device		<input type="checkbox"/> Connect...
> Video card	Specify custom settings		

CANCEL OK

### Edit Settings | admin01

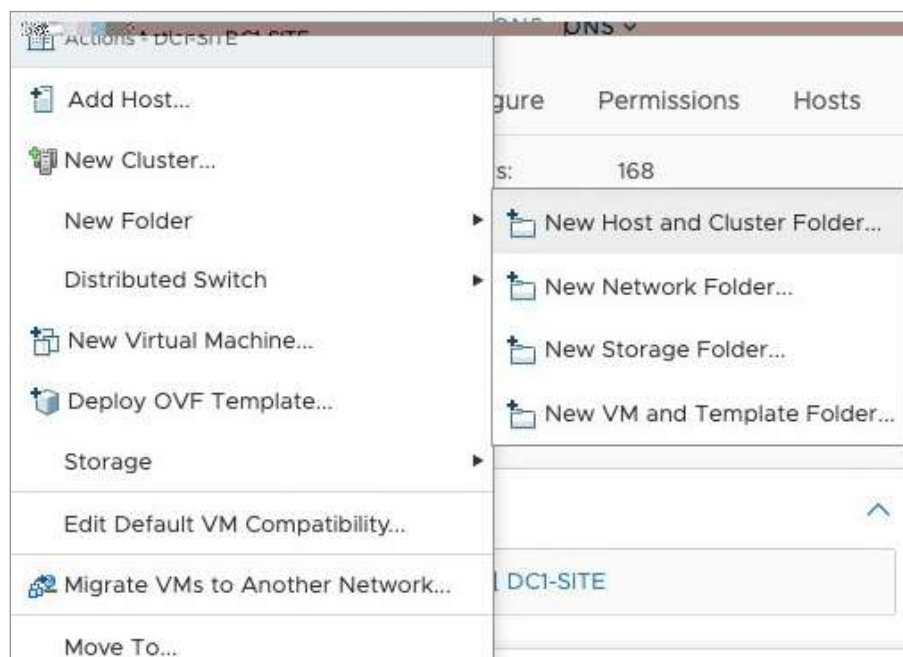
Virtual Hardware | VM Options

Encryption	Expand for encryption settings	
Encrypt VM	Datstore Default	(Requires Key Management Server)
Encrypted vMotion	Opportunistic	
> Power management	Expand for power management settings	
> VMware Tools	Expand for VMware Tools settings	
Boot Options		
Firmware	BIOS (recommended)	
Boot Delay	When powering on or resetting, delay boot order by 0 milliseconds	
Force BIOS setup	<input type="checkbox"/> During the next boot, force entry into the BIOS setup screen	

CANCEL OK

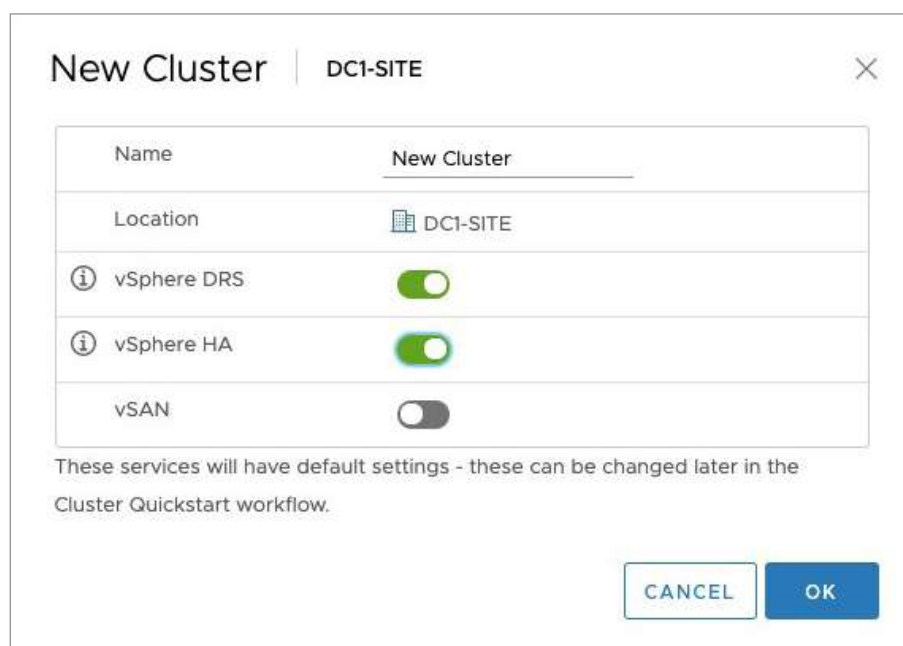


Create vSphere inventory folders for VMs and templates, network, storage, or host and cluster. Just right-click on your data center object > **New folder**



### Create and configure vSphere clusters

Clusters were fully discussed in Objectives 1.6 and 1.9, so I won't review specifics of cluster configuration here. You can review those two objectives, as well as VMware Guides, for more in-depth descriptions of cluster features. To create a new vCenter cluster, simply right-click the **data center** object > **New Cluster**







Once you have created and enabled the cluster features you want, and after creation, select your new cluster in vCenter, then the **Configure** tab and **Edit** button to configure items and features to meet your business needs.

Service	Status
DRS Automation	Fully Automated
Additional Options	Expand for policies
Power Management	Off
Advanced Options	None

### Create and configure resource pools

Resource pools were discussed in great detail in Objective 1.7, so I will not cover them again here. Creating resource pools are quite simple. The only part of creation you need to give thought to is the concept of resource allocation you want to give your workloads in each resource pool. Otherwise, creating them is as simple as right-clicking a **Host** or **Cluster** > **New Resource Pool**, then configure the desired metrics to meet your needs:

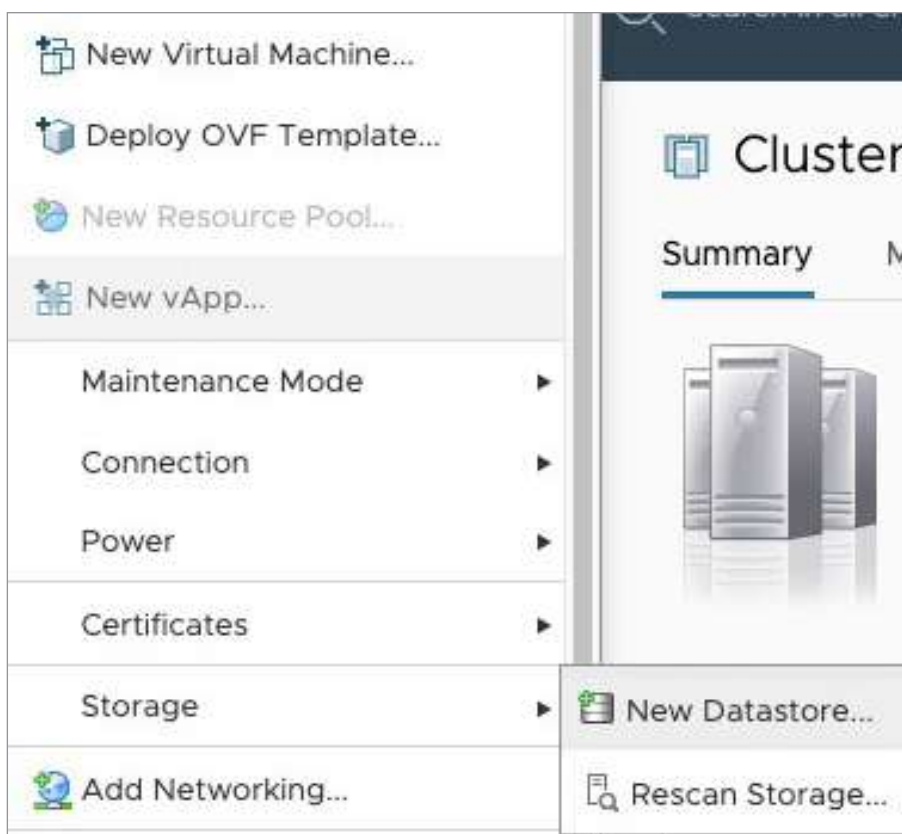
Category	Property	Value
CPU	Name	New Resource Pool
	Shares	Normal (4000)
	Reservation	0 MHz (Max: 324,990 MHz)
	Reservation Type	Expandable
Memory	Limit	Unlimited MHz (Max: 387,262 MHz)
	Shares	Normal (163840)
	Reservation	0 MB (Max: 1,402,617 MB)
	Reservation Type	Expandable



## Create and configure datastores

Storage was covered in great detail back in Objective 1.3. Make sure you meet the requirements discussed in that objective before adding a storage object in your infrastructure, whether it be a VMFS or NFS datastore, or VVol, or vSAN.

To create, make sure you have your storage set up and your prerequisites are met. Then, from the vSphere Client, either right-click on a host or select the host and from the actions menu click **Storage > New Datastore**.



### New Datastore

#### 1 Type

2 Name and device selection

3 VMFS version

4 Partition configuration

5 Ready to complete

#### Type

Specify datastore type.

VMFS

Create a VMFS datastore on a disk/LUN.

NFS

Create an NFS datastore on an NFS share over the network.

VVol

Create a Virtual Volumes datastore on a storage container connected to a storage provider.

- To edit datastore properties after you've created it, from the cluster, go to the Datastores tab, click the datastore you're wanting to manage, then the **Configure** tab. Select **General**, then the Edit button under any datastore option you are wanting to configure – **Capacity**, **Capabilities** (SIOC control) or **Space Reclamation**. To modify storage adapter and path capabilities, do so from each: Host > **Configure** > **Storage** section > **Storage Adapters**
- Properties tab – Modify the IQN, Authentication (CHAP), or enable/disable the iSCSI Storage Adapter
- Dynamic (or Static) Discovery tab – Add a storage device
- Port-binding tab – Add vmk ports for multipathing
- To change a device's multi-pathing PSP, select **Storage Devices** (instead of Storage Adapters), click on device in the list, then from the **Properties** tab below, scroll down to **Edit Multipathing**, and select the desired PSP from the drop-down menu



**Edit Multipathing Policies** | eui.21d29dcddfd05e0b6c9ce900af4b4ba2

Path selection policy:

- ✓ NIMBLE\_PSP\_DIRECTED
- Most Recently Used (VMware)
- Round Robin (VMware)
- Fixed (VMware)

Runtime Name	Status	Target	LUN	Preferred
vmhba64:C1:T2:L0	Active (I/O)	iqn.2007-11.com.nimblestorage:dc1...	0	
vmhba64:C0:T2:...	Active (I/O)	iqn.2007-11.com.nimblestorage:dc1...	0	

- To mount/unmount a datastore (for maintenance), right-click the datastore and choose **Unmount Datastore**. When maintenance is completed, right-click it again and select **Mount Datastore**

## Objective 4.3: Set up a content library

First, let me share what a content library is, then I will move on with a brief architectural overview and configuration details. A content library is a container object for VM and vApp templates, as well as files such as ISOs, images, txt files, etc. Using templates, you can deploy VMs or vApps in the vSphere inventory. Doing so across vCenter Server instances in the same or different location results in consistency, compliance, efficiency, and automation in deploying workloads at scale. Content libraries store and manage content in the form of library items. A single item can contain one or multiple files. As an example, an OVF consists of `.ovf`, `.vmdk`, and `.mf` files. When uploading an OVF template, all files are uploaded, but you only see one library item in the content library UI. Before vSphere 6.7U1, content libraries only supported the OVF template format. Now they also support VM templates (note: vApp templates are still converted to OVF templates).

### Resource:



[vSphere 6.7 Virtual Machine Administration Guide, Update 2 \(April 2019\)](#), pages 61-80

Content libraries are managed from a single vCenter Server, but you can share them with other vCenters if HTTPS traffic is allowed between vCenter instances. There are two types of content libraries:

- **Local Library** is used to store items in a single vCenter Server. You can publish them so other vCenter Server users can subscribe to them, as well as configure a password for authentication to the library. A caveat is if you have a VM template in the Local Library, you cannot publish it.
- **Subscribed Library** is a library you create when you subscribe to a published library. The Subscribed Library can be created in the same or remote vCenter than the published library. When creating a Subscribed Library, you can choose to immediately download the full contents of the library upon creation or download only the published library metadata and later download the full content of the library. Keep in mind, when using a Subscribed Library, you can only **use** the library content, not add items to it.

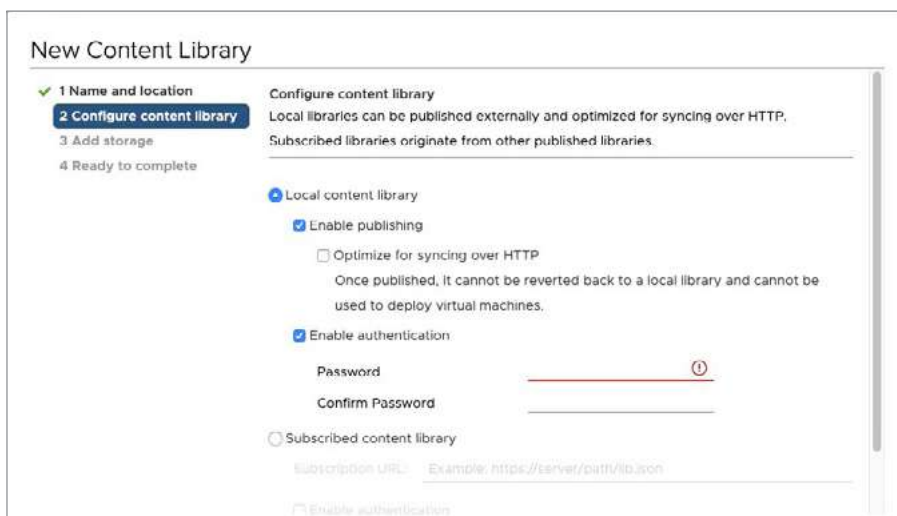
Source Object	Download library content immediately	Download library content when needed
A library running in a vCenter Server 6.x instance.	Supported	Supported
A catalog running in a vCloud Director 5.5 and later instance.	Supported	Not supported
A third-party library.	Supported for third-party libraries that require authentication, if the username of the third-party library is <b>vcsp</b> . If the username of the source third-party library is different than <b>vcsp</b> , you can subscribe to it by using VMware vCloud Suite API.	Supported for third-party libraries that require authentication, if the username of the third-party library is <b>vcsp</b> . If the username of the source third-party library is different than <b>vcsp</b> , you can subscribe to it by using VMware vCloud Suite API.

## Content library prerequisites

The only prerequisites for creating a content library is to at least be running vCenter Server 6.5 and to have **Content library.Create local library** or **Content library.Create subscribed library** privileges.

## Create a content library

To create a content library from the vSphere Client, select **Menu > Content Libraries**, then from the content library window click + to **Create New Content Library**.



**New Content Library**

1 Name and location  
 2 Configure content library  
 3 Add storage  
 4 Ready to complete

**Configure content library**  
 Local libraries can be published externally and optimized for syncing over HTTP. Subscribed libraries originate from other published libraries.

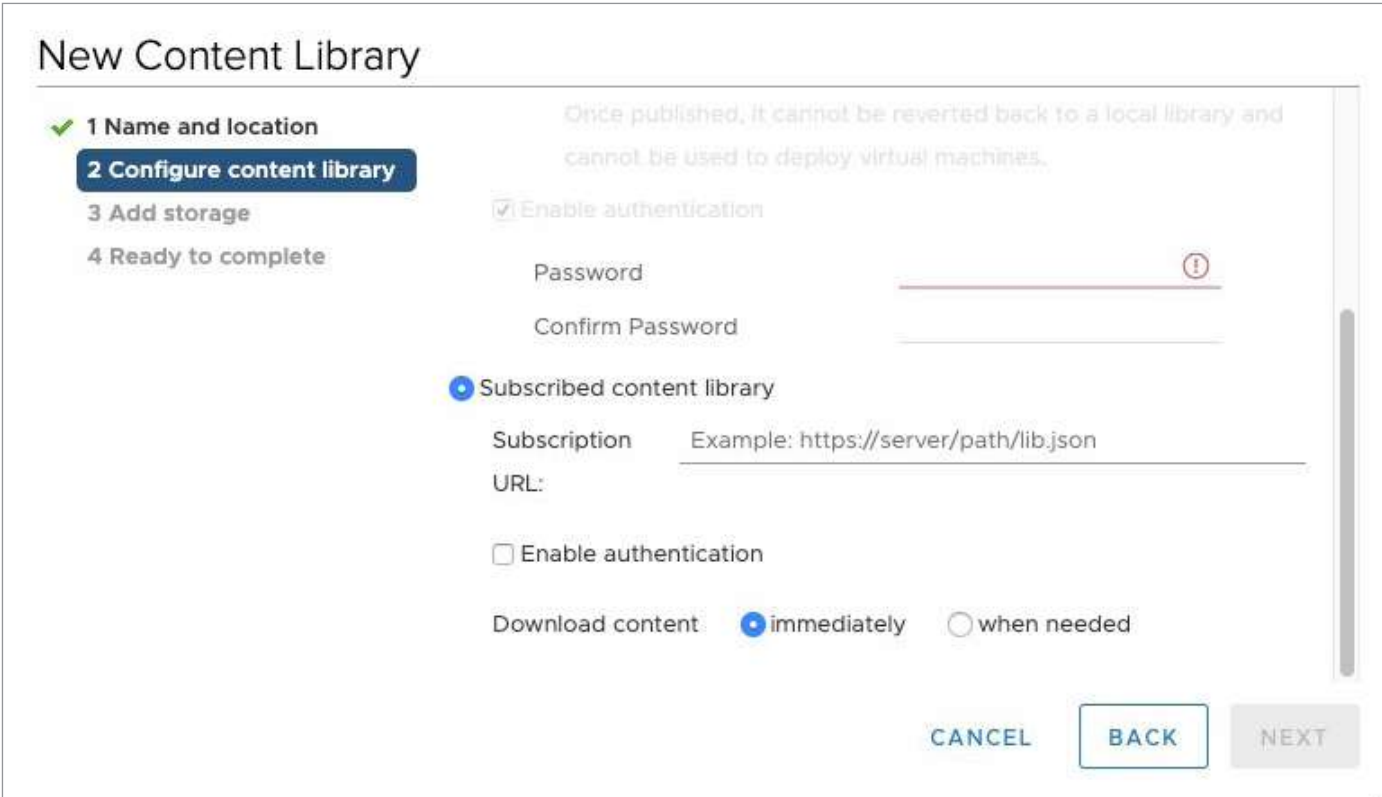
Local content library
 

- Enable publishing
  - Optimize for syncing over HTTP  
Once published, it cannot be reverted back to a local library and cannot be used to deploy virtual machines.
- Enable authentication
  - Password: \_\_\_\_\_
  - Confirm Password: \_\_\_\_\_

Subscribed content library
 

- Subscription URL: \_\_\_\_\_ Example: https://eserver/pa1tn/lib.json
- Enable authentication

Choose the library type you want to create and optionally publish it and/or enable password authentication to the library.



**New Content Library**

1 Name and location

**2 Configure content library**

3 Add storage

4 Ready to complete

Once published, it cannot be reverted back to a local library and cannot be used to deploy virtual machines.

Enable authentication

Password !

Confirm Password

Subscribed content library

Subscription Example: `https://server/path/lib.json`

URL:

Enable authentication

Download content:  immediately  when needed

CANCEL BACK NEXT

To make sure you have the latest library content if you are using a Subscribed Library, you can initiate a manual synchronization by right-clicking Library > Synchronize. Make sure you have the **Content library.Sync subscribed library** privilege.

When delegating permissions for content libraries, it's important to understand content library hierarchical structure. They work in vCenter Server but are not direct children under vCenter. They are under the **global root**. So, if you grant a user a content library permission at the vCenter Server level, the user will not be able to see nor manage content libraries. You need to assign permissions at the global root level. There is a default content library administrator role that can be assigned or you can copy that role and modify the permissions (and rename it) to fit your organizational needs. Administrators at the vCenter Server level can manage content libraries, but only if also given the read-only role at the global root.

### Importing and exporting content to/from a content library

A user must have at least the **Content library.Add library item** or **.Update files** privileges.

To add an item, simply right-click the content library and choose **Import Library Item**. You can choose to import from a URL or local file. Conversely, to export, simply select **Export Item** instead of Import.

For further content library management tasks, review pages 73-87.

## Objective 4.4: Set up ESXi hosts

### Resource:



[VMware ESXi 6.7 Installation and Setup Guide, Update 2 \(April 2019\)](#)

ESXi prerequisites were discussed in Objective 1.1. Review those requirements before performing one of the below ESXi installation methods. Also keep in mind we cannot cover every detail of the ESXi installation and configuration process. We highly recommend you read the VMware ESXi Installation Guide to fill in the gaps.

### Installing ESXi

**Interactive Installation** can be done by simply creating a bootable ISO on a CD-ROM, DVD drive, USB, or via PXE (network) boot. Once the bootable device is created, place it on/in your host and boot it. Follow the prompts to configure the host for installation. If you don't have an ISO burn tool (PowerISO, for example), you can use a Linux system to create a bootable USB using the following procedures:

- Connect the USB to the system. In the below steps, the USB is `/sdb`
- Create a partition using `fdisk`: `/sbin/fdisk /dev/sdb`
- Create a file system on the USB (FAT32):  
`/sbin/mkfs.vfat -F 32 -n USB /dev/sdb/`
- Install Syslinux bootloader on the USB:  
`usr/bin/syslinux /dev/sdb ;`  
`cat /usr/lib/syslinux/mbr/mbr.bin > /dev/sdb`
- Create a directory and mount a directory for both the USB and ESXi ISO
- Copy ISO directory contents to USB: `cp -r /esx-iso/* /usbdir`
- Rename the `isolinux.cfg` to `syslinux.cfg` using "mv" linux command
- Edit `syslinux.cfg`: `APPEND -c boot.cfg -p 1`
- Unmount both the USB and ISO image; USB is ready for use

**Scripting** can be used to install ESXi by using a `ks.cfg` file. This is a text file with various commands. The **command section** of the file contains ESXi install options, is required, and must appear first in the file. Keep in mind, you cannot place this file on the same USB you are using as a bootable installer. It needs to be on its own USB device or other supported locations – FTP server, HTTP/S server, USB, CD, or NFS server. The default location is in the initial RAMDISK location at: `/etc/vmware/weasel/ks.cfg`

To begin the install, start the ESXi installer:

- Press SHIFT+O to change any **kickstart** file (`ks.cfg`) boot options needed



- You are prompted by the runweasel command prompt. Enter `ks.cfg` information. Below is an example.:

```
ks=http://00.00.00.00/kickstart/ks-osdc-pdp101.cfg nameserver=00.00.0.0 ip=00.00.00.000
netmask=255.255.255.0 gateway=00.00.00.000
```

- Refer to the ESXi Install Guide for detailed description of `ks` file options

**PXE Booting** has been supported since vSphere 6.0. All that's required is a DHCP server and TFTP server, and a host that is PXE-bootable (configured in the BIOS). Below is the process involved in PXE-boot:

- ESXi host is booted
- ESXi host broadcasts for DHCP request
- DHCP responds with IP and location of TFTP server
- ESXi host contacts TFTP server and requests the file DHCP specifies
- TFTP sends boot loader and ESXi host executes it
- Boot loader searches TFTP server for configuration file, downloads the kernel, and other components
- Installer runs interactively or via kickstart script

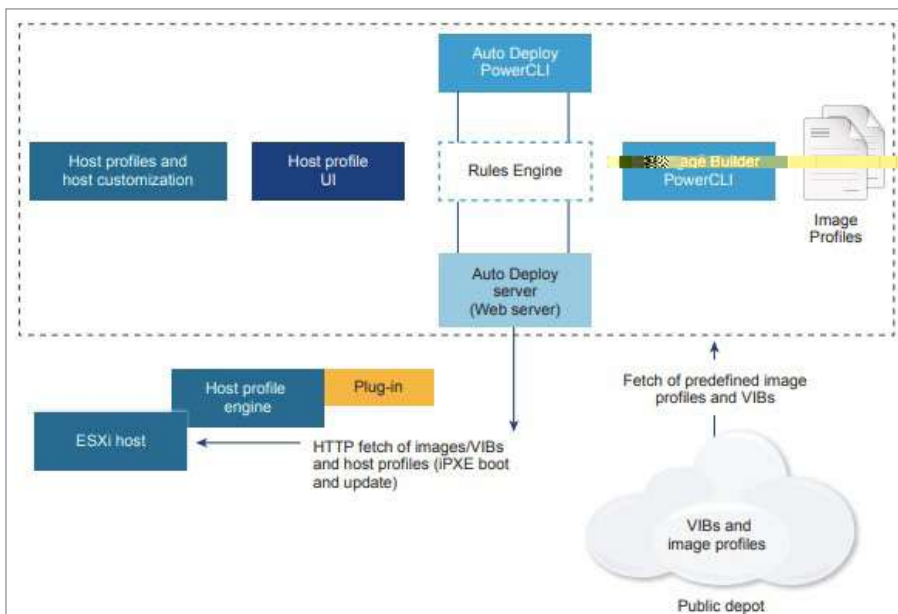
**Auto-Deploy** installation, though still requiring several steps, has gotten easier since VMware instituted this method within the vSphere Client. With Auto Deploy, hosts are network-booted from a central Auto Deploy server, optionally configured with a host profile (discussed in detail in Objective 7.16), and then once configuration is completed, is managed by vCenter Server like any other host. You can use Auto Deploy in one of two ways:

- **Stateless caching** is the default and the Auto Deploy image profile defines the image the host is provisioned with
- **Stateful install** is provisioning a host with Auto Deploy and installing the image to a local disk. The subsequent host reboots and the host boots from the disk

Auto Deploy consists of several components:

- **Auto Deploy server** serves image profiles and host profiles to hosts
- **Auto Deploy rules engine** sends information to the Auto Deploy server which image and host profile to serve which host
  - **Rules** assign image, host profiles, or specify the location of ESXi hosts. Rules are created using vSphere Client or PowerCLI
  - **Active rule set** is used by hosts first booted for matching rules
  - **Working rule set** allows you to test changes before making them active
- **Image profile** defines a set of VIBs to boot ESXi hosts with
- **Host profile** defines machine-specific configurations
- **Host customization** stores information a user provides when host profiles are applied to a host





The Auto Deploy workflow process is different for **first-boot** hosts vs **subsequent-boot** hosts:

- First boot is similar to a PXE install:
  - Host is booted
  - Host contacts TFTP server and downloads iPXE bootloader and configuration file
  - iPXE executes, instructing the host to make a boot request to Auto Deploy the server
  - Auto Deploy server queries the rules engine for information about host and streams components specified in the image and host profile
  - Host boots using image profile and optionally the host profile is applied
  - Auto Deploy adds the host to vCenter Server if there's a rule for folder or cluster; if there isn't a rule, then it goes to the data center object
  - Host customization, if needed
- Subsequent boot:
  - Host is rebooted
  - Auto Deploy provisions host with image and optional host profile
  - VMs are powered on (standalone) or migrated to host if in the cluster

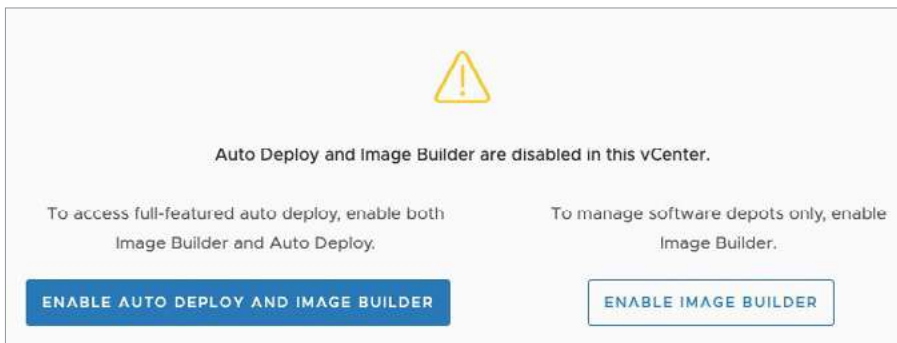
The Auto Deploy process is as follows:

- Verify that hosts are compatible for Auto Deploy (check HCL and enable hardware for PXE boot)
- Set up your DHCP and TFTP environment (refer to PXE boot above)
  - Replace `gpxelinux.0` with either `snponly64.efi.vmw-hardwired` for UEFI boot or `undionly.kpxe.vmw-hardwired` for legacy BIOS





- Enable both the Auto Deploy and Image Builder services in vCenter **vSphere Client** > **Menu** at top > **Auto Deploy**, then click to **Enable**



- Add a Software Depot, a collection of VIBs, and image profiles, either an online one or custom

**NOTE: Auto Deploy requires minimum of 2GB for its repository (about 350 MB per image profile).**

- Create an image profile
- Create a deploy rule assigning an image profile to hosts
- Power on the host you're wanting to provision
- Extract the host profile from a reference host
- If using PowerCLI, after enabling Auto Deploy in vCenter, the high-level PCLI commands to set up Auto Deploy is as follows:
  - `Add-EsxSoftwareDepot c:/<path>/<file.zip>`
  - `Get-EsxImageProfile`
  - `New-DeployRule -Name testRule -Item <img-prof-name> -Pattern "ipv4=192.x.x.x-192.x.x.x" (or AllHosts if no range or pattern)`
  - `Add-DeployRule testRule (NoActivate to not place rule in the Active Rule Set)`

## Configuring ESXi

Host profiles can be used to set up your hosts and are mainly used for large vSphere environments to provide a consistent, compliant, efficient host configuration. Further details on host profiles is discussed in Objective 7.16.

Otherwise, initial configuration of a host is done through the DCUI. Log in by pressing F2 and configure the network, including IP, DNS, domain, VLAN (optional), and then it's recommended to **Test the management network** when finished with configuration. The DNS check will fail if you forgot to add the host "A" record to DNS or if there are VLAN issues.

Another option to configure your host is by creating a PowerCLI script, then deploying it against one or more hosts. You can configure your `.ps1` script to prompt you for host-specific information, like IP configuration and/or VLAN ID assignment.



## Objective 4.5: Configure virtual networking



Resource:



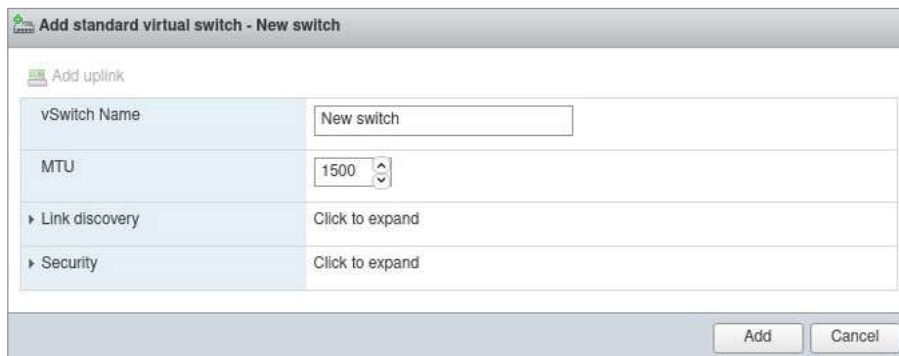
Resource: [vSphere 6.7 Networking Guide](#)

In VMware vSphere, there are two types of virtual switches:

- vSphere Standard Switch (vSS) – Available on every ESXi host; one vSS is created during installation on every deployed host; a vSS acts like a physical switch and connects VMs through ESXi host physical ports to the physical infrastructure. Each vSS is a standalone entity, applicable only to a single ESXi host – this means that once we decide to, for example, change the VLAN ID on our main network used by virtual machines, we will have to apply that change manually on each vSS used for VM networking on each ESXi host.
- vSphere Distributed Switch (vDS) – Available once we deploy a vCenter Server and requires an Enterprise *Plus* license; vDS brings more features than a vSS and allows for easier management of virtual networks in vSphere – changes applied to a vDS will be automatically carried over to the connected hosts, requiring little to no extra input.

To **create a new vSphere Standard Switch (vSS)**, log into the Host Client for the ESXi host in question and navigate to **Networking > Virtual Switches**. A vSS can also be created in the vSphere Client.

Select the **Add standard virtual switch** button. The following screen is similar to editing an existing vSS:



Creating a vSS is easy and requires just a name for the new virtual switch. Everything else is optional.

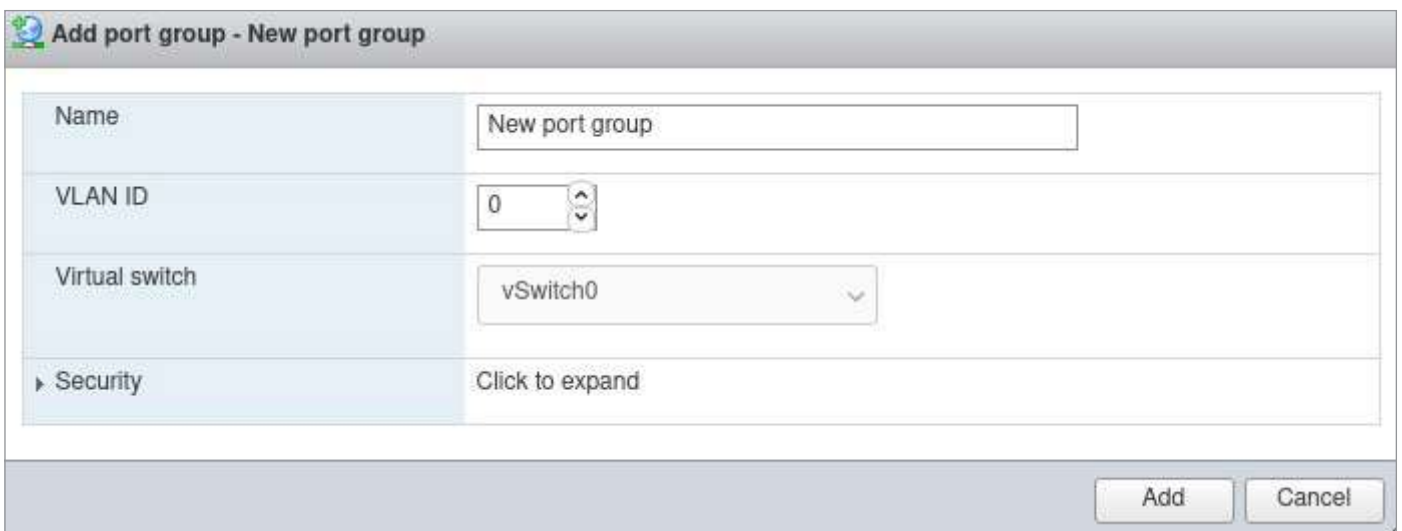
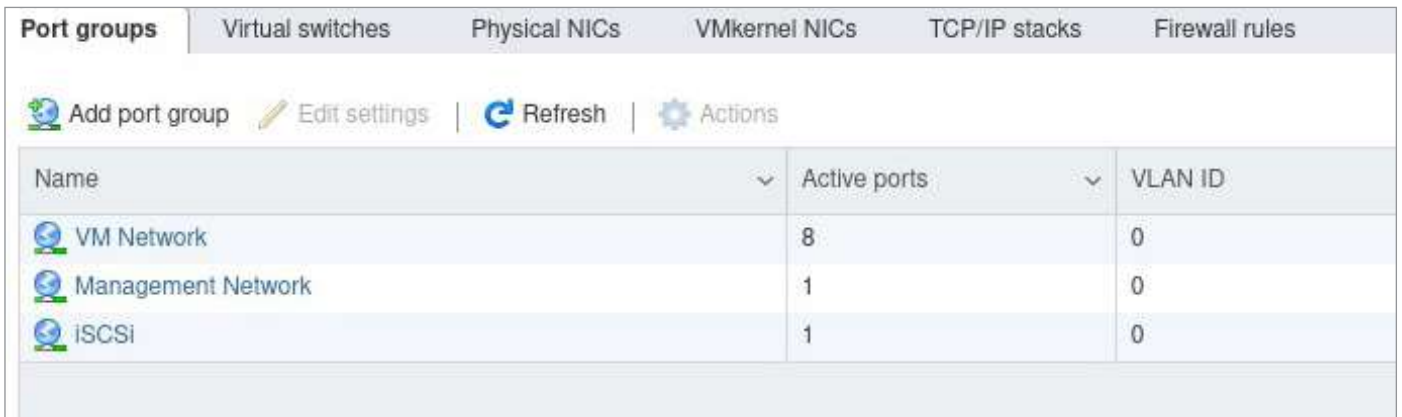




After adding a new virtual switch, we can **add uplinks** to it by using the button **Add uplink** in the same area of the ESXi Host Client.



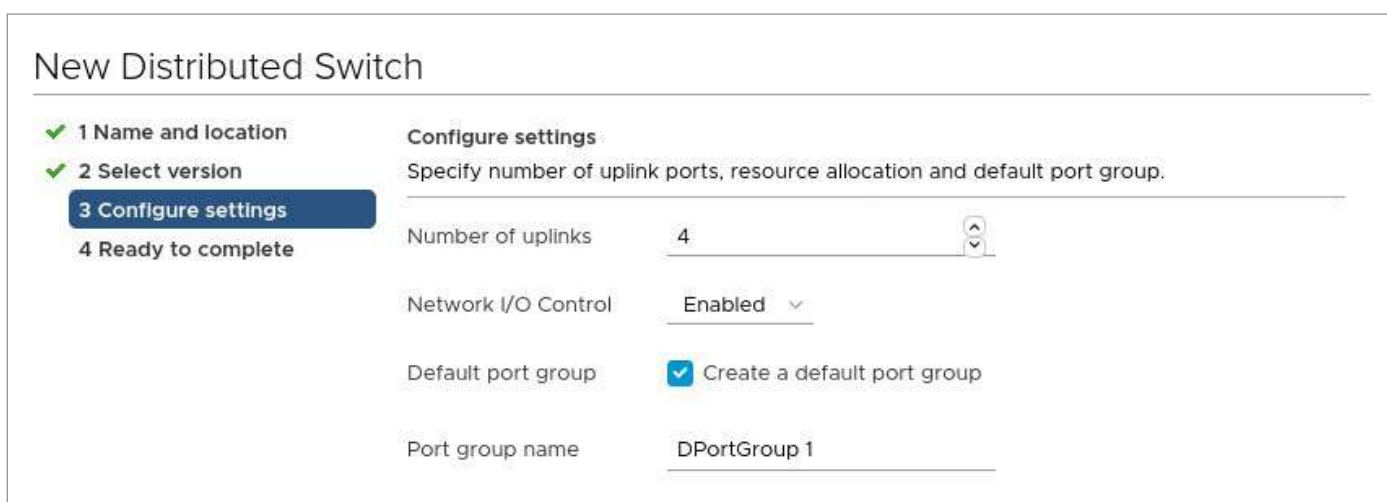
Lastly, we can add a new standard port group to the vSS by going to the **Port groups** tab in the ESXi Host Client. Pressing the **Add port group** button will launch a **New Port Group** wizard. The only required parameter here is a name for the new port group.



With that said we can move on to the **vSphere Distributed Switch (vDS)**. Before we begin discussing different configuration options available on a vDS, we need to create one. To do so, navigate to the **Networking** tab in the vSphere Client, right-click a data center object and select **Distributed Switch > New Distributed Switch**:



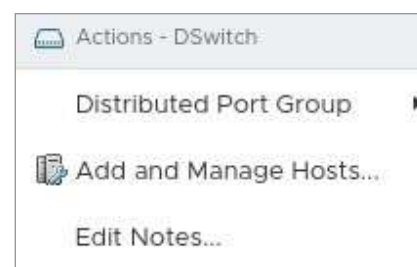
Creating a vDS is a bit more complicated than setting up a vSS on a host, aside from giving it a name and selecting a version (vDS version 5 can be connected to ESXi 5.x and up, vDS version 6 to ESXi 6.x and up, etc.), **you need to specify the number of allowed uplinks**, name of the first port group (optional), and if network I/O control should be enabled or not:



Once the new vDS is created, **connect ESXi hosts** to it, then migrate the virtual machines to the vDS port groups. This can be done by right-clicking on the new vDS and selecting the option **Add or Manage Hosts...**

**This wizard also allows you to manage already connected hosts** or remove hosts which should no longer use the vDS.

- While removing hosts from a vDS, you need to migrate the network adapters that are in use to a different switch – this could be a different vDS or a vSS.
- While migrating from different virtual switch to another, the important thing to keep in mind is to always make sure the ESXi hosts retain network connectivity throughout the whole process. To do so, it is recommended to migrate the adapters in stages; for example – migrate only one of the two management interfaces at a time.





## Objective 4.6: Deploy and configure VMware vCenter Server Appliance (VCSA)



### Resource:



[vCenter Server Installation and Setup](#)

**Deploying a vCenter Server Appliance** begins with downloading an official vCenter Server Appliance **Installation ISO** from the official VMware website. This ISO file contains graphical and CLI-based installers for the three big operating systems, which you can use to carry out the deployment (Microsoft Windows, Apple MacOS, and GNU/Linux).

To begin the deployment, mount the ISO on your client machine and browse to the location of the installer executable – for all the operating systems, this file will be called **vsca-ui-installer**.

The installation is very simple and should not cause any issues in most cases.

After starting the installer script, you will be presented with four options:

- Install – Starts the installation wizard for new VCSA deployments
- Upgrade – This option allows you to upgrade an existing VCSA deployments to the current version
- Migrate – VCSA deployments created with an external PSC (Platform Service Controller) can be converted to an embedded setup using this option
- Restore – Recover a failed VCSA or PSC deployment using a backup

As we are only discussing an initial deployment of VCSA, we are going to choose the first option; the next choice we are going to make is whether to deploy a vCenter Appliance with an embedded or separate (external) PSC. Since external PSC deployments are getting deprecated, we are going to discuss the Embedded deployment type only.

Embedded PSC deployment type also has some additional advantages, aside from cutting the cost of having two management VMs in our infrastructure:

- Simplified backup and restore process
- Simplified vCenter High Availability
- Enhanced Linked Mode – As of vSphere 6.7; a feature of embedded VCSA deployments allowing to connect multiple vCenter servers together, which among many other benefits, allows you to view and manage objects belonging to separate vSphere inventories in a single interface.

After choosing the embedded deployment type, we need to specify the details of an ESXi host, which will provide compute and storage:

- IP address or FQDN (and a port number if the default port 443 was changed)
- User name for a root-level user (or just root)
- Password for said user (usually root password)

Next, we will be asked to set VCSA appliance settings:

- VM name
- Root password



And then comes selecting the deployment size; we have a few options depending on the environment, which is going to be managed by the VCSA we are deploying:

- Tiny (Two vCPU, 10 GB RAM, 300 GB disk) – Up to 10 ESXi hosts or 100 VMs
- Small (Four vCPU, 15 GB RAM, 340 GB disk) – Up to 100 ESXi hosts or 1,000 VMs
- Medium (Eight vCPU, 24 GB RAM, 525 GB disk) – Up to 400 ESXi hosts or 4,000 VMs
- Large (16 vCPU, 32 GB RAM, 740 GB disk) – Up to 1,000 ESXi hosts or 10,000 VMs
- Extra-large (24 vCPU, 48 GB RAM, 1180 GB disk) – Up to 2,000 ESXi hosts or 35,000 VMs

It is possible to select a bigger size of the disk for smaller deployments – i.e. select a tiny deployment (two vCPU and 10 GB RAM) with a medium size disk (525 GB) – but not vice-versa.

After selecting the deployment size, we can choose which datastore should be used for hosting VCSA disks. It is possible to store VCSA disks in thin mode, but that should be used in caution and only for very specific deployment types (home lab, developer environment, etc.)

On the last screen of this section of the deployment wizard, we need to specify VCSA network settings:

- Network – Select a port group (or distributed port group) to which VCSA VM should be connected
- IP version – Choose between IPv4 or IPv6
- IP assignment – Static or DHCP
- FQDN – (optional) Specify an FQDN for the vCenter Server

**NOTE: FQDN must have a DNS record that is present in both forward and reverse lookup zones (A record and PTR record).**

- IP address, network mask, default gateway, DNS servers (only if static IP assignment was selected)
- Common ports – Allows you to change default ports used by the vCenter Server

After saving the network settings, the VCSA VM setup will begin and after some time you will be asked to proceed to stage two of VCSA deployment.

Stage two begins with configuring time synchronization and SSH access:

- Time synchronization – You can pick from the possible options:
  - Synchronize time with the ESXi host (no extra input is needed, VCSA will pull system time from the ESXi host it is running on)
  - Synchronize with an NTP server (IP address or FQDN of the NTP server will be required to proceed)
- SSH access – Select whether SSH access to the appliance should be enabled or disabled

Next comes a very important part – SSO configuration; on this page we can select one of two options:

- Create a new Single Sign-On (SSO) domain – Default and used in most of VCSA deployments; this option will create a new VCSA-based SSO domain for managing logins to the vCenter Server.; the default SSO domain is vsphere.local. SSO domain name – Can be any FQDN-compatible domain name and does not have to be included in DNS; the SSO domain name doesn't and shouldn't be identical to, for example, a Windows Active Directory domain name. Consider using .local TLD i.e. ADDomainName.TLD.local
- Join an existing SSO domain – This option should be used if the new VCSA server should join an already deployed VCSA server in Enhanced Linked Mode.

Lastly, we can decide whether we would like to join the CEIP – Customer Experience Improvement Program (optional).

After saving the changes, we will again have to wait for a few minutes. Once the appliance is fully configured, a message will be displayed informing us we can now log in to our newly deployed VCSA by pointing any supported Internet Browser to [https://VCSA\\_IP\\_or\\_FQDN/UI](https://VCSA_IP_or_FQDN/UI).

Before we wrap up this objective, it is worth mentioning that VCSA can also be deployed using a CLI. This type of deployment requires filling in a JSON file\* with all the necessary information and starting the installation from a Windows PowerShell, Linux Bash shell, or MacOS Terminal.

**NOTE: \*JSON templates for CLI deployments are also available on the VCSA ISO in the vcsa-cli-installer directory.**

## Objective 4.7: Set up identity sources

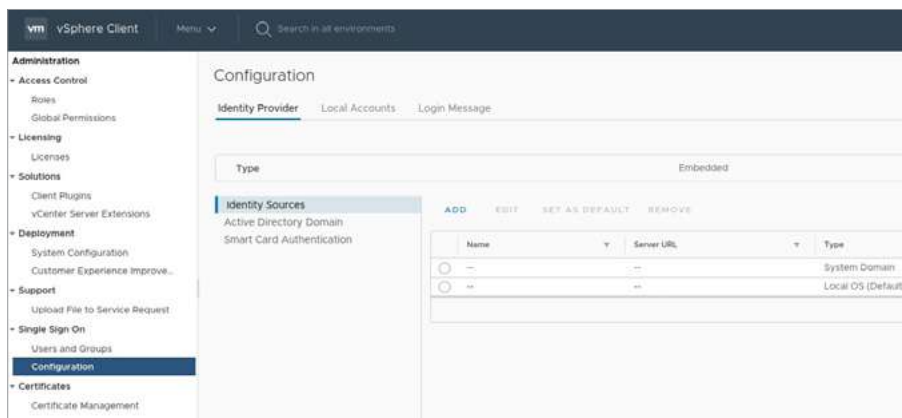
 Resource:



[vCenter Single Sing-On Identity Sources](#)

Setting up additional identity sources on a vCenter Server allows us to use accounts and groups we already manage using an existing LDAP/Kerberos solution such as Microsoft Active Directory. This reduces the need of managing local vCenter Server SSO accounts.

Configuring an additional identity source can be done via vSphere Client in **Administration > Single Sign-On > Configuration > Identity Sources**.

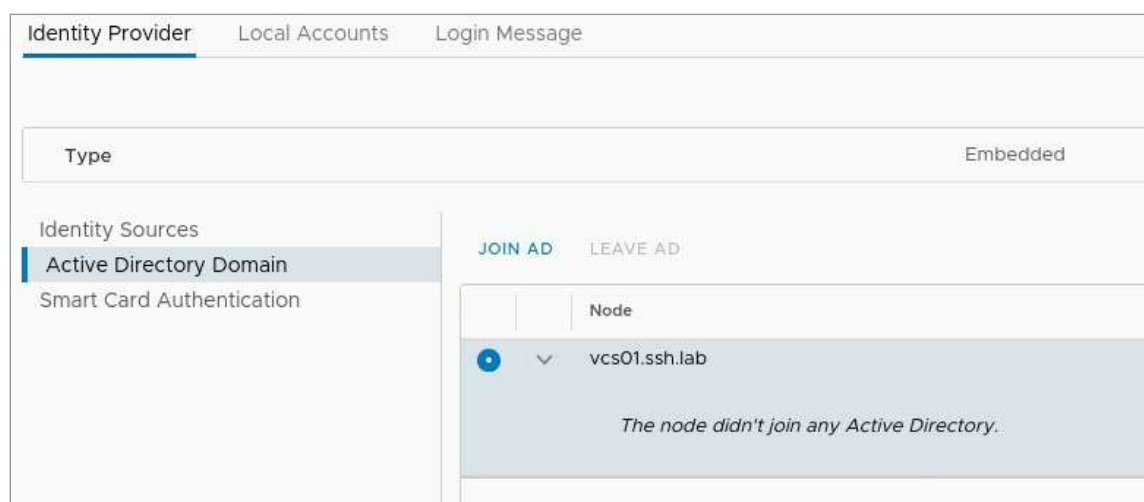


New identity sources can be added by pressing the **Add** button; there are four options available:

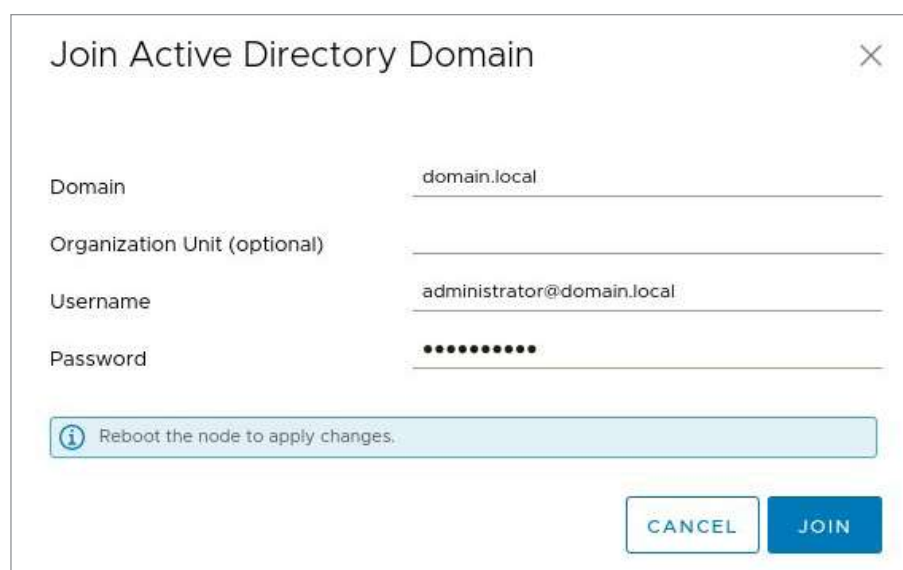
- Active Directory Windows Integrated Authentication
- Active Directory over LDAP
- Open LDAP
- Local operating system of SSO server

The most common option chosen here is the Active Directory Windows Integrated Authentication, so that's what we are going to discuss in this chapter.

Before we add Active Directory as an identity source, we need to join the vCenter Server to AD. To do so, select the **Active Directory Domain** tab, which is just underneath **Identity Sources** on the same page.



Click the **Join AD** button to begin the process. A new pop-up will open where you need to specify a domain name, organizational unit (OU, optional), username, and password for the user allowed to add new systems to AD.





After pressing the **Join** button and rebooting the vCenter Server appliance, your VCSA will be joined to the Active Directory domain and adding a new identity source will be possible.

Again, to do so we need to Go back to the **Identity Sources** tab; select the **Add** button to begin.

The only required bit of info on that page is the domain name. After saving the changes, AD-sourced users, and passwords will be available for use in the vCenter Server.

## Objective 4.8: Configure an SSO domain

During every vCenter Server Appliance deployment, you need to either join or create a Single Sign-On domain. This domain can be described as a glue holding together the vCenter Server, Platform Service Controller (PSC), and many other vSphere components and add-ons.

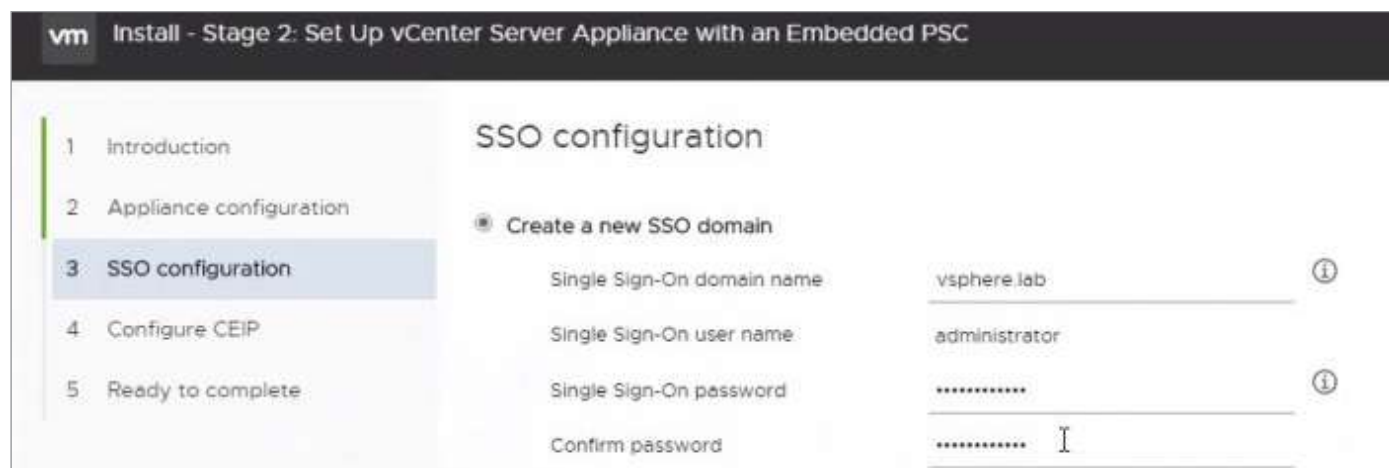
An SSO domain should not be confused with a Microsoft Active Directory domain or any other LDAP/Kerberos identity source – these can be added following a VCSA/PSC deployment and are optional, while an SSO is mandatory.

As already stated above, creating an SSO domain can be done while deploying a VCSA. Only a domain name and a password for the default user (administrator) is required to create the SSO.

### Resource:



[Understanding vCenter Single Sign-On](#)



Once the VCSA/PSC deployment is complete, you can manage or adjust some SSO settings in **Administration > Single Sign-On** part of the vSphere Client; there are two available tabs:

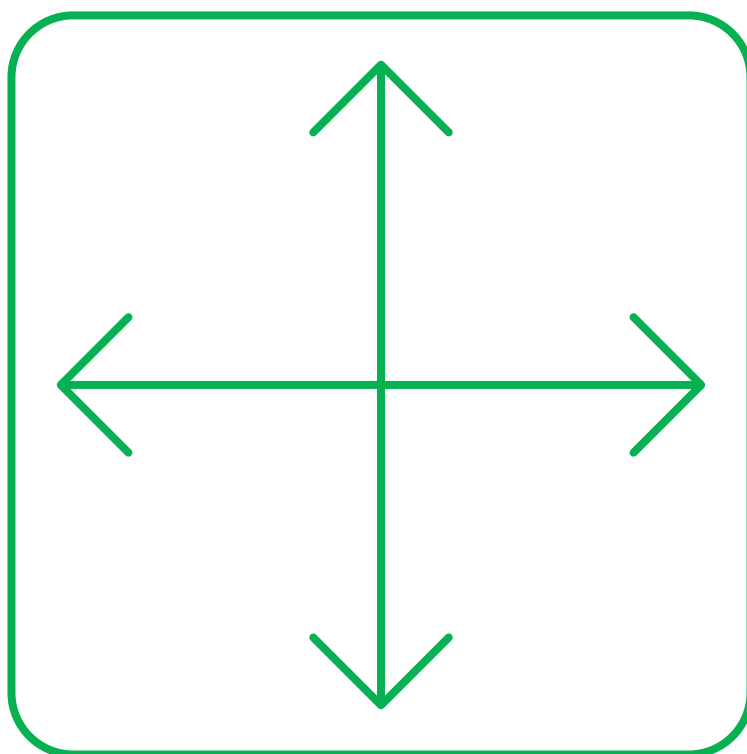
- **Users and Groups** – Allows for managing users and groups provided by vCenter SSO; it also allows you to view users and groups retrieved by vCenter from other identity sources.

**NOTE: The "locals" domain visible in some parts of the vCenter Server GUI refers to local users and groups setup on the underlying OS installed on VCSA – PhotonOS.**



- Configuration – Allows you to configure some additional SSO options:
  - Identity Sources – Add, remove, or manage identity sources available to the vCenter Server.
  - Local Accounts – Change SSO domain settings; **NOTE: It is not possible to change SSO domain name.**
    1. Password Policy – Adjust password lifetime and required complexity.
    2. Lockout Policy – How many logon attempts should be allowed before a lockout and how long should it last.
    3. Token Trustworthiness – Advanced configuration options for logon tokens.
  - Login Message – Set a login message displayed when user logs in.

**NOTE: Only members of the SSO administrator group or the default SSO administrator account can manage SSO settings. Other users will not be able to access the SSO tab.**



## SECTION 5: Performance-tuning and optimizing a VMware vSphere solution

Objective 5.1: Determine effective snapshot use cases . . . . .	83
Objective 5.2: Monitor resources of VCSA in a vSphere environment. . . . .	84
Objective 5.3: Identify impacts of VM configurations . . . . .	85



## Objective 5.1: Determine effective snapshot use cases



### Resource:



[VMware vSphere Product Documentation – Managing Snapshots](#)

**Snapshots** are best described as specific points in time chosen by the VMware vSphere administrator for the virtual machine to be rolled back to in the case of an emergency. Snapshots allow you to return the virtual machine to that particular state by saving its virtual memory, settings, and virtual disks.

Backup software (either native or third party) can also leverage the snapshot process to complete backups. Usually such a process involves taking the snapshot of a VM, saving that snapshot into a storage repository specified in the backup chain, and removing the snapshot once the process completes.

Because of how snapshots are designed, **they were never meant to replace long-term backups**. These should ideally only be used for specific reasons, which we are going to discuss in this objective.

When you take a snapshot, **the state of the virtual disk is preserved**, which prevents the guest operating system from writing to it – a delta or child disk is created.

The delta represents the difference between the current state of the VM disk and the state that existed when you took the previous snapshot. On the VMFS datastore, the delta disk is a sparse disk. **As the VM continues to operate, the delta disk grows – the more I/Os, the quicker delta grows.**

In VMware vSphere 6.7, there are two types of snapshots:

- **VMFSsparse** – This is a redo-log that starts empty, immediately after a VM snapshot is taken. The redo-log expands to the size of its base .vmdk, when the entire .vmdk is rewritten with new data after the VM snapshotting. This redo-log is a file in the VMFS datastore. Upon snapshot creation, the base .vmdk attached to the VM is changed to the newly created sparse .vmdk.
- **SEsparse** – This is a default format for all delta disks on the VMFS6 datastores. On VMFS5, SEsparse is used for virtual disks that are two TB and larger. SEsparse is a format similar to VMFSsparse with some enhancements. This format is space efficient and supports the space reclamation technique. With space reclamation, blocks that the guest OS deletes are marked. The system sends commands to the SEsparse layer in the hypervisor to un-map these blocks. The un-mapping helps to reclaim space allocated by SEsparse once the guest operating system has deleted that data.

**NOTE: Migrating virtual machines with present VMFSsparse snapshots to VMFS6 will cause the snapshot format to change to SEsparse**

In terms of valid use-cases, **snapshots should be mostly used in very specific scenarios** where a quick restore to the original state of a virtual machine following a change could be required.

For example, an administrator could consider taking a snapshot of a virtual machine before implementing a change to its configuration, which could potentially impact either its own performance or the operational status of other workloads in the environment. Concurrent snapshots could be then taken after each change until all planned changes are implemented and the environment is found to be stable enough for the regular backups to suffice.



## Objective 5.2: Monitor resources of VCSA in a vSphere environment




Resource:



[vSphere Monitoring and Performance](#)

Viewing VMware **vCenter Server Appliance health** is done via vCenter Server Appliance Management Interface (VAMI) available at [https://VCSA\\_IP\\_OR\\_FQDN:5480/](https://VCSA_IP_OR_FQDN:5480/).

After logging in, the overall health of the appliance is displayed on the main overview screen:


	Hostname: vcs01.ssh.lab Type: vCenter Server with an embedded ESXi Product: VMware vCenter Server Appliance Version: 7.0.0.10100 Build number: 15952498
---	---

Health Status	
Overall Health	✔ Good (Last checked Apr 12, 2020, 12:26:55 PM)
CPU	✔ Good
Memory	✔ Good
Database	✔ Good
Storage	✔ Good
Swap	✔ Good

To dig deeper into the available stats about the vCenter appliance, we can navigate to the **Monitor** tab. In this section, we can select one of the following views:

- CPU & Memory
- Disks
- Network
- Datastore

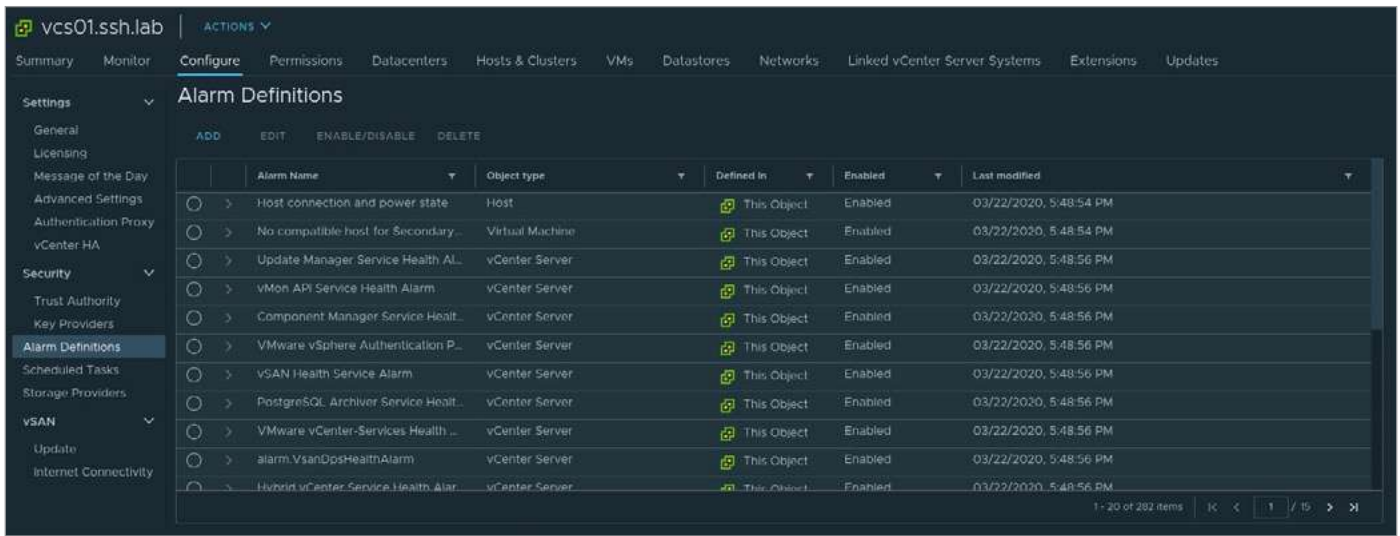
<ul style="list-style-type: none"> <li>Summary</li> <li><b>Monitor</b></li> <li>Access</li> <li>Networking</li> <li>Firewall</li> <li>Time</li> <li>Services</li> <li>Update</li> <li>Syslog</li> </ul>	<table border="1"> <tr> <td><b>CPU &amp; Memory</b></td> <td>Disks</td> <td>Network</td> <td>Database</td> </tr> </table> <p>Apr 12, 2020, 11:31:28 AM - Apr 12, 2020, 12:31:28 PM <span>Last hour</span> <span>↕</span> <span>↻</span></p> 	<b>CPU &amp; Memory</b>	Disks	Network	Database
<b>CPU &amp; Memory</b>	Disks	Network	Database		



The built-in monitoring system of VCSA would trigger an alert in one of the following scenarios by default:

- Disks – Warning 75%, Critical 85%
- Memory – Warning 85%, Critical 95%
- CPU – Warning 75%, Critical 90%

These values can be adjusted in the **Alarm Definitions** in the **Configure** tab of the VCSA object in the main vSphere client.



## Objective 5.3: Identify impacts of VM configurations



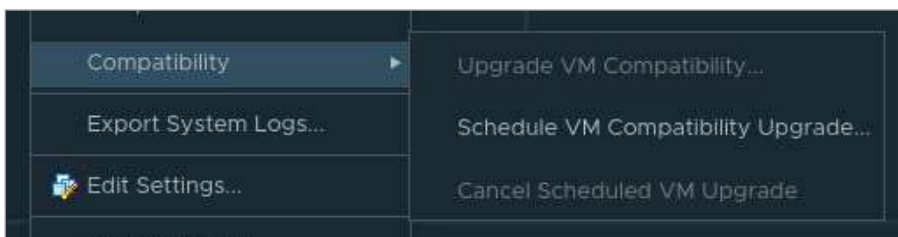
Resource:

**VM configuration** can be split into three main parts – all of them can have a different impact on the VM's performance, its stability, and the overall health of the vSphere environment:

- **Compatibility** – Compatibility settings assigned during the VM's creation determine which ESXi host versions can be used to **provide its compute resources and which hardware features are available** (VMs are not backwards-compatible and you cannot run a VM with hardware version 14 on an ESXi host 6.0 (which supports up to hardware version 11)). VM Compatibility Upgrade can be scheduled from the vSphere client. This upgrade would be then carried out during the next reboot or power-on operation against the machine in question.



[vSphere Virtual Machine Administration](#)





**NOTE: For virtual machines running the Microsoft Windows guest operating system, you should always upgrade VMware Tools first before scheduling a VM compatibility upgrade; otherwise the system might lose its network settings (the currently configured network adapter will be disabled and a brand new one will take over).**

- **Virtual hardware** – This is all the hardware visible to the guest OS installed on the VM. Each VM must have three main hardware components (CPU, RAM, disk).
- **CPU** – Adjusting the CPU settings of the VM tunes the way the virtual machine accesses ESXi host's CPU (threads and processes)
  1. Sockets and cores – Each new vCPU socket or vCPU core increases the maximum available time slots available to the VM to access the physical host's CPU

**NOTE #1: Adding multiple vCPUs or increasing the core count per socket should only be done with extra caution – multiple vCPUs do not increase the virtual machine's performance in a way a physical workload would benefit from an additional CPU core.**

**NOTE #2: Over-provisioning vCPUs is NOT possible – the upper limit of vCPU sockets and/or cores is the total number of logical CPUs on the physical host (i.e. if a host has eight logical CPUs, you can only configure a virtual machine with eight vCPUs).**

2. Reservation – This is a set amount of ESXi host's CPU's processing power bound to the virtual machine
  3. Limit – This is a maximum amount of ESXi host's CPU's processing power which the virtual machine is allowed to consume
  4. Shares – Controls the priority of virtual machines accessing the same physical resource – more shares equal a higher priority to a given resource – in this instance – ESXi host's CPU's cores
  5. CPU Hot Add – Allows for adding additional CPU resources to a virtual machine while it's in a power-on state. **Before enabling CPU Hot Add**, make sure the VM is turned off, is on the compatibility level of ESXi 4.x or later, has the latest VMware Tools version installed, and its guest operating system support CPU hotplug.
- **Memory (RAM)** – Defines the available RAM memory for the virtual machine.
    6. Reservation – This is a set amount of ESXi host's memory bound to the virtual machine. While memory over-provisioning is possible, **it is not recommended since it may cause swapping**, which can greatly impact the virtual machine performance
    7. Limit – This is a maximum amount of ESXi host's RAM which the virtual machine is allowed to consume
    8. Shares – Controls the priority of virtual machines accessing the same physical resource – more shares equal a higher priority to a given resource – in this instance – ESXi host's memory
  - **Disk** – This is a logical object created on the datastores used to store the persistent data used by the guest operating system installed on the virtual machine.
    1. Provisioning type – Controls the way the virtual disk is pre-allocated, created, and stored on a datastore



- a. Thin provisioned – Consumes only the space that it needs initially and grows with time according to demand

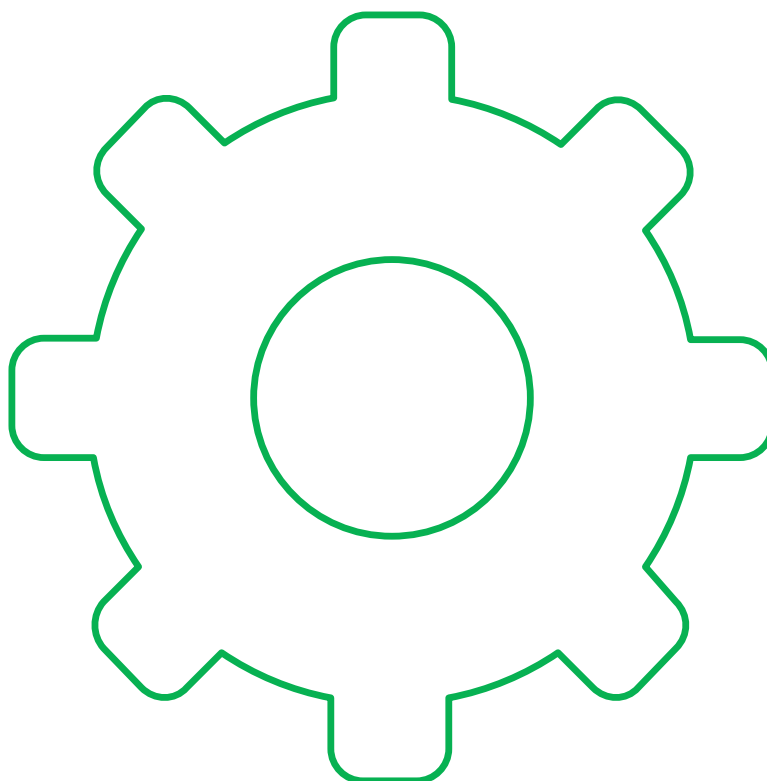
**NOTE: While thin provisioning allows for storage over-provisioning, it is not recommended as it may cause VM failure and severe data loss once the available storage space on the datastores get exhausted.**

- b. Thick provisioned lazily zeroed – Consumes all of its space at the time of its creation to avoid over-subscription but still needs to write the blocks on the storage as space is consumed.
  - c. Thick provisioned eagerly zeroed – Consumes all of the required space still at the time of its creation, and the space is wiped clean of any previous data on the physical media. This can lead to a slow creation time of the virtual disk but increase operation performance.
2. Limit – This is a maximum amount of I/O operations per second allowed against the virtual disk
  3. Shares – Controls the priority of virtual machines accessing the same physical resource – more shares equal a higher priority to a given resource – in this instance – I/O operations on the storage device
- **Virtual Machine Options** – Available under the **VM Options** tab on the virtual machine's settings page.
    - General Options – Virtual machine name and location of the virtual machine configuration file and virtual machine working location
    - Encryption Options – Controls the encryption of virtual machines managed by a vCenter server; generally speaking, there are two types of VM encryption available: VM files encryption (vmx, virtual disks, etc.) and encrypted vMotion

**NOTE: Encryption is only supported for virtual machines managed by a vCenter in a trusted relationship with a KMS server**

- VMware Remote Console Options – Locking behaviour and settings for simultaneous connections
- VMware Tools – Power Controls behaviour, VMware Tools scripts, automatic upgrades, and time synchronization between the guest and host. Similar to compatibility version, a VMware Tools upgrade can also be carried out while power-cycling the virtual machine
- Power Management – Virtual machine suspend behavior (suspend the virtual machine or leave it running while the guest operating system enters standby mode) and wake on LAN.
- Boot Options – Managing virtual machine boot options such as adding a delay before booting, forcing entry into the BIOS, or EFI setup screen (useful for recovering or reinstalling the guest operating system) and adjusting reboot options.
- Fibre Channel NPIV – Control virtual machine access to LUNs on per-virtual machine basis. N-port ID virtualization (NPIV) provides the ability to share a single physical Fibre Channel HBA port among multiple virtual ports, each with unique identifiers.
- Advanced Options – Enable and disable logging, configure debugging and statistics, and change the swap file location. This also allows for changing the latency sensitivity and adding custom configuration parameters.





## SECTION 7: Administrative and operational tasks in a VMware vSphere solution

Objective 7.1: Manage virtual networking . . . . .	89	Objective 7.10: Create and manage templates. . . . .	113
Objective 7.2: Manage datastores. . . . .	93	Objective 7.11: Manage different VMware vCenter Server objects . . . . .	114
Objective 7.3: Configure a storage policy . . . . .	97	Objective 7.12: Set up permissions on datastores, clusters, vCenter and hosts . . . . .	116
Objective 7.4: Configure host security . . . . .	99	Objective 7.13: Identify and interpret affinity/anti-affinity rules . . . . .	119
Objective 7.5: Configure role-based user management . . . . .	101	Objective 7.14: Understand use cases for alarms. . . . .	123
Objective 7.6: Configure and use vSphere compute and storage cluster options . . . . .	104	Objective 7.15: Utilize VMware vSphere Update Manager (VUM) . . . . .	124
Objective 7.7: Perform different types of migrations. . . . .	108	Objective 7.16: Configure and manage host profiles . . . . .	127
Objective 7.8: Manage resources of a vSphere environment . . . . .	110		
Objective 7.9: Create and manage VMs using different methods . . . . .	111		

## Objective 7.1: Manage virtual networking

### Resource:



[vSphere Networking](#)

Managing **vSphere virtual networking** can generally be discussed in two parts:

- **Basic virtual networking configuration** – This is applicable to both the default vSphere Standard Switch (vSS) and the cluster-focused vSphere Distributed Switch (vDS)
  - Assigning physical network interfaces (uplinks) to a virtual switch
  - Creating (distributed) port groups
  - Setting up a VLAN ID on a (distributed) port group
  - Changing the MTU value on a virtual switch
  - Setting up teaming and failover on the physical network interfaces connected to the virtual switch
    1. Route based on IP hash
    2. Route based on source MAC hash
    3. Route based on originating virtual port
    4. Explicit failover order
  - Basic traffic shaping (outbound only): average bandwidth, peak bandwidth, and burst size in KB
- **Advanced (or additional) virtual networking configuration** – which is only applicable to the vDS
  - Traffic filtering and marking
  - VLAN trunking, PVLANS
  - Monitoring (NetFlow)
  - Security
  - Traffic shaping (both inbound and outbound)
  - LACP
  - Port mirroring
  - Health check for VLAN, MTU, failover, and overall configuration

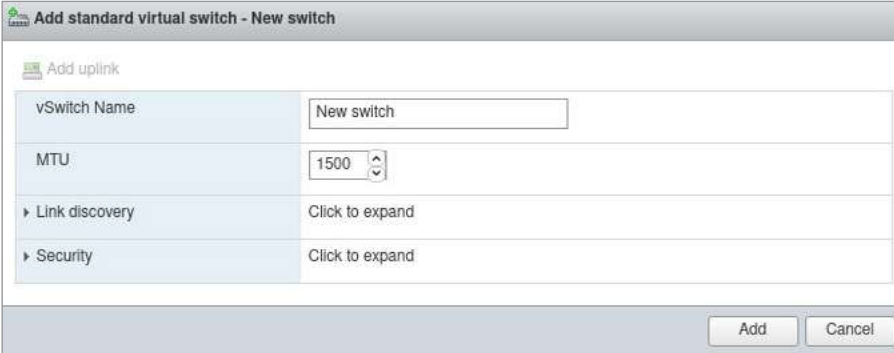
**NOTE: Some settings on the vSphere Standard Switch can only be changed in the ESXi Host Client.**

To **create a new vSphere Standard Switch (vSS)**, log into the Host Client for the ESXi host in question and navigate to **Networking > Virtual Switches**

To create a new vSS, select the **Add standard virtual switch** button.



The following screen is similar to editing an existing vSS.



**Add standard virtual switch - New switch**

Add uplink

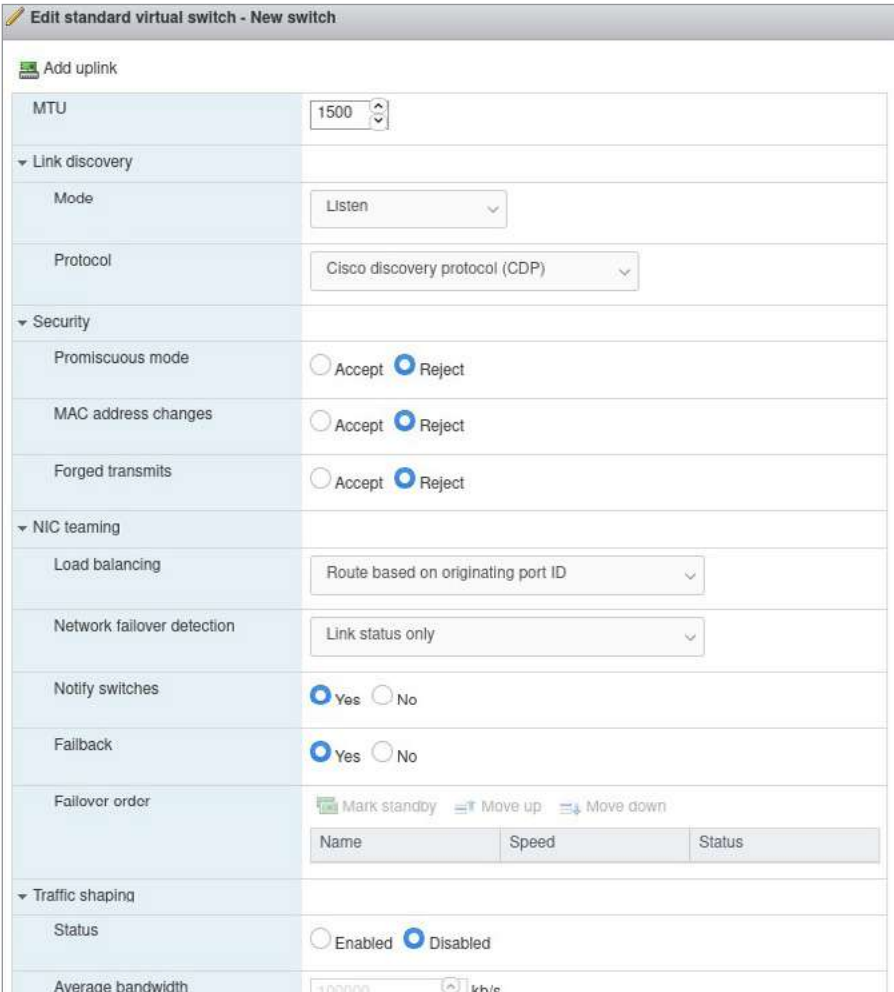
vSwitch Name	New switch
MTU	1500
▶ Link discovery	Click to expand
▶ Security	Click to expand

Add Cancel

Creating a vSS is easy and requires just a name for the new virtual switch. Everything else is optional.

Once the virtual switch is created, we can edit it to adjust some optional settings if required, such as MTU, security policies, NIC teaming, etc.

After adding a new virtual switch, we can **add uplinks** to it by using the button **Add uplink** in the same page of the ESXi web interface.



**Edit standard virtual switch - New switch**

Add uplink

MTU	1500						
▼ Link discovery							
Mode	Listen						
Protocol	Cisco discovery protocol (CDP)						
▼ Security							
Promiscuous mode	<input type="radio"/> Accept <input checked="" type="radio"/> Reject						
MAC address changes	<input type="radio"/> Accept <input checked="" type="radio"/> Reject						
Forged transmits	<input type="radio"/> Accept <input checked="" type="radio"/> Reject						
▼ NIC teaming							
Load balancing	Route based on originating port ID						
Network failover detection	Link status only						
Notify switches	<input checked="" type="radio"/> Yes <input type="radio"/> No						
Failback	<input checked="" type="radio"/> Yes <input type="radio"/> No						
Failover order	<input type="checkbox"/> Mark standby <input type="checkbox"/> Move up <input type="checkbox"/> Move down <table border="1"> <thead> <tr> <th>Name</th> <th>Speed</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	Name	Speed	Status			
Name	Speed	Status					
▼ Traffic shaping							
Status	<input type="radio"/> Enabled <input checked="" type="radio"/> Disabled						
Average bandwidth	1000000 kb/s						

Lastly, we can add a new standard port group to the vSS by going to the **Port groups** tab in the ESXi Host Client. Clicking the **Add port group** button will launch a **New Port Group** wizard.

Port groups			Virtual switches	Physical NICs	VMkernel NICs	TCP/IP stacks	Firewall rules
Add port group    Edit settings    Refresh    Actions							
Name	Active ports	VLAN ID					
VM Network	8	0					
Management Network	1	0					
iSCSI	1	0					

**Add port group - New port group**

Name	<input type="text" value="New port group"/>
VLAN ID	<input type="text" value="0"/>
Virtual switch	<input type="text" value="vSwitch0"/>
Security	<a href="#">Click to expand</a>

Again, the only required setting here is the name for the new port group; all other settings are optional but can be adjusted at a later stage if required (VLAN ID, security policy, NIC teaming, traffic shaping, etc.).

**Security**

Promiscuous mode:  Accept  Reject  Inherit from vSwitch

MAC address changes:  Accept  Reject  Inherit from vSwitch

Forged transmits:  Accept  Reject  Inherit from vSwitch

**NIC teaming**

Load balancing:

Network failover detection:

Notify switches:  Yes  No  Inherit from vSwitch

Failback:  Yes  No  Inherit from vSwitch

Override failover order:  Yes  No

Failover order

Mark standby
 Mark unused
 Move up
 Move down

Name	Speed	Status
vmnic1	1000 Mbps, full duplex	Active

**Traffic shaping**

Status:  Enabled  Disabled  Inherit from vSwitch

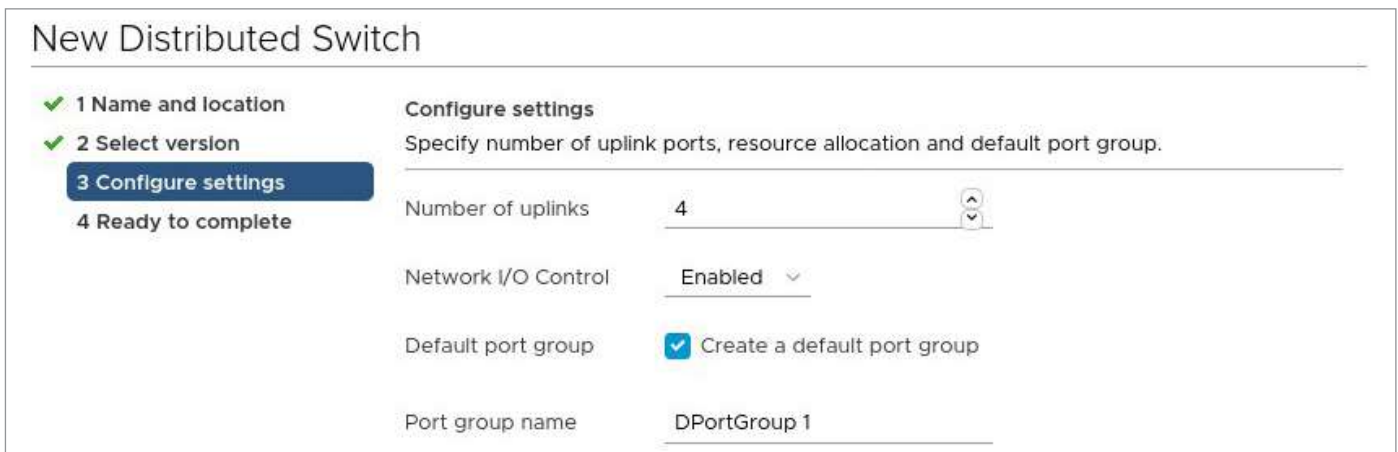
Average bandwidth:  kb/s

Peak bandwidth:  kb/s

Having said all this, we can move on to the **vSphere Distributed Switch (vDS)**. Before we begin discussing different configuration options available on a vDS, we need to create one. To do so, navigate to the **Networking** tab in the vSphere Client, right-click a data center object and select **Distributed Switch > New Distributed Switch**:



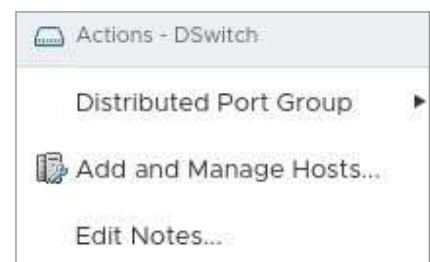
Creating a vDS is a bit more complicated than setting up a vSS on a host, aside from giving it a name and selecting a version (vDS version 5 can be connected to ESXi 5.x and up, vDS version 6 to ESXi 6.x and up, etc.), **you need to specify the number of allowed uplinks**, name of the first port group (optional), and if network I/O control should be enabled or not:



Once the new vDS is created, **you must connect to it ESXi hosts and virtual machines**. This can be done by right-clicking on the new vDS and selecting the option **Add or Manage Hosts...**

**This wizard also allows you to manage already connected hosts** or remove hosts which should no longer use the vDS.

**NOTE: Migrating from a vSS to a vDS can be tricky since the hosts can lose all network connectivity if you try to migrate a single management uplink currently assigned to vSS to a vDS; consider temporarily choosing a spare uplink first to simplify the migration.**



After connecting the hosts, you can configure some **additional settings on the vDS**. One of the most important settings is the **security policy** – this includes three options:

- Promiscuous Mode – Set to reject by default; setting this option to accept will enable the guest OS to receive all traffic observed on the connected vSwitch or Port Group.
- MAC address changes – Set to reject by default; setting this option to accept will cause the connected hosts to accept requests to change the effective MAC address to a different address than the initial MAC address.
- Forged transmits – Set to reject by default; setting this option to accept will stop the host from comparing the source and effective MAC addresses transmitted from a virtual machine.

Another useful, but optional setting on a vDS is the **load balancing** – in case of a vDS, there are five modes available:

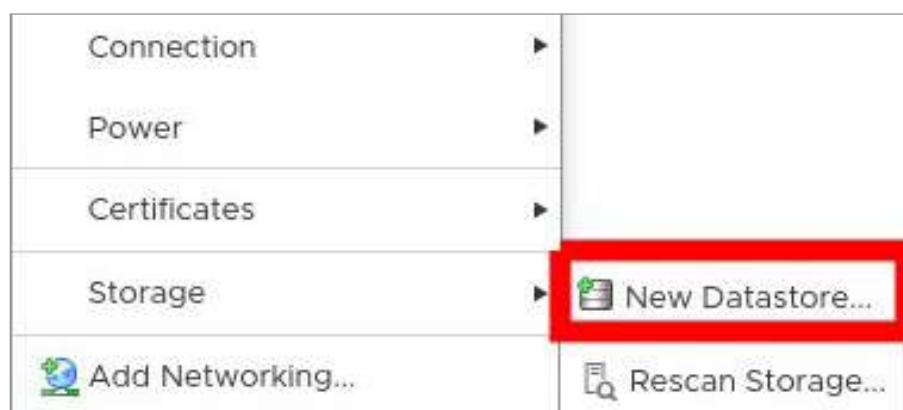
- Route based on IP hash – The virtual switch selects uplinks for virtual machines based on the source and destination IP address of each packet.
- Route based on source MAC hash – The virtual switch selects an uplink for a virtual machine based on the virtual machine MAC address.
- Route based on originating virtual port – Each virtual machine running on an ESXi host has an associated virtual port ID on the virtual switch. To calculate an uplink for a virtual machine, the virtual switch uses the virtual machine port ID and the number of uplinks in the NIC team. After the virtual switch selects an uplink for a virtual machine, it always forwards traffic through the same uplink for this virtual machine as long as the machine runs on the same port. The virtual switch calculates uplinks for virtual machines only once, unless uplinks are added or removed from the NIC team.
- Use explicit failover order – No actual load balancing is available with this policy. The virtual switch always uses the uplink that stands first in the list of active adapters from the failover order and that passes failover detection criteria. If no uplinks in the active list are available, the virtual switch uses the uplinks from the standby list.
- Route based on physical NIC load – Based on route based on Originating Virtual Port, where the virtual switch checks the actual load of the uplinks and takes steps to reduce it on overloaded uplinks. It's available only for vSphere Distributed Switch. The distributed switch calculates uplinks for virtual machines by taking their port ID and the number of uplinks in the NIC team. The distributed switch tests the uplinks every 30 seconds and if their load exceeds 75% of usage, the port ID of the virtual machine with the highest I/O is moved to a different uplink.

## Objective 7.2: Manage datastores

In VMware vSphere 6.7, you can carry out the following **operations against datastores**:

### Create a new datastore

Right-click on an ESXi host and select Storage > New Datastore:



### Resource:



[vSphere Resource Management](#)

In the New Datastore wizard, select the desired datastore type:

## New Datastore

**1 Type**

2 Name and device selection

3 VMFS version

4 Partition configuration

5 Ready to complete

**Type**

Specify datastore type.

VMFS  
Create a VMFS datastore on a disk/LUN.

NFS  
Create an NFS datastore on an NFS share over the network.

vVol  
Create a Virtual Volumes datastore on a storage container connected to a storage provider.

Then select the device:

✓ 1 Type

**2 Name and device selection**

3 VMFS version

4 Partition configuration

5 Ready to complete

**Name and device selection**

Select a name and a disk/LUN for provisioning the datastore.

Datastore name: NewDatastore

Name	LUN	Capacity	Hardware...	Drive T...	S
Local VMware Disk (mpx....	0	70.00 GB	Unknown	Flash	E

Then select the VMFS version:

- VMFS6 is recommended and should be used in most environments
- VMFS5 should only be used if compatibility with older ESXi versions is required

✓ 1 Type

✓ 2 Name and device selection

**3 VMFS version**

4 Partition configuration

5 Ready to complete

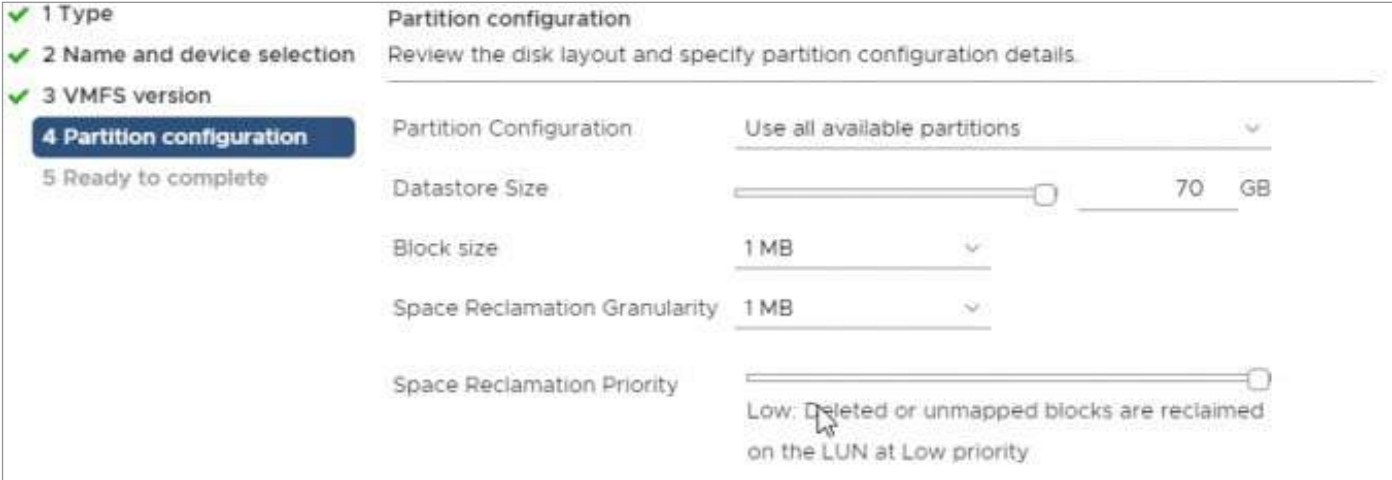
**VMFS version**

Specify the VMFS version for the datastore:

VMFS 6  
VMFS 6 enables advanced format (512e) and automatic space reclamation support.

VMFS 5  
VMFS 5 enables 2+TB LUN support.

And then adjust partitioning if needed:



**1 Type**  
**2 Name and device selection**  
**3 VMFS version**  
**4 Partition configuration**  
 5 Ready to complete

**Partition configuration**  
 Review the disk layout and specify partition configuration details.

Partition Configuration Use all available partitions

Datastore Size 70 GB

Block size 1 MB

Space Reclamation Granularity 1 MB

Space Reclamation Priority Low: Deleted or unmapped blocks are reclaimed on the LUN at Low priority

## Administrative operations

- Rename a datastore – Change the name of a datastore:
  - Right-click on a datastore and select **Rename...**
  - Input a new name for the datastore and select **OK**

**NOTE: This procedure is a low-risk operation and will not impact the environment, as the datastore name is only an alias for the actual datastore ID used by the ESXi OS in the backend.**

- Mount/unmount a datastore
  - To mount – Right-click on a datastore and select **Mount Datastore...**
  - To unmount – After making sure the datastore is NOT used by any running processes: right-click on a datastore and select **Unmount Datastore...**
- Delete a datastore
  - Right-click on a datastore and select **Delete Datastore**
  - A warning message will be displayed; select **Yes** if you wish to proceed

**NOTE: Deleting a datastore will not be possible in most high-risk situations, as vCenter Client will not allow for the operation to complete. However, you should still pay extra caution while deleting a datastore from an ESXi host.**

- The datastore in question will then disappear from the datastore list

**NOTE: Data saved on the datastore will be automatically deleted following this operation.**

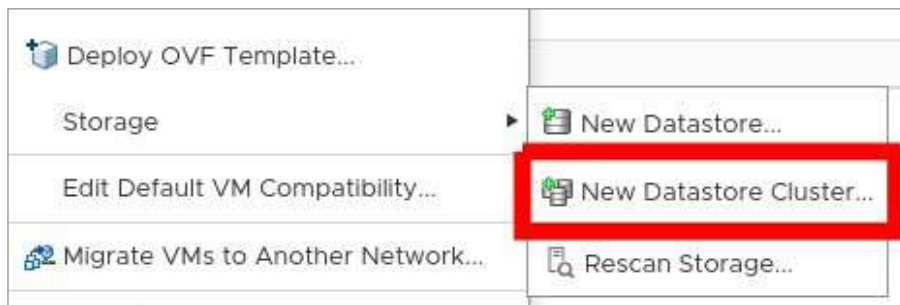
## Create a datastore cluster

Datastores can be grouped together into a datastore cluster, which allows for easier management of multiple datastores; for example, all datastores accessible to multiple hosts can be grouped together into a datastore cluster, so it would be easier to distinguish them from non-shared datastores in the environment. Datastore clusters also allow you to configure **Storage DRS**, which automatically manages storage resources by, for example, moving VMs to least used datastores.

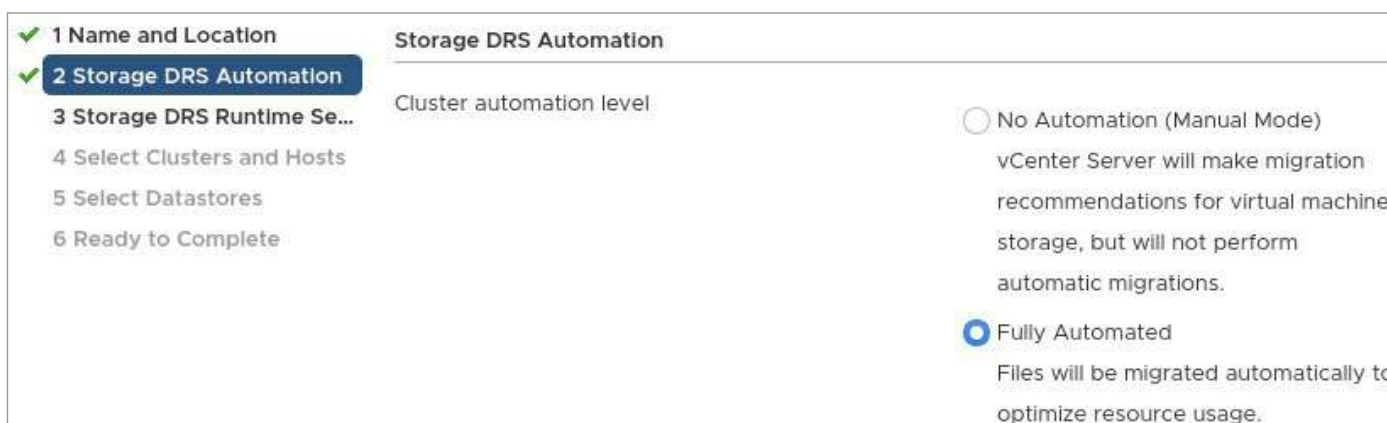
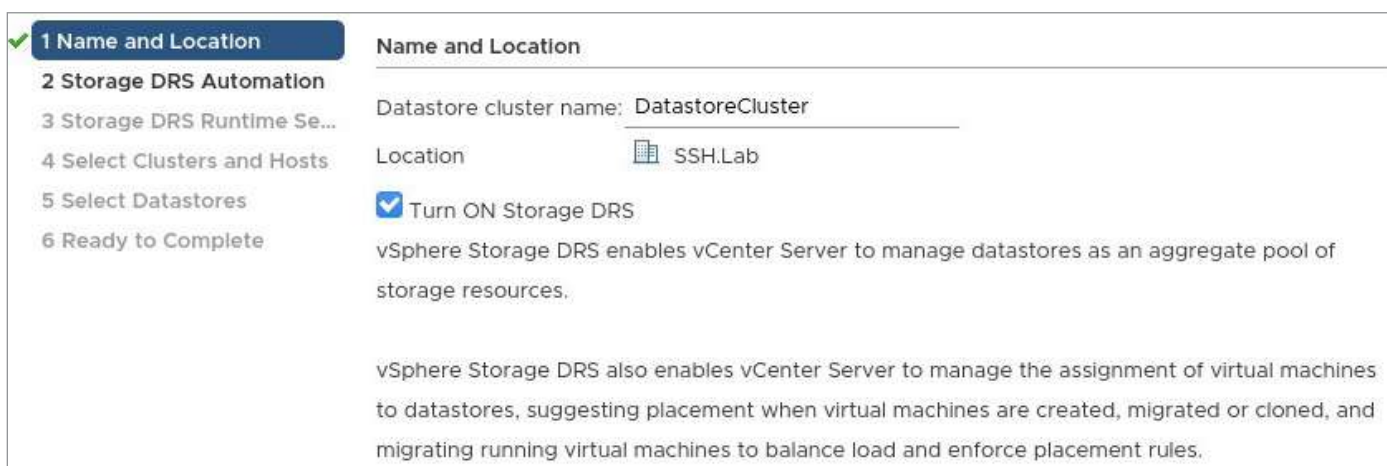


**To create a datastore cluster:**

Right-click on a data center object in vSphere Client and select **New Datastore Cluster**:



In the **New Datastore Cluster** wizard, select a datastore name and enable Storage DRS, if required. If we choose to enable Storage DRS, on the next page we need to select Storage DRS automation level and we can adjust some advanced settings, such as VM evacuation level or I/O balance automation level, to further tune Storage DRS workflow.





Next, select ESXi hosts, which should be a part of the datastore cluster:

**Select Clusters and Hosts**

Filter Selected (0)

Clusters Standalone Hosts

Filter

<input type="checkbox"/> Name ↑	State	Status
<input type="checkbox"/> esx01.ssh.lab	Connected	✓ Normal

Finally, select the datastores which should be added to the datastore cluster:

**Select Datastores**

Show datastores connected to all hosts

Filter Selected (0)

Filter

<input type="checkbox"/> Name ↑	Host Connection Status	Capacity
<input type="checkbox"/> esx01-local01	✓ All Hosts Connected	327.75 GB
<input type="checkbox"/> esx01-local02	✓ All Hosts Connected	698.5 GB
<input type="checkbox"/> nas01-iscsi01	✓ All Hosts Connected	4.9 TB

## Objective 7.3: Configure a storage policy

**Storage policies** are used to configure how virtual machines are placed on datastores, how VM files are managed, configuring data caching or replication, and much more.

You can assign a storage policy when creating a VM or by editing an existing VM and changing its storage policy under Virtual Hardware. **Storage policies are applied on a per virtual disk basis**, which means that each virtual disk associated with a VM can have a different storage policy, if required.



### Resource:

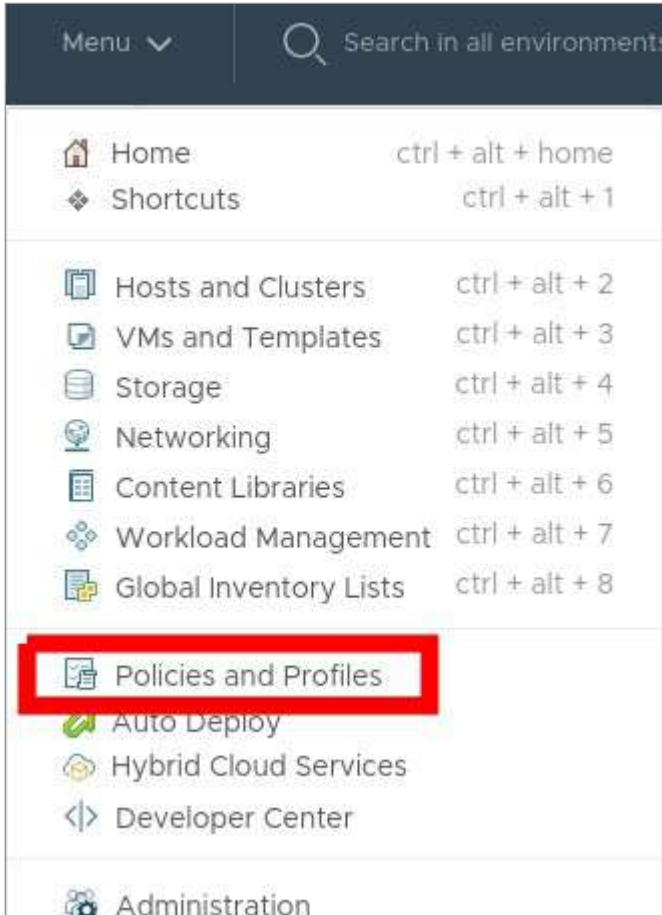


[vSphere Resource Management](#)



In vSphere 6.7, there are multiple predefined storage policies but you can also create your own. To create a new Storage Policy:

- Select **Policies and Profiles** from the main vSphere Client menu



- Select **VM Storage Policies** from the **Policies and Profiles** section
- Select the **Create VM Storage Policy** button located above the list of currently configured Storage Policies



- Select a vCenter Server on which the Storage Policy should be created
- Input a recognizable name for the Storage Policy and enter an optional description if required
- Select the Storage Policy structure options:
  - Enable Host base services – These include data encryption, I/O control, caching, etc.; these rules will be applied in addition to any datastore specific rules
  - Enable Datastore specific rules – These rules will be applied when VMs are placed on the specific storage type:
    1. vSAN specific rules – Site disaster tolerance, cache reservation, number of disk stripes per object, etc.
    2. Tag based placement rules – This allows you to associate VM tags with datastores tags; i.e. store VMs tagged with **Production Workload** tag only on datastores tagged with **Production Storage** tag
- Review storage compatibility (which storage would be in compliance with the new policy)
- Review the summary and save the Storage Policy

## Objective 7.4: Configure host security

The ESXi hosts have been designed with many **security features baked into the kernel**. Examples of such security features include security profiles, CPU and memory isolation, and certificate management. Some of these features come pre-configured and some require additional configuration to function, depending on the use case.

By default, freshly deployed ESXi hosts have some **security risks already mitigated**:

- SSH access is disabled and ESXi shell is only available via direct access;
- Only the bare minimum TCP/IP ports are open (for example, ports required for ESXi management by a vCenter server);
- Client communication is secured with SSL certificated based on PKCS#1 SHA-256 with RSA encryption as the signature algorithm;
- Insecure services, such as plain FTP, are not installed.

**The security of ESXi hosts can be increased** further by hardening. Some examples of security hardening available for ESXi hosts include:

- LDAP/Kerberos-based logon via Microsoft Active Directory or other compatible directory service;
  - in Microsoft Active Directory, a special group must be created – ESX Admins; members of this group will be automatically assigned administrator privileges on the ESXi host when it's joined to an AD.
  - Joining an ESXi host to a directory service can be done with a vSphere Client by navigating to Configure > System > Authentication Services > Join Domain
- Certificate replacement and smart card authentication;

## Resource:

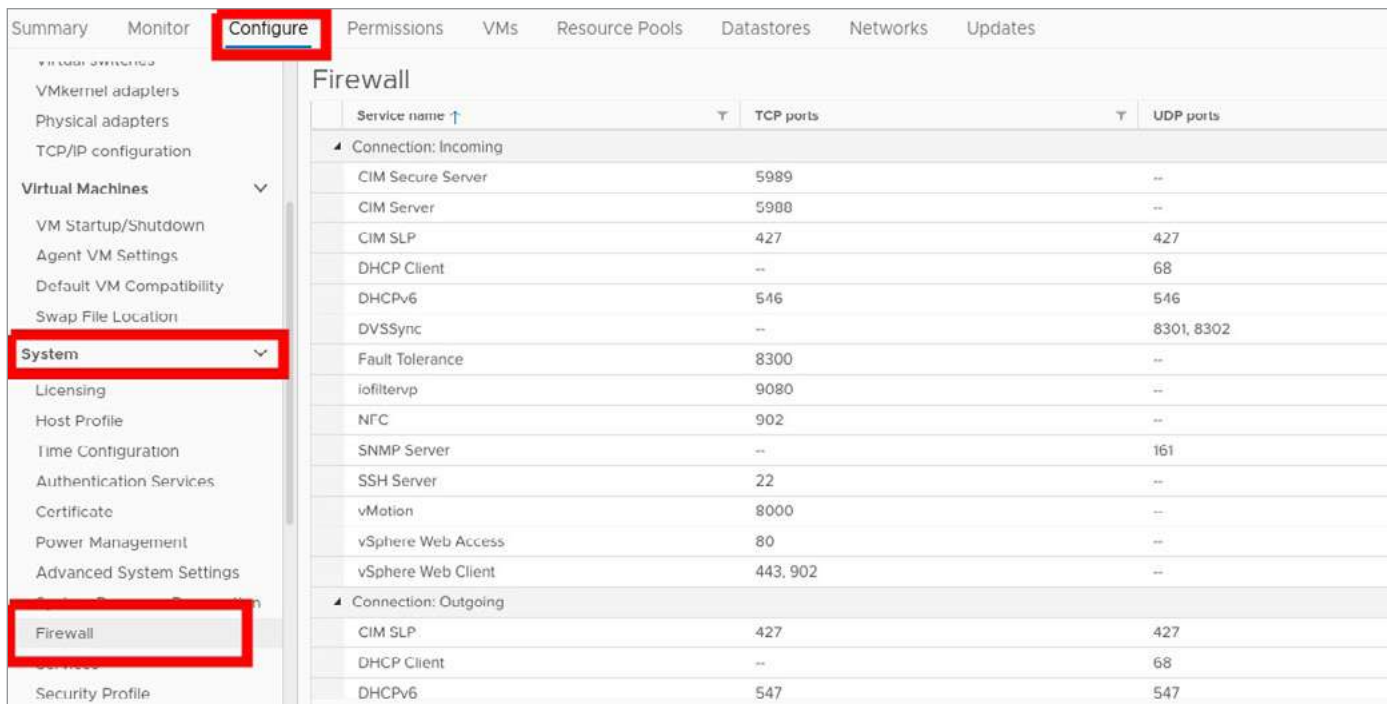


[vSphere Security](#)

- Locking down the firewall even further by disabling unused services and closing unnecessary ports;
- Implementing UEFI Secure Boot;
- Enabling ESXi **Lockdown mode**
  - Lockdown Mode allows you to completely disable direct access to the ESXi host; with Lockdown Mode on, ESXi hosts must be administered from a vCenter Server. This mode should be used in caution, as it is possible to lock yourself out – locked out hosts can only be “reopened” by redeployment (ESXi OS reinstall).
  - There are two Lockdown Modes available:
    1. Normal: The host can only be managed through vCenter Server with an exception of the users who are on the Exception Users list and also have administrator privileges in vSphere – these users can still use direct console access or connect via SSH to access the ESXi shell. Adding local users to an Exception Users list can be done by modifying the DCUI.Access parameter in **Advanced System Settings** on ESXi host
    2. Strict: The hosts can only be managed through vCenter Server with no exceptions by default. Some exceptions can be set via ESXi advanced options.
- Setting shell timeouts with ESXi.set-shell-timeout
- Restricting administrative privileges. Consider creating custom roles with the least amount of privileges for users who do not require access to all vCenter objects; keep in mind the default administrator role allows access to all objects

**Managing ESXi firewall rules** can be done via vSphere Client:

- Select an ESXi host to adjust a firewall on and navigate to **Configure > System > Firewall**



Service name ↑	TCP ports	UDP ports
Connection: Incoming		
CIM Secure Server	5989	--
CIM Server	5988	--
CIM SLP	427	427
DHCP Client	--	68
DHCPv6	546	546
DVSSync	--	8301, 8302
Fault Tolerance	8300	--
iofiltervpx	9080	--
NFC	902	--
SNMP Server	--	161
SSH Server	22	--
vMotion	8000	--
vSphere Web Access	80	--
vSphere Web Client	443, 902	--
Connection: Outgoing		
CIM SLP	427	427
DHCP Client	--	68
DHCPv6	547	547



- Click the **Edit** button to enter the **Edit Security Profile** menu – here you can enable and disable firewall rules, and adjust start-up policies to have a service in question started up with the host or once a port is active. It is also possible to configure allowed incoming IP addresses on some services.

Managing the firewall rules can also be done via ESXCLI: `esxcli network firewall ruleset set -e true -r sshClient`

## Objective 7.5: Configure role-based user management



Resource:



[vSphere Security](#)

**User roles** allow you to configure access levels to different parts of a vSphere environment. For example, one of the default and most used roles in vSphere is the administrator role – this role allows users to manage every single object, adjust all the available settings, and perform any action against any object in the inventory.

By creating **custom roles**, we can control what objects the users have access to, which parts of the vSphere deployment should not be available to them, and improve the security of the vSphere environment by requiring users to request their privilege level to be escalated once they hit a limitation enforced by the currently assigned role.

When designing a new role, there are a few things we need to take into consideration:

- How the role should be distributed among the users? Should it be assigned on a per user basis or should it be assigned to a group?
- Which objects should the new role be applicable to? If the environment in question has multiple clusters, data centers, resource pools, etc., it might be a good idea to focus on creating roles directly responsible for said object, i.e. **Production Administrator**, which would only allow Administrator (full) access to the "Production" ESXi cluster.
- What **permission level** should be granted by the role? Roles don't have to always come with a full level access to an object. Consider lowering the access level, depending on the role in question, i.e. a storage administrator might not need to be able to initiate a power down on the ESXi hosts.

Managing roles must be done via vSphere Client. To do so, navigate to **Administration > Access Control > Roles**

Roles
Roles provider: SSH.LAB
+ [Add] [Edit] [Delete]
Administrator
Read-only
No access
AppdApplianceUser



Here you can create, edit or clone roles. To create a role:

- Select the plus icon above the existing roles list
- Using the New Role wizard, browse, and select which privileges should be assigned to the role

**New Role**

Alarms  
AutoDeploy  
Certificate Authority  
Certificate Management  
Certificates  
Cns  
Compute Policy  
Content Library  
Cryptographic operations  
Datacenter  
Datastore  
Datastore cluster  
Distributed switch  
ESX Agent Manager  
Extension  
External stats provider  
Folder  
Global

All Alarms Privileges All | Selected | Unselected

Acknowledge alarm

Create alarm

Disable alarm action

Disable or enable alarm on entity

Modify alarm

Remove alarm

Set alarm status

CANCEL BACK NEXT

- In this example, the new role has all the privileges associated with the **Alarms** privilege category
- After choosing all the necessary privileges, select **Next** and pick a name for the new role. You can also add an optional description for the role, if required.

**New Role**

Role name \_\_\_\_\_

Description





Once the new role has been created, you can associate it with a user or a group in the **Global Permissions** menu accessible under **Administration > Access Control**.

A **global permission** is the base level of access to all vSphere objects. If a user is defined as an administrator in global permissions, that user will have administrator level access to all objects associated with that vCenter Server.

Permissions can also be granted to users or groups using roles on a per object level. For example, to assign a **Read-Only** role to a user or a group of users on a particular ESXi host:

- In vSphere Client, select a desired host and open the **Permissions** tab

User/Group ↑	Role
SSH.LAB\Administrator	Administrator
SSH.LAB\Administrators	Administrator
SSH.LAB\ApplVereset	ApplVereset
SSH.LAB\ju	Administrator
SSH.LAB\NsxAdministrators	NsxAdministrator
SSH.LAB\NsxAuditors	NsxAuditor
SSH.LAB\NsxViAdministrators	NsxViAdministrator

- Click the plus icon to add a new permission
- Select a user or a group and choose a role which should be assigned. Propagate to Children – Checking this option would propagate the newly added permission to all objects lower down in the hierarchy, i.e. all VMs running on said ESXi host

**Add Permission** | esx01.ssh.lab

Domain: localos

User/Group: root

Role: Read-only

Propagate to children





## Objective 7.6: Configure and use vSphere compute and storage cluster options



### Resource:



[vSphere Availability](#)

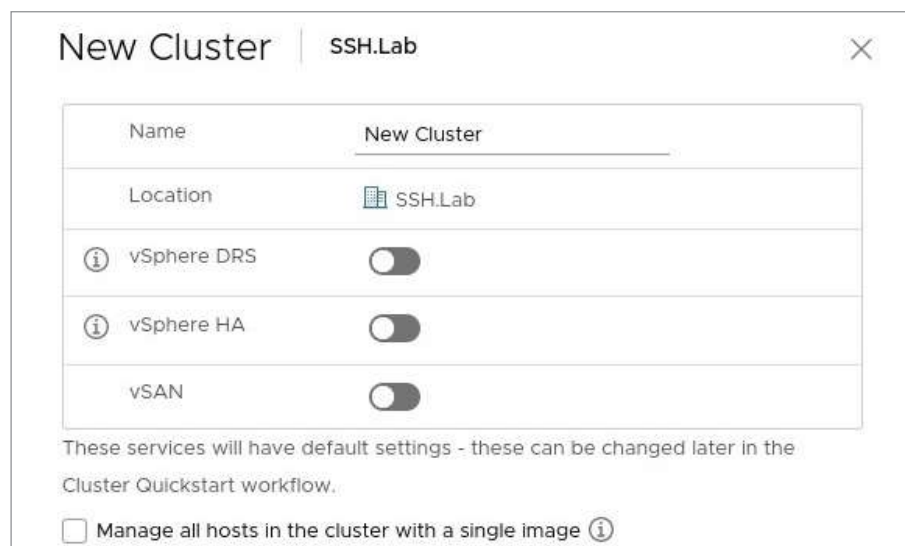
Before creating an **ESXi cluster**, you need to verify you meet several prerequisites:

- Your account has enough privileges to create a cluster object;
- The ESXi hosts you plan to add to the cluster are in the same data center and are on the **same ESXi version and patch level**; it is a best practice to ensure the hosts have similar hardware installed (namely CPU type and memory);
- You have a root account password for each of the ESXi hosts;
- There is no pre-existing configuration on the hosts, which would prohibit you from creating a cluster.

#### To create an ESXi cluster:

- Right-click on the relevant data center object and select **New Cluster**
- In the **New Cluster** wizard:
  - Enter a name for the new cluster
  - Select supplementary services to activate:
    1. vSphere DRS
    2. vSphere HA
    3. vSAN

Each of these services can also be turned on at a later stage once the cluster is fully populated with ESXi hosts – to do so, open the **Configure** tab on the cluster object.

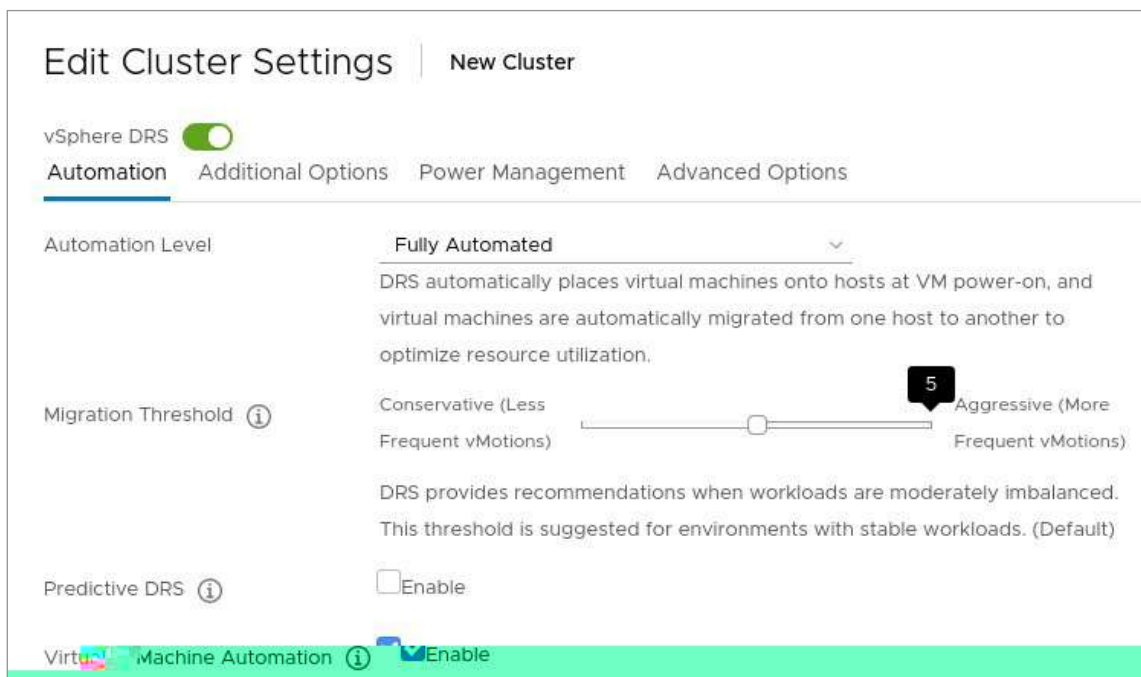


- After clicking the **OK** button, a new cluster object will be created. ESXi hosts can then be dragged into the cluster, which will automatically start a cluster configuration process (this might take a few minutes, depending on the additional options set during cluster creation)



Once the ESXi cluster is populated with target hosts, you can open its **Configure** tab to adjust additional settings for the supplementary services. Depending on the service you want to set up, navigate to either **vSphere DRS**, **vSphere Availability (aka vSphere HA)**, or **vSAN**.

Some configuration options available under **vSphere DRS**:



- **Automation Level** – How automated the vSphere DRS system should be on the cluster
  - Fully Automated – DRS automatically places virtual machines onto hosts at VM power-on and virtual machines are automatically migrated from one host to another to optimize resource utilization.
  - Partially Automated – DRS automatically places virtual machines onto hosts at VM power-on. Migration recommendations need to be manually applied or ignored.
  - Manual – DRS generates both power-on placement recommendations and migration recommendations for virtual machines. Recommendations need to be manually applied or ignored.
- **Migration Threshold** – Specifies how aggressively DRS recommends vMotions. Recommendations are generated automatically based on resources demanded by the virtual machines, resource allocation settings (reservations, limits, and shares), the resources provided by each host, and the cost of migrating VMs. The more conservative the setting, the less frequent the vMotions. This can be adjusted on a scale from 1 (conservative – less frequent vMotions) to 5 (aggressive – more frequent vMotions):
  - Threshold 1 – DRS will only apply recommendations that must be taken to satisfy cluster constraints, like affinity rules and host maintenance. DRS will not try to correct host imbalance at this threshold.



- **Threshold 2** – DRS only provides recommendations when workloads are extremely imbalanced or virtual machine demand is not being satisfied on the current host.
- **Threshold 3** – DRS provides recommendations when workloads are moderately imbalanced. This threshold is suggested for environments with stable workloads. This is also the default migration threshold level for new vSphere DRS deployments.
- **Threshold 4** – DRS provides recommendations when workloads are fairly imbalanced. This threshold is suggested for environments with bursting workloads.
- **Threshold 5** – DRS provides recommendations when workloads are even slightly imbalanced and marginal improvement may be realized. For dynamic workloads, this may generate frequent vMotion recommendations.
- **Predictive DRS** – This requires you to deploy and configure vRealize Operations Manager. If this option is selected, DRS will respond to forecasted metrics provided by vRealize Operations Manager. Only forecasted metrics with high confidence will be considered by DRS to balance the cluster’s workloads prior to predicted utilization spikes and resource contention.
- **Virtual Machine Automation** – Enables or disables overriding vSphere DRS configuration on VM-level (for example, preventing some VMs from being automatically migrated in Fully Automated mode)
- **Power Management** (also known as DPM) – vSphere DRS can use Wake-on-LAN, IPMI, or iLO to power on hosts. When using IPMI or iLO, configure IPMI or iLO separately for each participating host prior to enabling DPM. For all power-on methods, test exit standby for each participating host prior to enabling DPM. Overrides for individual hosts can be set from the Host Options page. DPM can be also configured to be fully automated, partially automated, or manual; it also has five threshold levels to choose from.

Some configuration options available under **vSphere Availability (HA)**:

The screenshot shows the 'Edit Cluster Settings' interface for vSphere HA. At the top, there are tabs for 'New Cluster' and 'vSphere HA', with 'vSphere HA' being the active tab. Below the tabs, there are sub-tabs for 'Failures and responses', 'Admission Control', 'Heartbeat Datastores', and 'Advanced Options'. The 'Failures and responses' sub-tab is selected. A descriptive text states: 'You can configure how vSphere HA responds to the failure conditions on this cluster. The following failure conditions are supported: host, host isolation, VM component protection (datastore with PDL and APD), VM and application.' Below this text, there is a toggle for 'Enable Host Monitoring' which is turned on. A table lists the following failure conditions and their configured responses:

Failure Condition	Response
Host Failure Response	Restart VMs
Response for Host Isolation	Disabled
Datastore with PDL	Power off and restart VMs
Datastore with APD	Power off and restart VMs - Conservative restart policy
VM Monitoring	Disabled

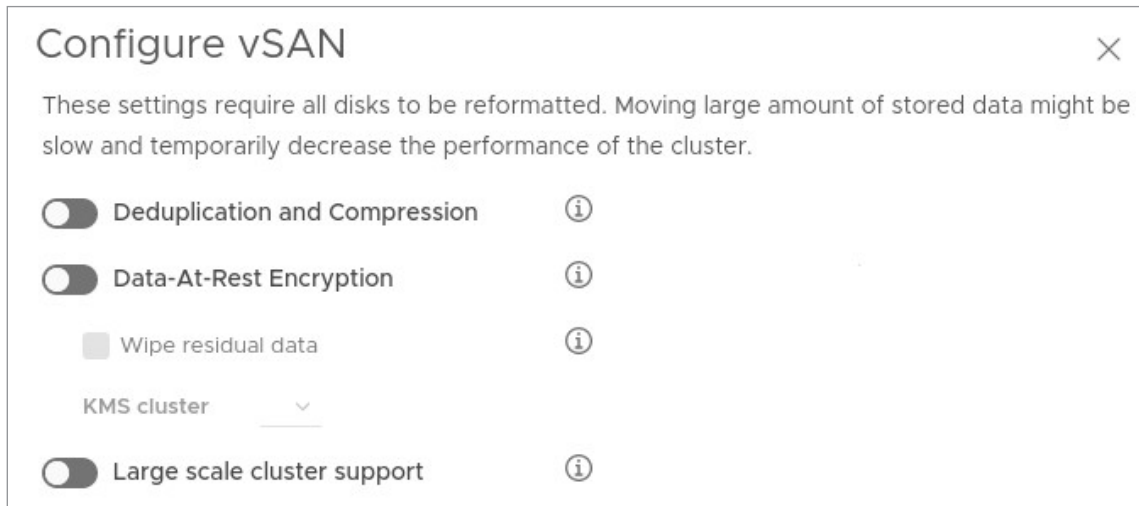


- Enable or disable **Host Monitoring** – This feature makes ESXi hosts exchange network heartbeats for failure detection; only disable this feature during planned maintenance or known network issues, which might cause unnecessary HA failovers.
- Host Failure Response – How vSphere Availability service should react in an event of a host failure
  1. Disabled – vSphere Availability will not react to host failures
  2. Restart VMs – VMs that were running on the failed host failed will be automatically restarted on other hosts in the cluster
- Response to Host Isolation – How should vSphere Availability service react in the event of a host becoming isolated
  1. Disabled – vSphere Availability will not react to host isolation
  2. Power off and restart VMs – Initiate a forced (instant) shutdown on VMs running on the isolated host and then restart them on non-isolated hosts in the cluster
  3. Shut down and restart VMs – Initiate a guest operating system shutdown (graceful) via VMware Tools on VMs running on the isolated host and then restart them on non-isolated hosts in the cluster
- Datastore with PDL (permanent device loss) – How vSphere Availability service should react in the event of datastore PDL failure
  1. Disabled – vSphere Availability will not react to datastore PDL failure.
  2. Issue events – vSphere Availability will not react to datastore PDL failure; events will be generated.
  3. Power off and restart VMs – All affected VMs will be terminated and vSphere HA will attempt to restart the VMs on hosts that still have connectivity to the datastore.
- Datastore with APD (all paths down) – How vSphere Availability service should react in the event of a datastore APD failure
  1. Disabled – vSphere Availability will not react to datastore APD failure.
  2. Issue events – vSphere Availability will not react to datastore APD failure; events will be generated.
  3. Power off and restart VMs – Conservative restart policy – A VM will be powered off, if HA determines the VM can be restarted on a different host.
  4. Shut down and restart VMs – Aggressive restart policy – A VM will be powered off, if HA determines the VM can be restarted on a different host or if HA cannot detect the resources on other hosts because of network connectivity loss (network partition).
- VM Monitoring – Resets individual VMs if their VMware Tools heartbeats are not received within a set time. Application monitoring resets individual VMs if their in-guest heartbeats are not received within a set time.
- **Admission Control** – This is a policy used by vSphere HA to ensure failover capacity within a cluster. Raising the number of potential host failures will increase the availability constraints and capacity reserved. You can adjust the number of host failures tolerated, define host failover capacity, and determine how much VM performance can degrade before vSphere Availability should kick in the HA.



- **Heartbeat Datastores** – This determines which datastores should be used to monitor hosts and virtual machines when the HA network has failed (management network by default). vCenter Server selects two datastores for each host using the policy and datastore preferences.

Some configuration options available under vSAN:



**NOTE: Changing any of these settings will cause the disks used in a vSAN to be reformatted.**

- Deduplication and compression – Improves the total cost of ownership by reducing the data stored on the physical disks (all flash vSAN only)
- Data-At-Rest Encryption – Prevents data visibility in the event of its unauthorized access or theft
- Large scale cluster support – By default, the vSAN cluster can only grow up to 32 nodes; by setting up this option, vSAN cluster is allowed to grow up to large scale, at a maximum of 64 nodes

## Objective 7.7: Perform different types of migrations



Resource:



[Migration with vMotion](#)

In general, there are two types of **migration supported by VMware vSphere**:

- **Compute migration or vMotion** – Changing which compute resource (ESXi host, cluster, or resource pool) should be used for running a virtual machine;
  - Is it also possible to migrate a VM from one virtual switch to another without changing its compute resource; this can be achieved by using the **Migrate VMs to Another Network...** option available by right-clicking on a port group available on a vSphere Standard Switch or vSphere Distributed Switch.
  - When migrating a VM between two ESXi hosts, you can specify the type of a target virtual switch different from the source virtual switch.



- **Storage migration or Storage vMotion** – Changing which storage resource (datastore) should be used for hosting virtual machine data (virtual disks, configuration files, etc.)

If the virtual machine that you migrate has an NVDIMM device and virtual PMem hard disks, the destination host or cluster must have available PMem resource; otherwise, the compatibility check will fail and you won't be able to start the storage vMotion procedure.

Both migrations can be initiated independently or simultaneously. You can move a VM to any ESXi host and to any datastore in the environment as long as the connectivity between the compute and storage resources will not be impacted as the result of said migration (for example, you cannot vMotion a VM from one host to another if the virtual disks of that particular VM are saved on the local storage connected to the source host).

Each of these migration types can also be sub-divided into:

- Hot migrations – A VM migration performed against a powered-on workload
- Cold migrations – A VM migration performed against a powered-off or suspended workload

When migrating a VM, you might also select target resources (both compute and storage), which are associated with a different data center or vCenter Server – the latter requires you to have the two vCenters (source and destination) connected in an Enhanced Linked Mode.

**NOTE: It is not possible to host a virtual machine on a storage resource associated with a data center or vCenter Server different from a compute resource or vice-versa.**

Each type of migration, such as vMotion or Storage vMotion, is assigned a **resource cost**. In vSphere 6.x, there are some limitations on how many migrations can happen at any given point in time – once the total resource cost of all migrations taking place exceeds the limit for a given resource, the migrations scheduled last will be placed in a queue.

There are three **main types of migration limits** that you need to keep in mind:

- **Network limit** – Only applies to vMotion migrations – all network migrations are assigned a resource cost of one; the exact resource limit for network migrations depends on the version of ESXi and type of network adapters installed on the hosts:
  - ESXi 5.x and ESXi 6.x with one GigE adapters – up to four migrations at the same time
  - ESXi 5.x and ESXi 6.x with 10 GigE adapters – up to eight migrations at the same time
- **Datastore limit** – Applies to both regular (compute) vMotions and Storage vMotions; the limit is 128.
  - If a vMotion was started against a VM hosted on a shared datastore, the storage resource cost assigned to it is one (against the shared datastore limit). This means that there can be up to 128 simultaneous vMotions of virtual machines hosted on a single, shared datastore.
  - Storage vMotion cost for moving a VM between two datastores is 16 per datastore (source & target). This means each datastore can support up to eight simultaneous Storage vMotions.



- **Host limit** – Applies to all provisioning operations (vMotion, Storage vMotion, cloning, deployment, etc.) – the limit is eight and the cost of each of the provisioning operations is as follows:
  - vMotion – host resource cost: one; up to eight vMotions per ESXi host
  - Storage vMotion – host resource cost: two; up to four Storage vMotions per ESXi host
  - vMotion without shared storage – host resource cost: four; up to two simultaneous VM migrations per ESXi host
  - Other provisioning operations (cloning, deployment, etc.) – host resource cost: one; up to eight other provisioning operations per ESXi host

## Objective 7.8: Manage resources of a vSphere environment

**Resource or capacity management** is one of the most important topics when administering or designing a vSphere environment – the two main things to keep in mind are:

- What are my resource providers (ESXi hosts) and how much resources/capacity is available for the virtual machines:

**NOTE: Keep in mind the ESXi hypervisor is consuming some of the resources to operate, so the sum of compute resources available for the virtual machines will be a bit lower than what's the theoretical maximum capacity of the installed hardware.**

- How many logical CPU cores are available on each ESXi host and what's the total logical CPU count; while under maximum load, how many operations per second can the physical CPUs handle (what's the maximum available CPU capacity)?
- How much physical RAM memory is available on each ESXi host; what's the total sum of RAM memory across all the ESXi hosts?
- How many resources are required by my resource consumers (or to put it simply – virtual machines) to operate?
  - vCPUs assigned to the virtual machines and CPU reservations
  - Virtual RAM memory assigned to the virtual machines and reservations

In VMware vSphere, compute resources can be divided into **resource pools** – these logical objects can be used to limit or reserve compute capacity for virtual machines. VMs created within a resource pool will be subject to an additional check before being powered on – this check will verify if there's enough capacity in said resource pool to accumulate the VM's compute requirements.



### Resource:



[vSphere Resource Management](#)





## Objective 7.9: Create and manage VMs using different methods



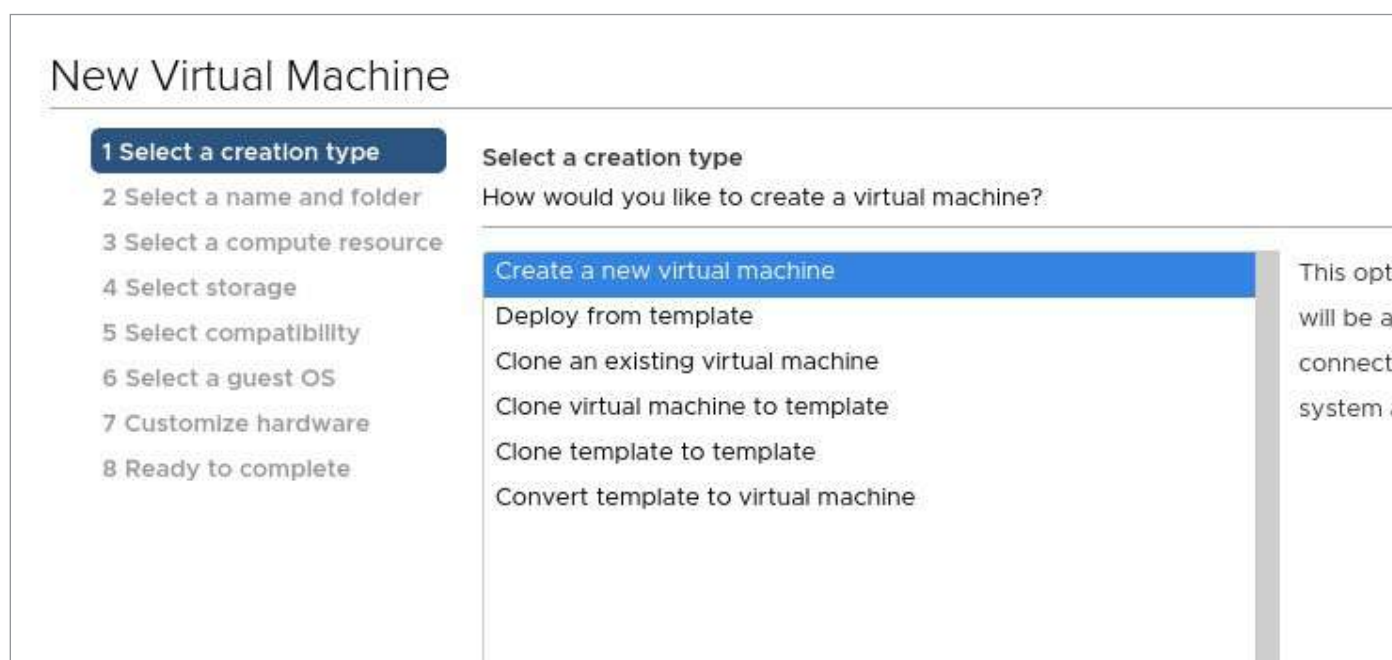
### Resource:



[vSphere Virtual Machine Administration](#)

When **creating a virtual machine**, you have many options on how to carry out the actual deployment. The easiest and most common way of creating VMs is to start fresh (specifying the VM name, which hosts/resource pools to use, what should be the guest operating system, what virtual hardware should be assigned, etc.), however there are other ways to go about it, which we are going to discuss in this section.

After starting the **New Virtual Machine** wizard from any valid parent object (data center, ESXi cluster, stand-alone ESXi server, etc.), you can pick one of the following options on the very first page displayed:



- Create a new virtual machine – The start fresh option; allows you to select every single customization option and create a brand new VM like no other in your environment.
- Deploy from template – Deploy a virtual machine based on a template image; template is a saved state of a VM, which can be used to speed up the deployment process; this option skips some of the steps required when creating a brand new VM, but it requires you to create a VM template object beforehand.
- Clone an existing virtual machine – Deploy a virtual machine as a copy of an existing VM in your environment.
- Clone virtual machine to template – Copy an existing VM in your environment into a VM template, which will allow you to speed up the deployment process of similar systems in the future.
- Clone template to template – Make a copy of an existing VM template.
- Convert template to virtual machine – Revert a VM template back to a virtual machine; this could be useful to update some configuration choices made when originally designing the template.





Another option of creating virtual machines is to use an **OVF (Open Virtual Format) or OVA (Open Virtual Appliance)** file.

**Exporting** a virtual machine to an OVF/OVA template can easily be done via vSphere Client:

- Right-click on the VM you wish to export and select Template > Export OVF Template



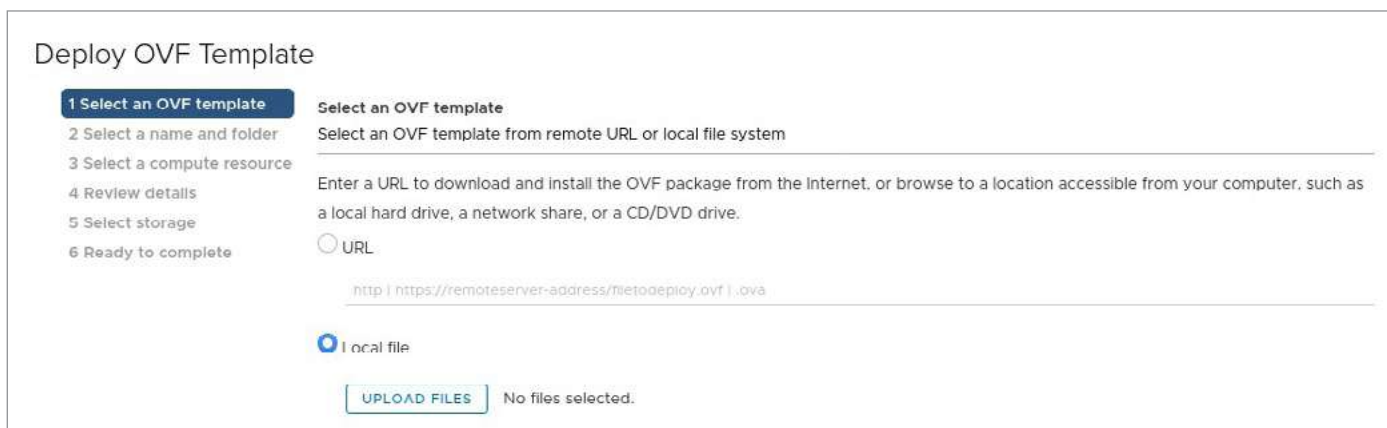
- In the Export OVF Template wizard, enter a name for the template and type in an optional template annotation

**Importing** OVF/OVA files as virtual machines is similarly simple:

- Right-click on the compute resource you wish to use for running the new virtual machine and select Deploy OVF Template...



- On the following screen, you can either choose an OVF/OVA file located somewhere on one of your datastores or type in an URL for an OVF/OVA file hosted on a server



- The next steps in the process heavily depend on the OVA/OVF itself – sometimes all you have to do is to choose a VM name, target compute, and storage resources, while some other times (for example, while deploying a NSX-T appliance) you might have to provide further details, such as the IP address or hostname, for the deployment to start.



## Objective 7.10: Create and manage templates



### Resource:



[vSphere Virtual Machine Administration](#)

Creating a **VM template** is one of the easiest tasks in VMware vSphere:

- Converting a VM to a template (source VM will be removed):
- Right-click on the VM
- Select Template > Convert to Template

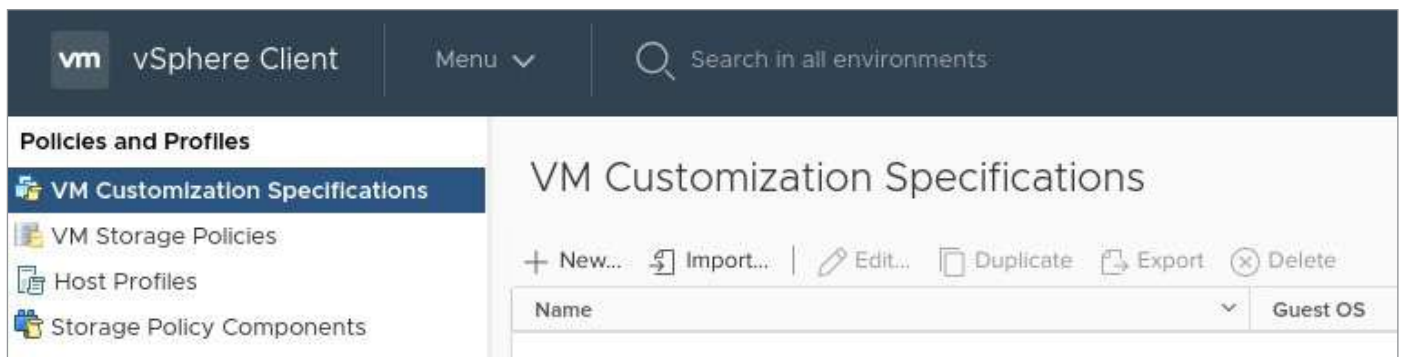


- Cloning a VM to a template (source VM won't be affected):
- Right-click on the VM
- Select Clone > Clone to Template... (Clone as Template to Library... performs the same exact action; the only difference is where the template will be saved and stored)



Now with the templates created, we can look at how to set up **VM Customization Specification**. This workflow allows you to speed up the new VM deployment process by preparing a script, which can run against the VM guest operating system (Windows and Linux with Perl only) to set up some basic parameters on first boot (administrator account password, OS license key or network addressing, etc.).

To start creating a VM Customization Specification, navigate to Menu > Policies and Profiles > VM Customization Specifications and select New.





This will open a **New VM Customization Specification** wizard, which can be used to quickly set all the different parameters, which we would like to be set on the newly deployed virtual machines.

### New VM Customization Specification

- 1 Name and target OS**
- 2 Registration Information
- 3 Computer name
- 4 Windows license
- 5 Administrator password
- 6 Time zone
- 7 Commands to run once
- 8 Network
- 9 Workgroup or domain
- 10 Ready to complete

**Name and target OS**  
Specify a unique name for the VM customization specification and select the OS of the target VM.

#### VM Customization Specification

Name

Description

vCenter Server

Guest OS

Target guest OS  Windows  Linux

Use custom SysPrep answer file

Generate a new security identity (SID)

## Objective 7.11: Manage different VMware vCenter Server objects

One of the main views of the vSphere Client is the **vCenter Server inventory** – this view can be used to view a collection of virtual and physical objects on which you can set permissions, monitoring, alerting, and carry out many other different actions. The main object, which is added to the vCenter Server inventory first, is the data center.

A data center object is a collection of all different types of objects present in the virtual infrastructure. Most vCenter deployments will only have a single data center object present but it is not uncommon to see more data centers defined in the inventory – for example, a company could decide to completely divide their production, test, and development workloads into three distinct data center objects in vCenter.



### Resource:



[vSphere Managed Inventory Objects](#)



Within each data center object, there are four separate **hierarchies**:

### Hosts and clusters

- Host – An object representing an ESXi host; virtual machines use ESXi hosts for compute resources.
- Cluster – A group of ESXi hosts; clusters in VMware vSphere allow the hosts to share their resources more freely. With vSphere DRS, the virtual machines running off clusters can be automatically placed on the hosts with most compute resources available for easier capacity management, while vSphere Availability makes sure VMs will failover onto healthy hosts in the event of a host failure.
- Resource pools – These objects can be used to compartmentalize the CPU and memory resources of a host or cluster; this allows you to split or reserve compute capacity and ensure the CPU and memory resources of the host(s) are properly utilized between the virtual machines.

**NOTE: Resource pools ARE NOT vSphere folders. This might be something obvious but it's better to mention that again than not.**

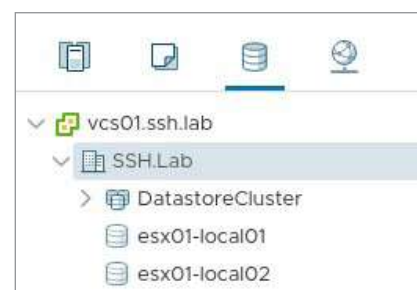
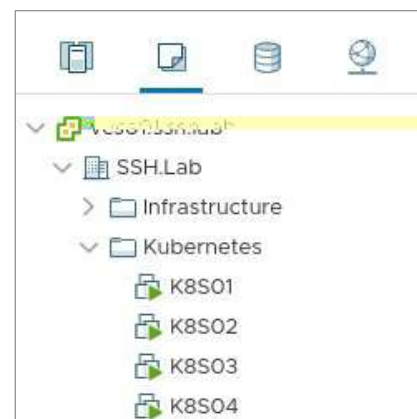
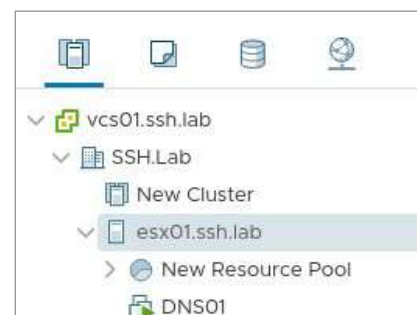
- vApps – A group of resources and virtual machines; vApps can be used for easier management of a multi-tier application spanning multiple systems.
- Host, cluster, and vApp objects can be added to a **Host and Clusters** folder; such a folder object allows you to set up alarms, permissions, and organize objects in a logical way to an administrator without redesigning any virtualization aspects (i.e. it is possible to group together one host out of two clusters without reconfiguring any of the affected vSphere components)

### VMs and templates

- VMs – A virtualized computer object in which a guest operating system operates, like on physical hardware.
- Templates – Virtual machines' "recipes" or master images, allowing for quicker deployment of concurrent VMs; for example, you could have a Windows RDSH template, which already has an up-to-date version of Windows Server installed and configured to act as a Remote Desktop Server Host.
- VMs, VM Templates, and vApps can be added to a **VMs and Templates** folder; such a folder object allows you to set up alarms, permissions, and organize objects in a logical way to an administrator without redesigning any virtualization aspects (i.e. it is possible to group together one VM out of two vApps without reconfiguring any of the affected vSphere components)

### Datstores

- Datastore – Storage location for virtual machine files; a datastore could be a local disk connected directly to the ESXi host, an NFS share hosted somewhere on the network, or an iSCSI/Fibre Channel LUN physically located on a storage array.
- Datastore cluster – A collection of datastores grouped together; vSphere Storage DRS allows for easier management of the data saved on all datastores in a cluster.
- Datastores and datastore clusters can be added to a **Datastore** folder; such a folder object allows you to set up alarms, permissions, and organize objects in a logical way to an administrator without redesigning any virtualization aspects (i.e. it is possible to group together one datastore out of two datastore clusters without reconfiguring any of the affected vSphere components)





## Networks

- vSphere Virtual Standard Switch (vSS) – Standard switches provide network connectivity to hosts and virtual machines. A standard switch can bridge traffic internally between virtual machines in the same VLAN and link to external networks.
- Standard Port Groups – Aggregate multiple ports under a common configuration and provide a stable anchor point for virtual machines connecting to labelled networks.
- vSphere Distributed Switches (vDS) – Aside from acting as a vSphere Standard Switch, a vDS provides a centralized interface from which you can configure, monitor, and administer virtual machine access switching for the entire data center; vDS simplifies VM network configuration, enhances network monitoring and troubleshooting, and provides support for advanced vSphere networking features.

A vDS is a vCenter construct which can only be created, managed, edited, or removed from a vCenter GUI or CLI interface – a host disconnected from a vCenter will retain vDS settings, but you will not be able to manage it using ESXi only.

- Uplink port groups – An uplink port group or dvuplink port group is defined during the creation of the distributed switch and can have one or more uplinks. An uplink is a template that you use to configure physical connections of hosts, as well as failover and load balancing policies. You map physical NICs of hosts to uplinks on the distributed switch.
- Distributed port group – Distributed port groups provide network connectivity to virtual machines and accommodate VMkernel traffic. You identify each distributed port group by using a network label, which must be unique to the current data center. You configure NIC teaming, failover, load balancing, VLAN, security, traffic shaping, and other policies on distributed port groups. The virtual ports that are connected to a distributed port group share the same properties that are configured to the distributed port group.
- Standard port groups and vSphere Distributed Switches can be added to a **Network** folder; such a folder object allows you to set up alarms, permissions, and organize objects in a logical way to an administrator without redesigning any virtualization aspects.



## Objective 7.12: Set up permissions on datastores, clusters, vCenter and hosts

Every single object managed by a vCenter Server has some sort of **permissions** assigned to it. These permissions can either be configured statically on a per object basis or can be inherited from the objects higher up in the object hierarchy.

To fully understand how the vSphere permissions work, we need to see how the **object hierarchy** really looks like.

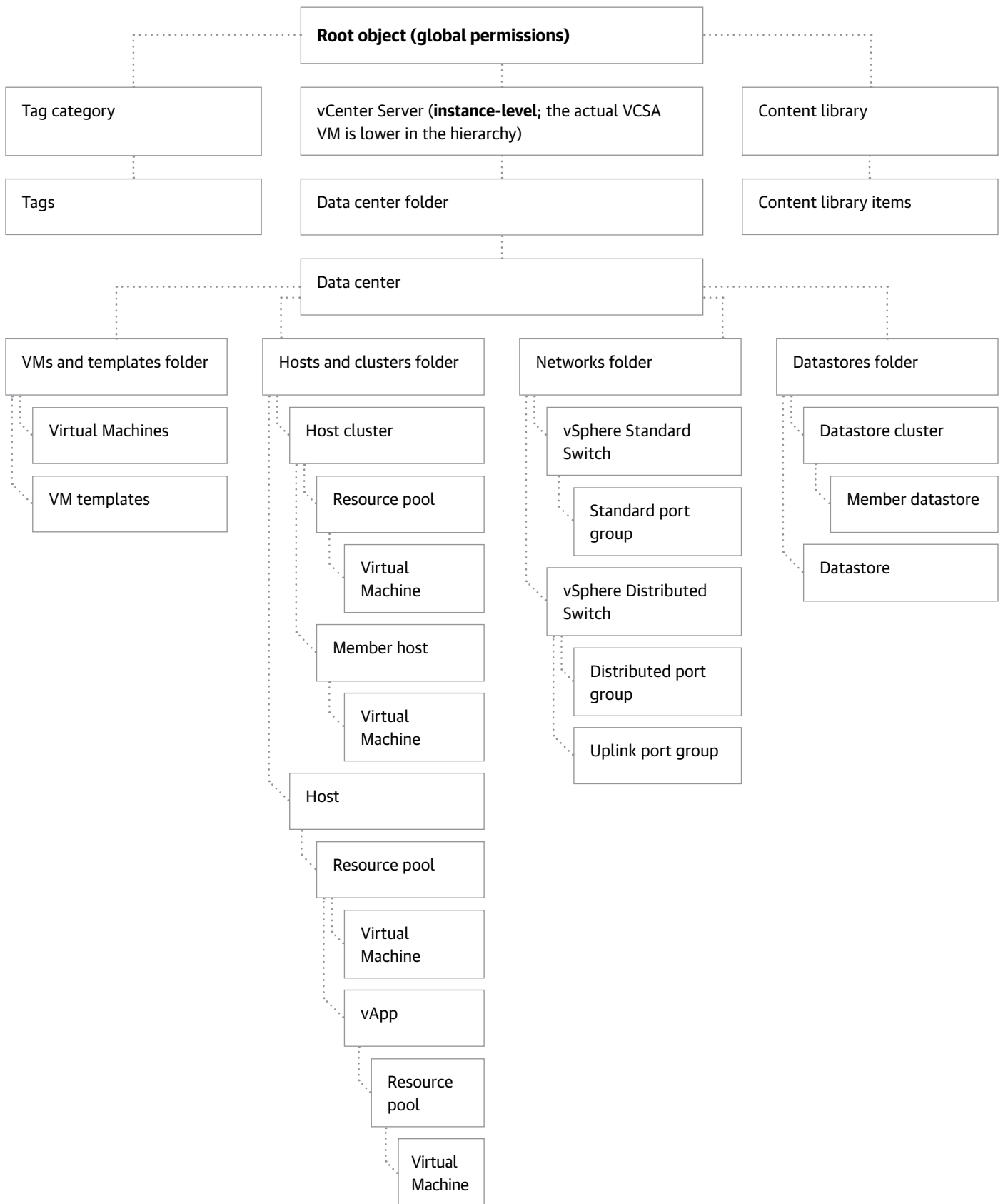
The vSphere hierarchy begins with a root object – it is not a visible item in the inventory; it is rather a logical concept than a regular inventory object. The root object is where the **global permissions** are applied. Every other object in vSphere inventory can inherit permissions from the root.



## Resource:



[vSphere Security](#)





This hierarchy overview should make understanding permission inheritance easier. Let's say we assign a read-only permission for a local user, "Reader", to an ESXi host cluster and enable the option to propagate that change to children – this will cause a read-only permission for the "Reader" user to be applied to all ESXi hosts, resource pools, and virtual machines associated with that cluster.

We could then override that change on a resource pool level and grant a higher level of access for the "Reader" user to a single resource pool. That would cause the permissions to remain on a read-only level on all objects in a cluster except for the objects associated with that particular resource pool.

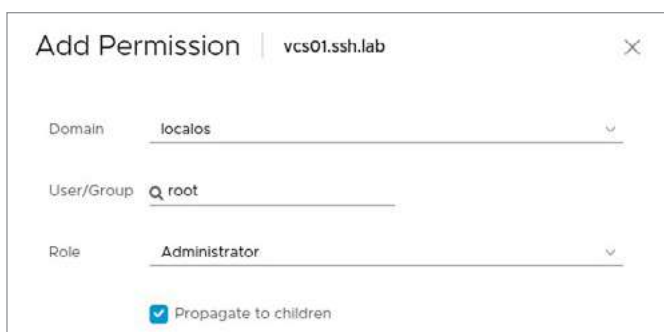
- Permissions defined on any object high in the hierarchy can be **propagated to children** objects (objects lower in hierarchy).
- Permissions defined for a **child object always override the permissions that are propagated from parent objects**.
- It is possible to inherit permissions from multiple parent objects – for example, a virtual machine parent objects could be an ESXi host, ESXi host cluster, and a resource pool; permissions would be inherited in that case from all of three parent objects simultaneously.

**Assigning permissions** to objects in the vSphere Client is easy and done the same way for all types of objects.

- First, navigate to the particular object in the vSphere Client inventory navigator



- Open the **Permissions** tab and click the **Add Permission** icon
- Select the user or the group, which should be affected by the change
- Select a new role for said user or group. A role is a set of permissions grouped together, which can be assigned to users.



- Optionally, select **Propagate to Children** to make sure all objects lower down in the hierarchy will inherit the new permissions





## Objective 7.13: Identify and interpret affinity/anti-affinity rules



### Resource:



[Using DRS Affinity Rules](#)

In many designs, it is critical to ensure some virtual machines are either always running on the same compute resource or never together on a single ESXi host. Since some VM migrations can be carried out automatically by vSphere (fully automated vSphere DRS, vSphere High Availability, vSphere Fault Tolerance...), **the affinity and anti-affinity rules** could be used to stop the virtual machines from ending up running on hosts we don't want them to run on.

There are **two types of affinity/anti-affinity rules** available in vSphere:

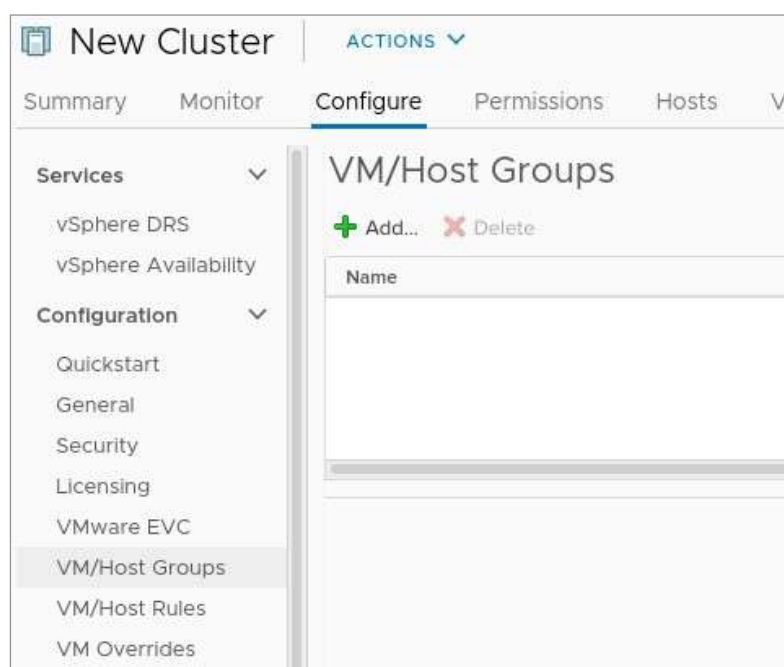
- VM Affinity/Anti-affinity – These can be used to force the virtual machine to either stay apart or remain together during failover actions.
- VM-Host affinity rules – These can be used to force virtual machines to only utilize a certain host or a group of hosts during failover actions.

In vSphere HA advanced options, we can select one of the two additional options to further tune **how affinity/anti-affinity rules work** in the environment:

- HA must respect VM anti-affinity rules during failover – This setting would stop an HA failover if carrying it out would violate an affinity rule. This could be useful if a VM outage is considered by the business as a lesser issue than a VM ending up on a wrong host.
- HA should respect VM anti-affinity rules during failover – HA will attempt to stay in compliance with the affinity rules, but it will ignore them if necessary, to carry out the failover.

To create any type of an affinity or anti-affinity rule, we need to configure **VM and host groups** first.

- This is done on an ESXi cluster level in **Configure > VM/Host Groups**:







- To create a new VM/Host group, select the **Add...** icon. This will open a **Create VM/Host Group** wizard:

Create VM/Host Group | New Cluster

Name: \_\_\_\_\_

Type: VM Group

+ Add... - Remove

Members ↑

CANCEL OK

- All we must do now is give the new group a name, select the group type, and add the actual group members.

Once we have created the required groups, we can move on to **setting up the affinity/anti-affinity rules**.

- To begin, we need to again navigate to the **Configure** tab on the cluster object
- Select **VM/Host Rules** underneath **Configuration** to see the list of currently configured affinity/anti-affinity rules

New Cluster | ACTIONS

Summary Monitor **Configure** Permissions Hosts VMs

Services

- vSphere DRS
- vSphere Availability

Configuration

- Quickstart
- General
- Security
- Licensing
- VMware EVC
- VM/Host Groups
- VM/Host Rules**
- VM Overrides
- I/O Filters
- Host Options

VM/Host Rules

+ Add... Edit... Delete

Name	Type
------	------



- Select the **Add...** button to add a new affinity or anti-affinity rule
- To create a VM affinity rule, select **Keep Virtual Machines Together** from the drop-down. Select the **Add...** button to pick VMs, which should be affected by the rule

The screenshot shows a dialog box titled "Create VM/Host Rule" with a close button (X) in the top right corner. The dialog is for a "New Cluster". It contains the following fields and controls:

- Name:** "Affinity Rule" (text input)
- Enable rule:** A checked checkbox.
- Type:** A dropdown menu set to "Keep Virtual Machines Together".
- Description:** "The listed Virtual Machines must be run on the same host."
- Buttons:** "+ Add..." (green) and "X Remove" (red).
- Members:** A list box currently empty.

- To create a VM anti-affinity rule, select **Separate Virtual Machines** from the drop-down menu. Select the **Add...** button to pick VMs, which should be affected by the rule

The screenshot shows a dialog box titled "Create VM/Host Rule" with a close button (X) in the top right corner. The dialog is for a "New Cluster". It contains the following fields and controls:

- Name:** "Anti-Affinity Rule" (text input)
- Enable rule:** A checked checkbox.
- Type:** A dropdown menu set to "Separate Virtual Machines".
- Description:** "The listed Virtual Machines must be run on separate hosts."
- Buttons:** "+ Add..." (green) and "X Remove" (red).
- Members:** A list box currently empty.



- To create a VM-Host rule, select **Virtual Machines to Hosts** from the drop-down menu. Using the **VM Group** and **Host Group** drop-down menus, select VM and Host groups, which should be affected by the rule. Using the middle drop-down menu, you can pick one of the additional options for VM-Host affinity:
  1. Must run on hosts in group – HA will not carry out a failover if the VMs would end up on a host not allowed by the rule
  2. Should run on hosts in group – HA will try to place the VMs on the hosts mentioned in the rule but will ignore that restriction to carry out the failover if necessary
  3. Must not run on hosts in group – HA will not carry out a failover if the VMs would end up on a host specified by the rule
  4. Should not run on hosts in group – HA will try to avoid placing the VMs on the hosts mentioned in the rule but will ignore that restriction to carry out the failover, if necessary

Create VM/Host Rule | New Cluster

Name: VM-Host Rule  
 Enable rule.

Type: Virtual Machines to Hosts

Description:  
Select cluster VM group

VM Group:

Must run on hosts in group

Host Group:

**NOTE:** It is possible to cause an affinity rule conflict when dealing with a huge number of rules defined in the cluster. If a conflict is detected, the older rule will be honored and the newer rule will be disabled until the conflict is resolved by the administrator. Also, anti-affinity rules take precedence over affinity rules.



## Objective 7.14: Understand use cases for alarms

**Alarms** are considered by some to be these annoying little alerts, which tend to show up at the least convenient times, but in reality they are very useful for monitoring many aspects of a vSphere infrastructure. Alarms can be adjusted to our needs and take a lot of work off our shoulders, allowing us to focus on more important tasks.

There are many types of **built-in alarms**, which can be tweaked to our needs. Each vCenter object has a set of default alarms ranging from very basic ones, such as a CPU usage alert on hosts, to far more sophisticated ones like a warning about a potential security risk discovered somewhere in the infrastructure.

When **editing an already defined alarm or creating a completely new one**, there are a few basic elements that are common across all of them:

- **Name and description** – Pretty self-explanatory; each alarm has a name and a description, allowing an administrator to quickly identify its purpose.
- **Targets** – Defines which objects are monitored by the alarm.
- **Alarm Rules** – Defines the event, condition, or a state, which when met will trigger an alarm. It also controls the severity level of said alarm and what response should be taken in response:
  - Type – What kind of object can be monitored by the alarm.
  - Triggers – What events, conditions or state should trigger the alarm; each alarm trigger has a severity level assigned – severity allows you to categorize the alarms and quickly distinguish which alarms should be actioned first:
    1. Normal – green
    2. Warning – yellow
    3. Alert – red
  - Tolerance Threshold (Reporting) – Provides additional restrictions on the triggers that must be exceeded before the alarm is triggered; for example, a trigger could be the CPU usage on a host, while a tolerance threshold would define which percentage of the CPU resources have to be consumed before vCenter would generate an alert.
  - Actions – Defines operations, which should be started once the alarm is triggered; there are many pre-defined actions from which we can choose from when creating a new alert:
    1. Send email notification
    2. Send SNMP traps
    3. Run a script



## Resource:



[vSphere Events, Alarms and Automated Actions](#)



## Objective 7.15: Utilize VMware vSphere Update Manager (VUM)

One of the most important jobs of a vSphere administrator is to ensure every vSphere component is up-to-date and fully patched up in compliance with security recommendations and policies. **VMware vSphere Update Manager (aka VUM)** can be a great help with that since it is a tool for centralized and automated patch and version management of ESXi hosts and virtual machines (VMware Tools and virtual hardware).

**VUM** allows you to:

- Upgrade, update and patch ESXi hosts
- Install and update third party software on the hosts
- Upgrade virtual machine hardware and install or update VMware Tools

With vSphere 6.7, VMware Update Manager comes **preinstalled on the vCenter Server Appliance**; while in the past it was required to install it separately, it is now a built-in part of the vCenter Server solution.

**NOTE: While on vSphere 6.7 it is still possible to have a vCenter Server running on a Windows Server, this deployment type is deprecated. Customers running vCenter Server for Windows should consider migrating to vCenter Server Appliance instead of installing additional roles on their vCenter Server for Windows system.**

You can access VUM from the main vSphere Client menu by selecting the **Update Manager** link. The main component of VUM are baselines. Baselines can be described as a set of update packages, which can be attached to hosts or virtual machines.

### Virtual Machine baseline

These baselines are always predefined; it is not possible to mix a virtual machine hardware version with an incompatible version of VMware Tools. In other words, VUM will only allow you to update virtual machines hardware and VMware Tools to the most recent version supported by the host or the host cluster.

- The advantage of using VUM to update the hardware/VMware Tools version on the virtual machines is that VUM allows for an instant rollback to the previous version. This could be very useful when a badly timed update causes issues with the VM performance, etc.
- Updating virtual machines with VUM is far simpler than updating the ESXi hosts; to carry out a virtual machine upgrade with VUM, we need to navigate to the ESXi host or cluster object, select the **Updates** tab and open:
  - The VMware Tools section to update VMware Tools to the most up-to-date version supported by the host/cluster
  - The VM Hardware section to update VM Hardware to the most up-to-date version supported by the host/cluster

### Hosts baseline

- Predefined baselines – Official baselines created using official VMware ISOs or update bundles; you can only attach or detach them to the respective inventory objects:
  - Critical Host Patches – Checks ESXi hosts for compliance with all critical patches.



### Resource:



[vSphere Update Manager Administration Guide](#)

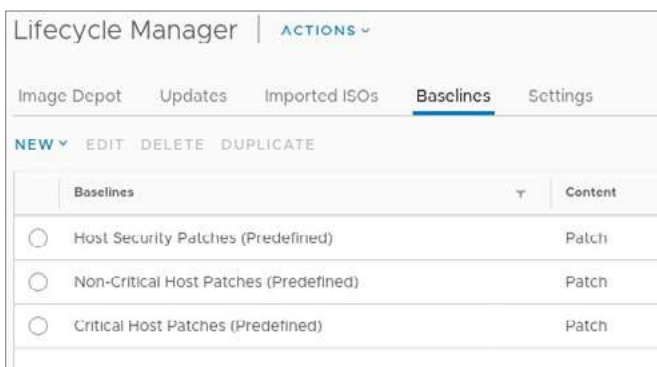


- Non-Critical Host Patches – Checks ESXi hosts for compliance with all optional patches.
- **Custom Baselines** – Baselines created by the administrator; usually these are used to apply non-standard updates, third party software, etc.

Different baselines can be grouped together into **Baseline Groups**; for example, a predefined critical host patch baseline can be grouped with a custom baseline patching the third party software. That way, after associating such a custom baseline with a host, both the ESXi patches and third party patches will be applied at the same time.

#### To create a new baseline:

- In VUM, select the **Baselines** tab and click the **New** button:



- On the following page, assign a name to the new baseline, add optional description and choose which type of content comes with the baseline:
  - Upgrade – For attaching upgrade ISOs; for example, an ESXi 6.7 ISO to be used against ESXi hosts running version 6.5
  - Patch – ESXi patch bundles;
  - Extension – For attaching third party software packages.

**Create Baseline** [X]

1 Name and Description

Name and description  
Enter a name and select the baseline type.

Name: new baseline

Description: [Empty text area]

Content:  Upgrade,  Patch,  Extension

2 Select ISO

3 Summary



- Next, upload or select an existing ISO containing the actual update data and save the changes.

Once the baseline is created, we can **apply it to a host** by using the **Updates** tab on the host object. Then we can navigate to **Host Updates** and select the **Attach** button to select a baseline to use.

Attached Baselines	Status	Content	Type	ESXi version	Last Modified
<input type="checkbox"/> Critical Host Patches (Predefined)	Unknown	Patch	Predefined	6.5.0, 6.7.0	3 weeks ago
<input type="checkbox"/> Host Security Patches (Predefined)	Unknown	Patch	Predefined	6.5.0, 6.7.0	1 week ago

Once the baselines are attached to the host, we can check the boxes next to their names to select them and click the **Remediate** button to start the update procedure:

- First, the baseline (or the updated/patches in it) is checked by vCenter. If any issues are found, VUM will generate a precheck report, which we can analyse before proceeding.
- Second, we can select scheduling options – this controls how and when the updates should be applied.
- Finally, we can tweak some additional settings under **Remediation settings** and begin the process by pressing the **Remediate** button.

Once that's all done, VUM will start the update process on the selected hosts. By default, **this process is sequential, but it is possible to run updates in parallel** as well using advanced options. If any of the hosts in a cluster fails to update, all hosts which were not updated yet will remain on their existing patch level and the hosts which were successfully updated before will not be rolled back.

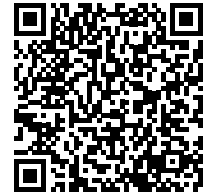
**NOTE: VUM will not allow you to proceed with an update of ESXi hosts in a cluster if updating some of the hosts will not be possible (for example, hardware limitation); VUM will also not proceed if the hosts are not in maintenance mode with Storage DRS disabled (with Storage DRS, VUM would automatically evacuate the VMs and turn the maintenance mode on).**



## Objective 7.16: Configure and manage host profiles



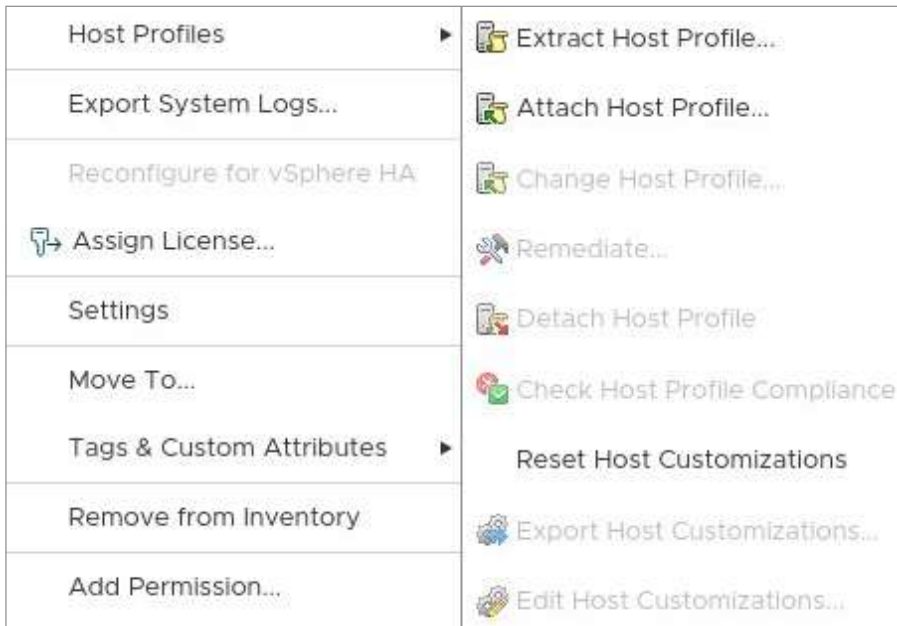
### Resource:



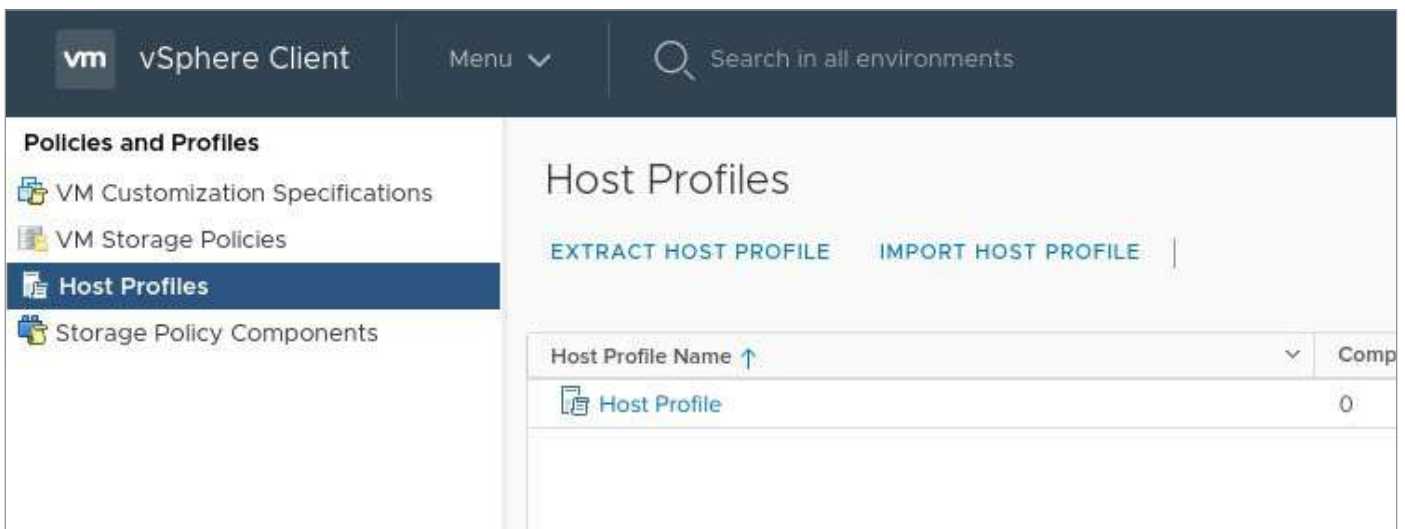
[vSphere Host Profiles](#)

Broadly speaking, **host profiles** are a way to copy configuration of one ESXi host to another. After creating a host profile using vSphere Client, we can apply it to the hosts to quickly apply different host-level settings. This feature of vSphere is very useful when a similar ESXi configuration change must be applied to multiple ESXi hosts – for example, updating the DNS server IP address.

The easiest way to start working on a host profile is to use one of the ESXi hosts in the environment as a template. To do so, we can right-click an ESXi host in the vCenter Inventory and select the **Extract Host Profile...** option:



After a few minutes the host profile should be ready for us in the **Policies in Profiles > Host Profiles** section of the vSphere Client.



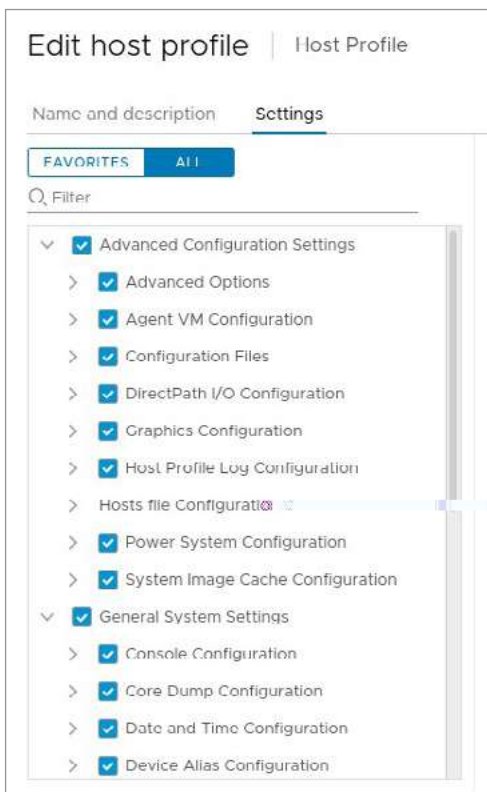




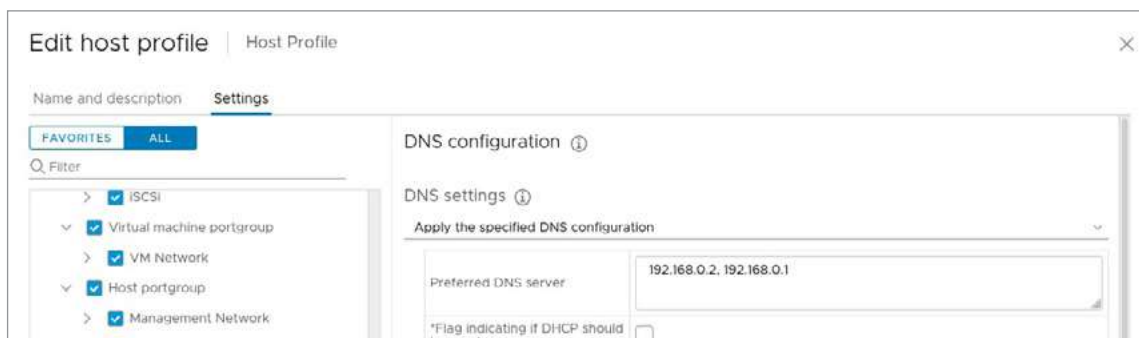
At this point we can start editing it to tweak some settings. Right-click on the host profile you have extracted and select **Edit Host Profile....** to begin.



The host profile editor allows us to change every single bit of ESXi configuration. For ease of use, these settings can be filtered or browsed based on some obvious categories:



Unchecking the boxes in the configuration browser means **these settings will not be affected by the host profile**. For example, if we were to configure a new DNS server on all our ESXi hosts using host profiles, we would have to uncheck all settings on that list except **DNS Configuration** in **Network Configuration** > **Netstack Instance** > **defaultTcpipStack**.

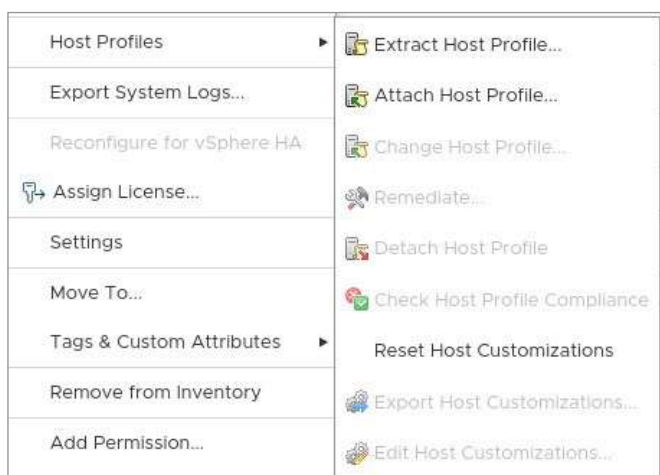


Speaking of applying a host profile – **this process actually involves three steps:**

- Attach a host profile to an ESXi host
- Check the host profile compliance
- Remediate the host with the attached profile

First, **attaching the host profile:**

- Navigate to the hosts and clusters view in the vSphere Client and right click the desired host:



- Select **Attach Host Profile...** and select the desired profile from the list





Next, **checking the compliance:**

- Select the host object to which you attached a host profile and open the **Configure** tab and scroll down to the **Host Profile** section:

Property Name	Path	Value
Host IPv4 address	Networking configuration > Host portgroup > iSCSI > IP address settings ...	192.168.110
Subnet mask	Networking configuration > Host portgroup > iSCSI > IP address settings ...	255.255.255.0
Name for this host	Networking configuration > Netstack instance > defaultTcpipStack > DNS ...	esx01
MAC Address	Networking configuration > Host portgroup > iSCSI > Determine how MA...	00:50:56:6b:51aa

- Select the **Check Compliance** button to verify if the host profile can be applied to the system
  - If the host is not compliant, it means the host is missing some settings, which are present in the profile and the profile can be applied to correct that.
  - Applying a profile to an already compliant host will not make any difference.

**Finally**, if the host in question was found to be non-compliant, we can select the **Remediate** button in the same menu to start the process. Depending on the host profile we attached, we might be asked to provide some additional details or we might have to wait a bit longer for the changes to be fully applied (for example, some settings might require the host to be fully restarted, which might take a few minutes, depending on the underlying hardware).

**NOTE: Only hosts in a cluster with fully-automated DRS will be restarted during the remediation process. For all other hosts, you will need to turn maintenance mode on manually before starting remediation for the reboot to take place automatically.**

## About the Authors



**Shane Williford** is a Sr. Systems Engineer and has been working in the IT industry for over 20 years across multiple business disciplines. He has been awarded several industry certifications, some of which include Veeam® Certified Architect, VMware Certified Advanced Professional, and AWS Solutions Architect – Associate, just to name a few. He also holds two technology community awards – Veeam Vanguard for the past three years and VMware vExpert for 10 years – for his many contributions, involvements, and interactions with the technology community. He has also authored numerous technology certification study guides over the years, including several VCP-DCV Study Guides.



**Paul Wilk** is a VMware Consultant at Triangle Computer Services, a vExpert, vExpert PRO and NSX, and Ireland VMware User Group Leader. Despite still being in his 20s, he has already acquired three VCPs certifications (DCV, NV, CMA) and RHCSA title twice (RHEL5 and 8).



## About Veeam Software

Veeam® is the leader in Backup solutions that deliver Cloud Data Management™. Veeam provides a single platform for modernizing backup, accelerating hybrid cloud, and securing data. With 375,000+ customers worldwide, including 82% of the Fortune 500 and 67% of the Global 2,000, Veeam customer-satisfaction scores are the highest in the industry at 3.5x the average. Veeam's 100-percent channel ecosystem includes global partners, as well as HPE, NetApp, Cisco and Lenovo as exclusive resellers. Veeam has offices in more than 30 countries. To learn more, visit <https://www.veeam.com> or follow Veeam on Twitter [@veeam](https://twitter.com/veeam).

veeam

# A single solution that delivers unparalleled data Availability

[Download](#) a FREE 30-Day Trial  
of Veeam Availability Suite today!

