VMware ESXi Upgrade

8.0
VMware vSphere 8.0
VMware ESXi 8.0

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About VMware ESXi Upgrade

1

VMware ESXi Upgrade describes how to upgrade VMware ESXi™ to the current version.

At VMware, we value inclusion. To foster this principle within our customer, partner, and internal community, we have updated this guide to remove instances of non-inclusive language.

Intended Audience

VMware ESXi Upgrade is for anyone who needs to upgrade from earlier versions of ESXi. These topics are for experienced Microsoft Windows or Linux system administrators who are familiar with virtual machine technology and data center operations.

vCenter Server Upgrade Options 2

vCenter Server 8.0 provides many options for upgrading your vCenter Server deployment. For a successful vCenter Server upgrade, you must understand the upgrade options, the configuration details that impact the upgrade process, and the sequence of tasks.

The two core components of vSphere are VMware ESXi™ and VMware vCenter Server™. ESXi is the virtualization platform on which you can create and run virtual machines and virtual appliances. vCenter Server is a service that acts as a central administrator for ESXi hosts connected in a network. You use the vCenter Server system to pool and manage the resources of multiple hosts. vCenter Server appliance is a preconfigured virtual machine optimized to run vCenter Server.

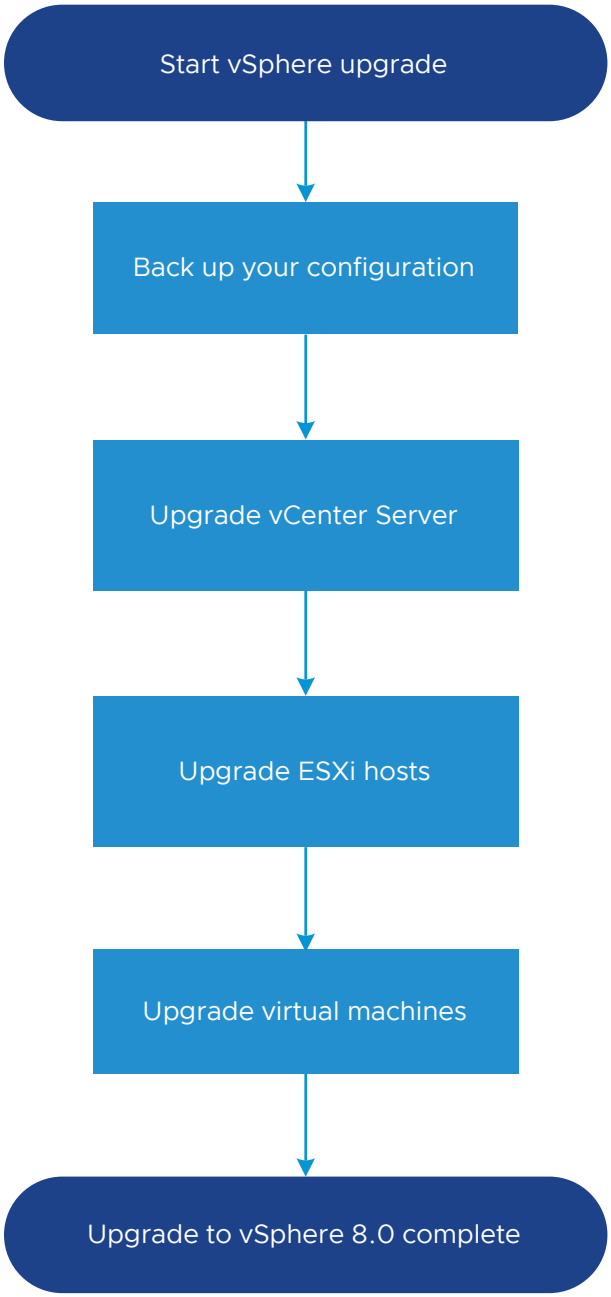
You can upgrade existing vCenter Server deployments that include either an embedded or an external Platform Services Controller to a deployment consisting of a vCenter Server appliance.

This chapter includes the following topics:

Overview of the vSphere Upgrade Process

vSphere is a sophisticated product with multiple components to upgrade. Understanding the required sequence of tasks is vital for a successful vSphere upgrade.

Figure 2-1. Overview of vSphere Upgrade Tasks



Upgrading vSphere includes the following tasks:

1 Read the vSphere release notes.

2 Verify that you have backed up your configuration.

3 If your vSphere system includes VMware solutions or plug-ins, verify that they are compatible with the vCenter Server appliance version to which you are upgrading. See *VMware Product Interoperability Matrix* at http://www.vmware.com/resources/compatibility/sim/interop_matrix.php.

4 Upgrade vCenter Server.

For detailed instructions, see *vCenter Server Upgrade*

5 Upgrade your ESXi hosts. See Overview of the ESXi Host Upgrade Process.

6 To ensure sufficient disk storage for log files, consider setting up a syslog server for remote logging. Setting up logging on a remote host is especially important for hosts with a limited amount of local storage.

See Required Free Space for System Logging and Configure Syslog on ESXi Hosts.

7 Upgrade your VMs manually or by using vSphere Lifecycle Manager to perform an orchestrated upgrade.

See Upgrading Virtual Machines and VMware Tools

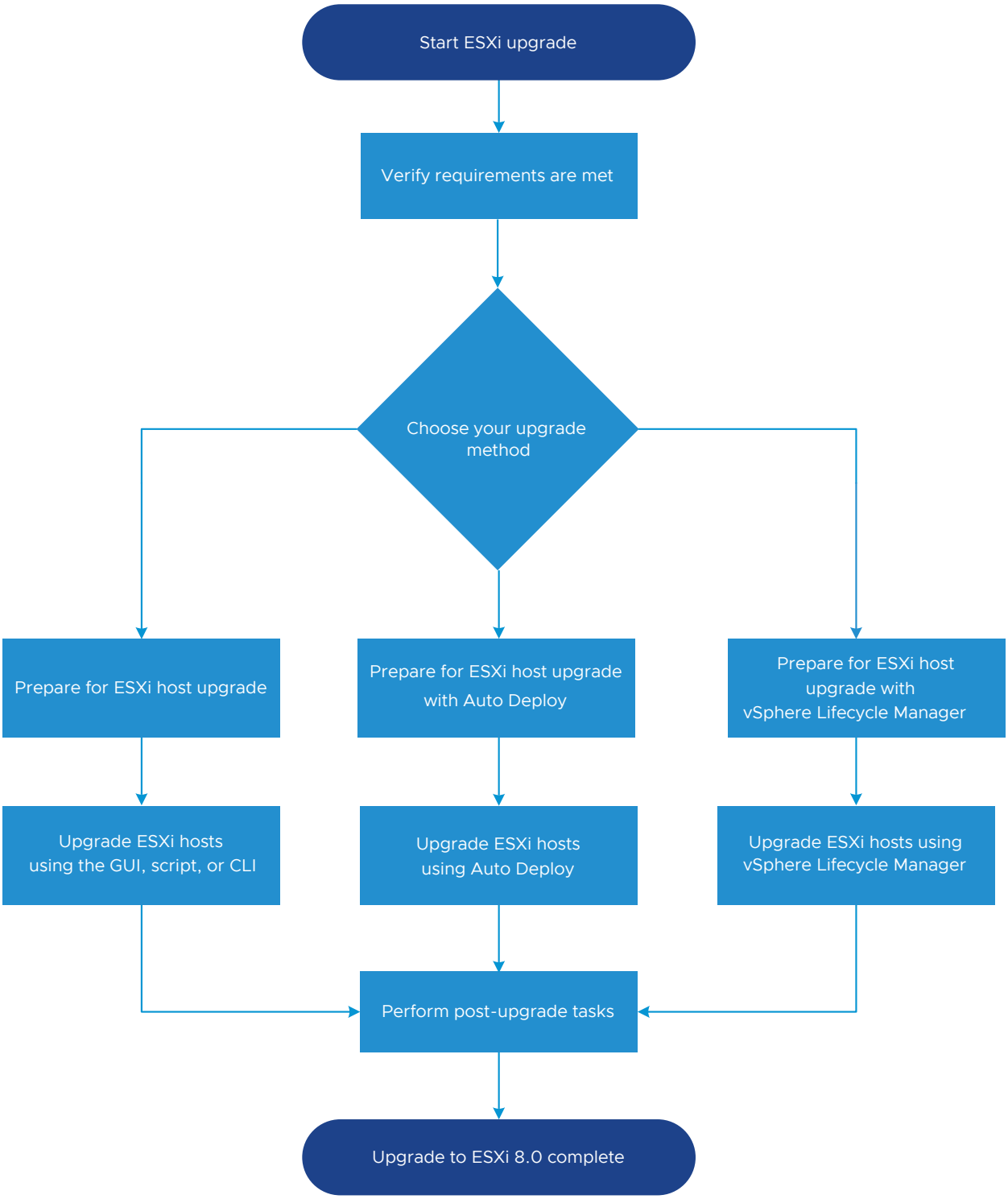
Overview of the ESXi Host Upgrade Process

VMware provides several ways to upgrade ESXi hosts with earlier versions to ESXi version 8.0.

The details and level of support for an upgrade to ESXi 8.0 depend on the host to be upgraded and the upgrade method that you use. Verify that the upgrade path from your current version of ESXi to the version to which you are upgrading, is supported. For more information, see the VMware Product Interoperability Matrices at http://www.vmware.com/resources/compatibility/sim/interop_matrix.php.

You can upgrade an ESXi host to version 8.0 by using an interactive upgrade from a CD, DVD, or USB, a scripted upgrade, ESXCLI, or vSphere Lifecycle Manager. When you upgrade an ESXi host that has custom VIBs to version 8.0, all supported custom VIBs are migrated. For more information, see Upgrading Hosts That Have Third-Party Custom VIBs.

Figure 2-2. Overview of the ESXi Host Upgrade Process



The following high-level steps are for upgrading ESXi.

1 Verify that your system meets the upgrade requirements. See ESXi Requirements.

2 Prepare your environment before upgrading. See Before Upgrading ESXi Hosts.

3 Determine where you want to locate and boot the ESXi installer. See Media Options for Booting the ESXi Installer. If you are network booting the installer, verify that your network boot infrastructure is properly set up. See Network Booting the ESXi Installer.

4 Upgrade ESXi. See Chapter 3 Upgrading ESXi Hosts

5 After upgrading ESXi hosts, you must reconnect the hosts to the vCenter Server and reapply the licenses. See After You Upgrade ESXi Hosts.

The following methods are supported for a direct upgrade to ESXi 8.0.

- Use the interactive graphical user interface (GUI) installer from a CD, DVD, or USB drive.
- Perform a scripted upgrade.
- Use ESXCLI.
- Use vSphere Auto Deploy. If the ESXi host is deployed by using vSphere Auto Deploy, you can use vSphere Auto Deploy to reprovision the host with a 8.0 image.
- Use the vSphere Lifecycle Manager.

Graphical User Interface (GUI) Installer

You can upgrade interactively by using an ESXi installer ISO image on a CD, DVD, or USB flash drive or by network booting the installer. This method is appropriate for deployments with a few hosts. If during the installation process, you select a target disk that contains an ESXi installation, the installer upgrades the host to ESXi version 8.0. The installer also gives you the option to migrate some existing host settings and configuration files and to preserve the existing VMFS datastore. See Upgrade Hosts Interactively.

Perform a Scripted Upgrade

To do a scripted upgrade, you can use the ESXi 8.0 installer from a CD, DVD, or USB flash drive or by network booting the installer. This method is an efficient way to deploy multiple hosts. For more information, see Installing or Upgrading Hosts by Using a Script.

ESXCLI

You can use ESXCLI to upgrade ESXi 6.7 hosts or ESXi 7.0 hosts to ESXi 8.0 hosts.

vSphere 8.0 introduces configuration files, components, base images, and add-ons as new software deliverables that you can use to update or patch ESXi 8.0 hosts. For information about managing components, base images, and add-ons on ESXi, see *ESXCLI Concepts and Examples*.

To use ESXCLI commands, you must install Standalone ESXCLI. For more information about installing and using ESXCLI, see the following documents.

- *Getting Started with ESXCLI*
- *ESXCLI Reference*

See Upgrading Hosts by Using ESXCLI Commands .

vSphere Auto Deploy

If an ESXi host is deployed with vSphere Auto Deploy, you can use vSphere Auto Deploy to reprovision the host and reboot it with a new image profile or a configuration that you manage on a cluster level. An image profile contains an ESXi upgrade or patch, a host configuration profile, and optionally, third-party drivers or management agents that are provided by VMware partners. To add ESXi hosts to a cluster that manages ESXi configuration at a cluster level, you create a rule in Auto Deploy that assigns such a cluster as the host location for newly added hosts, which inherit the same settings and do not require manual configuration.You can build custom images by using vSphere ESXi Image Builder CLI. For more information, see Chapter 4 Using vSphere Auto Deploy to Reprovision Hosts .

vSphere Lifecycle Manager

vSphere Lifecycle Manager is a vCenter Server service for installing, upgrading, and updating ESXi hosts. By using images and baselines, vSphere Lifecycle Manager enables centralized and simplified lifecycle management for multiple ESXi hosts at a cluster level. For more information about performing orchestrated installations, upgrades, and updates, see the *Managing Host and Cluster Lifecycle* documentation.

Upgrading Virtual Machines and VMware Tools

After you upgrade an ESXi host, you can upgrade the virtual machines on the host to take advantage of new features.

You have the following tools for upgrading virtual machines.

vSphere Client

You can use the vSphere Client to upgrade a virtual machine step by step. For more information about upgrading virtual machines, see the *vSphere Virtual Machine Administration* documentation.

vSphere Lifecycle Manager

You can use the vSphere Lifecycle Manager to upgrade the virtual machine hardware and VMware Tools versions of the virtual machines in your environment. The vSphere Lifecycle Manager automates the upgrade process and verifies that the steps occur in the correct order. For more information, see the *Managing Host and Cluster Lifecycle* documentation.

Upgrading ESXi Hosts

3

After you upgrade vCenter Server, upgrade your ESXi hosts. You can upgrade ESXi 6.7 and 7.0 hosts directly to ESXi 8.0.

To upgrade hosts, you can use the tools and methods that are described in Overview of the ESXi Host Upgrade Process.

Caution If you upgrade hosts managed by vCenter Server, you must upgrade vCenter Server before you upgrade the ESXi hosts. If you do not upgrade your environment in the correct order, you can lose data and lose access to servers.

This chapter includes the following topics:

ESXi Requirements

To install or upgrade ESXi, your system must meet specific hardware and software requirements.

ESXi System Storage Overview

ESXi 8.0 has a system storage layout that allows flexible partition management and support for large modules, and third-party components, while facilitating debugging.

ESXi System Storage

The ESXi 8.0 system storage layout consists of four partitions:

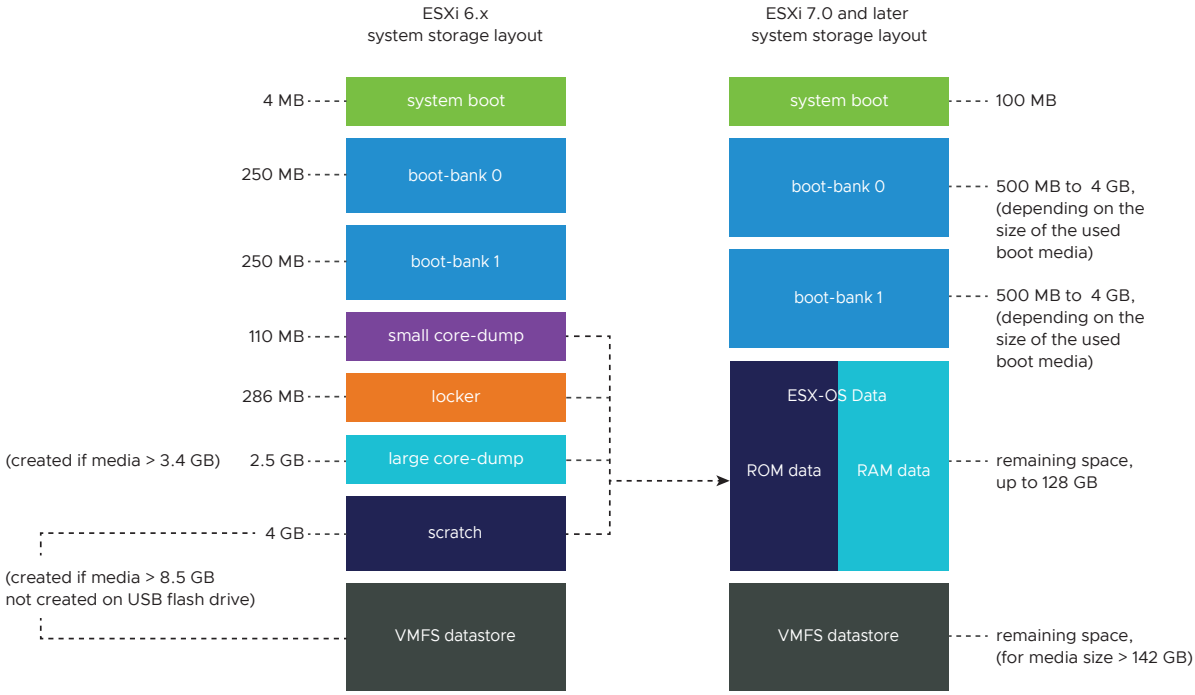
Table 3-1. ESXi system storage partitions:

Partition	Use	Type
System Boot	Stores boot loader and EFI modules.	FAT16
Boot-bank 0	System space to store ESXi boot modules.	FAT16
Boot-bank 1	System space to store ESXi boot modules.	FAT16
ESX-OSData	Acts as the unified location to store additional modules. Not used for booting and virtual machines. Consolidates the legacy `/scratch` partition, locker partition for VMware Tools, and core dump destinations. **Caution** In case the installation media is a USB or an SD card device, best practice is to create ESX-OSData partitions on persistent storage device that is not shared between ESXi hosts.	VMFS-L

The ESX-OSData volume is divided into two high-level categories of data, persistent and non-persistent data. Persistent data contains of data written infrequently, for example, VMware Tools ISOs, configurations, and core dumps.

Non-persistent data contains of frequently written data, for example, logs, VMFS global traces, vSAN Entry Persistence Daemon (EPD) data, vSAN traces, and real-time databases.

Figure 3-1. Consolidated system storage in ESXi 8.0



ESXi System Storage Sizes

Partition sizes, except for the system boot partition, can vary depending on the size of the boot media used. If the boot media is a high-endurance one with capacity larger than 142 GB, a VMFS datastore is created automatically to store virtual machine data.

You can review the boot media capacity and the automatic sizing as configured by the ESXi installer by using the vSphere Client and navigating to the **Partition Details** view. Alternatively, you can use ESXCLI, for example the `esxcli storage filesystem list` command.

Table 3-2. ESXi System Storage Sizes, Depending on the Used Boot Media and Its Capacity.

Boot Media Size	8-10 GB	10-32 GB	32-128 GB	>128 GB
System Boot	100 MB	100 MB	100 MB	100 MB
Boot-bank 0	500 MB	1 GB	4 GB	4 GB
Boot-bank 1	500 MB	1 GB	4 GB	4 GB
ESX-OSData	remaining space	remaining space	remaining space	up to 128 GB
VMFS datastore				remaining space for media size > 142 GB

You can use the ESXi installer boot option `systemMediaSize` to limit the size of system storage partitions on the boot media. If your system has a small footprint that does not require the maximum of 128 GB of system storage size, you can limit it to the minimum of 32 GB. The `systemMediaSize` parameter accepts the following values:

- min (32 GB, for single disk or embedded servers)
- small (64 GB, for servers with at least 512 GB of RAM)
- default (128 GB)
- max (consume all available space, for multi-terabyte servers)

The selected value must fit the purpose of your system. For example, a system with 1 TB of memory must use the minimum of 64 GB for system storage. To set the boot option at install time, for example `systemMediaSize=small`, refer to Enter Boot Options to Start an Installation or Upgrade Script. For more information, see Knowledge Base article 81166.

ESXi System Storage Links

The sub-systems that require access to the ESXi partitions, access these partitions by using the following symbolic links:

Table 3-3. ESXi system storage symbolic links.

System Storage Volume	Symbolic Link
Boot-bank 0	`/bootbank`
Boot-bank 1	`/altbootbank`

Persistent data	`/productLocker`
	`/locker`
	`/var/core`
	`/usr/lib/vmware/isoimages`
	`/usr/lib/vmware/floppies`

Non-persistent data	`/var/run`
	`/var/log`
	`/var/vmware`
	`/var/tmp`
	`/scratch`

Storage Behavior

When you start ESXi, the host enters an autoconfiguration phase during which system storage devices are configured with defaults.

When you reboot the ESXi host after installing the ESXi image, the host configures the system storage devices with default settings. By default, all visible blank internal disks are formatted with VMFS, so you can store virtual machines on the disks. In ESXi Embedded, all visible blank internal disks with VMFS are also formatted by default.

Caution ESXi overwrites any disks that appear to be blank. Disks are considered to be blank if they do not have a valid partition table or partitions. If you are using software that uses such disks, in particular if you are using logical volume manager (LVM) instead of, or in addition to, conventional partitioning schemes, ESXi might cause local LVM to be reformatted. Back up your system data before you power on ESXi for the first time.

On the hard drive or USB device that the ESXi host is booting from, the disk-formatting software retains existing diagnostic partitions that the hardware vendor creates. In the remaining space, the software creates the partitions described below.

Partitions Created by ESXi on the Host Drive

For fresh installations, several new partitions are created for the boot banks, scratch partition, locker, and core dump. Fresh ESXi installations use GUID Partition Tables (GPT) instead of MSDOS-based partitioning. The installer creates boot banks of varying size depending on the size of the disk. For more information on the scratch partition see About the Scratch Partition.

The installer affects only the installation disk. The installer does not affect other disks of the server. When you install on a disk, the installer overwrites the entire disk. When the installer autoconfigures storage, the installer does not overwrite hardware vendor partitions.

To create the VMFS datastore, the ESXi installer requires a minimum of 128 GB of free space on the installation disk.

You might want to override this default behavior if, for example, you use shared storage devices instead of local storage. To prevent automatic disk formatting, detach the local storage devices from the host under the following circumstances:

- Before you start the host for the first time.
- Before you start the host after you reset the host to the configuration defaults.

To override the VMFS formatting if automatic disk formatting already occurred, you can remove the datastore. See the *vCenter Server and Host Management* documentation.

About the Scratch Partition

For new installations of ESXi, during the autoconfiguration phase, a scratch partition is created on the installation disk if it is a high-endurance device such as a hard drive or SSD.

Note Partitioning for hosts that are upgraded to ESXi 7.0 and later from earlier versions differs significantly from partitioning for new installations of ESXi. The size of bootbank partitions is different and autoconfiguration might not configure a coredump partition on the boot disk due to size limitations.

When ESXi boots, the system tries to find a suitable partition on a local disk to create a scratch partition.

The scratch partition is not required. It is used to store system logs, which you need when you create a support bundle. If the scratch partition is not present, system logs are stored in a ramdisk. In low-memory situations, you might want to create a scratch partition if one is not present.

The scratch partition is created during installation. Do not modify the partition.

If no scratch partition is created, you can configure one, but a scratch partition is not required. You can also override the default configuration. You can create the scratch partition on a remote NFS-mounted directory.

ESXi Hardware Requirements

Make sure that the host meets the minimum hardware configurations supported by ESXi 8.0.

Hardware and System Resources

To install or upgrade ESXi, your hardware and system resources must meet the following requirements:

- Supported server platform. For a list of supported platforms, see the *VMware Compatibility Guide* at http://www.vmware.com/resources/compatibility.
- ESXi 8.0 requires a host with at least two CPU cores.
- ESXi 8.0 supports a broad range of multi-core of 64-bit x86 processors. For a complete list of supported processors, see the VMware compatibility guide at http://www.vmware.com/resources/compatibility.
- ESXi 8.0 requires the NX/XD bit to be enabled for the CPU in the BIOS.
- ESXi 8.0 requires a minimum of 8 GB of physical RAM. Provide at least 12 GB of RAM to run virtual machines in typical production environments.
- To support 64-bit virtual machines, support for hardware virtualization (Intel VT-x or AMD RVI) must be enabled on x64 CPUs.
- One or more Gigabit or faster Ethernet controllers. For a list of supported network adapter models, see the *VMware Compatibility Guide* at http://www.vmware.com/resources/compatibility.
- ESXi 8.0 requires a boot disk of at least 32 GB of persistent storage such as HDD, SSD, or NVMe. A boot device must not be shared between ESXi hosts.
- SCSI disk or a local, non-network, RAID LUN with unpartitioned space for the virtual machines.

- For Serial ATA (SATA), a disk connected through supported SAS controllers or supported on-board SATA controllers. SATA disks are considered remote, not local. These disks are not used as a scratch partition by default because they are seen as remote.

  Note You cannot connect a SATA CD-ROM device to a virtual machine on an ESXi host. To use the SATA CD-ROM device, you must use IDE emulation mode.

Storage Systems

For a list of supported storage systems, see the *VMware Compatibility Guide* at http://www.vmware.com/resources/compatibility. Starting with ESXi 8.0, you cannot use software adapters for Fibre Channel over Ethernet (FCoE), only hardware FCoE adapters.

ESXi Booting Requirements

In vSphere 8.0, support for legacy BIOS is limited and booting ESXi hosts from the Unified Extensible Firmware Interface (UEFI) is recommended. With UEFI, you can boot systems from hard drives, CD-ROM drives, or USB media. vSphere Auto Deploy supports network booting and provisioning of ESXi hosts with UEFI. If your system has supported data processing units (DPU), you can only use UEFI to install and boot ESXi on the DPUs. For more information on VMware plans to deprecate support for legacy BIOS in server platforms, see Knowledge Base article https://kb.vmware.com/s/article/84233.

ESXi can boot from a disk larger than 2 TB if the system firmware and the firmware on any add-in card that you are using support it. See the vendor documentation.

Storage Requirements for ESXi 8.0 Installation or Upgrade

For best performance of an ESXi 8.0 installation, use a persistent storage device that is a minimum of 32 GB for boot devices. Upgrading to ESXi 8.0 requires a boot device that is a minimum of 8 GB. When booting from a local disk, SAN or iSCSI LUN, at least a 32 GB disk is required to allow for the creation of system storage volumes, which include a boot partition, boot banks, and a VMFS-L based ESX-OSData volume. The ESX-OSData volume takes on the role of the legacy `/scratch` partition, locker partition for VMware Tools, and core dump destination.

Note In ESXi 8.0, the ESX-OSData volume is considered a unified partition and the separate components, such as `/scratch` and VMware Tools, are consolidated into a single persistent OSDATA partition.

Other options for best performance of an ESXi 8.0 installation are the following:

- A local disk of 128 GB or larger for optimal support of ESX-OSData. The disk contains the boot partition, ESX-OSData volume and a VMFS datastore.
- A device that supports the minimum of 128 terabytes written (TBW).
- A device that delivers at least 100 MB/s of sequential write speed.

- To provide resiliency in case of device failure, a RAID 1 mirrored device is recommended.

Note GB units are 2^30 bytes or 1024*1024*1024 byte multiples.

Legacy SD and USB devices are supported with the following limitations:

- SD and USB devices are supported for boot bank partitions. The use of SD and USB devices for storing ESX-OSData partitions is being deprecated and best practice is to provide a separate persistent local device with a minimum of 32 GB to store the ESX-OSData volume. The persistent local boot device can be an industrial grade M.2 flash (SLC and MLC), SAS, SATA, HDD, SSD, or a NVMe device. The optimal capacity for persistent local devices is 128 GB.
- If you do not provide persistent storage, you see an alarm such as `Secondary persistent device not found. Please move installation to persistent storage as support for SD-Card/USB only configuration is being deprecated.`
- You must use an SD flash device that is approved by the server vendor for the particular server model on which you want to install ESXi on an SD flash storage device. You can find a list of validated devices on partnerweb.vmware.com.
- See Knowledge Base article 85685 on updated guidance for SD card or USB-based environments.
- To chose a proper SD or USB boot device, see Knowledge Base article 82515.

The upgrade process to ESXi 8.0 from versions earlier than 7.x repartitions the boot device and consolidates the original core dump, locker, and scratch partitions into the ESX-OSData volume.

The following events occur during the repartitioning process:

- If a custom core dump destination is not configured, then the default core dump location is a file in the ESX-OSData volume.
- If the syslog service is configured to store log files on the 4 GB VFAT scratch partition, the log files in `var/run/log` are migrated to the ESX-OSData volume.
- VMware Tools are migrated from the locker partition and the partition is wiped.
- The core dump partition is wiped. The application core dump files that are stored on the scratch partition are deleted.

Note Rollback to an earlier version of ESXi is not possible due to the repartitioning process of the boot device. To use an earlier version of ESXi after upgrading to version 8.0, you must create a backup of the boot device before the upgrade, and restore the ESXi boot device from the backup.

If you use USB or SD devices to perform an upgrade, best practice is to allocate an ESX-OSData region on an available persistent disk or a SAN LUN. If persistent storage or a SAN LUN are not available, ESX-OSData is automatically created on a RAM disk. VMFS can also be used for ESX-OSData partition.

After upgrade, if ESX-OSData resides on a RAM disk and a new persistent device is found on subsequent boots, and this device has the setting `autoPartition=True`, ESX-OSData is automatically created on the new persistent device. ESX-OSData does not move between persistent storage automatically, but you can manually change the ESX-OSData location on a supported storage.

For more information on reconfiguring the `/scratch` partition, see the *vCenter Server Installation and Setup* documentation.

To configure the size of ESXi system partitions, you can use the `systemMediaSize` option. For more information, see Knowledge Base article https://kb.vmware.com/s/article/81166.

In Auto Deploy installations, the installer attempts to allocate a scratch region on an available local disk or datastore. If no local disk or datastore is found, installation fails.

For environments that boot from a SAN or use Auto Deploy, the ESX-OSData volume for each ESXi host must be set up on a separate SAN LUN.

Using Remote Management Applications

Remote management applications allow you to install ESXi on servers that are in remote locations.

Remote management applications supported for installation include HP Integrated Lights-Out (iLO), Dell Remote Access Card (DRAC), IBM management module (MM), and Remote Supervisor Adapter II (RSA II). For support on remote management applications, contact the vendor.

You can use remote management applications to do both interactive and scripted installations of ESXi remotely.

If you use remote management applications to install ESXi, the virtual CD might encounter corruption problems with systems or networks operating at peak capacity. If a remote installation from an ISO image fails, complete the installation from the physical CD media.

Supported Remote Management Server Models and Firmware Versions

You can use remote management applications to install or upgrade ESXi, or to manage hosts remotely.

Table 3-4. Supported Remote Management Server Models and Minimum Firmware Versions

Remote Management Server Model	Firmware Version	Java
Dell DRAC 9	6.0.30.00	N/A
Dell DRAC 7	1.30.30 (Build 43)	1.7.0_60-b19
Dell DRAC 6	1.54 (Build 15), 1.70 (Build 21)	1.6.0_24
Dell DRAC 5	1.0, 1.45, 1.51	1.6.0_20,1.6.0_203
Dell DRAC 4	1.75	1.6.0_23
HP ILO	1.81, 1.92	1.6.0_22, 1.6.0_23

Table 3-4. Supported Remote Management Server Models and Minimum Firmware Versions (continued)

Remote Management Server Model	Firmware Version	Java
HP ILO 2	1.8, 1.81	1.6.0_20, 1.6.0_23
HP ILO 3	1.28	1.7.0_60-b19
HP ILO 4	1.13	1.7.0_60-b19
HP ILO 5	2.72	N/A
IBM RSA 2	1.03, 1.2	1.6.0_22

Recommendations for Enhanced ESXi Performance

To enhance performance, install or upgrade ESXi on a robust system with more RAM than the minimum required and with multiple physical disks.

For ESXi system requirements, see ESXi Hardware Requirements.

Table 3-5. Recommendations for Enhanced Performance

System Element	Recommendation
RAM	ESXi hosts require more RAM than typical servers. ESXi 8.0 requires a minimum of 8 GB of physical RAM. Provide at least 12 GB of RAM to take full advantage of ESXi features and run virtual machines in typical production environments. An ESXi host must have sufficient RAM to run concurrent virtual machines. The following examples are provided to help you calculate the RAM required by the virtual machines running on the ESXi host. Operating four virtual machines with Red Hat Enterprise Linux or Windows XP requires at least 3 GB of RAM for baseline performance. This figure includes 1024 MB for the virtual machines, 256 MB minimum for each operating system as recommended by vendors. Running these four virtual machines with 512 MB RAM requires that the ESXi host have 4 GB RAM, which includes 2048 MB for the virtual machines. These calculations do not include possible memory savings from using variable overhead memory for each virtual machine. See *vSphere Resource Management*.
Dedicated Fast Ethernet adapters for virtual machines	Place the management network and virtual machine networks on different physical network cards. Dedicated Gigabit Ethernet cards for virtual machines, such as Intel PRO 1000 adapters, improve throughput to virtual machines with high network traffic.

Table 3-5. Recommendations for Enhanced Performance (continued)

System Element	Recommendation
Disk location	Place all data that your virtual machines use on physical disks allocated specifically to virtual machines. Performance is better when you do not place your virtual machines on the disk containing the ESXi boot image. Use physical disks that are large enough to hold disk images that all the virtual machines use.
VMFS6 partitioning	The ESXi installer creates the initial VMFS volumes on the first blank local disk found. To add disks or modify the original configuration, use the vSphere Client. This practice ensures that the starting sectors of partitions are 64K-aligned, which improves storage performance. **Note** For SAS-only environments, the installer might not format the disks. For some SAS disks, it is not possible to identify whether the disks are local or remote. After the installation, you can use the vSphere Client to set up VMFS.
Processors	Faster processors improve ESXi performance. For certain workloads, larger caches improve ESXi performance.
Hardware compatibility	Use devices in your server that are supported by ESXi drivers. See the *Hardware Compatibility Guide* at http://www.vmware.com/resources/compatibility.

Incoming and Outgoing Firewall Ports for ESXi Hosts

The vSphere Client, vSphere Web Client, and VMware Host Client allow you to open and close firewall ports for each service or to allow traffic from selected IP addresses.

ESXi includes a firewall that is enabled by default. At installation time, the ESXi firewall is configured to block incoming and outgoing traffic, except traffic for services that are enabled in the host's security profile. For the list of supported ports and protocols in the ESXi firewall, see the VMware Ports and Protocols Tool™ at https://ports.vmware.com/.

The VMware Ports and Protocols Tool lists port information for services that are installed by default. If you install other VIBs on your host, additional services and firewall ports might become available. The information is primarily for services that are visible in the vSphere Client and vSphere Web Client but the VMware Ports and Protocols Tool includes some other ports as well.

Required Free Space for System Logging

If you used Auto Deploy to install your ESXi 8.0 host, or if you set up a log directory separate from the default location in a scratch directory on the VMFS volume, you might need to change your current log size and rotation settings to ensure that enough space is available for system logging .

All vSphere components use this infrastructure. The default values for log capacity in this infrastructure vary, depending on the amount of storage available and on how you have configured system logging. Hosts that are deployed with Auto Deploy store logs on a RAM disk, which means that the amount of space available for logs is small.

If your host is deployed with Auto Deploy, reconfigure your log storage in one of the following ways:

- Redirect logs over the network to a remote collector.
- Redirect logs to a NAS or NFS store.

If you redirect logs to non-default storage, such as a NAS or NFS store, you might also want to reconfigure log sizing and rotations for hosts that are installed to disk.

You do not need to reconfigure log storage for ESXi hosts that use the default configuration, which stores logs in a scratch directory on the VMFS volume. For these hosts, ESXi 8.0 configures logs to best suit your installation, and provides enough space to accommodate log messages.

Table 3-6. Recommended Minimum Size and Rotation Configuration for hostd, vpxa, and fdm Logs

Log	Maximum Log File Size	Number of Log Files to Preserve	Minimum Disk Space Required
Management Agent (hostd)	10 MB	10	100 MB
VirtualCenter Agent (vpxa)	5 MB	10	50 MB
vSphere HA agent (Fault Domain Manager, fdm)	5 MB	10	50 MB

For information about setting up and configuring syslog and a syslog server and installing vSphere Syslog Collector, see the *vCenter Server Installation and Setup* documentation.

VMware Host Client System Requirements

Make sure that your browser supports the VMware Host Client.

The following guest operating systems and Web browser versions are supported for the VMware Host Client.

Supported Browsers	Mac OS	Windows 32-bit and 64-bit	Linux
Google Chrome	89+	89+	75+
Mozilla Firefox	80+	80+	60+
Microsoft Edge	90+	90+	N/A
Safari	9.0+	N/A	N/A

ESXi Passwords and Account Lockout

For ESXi hosts, you must use a password with predefined requirements. You can change the required length and the character class requirement or allow pass phrases using the `Security.PasswordQualityControl` advanced system setting. You can also set the number of passwords to remember for each user using the `Security.PasswordHistory` advanced system setting.

Note The default requirements for ESXi passwords can change from one release to the next. You can check and change the default password restrictions by using the `Security.PasswordQualityControl` advanced system setting.

ESXi Passwords

ESXi enforces password requirements for access from the Direct Console User Interface, the ESXi Shell, SSH, or the VMware Host Client.

- By default, you must include a mix of at least three from the following four character classes: lowercase letters, uppercase letters, numbers, and special characters such as underscore or dash when you create a password.
- By default, password length is at least 7 characters and less than 40.
- Passwords must not contain a dictionary word or part of a dictionary word.
- Passwords must not contain the user name or parts of the user name.

Note An uppercase character that begins a password does not count toward the number of character classes used. A number that ends a password does not count toward the number of character classes used. A dictionary word used inside a password reduces the overall password strength.

Example ESXi Passwords

The following password candidates illustrate potential passwords if the option is set as follows.

```
retry=3 min=disabled,disabled,disabled,7,7
```

With this setting, a user is prompted up to three times (retry=3) for a new password that is not sufficiently strong or if the password was not entered correctly twice. Passwords with one or two character classes and pass phrases are not allowed, because the first three items are deactivated. Passwords from three- and four-character classes require seven characters. See the `pam_passwdqc` man page for details on other options, such as `max`, `passphrase`, and so on.

With these settings, the following passwords are allowed.

- xQaTEhb!: Contains eight characters from three character classes.
- xQaT3#A: Contains seven characters from four character classes.

The following password candidates do not meet requirements.

- Xqat3hi: Begins with an uppercase character, reducing the effective number of character classes to two. The minimum number of required character classes is three.
- xQaTEh2: Ends with a number, reducing the effective number of character classes to two. The minimum number of required character classes is three.

ESXi Pass Phrase

Instead of a password, you can also use a pass phrase. However, pass phrases are deactivated by default. You can change the default setting and other settings by using the `Security.PasswordQualityControl` advanced system setting from the vSphere Client.

For example, you can change the option to the following.

```
retry=3 min=disabled,disabled,16,7,7
```

This example allows pass phrases of at least 16 characters and at least three words.

For legacy hosts, changing the `/etc/pam.d/passwd` file is still supported, but changing the file is deprecated for future releases. Use the `Security.PasswordQualityControl` advanced system setting instead.

Changing Default Password Restrictions

You can change the default restriction on passwords or pass phrases by using the `Security.PasswordQualityControl` advanced system setting for your ESXi host. See the *vCenter Server and Host Management* documentation for information on changing ESXi advanced system settings.

You can change the default, for example, to require a minimum of 15 characters and a minimum number of four words (`passphrase=4`), as follows:

```
retry=3 min=disabled,disabled,15,7,7 passphrase=4
```

See the man page for `pam_passwdqc` for details.

Note Not all possible combinations of password options have been tested. Perform testing after you change the default password settings.

This example sets the password complexity requirement to require eight characters from four character classes that enforce a significant password difference, a remembered history of five passwords, and a 90 day rotation policy:

```
min=disabled,disabled,disabled,disabled,8 similar=deny
```

Set the `Security.PasswordHistory` option to 5 and the `Security.PasswordMaxDays` option to 90.

ESXi Account Lockout Behavior

Account locking is supported for access through SSH and through the vSphere Web Services SDK. The Direct Console Interface (DCUI) and the ESXi Shell do not support account lockout. By default, a maximum of five failed attempts is allowed before the account is locked. The account is unlocked after 15 minutes by default.

Configuring Login Behavior

You can configure the login behavior for your ESXi host with the following advanced system settings:

- `Security.AccountLockFailures`. Maximum number of failed login attempts before a user's account is locked. Zero deactivates account locking.
- `Security.AccountUnlockTime`. Number of seconds that a user is locked out.
- `Security.PasswordHistory`. Number of passwords to remember for each user. Zero deactivates password history.

See the *vCenter Server and Host Management* documentation for information on setting ESXi advanced options.

Before Upgrading ESXi Hosts

For a successful upgrade of your ESXi hosts, understand and prepare for the changes that are involved.

For a successful ESXi upgrade, follow these best practices:

1 Make sure that you understand the ESXi upgrade process, the effect of that process on your existing deployment, and the preparation required for the upgrade.
   - If your vSphere system includes VMware solutions or plug-ins, make sure they are compatible with the vCenter Server version that you are upgrading to. See the VMware Product Interoperability Matrix at http://www.vmware.com/resources/compatibility/sim/interop_matrix.php.
   - Read Overview of the ESXi Host Upgrade Process to understand the upgrade scenarios that are supported, and the options and tools that are available to perform the upgrade.
   - Read the VMware vSphere Release Notes for known installation issues.

2 Prepare the system for the upgrade.
   - Make sure that the current ESXi version is supported for the upgrade. See Overview of the ESXi Host Upgrade Process.
   - Make sure that the system hardware complies with ESXi requirements. See ESXi Requirements and VMware Compatibility Guide at http://www.vmware.com/resources/compatibility/search.php. Check for system compatibility, I/O compatibility with network and host bus adapter (HBA) cards, storage compatibility, and backup software compatibility.

- Make sure that sufficient disk space is available on the host for the upgrade.
- If a SAN is connected to the host, detach the Fibre Channel system before continuing with the upgrade. Do not deactivate HBA cards in the BIOS.

3 Back up the host before performing an upgrade. If the upgrade fails, you can restore the host.

4 If you are using Auto Deploy to provision hosts, the user who is running the process must have local administrator privileges on the ESXi host that is being provisioned. By default the installation process has these privileges and certificate provisioning happens as expected. However, if you are using another method than the installer, you must run it as a user who has the local administrator privileges.

5 Depending on the upgrade option you choose, you might need to migrate or power off all virtual machines on the host. See the instructions for your upgrade method.

- For an interactive upgrade from CD, DVD, or USB drive: see Upgrade Hosts Interactively.
- For a scripted upgrade: see Installing or Upgrading Hosts by Using a Script.
- For vSphere Auto Deploy: see Chapter 4 Using vSphere Auto Deploy to Reprovision Hosts . If the ESXi 6.7x or 7.0.x host was deployed by using vSphere Auto Deploy, you can use vSphere Auto Deploy to reprovision the host with a 8.0 image.
- For the `esxcli` command method: see Upgrading Hosts by Using ESXCLI Commands .

6 Plan for the tasks that must be performed after the ESXi host upgrade:

- Test the system to ensure that the upgrade completed successfully.
- Apply a host's licenses. See Licensing ESXi Hosts After Upgrade.
- Consider setting up a syslog server for remote logging, to ensure sufficient disk storage for log files. Setting up logging on a remote host is especially important for hosts with limited local storage. vSphere Syslog Collector is included as a service in vCenter Server 6.0 and can be used to collect logs from all hosts. See Required Free Space for System Logging. For information about setting up and configuring syslog and a syslog server, setting up syslog from the host profiles interface, and installing vSphere Syslog Collector, see the *vCenter Server Installation and Setup* documentation.

7 If the upgrade was unsuccessful and you backed up the host, you can restore the host.

Upgrading Hosts That Have Third-Party Custom VIBs

A host can have custom vSphere Installation Bundles (VIBs) installed, for example, third-party drivers or management agents. When you upgrade an ESXi host to 8.0, all supported custom VIBs are migrated, regardless of whether the VIBs are included in the installer ISO.

If the host or the installer ISO image contains a VIB that creates a conflict and prevents the upgrade, an error message identifies the VIB that created the conflict. To upgrade the host, take one of the following actions:

- Remove the VIB that created the conflict from the ESXi host and retry the upgrade. You can remove a VIB from the host by using `esxcli` commands. For more information, see Remove VIBs from a Host .
- Use the vSphere ESXi Image Builder CLI to create a custom installer ISO image that resolves the conflict. For more information about vSphere ESXi Image Builder CLI, see the *vCenter Server Installation and Setup* documentation.

Upgrading ESXi Hosts in an Environment With VMware NSX

If your vSphere system includes VMware NSX, before you start an upgrade of your ESXi hosts, you must ensure that the NSX kernel module is part of the desired software specification or baseline that you use for the upgrade.

When you upgrade an ESXi host to 8.0 or later, all supported custom VIBs are migrated, regardless of whether the VIBs are included in the installer ISO. However, the NSX kernel module is not automatically migrated to the installer ISO image. Before you proceed to the upgrade operation, you must take one of the following actions:

- Create an extension baseline with a newly uploaded NSX kernel module. For more information, see Managing Host and Cluster Lifecycle.
- Create a custom image profile with the NSX kernel module. For more information, see Create a Custom Image Profile to Upgrade ESXi Hosts in an Environment With VMware NSX.
- Use PowerCLI to create a new ISO image. For more information, see Create a New ISO Image to Upgrade ESXi Hosts in an Environment With VMware NSX.
- Use ESXCLI. For more information, see Use ESXCLI to Upgrade ESXi Hosts in an Environment With VMware NSX.

Create a Custom Image Profile to Upgrade ESXi Hosts in an Environment With VMware NSX

If your vSphere system includes VMware NSX, before you start an upgrade of your ESXi hosts to 8.0 and later from an earlier version of ESXi, you must ensure that the NSX kernel module is part of the baseline that you use for the upgrade. For this purpose, you can create a custom image profile with an ESXi base image and a newly uploaded NSX kernel module.

Prerequisites

- Download from VMware Customer Connect the `NSX Kernel Module for VMware ESXi 8.0` zip file for the version of VMware NSX deployed in your environment. For example, `nsx-lcp-4.0.1.0.0.xxx-esx80.zip` for VMware NSX 4.0.1.
- Make sure that Auto Deploy and Image Builder are enabled in your vCenter Server system.

Procedure

1. Log in to a vCenter Server 8.x system.
2. Navigate to **Home > Autodeploy > Software Depots** to import to the vSphere ESXi Image Builder inventory an ESXi 8.x base image, if it is not already available, and the ZIP file for the NSX kernel module.
3. Create an image profile that combines the VMware NSX Kernel Module and the base image for ESX 8.x. For detailed steps, see Create an Image Profile.
4. Export the custom image profile to an ISO image.
5. Import the ISO image to the vSphere Lifecycle Manager depot.

   You can now create an upgrade baseline based on the imported ISO image by using the vSphere Lifecycle Manager. For more information on vSphere Lifecycle Manager upgrades workflow with baselines, see the Managing Host and Cluster Lifecycle guide.

Create a New ISO Image to Upgrade ESXi Hosts in an Environment With VMware NSX

If your vSphere system includes VMware NSX, before you start an upgrade of your ESXi hosts to 8.0 and later from an earlier version of ESXi, you must ensure that the NSX kernel module is part of the software specification or baseline that you use for the upgrade. For this purpose, you can use the `New-IsoImage` PowerCLI cmdlet to create a new ISO image and perform the ESXi upgrade in your preferred way.

Prerequisites

- Download from VMware Customer Connect the `NSX Kernel Module for VMware ESXi 8.0` zip file for the version of VMware NSX deployed in your environment. For example, `nsx-lcp-4.0.1.0.0.xxx-esx80.zip` for VMware NSX 4.0.1.
- Install the PowerCLI and all prerequisite software. See vSphere ESXi Image Builder Installation and Usage.
- Verify that you have access to the software depot that contains the software specification you want to use.

Procedure

- In a PowerCLI session, run the `New-IsoImage` cmdlet to generate an ISO image by passing the parameters `Depots`,`Destination` and `SoftwareSpec`. For example, `PS C:\Users\Administrator> New-IsoImage -Depots "C:\VMware-ESXi-8.x.x-xxx-depot.zip","C:\nsx-lcp-4.0.1.0.0.xxx-esx80.zip", -Destination C:\<your new ISO image name>.iso -SoftwareSpec C:\<your file name>.json`. This command creates a new

ISO image by using the ESXi base image and the NSX kernel zip files, and the software specification of your desired image in a JSON file. You can use any number and combination of software depots, offline and online. For upgrades to ESXi 8.0, the `New-IsoImage` cmdlet preserves additional metadata for ESXi 8.0 required by the vSphere Lifecycle Manager.

What to do next

Use the new ISO image to complete the ESXi upgrade in your preferred way. For more information on vSphere Lifecycle Manager upgrade workflows, see the Managing Host and Cluster Lifecycle guide.

Use ESXCLI to Upgrade ESXi Hosts in an Environment With VMware NSX

If your vSphere system includes VMware NSX, before you start an upgrade of your ESXi hosts to 8.0 and later from an earlier version of ESXi, you must ensure that the NSX kernel module is part of the software specification or baseline that you use for the upgrade. You can use ESXCLI commands to upgrade your ESXi hosts and re-install the NSX kernel module.

To use ESXCLI for the upgrade of an ESXi host in a vSphere system that includes NSX-T Data Center, you must follow the procedures described in Upgrading Hosts by Using ESXCLI Commands:

Prerequisites

- Download from VMware Customer Connect the `NSX Kernel Module for VMware ESXi 8.0` zip file for the version of VMware NSX deployed in your environment. For example, `nsx-lcp-4.0.1.0.0.xxx-esx80.zip` for VMware NSX 4.0.1.

Procedure

1. Place your ESXi host in maintenance mode. For more information, see Place a Host in Maintenance Mode.
2. Download an ESXi 8.x image profile in a software depot that is accessible through a URL or in an offline ZIP depot.
3. Run the ESXCLI command `esxcli software profile update --depot <path-to-depot-file> -p ESXi-X.X.X-XXXXXX-standard --allow-downgrades --no-sig-check`. For example: `esxcli software profile update --depot /vmfs/volumes/5e8fd197-68bce4dc-f8f1-005056af93cf/VMware-ESXi-8.0.0-xxx-depot.zip -p ESXi-8.0.0-xxx-standard --allow-downgrades --no-sig-check`. For more information, see Upgrade or Update a Host with Image Profiles.
4. Install the NSX kernel module by using the ESXCLI command `esxcli software vib install -d <path_to_kernel_module_file> --no-sig-check`. For example: `esxcli software vib install -d /tmp/nsx-lcp-4.0.1.0.0.xxx-esx80.zip`
5. Reboot the ESXi host.
6. Move your ESXi host out of maintenance mode.

Media Options for Booting the ESXi Installer

The ESXi installer must be accessible to the system on which you are installing ESXi.

The following boot media are supported for the ESXi installer:

- Boot from a CD/DVD. See Download and Burn the ESXi Installer ISO Image to a CD or DVD.
- Boot from a USB flash drive. See Format a USB Flash Drive to Boot the ESXi Installation or Upgrade.
- Boot from a network. Network Booting the ESXi Installer
- Boot from a remote location using a remote management application. See Using Remote Management Applications

Download and Burn the ESXi Installer ISO Image to a CD or DVD

If you do not have an ESXi installation CD/DVD, you can create one.

You can also create an installer ISO image that includes a custom installation script. See Create an Installer ISO Image with a Custom Installation or Upgrade Script.

Procedure

1. Follow the procedure Download the ESXi Installer.
2. Burn the ISO image to a CD or DVD.

Format a USB Flash Drive to Boot the ESXi Installation or Upgrade

You can format a USB flash drive to boot the ESXi installation or upgrade.

The instructions in this procedure assume that the USB flash drive is detected as `/dev/sdb`.

Note The `ks.cfg` file that contains the installation script cannot be located on the same USB flash drive that you are using to boot the installation or upgrade. The kickstart file does not have any dependency on BIOS or UEFI boot.

Prerequisites

- Linux machine with superuser access to it
- USB flash drive that can be detected by the Linux machine
- The ESXi ISO image, `VMware-VMvisor-Installer-`*`version_number`*`-`*`build_number`*`.x86_64.iso`, which includes the `isolinux.cfg` file

Procedure

1. Boot Linux, log in, and enter superuser mode by using a `su` or `sudo root` command.

2 If your USB flash drive is not detected as `/dev/sdb`, or you are not sure how your USB flash drive is detected, determine how it is detected.

   a Plug in your USB flash drive.

   b At the command line, run the command for displaying the current log messages.

   ```
   tail -f /var/log/messages
   ```

   You see several messages that identify the USB flash drive in a format similar to the following message.

   ```
   Oct 25 13:25:23 ubuntu kernel: [  712.447080] sd 3:0:0:0: [sdb] Attached SCSI
   removable disk
   ```

   In this example, `sdb` identifies the USB device. If your device is identified differently, use that identification in place of *sdb*.

3 Overwrite the entire USB drive with the ISO image. This overwrites the partition table and any previous content on the USB drive.

   ```
   dd bs=10M if=VMware-VMvisor-Installer-version_number-build_number.x86_64.iso
   of=/dev/sdb
   ```

4 Eject the USB drive.

   ```
   eject /dev/sdb
   ```

Results

You can use the USB flash drive to boot the ESXi installer.

Create a USB Flash Drive to Store the ESXi Installation Script or Upgrade Script

You can use a USB flash drive to store the ESXi installation script or upgrade script that is used during scripted installation or upgrade of ESXi.

When multiple USB flash drives are present on the installation machine, the installation software searches for the installation or upgrade script on all attached USB flash drives.

The instructions in this procedure assume that the USB flash drive is detected as `/dev/sdb`.

Note Do not store the `ks` file containing the installation or upgrade script on the same USB flash drive that you are using to boot the installation or upgrade.

Prerequisites

- Linux machine
- ESXi installation or upgrade script, the `ks.cfg` kickstart file
- USB flash drive

Procedure

1. Attach the USB flash drive to a Linux machine that has access to the installation or upgrade script.
2. Create a partition table.

   `/sbin/fdisk /dev/sdb`

   a. Type `d` to delete partitions until they are all deleted.

   b. Type `n` to create primary partition 1 that extends over the entire disk.

   c. Type `t` to set the type to an appropriate setting for the FAT32 file system, such as **`c`**.

   d. Type `p` to print the partition table.

   The result should be similar to the following text:

   ```
   Disk /dev/sdb: 2004 MB, 2004877312 bytes
   255 heads, 63 sectors/track, 243 cylinders
   Units = cylinders of 16065 * 512 = 8225280 bytes
      Device Boot      Start         End      Blocks   Id  System
   /dev/sdb1               1         243      1951866  c   W95 FAT32 (LBA)
   ```

   e. Type `w` to write the partition table and quit.
3. Format the USB flash drive with the FAT32 file system.

   `/sbin/mkfs.vfat -F 32 -n USB /dev/sdb1`
4. Create a destination directory and mount the USB flash drive to it.

   `mkdir -p /usbdisk`

   `mount /dev/sdb1 /usbdisk`
5. Copy the ESXi installation script to the USB flash drive.

   `cp ks.cfg /usbdisk`
6. Unmount the USB flash drive.

   `umount /usbdisk`

Results

The USB flash drive contains the installation or upgrade script for ESXi.

What to do next

When you boot the ESXi installer, point to the location of the USB flash drive for the installation or upgrade script. See Enter Boot Options to Start an Installation or Upgrade Script and PXELINUX Configuration Files.

Create an Installer ISO Image with a Custom Installation or Upgrade Script

You can customize the standard ESXi installer ISO image with your own installation or upgrade script. This customization enables you to perform a scripted, unattended installation or upgrade when you boot the resulting installer ISO image.

See also About Installation and Upgrade Scripts and About the boot.cfg File .

Prerequisites

- Linux machine
- The ESXi ISO image `VMware-VMvisor-Installer-x.x.x-XXXXXX.x86_64.iso`,where *`x.x.x`* is the version of ESXi you are installing, and *`XXXXXX`* is the build number of the installer ISO image
- Your custom installation or upgrade script, the `KS_CUST.CFG` kickstart file

Procedure

1. Download the ESXi ISO image from VMware Customer Connect.
2. Mount the ISO image in a folder:

   `mount -o loop VMware-VMvisor-Installer-x.x.x-XXXXXX.x86_64.iso / esxi_cdrom_mount`

   `XXXXXX` is the ESXi build number for the version that you are installing or upgrading to.
3. Copy the contents of `esxi_cdrom` to another folder:

   `cp -r /esxi_cdrom_mount/* /esxi_cdrom`
4. Copy the kickstart file to `/esxi_cdrom`.

   `cp KS_CUST.CFG /esxi_cdrom`
5. Modify the `boot.cfg` file in both `/esxi_cdrom/efi/boot/boot.cfg` (for UEFI boot) and `/esxi_cdrom/boot.cfg` (for legacy BIOS boot) to specify the location of the installation or upgrade script by using the `kernelopt` option.

   You must use uppercase characters to provide the path of the script, for example,

   ```
   kernelopt=runweasel ks=cdrom:/KS_CUST.CFG
   ```

   The installation or upgrade becomes completely automatic, without the need to specify the kickstart file during the installation or upgrade.

6 Recreate the ISO image using the `mkisofs` or the `genisoimage` command.

Command	Syntax
`mkisofs`	`mkisofs -relaxed-filenames -J -R -o custom_esxi.iso -b ISOLINUX.BIN -c BOOT.CAT -no-emul-boot -boot-load-size 4 -boot-info-table -eltorito-alt-boot -eltorito-platform efi -b EFIBOOT.IMG -no-emul-boot /esxi_cdrom`
`genisoimage`	`genisoimage -relaxed-filenames -J -R -o custom_esxi.iso -b ISOLINUX.BIN -c BOOT.CAT -no-emul-boot -boot-load-size 4 -boot-info-table -eltorito-alt-boot -e EFIBOOT.IMG -no-emul-boot /esxi_cdrom`

You can use this ISO installer image for regular boot or UEFI secure boot. However, the vSphere Lifecycle Manager cannot verify the checksum of such an ISO image and you cannot use it for upgrades by using vSphere Lifecycle Manager workflows.

Results

The ISO image includes your custom installation or upgrade script.

What to do next

Install ESXi from the ISO image.

Download the ESXi Installer

Download the installer for ESXi. You can obtain the software either from an OEM or from the VMware download portal at `https://customerconnect.vmware.com/`.

Prerequisites

Create a VMware Customer Connect account at https://customerconnect.vmware.com/.

Procedure

1 Log in to VMware Customer Connect.

2 Navigate to **Products and Accounts** > **All Products**.

3 Find VMware vSphere and click **Download Product**.

4 Select a VMware vSphere version from the **Select Version** drop-down menu.

5 Select a version of VMware vSphere Hypervisor (ESXi) and click **GO TO DOWNLOADS**.

6 Download an ESXi ISO image.

7 Confirm the SHA256 checksum.

Note vSphere 8.0 removes insecure default ciphers such as SHA1 and MD5 and replaces them with secure ciphers such as SHA256.

For an evaluation copy of ESXi, go to https://customerconnect.vmware.com/en/evalcenter?p=free-esxi8.

For more information on ESXi downloads, see VMware knowledge base article 2107518.

For product patches to ESXi, see VMware knowledge base article 1021623 or go to https://my.vmware.com/group/vmware/patch.

ESXi Storage Device Names and Identifiers

In the ESXi environment, each storage device is identified by several names.

Device Identifiers

Depending on the type of storage, the ESXi host uses different algorithms and conventions to generate an identifier for each storage device.

Storage-provided identifiers

The ESXi host queries a target storage device for the device name. From the returned metadata, the host extracts or generates a unique identifier for the device. The identifier is based on specific storage standards, is unique and persistent across all hosts, and has one of the following formats:

- naa.*xxx*
- eui.*xxx*
- t10.*xxx*

Path-based identifier

When the device does not provide an identifier, the host generates an mpx.*path* name, where *path* represents the first path to the device, for example, `mpx.vmhba1:C0:T1:L3`. This identifier can be used in the same way as the storage-provided identifies.

The mpx.*path* identifier is created for local devices on the assumption that their path names are unique. However, this identifier is not unique or persistent, and can change after every system restart.

Typically, the path to the device has the following format:

vmhba*Adapter*:C*Channel*:T*Target*:L*LUN*

- vmhba*Adapter* is the name of the storage adapter. The name refers to the physical adapter on the host, not to the SCSI controller used by the virtual machines.
- C*Channel* is the storage channel number.

  Software iSCSI adapters and dependent hardware adapters use the channel number to show multiple paths to the same target.

- T*Target* is the target number. Target numbering is determined by the host and might change when the mappings of targets visible to the host change. Targets that are shared by different hosts might not have the same target number.
- L*LUN* is the LUN number that shows the position of the LUN within the target. The LUN number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0).

For example, `vmhba1:C0:T3:L1` represents LUN1 on target 3 accessed through the storage adapter vmhba1 and channel 0.

Legacy identifier

In addition to the device-provided identifiers or mpx.*path* identifiers, ESXi generates an alternative legacy name for each device. The identifier has the following format:

vml.*number*

The legacy identifier includes a series of digits that are unique to the device. The identifier can be derived in part from the metadata obtained through the SCSI INQUIRY command. For nonlocal devices that do not provide SCSI INQUIRY identifiers, the vml.*number* identifier is used as the only available unique identifier.

Example: Displaying Device Names in the vSphere CLI

You can use the `esxcli storage core device list` command to display all device names in the vSphere CLI. The output is similar to the following example:

```
# esxcli storage core device list
naa.XXX
      Display Name: DGC Fibre Channel Disk(naa.XXX)
      ...
      Other UIDs: vml.000XXX
mpx.vmhba1:C0:T0:L0
   Display Name: Local VMware Disk (mpx.vmhba1:C0:T0:L0)
   ...
   Other UIDs: vml.0000000000XYZ
```

Upgrade Hosts Interactively

To upgrade ESXi 6.7 hosts or ESXi 7.0 hosts to ESXi 8.0, you can boot the ESXi installer from a CD, DVD, or USB flash drive.

Before upgrading, consider disconnecting the network storage. This action decreases the time it takes the installer to search for available disk drives. When you disconnect network storage, any files on the disconnected disks are unavailable at installation. Do not disconnect a LUN that contains an existing ESXi installation.

Note Interactive upgrade is not supported on ESXi hosts with a data processing unit (DPU).

Prerequisites

- Verify that the ESXi installer ISO in one of the following locations.
  - On CD or DVD. If you do not have the installation CD or DVD, you can create one. See Download and Burn the ESXi Installer ISO Image to a CD or DVD
  - On a USB flash drive. See Format a USB Flash Drive to Boot the ESXi Installation or Upgrade

  Note You can also use PXE to boot the ESXi installer to run an interactive installation or a scripted installation. See Overview of the Network Boot Installation Process.
- Verify that the server hardware clock is set to UTC. This setting is in the system BIOS or UEFI.
- ESXi Embedded must not be on the host. ESXi Installable and ESXi Embedded cannot exist on the same host.
- If you are upgrading an ESXi host, supported custom VIBs that are not included in the ESXi installer ISO are migrated. See Upgrading Hosts That Have Third-Party Custom VIBs
- See your hardware vendor documentation for information about changing the boot order.

Procedure

1. Insert the ESXi installer CD or DVD in the CD-ROM or DVD-ROM drive, or attach the Installer USB flash drive and restart the machine.
2. Set the BIOS or UEFI to boot from the CD-ROM device or the USB flash drive.
3. In the Select a Disk panel, select the drive on which to install or upgrade ESXi and press Enter.

   Press F1 for information about the selected disk.

   Note Do not rely on the disk order in the list to select a disk. The disk order is determined by the BIOS or UEFI. On systems where drives are continuously being added and removed, they might be out of order.
4. Upgrade or install ESXi if the installer finds an existing ESXi installation and VMFS datastore.

   If an existing VMFS datastore cannot be preserved, you can choose only to install ESXi and overwrite the existing VMFS datastore, or to cancel the installation. If you choose to overwrite the existing VMFS datastore, back up the datastore first.
5. Press F11 to confirm and start the upgrade.
6. Remove the installation CD or DVD or USB flash drive when the upgrade is complete.
7. Press Enter to reboot the host.
8. Set the first boot device to be the drive which you selected previously when you upgraded ESXi.

Installing or Upgrading Hosts by Using a Script

You can quickly deploy ESXi hosts by using scripted, unattended installations or upgrades. Scripted installations or upgrades provide an efficient way to deploy multiple hosts.

The installation or upgrade script contains the installation settings for ESXi. You can apply the script to all hosts that you want to have a similar configuration.

For a scripted installation or upgrade, you must use the supported commands to create a script. You can edit the script to change settings that are unique for each host.

The installation or upgrade script can reside in one of the following locations:

- FTP server
- HTTP/HTTPS server
- NFS server
- USB flash drive
- CD-ROM drive

Enter Boot Options to Start an Installation or Upgrade Script

You can start an installation or upgrade script by typing boot options at the ESXi installer boot command line.

At boot time you might need to specify options to access the kickstart file. You can enter boot options by pressing Shift+O in the boot loader. For a PXE boot installation, you can pass options through the `kernelopts` line of the `boot.cfg` file. See About the boot.cfg File and Network Booting the ESXi Installer.

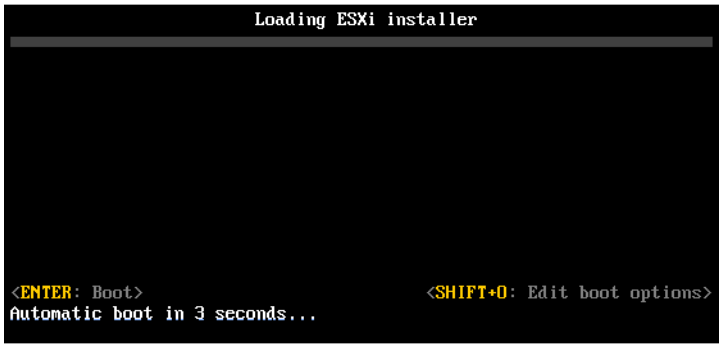
To specify the location of the installation script, set the `ks=filepath` option, where *filepath* indicates the location of your kickstart file. Otherwise, a scripted installation or upgrade cannot start. If `ks=filepath` is omitted, the text installer is run.

Supported boot options are listed in Boot Options .

Procedure

1 Start the host.

2 When the ESXi installer window appears, press Shift+O to edit boot options.



3 At the `runweasel` command prompt, type **ks=*location of installation script plus boot command-line options***.

Example: Boot Option

You type the following boot options:

```
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
```

Boot Options

When you perform a scripted installation, you might need to specify options at boot time to access the kickstart file.

Supported Boot Options

Table 3-7. Boot Options for ESXi Installation

Boot Option	Description
`BOOTIF=`*`hwtype-MAC address`*	Similar to the `netdevice` option, except in the PXELINUX format as described in the IPAPPEND option under SYSLINUX at the `syslinux.org` site.
`gateway=`*`ip address`*	Sets this network gateway as the default gateway to be used for downloading the installation script and installation media.
`ip=`*`ip address`*	Sets up a static IP address to be used for downloading the installation script and the installation media. Note: the PXELINUX format for this option is also supported. See the IPAPPEND option under SYSLINUX at the `syslinux.org` site.

Table 3-7. Boot Options for ESXi Installation (continued)

Boot Option	Description
`ks=cdrom:/path`	Performs a scripted installation with the script at *path*, which resides on the CD in the CD-ROM drive. Each CDROM is mounted and checked until the file that matches the path is found. **Important** If you have created an installer ISO image with a custom installation or upgrade script, you must use uppercase characters to provide the path of the script, for example, `ks=cdrom:/KS_CUST.CFG`.
`ks=file://path`	Performs a scripted installation with the script at *path*.
`ks=protocol://serverpath`	Performs a scripted installation with a script located on the network at the given URL. *protocol* can be `http`, `https`, `ftp`, or `nfs`. An example using NFS protocol is `ks=nfs://host/porturl-path`. The format of an NFS URL is specified in RFC 2224.
`ks=usb`	Performs a scripted installation, accessing the script from an attached USB drive. Searches for a file named `ks.cfg`. The file must be located in the root directory of the drive. If multiple USB flash drives are attached, they are searched until the `ks.cfg` file is found. Only FAT16 and FAT32 file systems are supported.
`ks=usb:/path`	Performs a scripted installation with the script file at the specified path, which resides on USB.
`ksdevice=device`	Tries to use a network adapter *device* when looking for an installation script and installation media. Specify as a MAC address, for example, 00:50:56:C0:00:01. This location can also be a vmnicNN name. If not specified and files need to be retrieved over the network, the installer defaults to the first discovered network adapter that is plugged in.
`nameserver=ip address`	Specifies a domain name server to be used for downloading the installation script and installation media.
`netdevice=device`	Tries to use a network adapter *device* when looking for an installation script and installation media. Specify as a MAC address, for example, 00:50:56:C0:00:01. This location can also be a vmnicNN name. If not specified and files need to be retrieved over the network, the installer defaults to the first discovered network adapter that is plugged in.
`netmask=subnet mask`	Specifies subnet mask for the network interface that downloads the installation script and the installation media.

Table 3-7. Boot Options for ESXi Installation (continued)

Boot Option	Description
`vlanid=vlanid`	Configure the network card to be on the specified VLAN.
`systemMediaSize=small`	Limits the size of system storage partitions on the boot media. The selected value must fit the purpose of your system. You can select from the following values: ■ *min* (32 GB, for single disk or embedded servers) ■ *small* (64 GB, for servers with at least 512 GB RAM) ■ *default* (128 GB) ■ *max* (consume all available space, for multi-terabyte servers) **Note** GB units are 2^30 bytes or 1024*1024*1024 byte multiples.

For more information on ESXi booting options post installation, see VMware knowledge base article 77009.

About Installation and Upgrade Scripts

The installation/upgrade script is a text file, for example `ks.cfg`, that contains supported commands.

The command section of the script contains the ESXi installation options. This section is required and must appear first in the script.

About the boot.cfg File

The boot loader configuration file `boot.cfg` specifies the kernel, the kernel options, and the boot modules that the `mboot.c32` or `mboot.efi` boot loader uses in an ESXi installation.

The `boot.cfg` file is provided in the ESXi installer. You can modify the `kernelopt` line of the `boot.cfg` file to specify the location of an installation script or to pass other boot options.

The `boot.cfg` file has the following syntax:

```
# boot.cfg -- mboot configuration file
#
# Any line preceded with '#' is a comment.

title=STRING
prefix=DIRPATH
kernel=FILEPATH
kernelopt=STRING
modules=FILEPATH1 --- FILEPATH2... --- FILEPATHn

# Any other line must remain unchanged.
```

The commands in `boot.cfg` configure the boot loader.

Table 3-8. Commands in `boot.cfg`.

Command	Description
`title=`*`STRING`*	Sets the boot loader title to *`STRING`*.
`prefix=`*`STRING`*	(Optional) Adds*`DIRPATH/`* in front of every *`FILEPATH`* in the `kernel=` and `modules=` commands that do not already start with / or with `http://`.
`kernel=`*`FILEPATH`*	Sets the kernel path to *`FILEPATH`*.
`kernelopt=`*`STRING`*	Appends *`STRING`* to the kernel boot options.
`modules=`*`FILEPATH1`* `---` *`FILEPATH2`*`...` `---` *`FILEPATHn`*	Lists the modules to be loaded, separated by three hyphens (`---`).

See Create an Installer ISO Image with a Custom Installation or Upgrade Script and Network Booting the ESXi Installer.

Locations Supported for Installation or Upgrade Scripts

In scripted installations and upgrades, the ESXi installer can access the installation or upgrade script, also called the kickstart file, from several locations.

The following locations are supported for the installation or upgrade script:

- CD/DVD. See Create an Installer ISO Image with a Custom Installation or Upgrade Script.
- USB Flash drive. See Create a USB Flash Drive to Store the ESXi Installation Script or Upgrade Script.
- A network location accessible through the following protocols: NFS, HTTP, HTTPS, FTP

Path to the Installation or Upgrade Script

You can specify the path to an installation or upgrade script.

`ks=http://`*`XXX.XXX.XXX.XXX`*`/kickstart/KS.CFG` is the path to the ESXi installation script, where *`XXX.XXX.XXX.XXX`* is the IP address of the machine where the script resides. See About Installation and Upgrade Scripts.

To start an installation script from an interactive installation, you enter the `ks=` option manually. See Enter Boot Options to Start an Installation or Upgrade Script.

Installation and Upgrade Script Commands

To modify the default installation or upgrade script or to create your own script, use supported commands. Use supported commands in the installation script, which you specify with a boot command when you boot the installer.

To determine which disk to install or upgrade ESXi on, the installation script requires one of the following commands: `install`, `upgrade`, or `installorupgrade`. The `install` command creates the default partitions, including a VMFS datastore that occupies all available space after the other partitions are created.

With vSphere 8.0, if your system has supported data processing units (DPU), always consider the installation, re-installation or upgrade of ESXi on the DPUs along with ESXi on hosts. However, ESXi update and upgrade on DPUs is not supported by the interactive or scripted method, you can only use vSphere Lifecycle Manager.

Note The use of SD and USB devices for storing ESX-OSData partitions is being deprecated. You can use SD and USB devices only to create boot bank partitions, `boot-bank 0` and `boot-bank 1`. Additionally, you can provide a persistent disk of minimum 32 GB on which to install the ESX-OSData partition. You define such disks by using the parameter `systemDisk` in the `install` command.

accepteula or vmaccepteula (Required)

Accepts the ESXi license agreement.

clearpart (Optional)

Clears any existing partitions on the disk. Requires the `install` command to be specified. Carefully edit the `clearpart` command in your existing scripts.

`--drives=`
Remove partitions on the specified drives.

`--alldrives`
Ignores the `--drives=` requirement and allows clearing of partitions on every drive.

`--ignoredrives=`
Removes partitions on all drives except those specified. Required unless the `--drives=` or `--alldrives` flag is specified.

`--overwritevmfs`
Allows overwriting of VMFS partitions on the specified drives. By default, overwriting VMFS partitions is not allowed.

`--firstdisk=`

`disk-type1`

`[`*`disk-type2`*`,...]`

Note In case you system has DPUs, you also specify a PCI slot.

Partitions the first eligible disk found. By default, the eligible disks are set to the following order:

1. Locally attached storage (`local`)
2. Network storage (`remote`)

You can change the order of the disks by using a comma-separated list appended to the argument. If you provide a filter list, the default settings are overridden. You can combine filters to specify a particular disk, including `esx` for the first disk with ESXi installed on it, model and vendor information, or the name of the VMkernel device driver. For example, to prefer a disk with the model name ST3120814A and

any disk that uses the mptsas driver rather than a normal local disk, the argument is `--firstdisk=ST3120814A,mptsas,local`. You can use `localesx` for local storage that contains ESXi image or `remoteesx` for remote storage that contains ESXi image.

dryrun (Optional)

Parses and checks the installation script. Does not perform the installation.

install

Specifies that this is a fresh installation. Either the `install`, `upgrade`, or `installorupgrade` command is required to determine which disk to install or upgrade ESXi on.

`--disk= or --drive=`

Specifies the disk to partition. In the command `--disk=`*`diskname`*, the *diskname* can be a disk name or a full disk filesystem path in ESXi, for example:

- Disk name: `--disk=naa.6d09466044143600247aee55ca2a6405` or
- Device path: `--disk=`*`/vmfs/devices/disks/mpx.vmhba1:C0:T0:L0`*

For accepted disk name formats, see Disk Device Names.

`--firstdisk=`

`disk-type1,`

`[`*`disk-type2`*`,...]`

Note In case you system has DPUs, you also specify a PCI slot:
`install --firstdisk --overwritevmfs --dpuPciSlots=<PCIeSlotID>`

Partitions the first eligible disk found. By default, the eligible disks are set to the following order:

1 Locally attached storage (`local`)

2 Network storage (`remote`)

You can change the order of the disks by using a comma-separated list appended to the argument. If you provide a filter list, the default settings are overridden. You can combine filters to specify a particular disk, including `esx` for the first disk with ESX installed on it, model and vendor information, or the name of the VMkernel device driver. For example, to prefer a disk with the model name ST3120814A and any disk that uses the mptsas driver rather than a normal local disk, the argument is `--firstdisk=ST3120814A,mptsas,local`. You can use `localesx` for local storage that contains ESXi image or `remoteesx` for remote storage that contains ESXi image.

`--ignoressd`

Excludes solid-state disks from eligibility for partitioning. This option can be used with the `install` command and the `--firstdisk` option. This option takes precedence over the `--firstdisk` option. This option is invalid with the `--drive` or `--disk` options and with

	the `upgrade` and `installorupgrade` commands. See the *vSphere Storage* documentation for more information about preventing SSD formatting during auto-partitioning.
`--overwritevsan`	You must use the `--overwritevsan` option when you install ESXi on a disk, either SSD or HDD (magnetic), that is in a vSAN disk group. If you use this option and no vSAN partition is on the selected disk, the installation fails. When you install ESXi on a disk that is in vSAN disk group, the result depends on the disk that you select: ■ If you select an SSD, the SSD and all underlying HDDs in the same disk group is wiped. ■ If you select an HDD, and the disk group size is greater than two, only the selected HDD is wiped. ■ If you select an HDD disk, and the disk group size is two or less, the SSD and the selected HDD is wiped. For more information about managing vSAN disk groups, see the *vSphere Storage* documentation.
`--overwritevmfs`	Required to overwrite an existing VMFS datastore on the disk before installation.
`--preservevmfs`	Preserves an existing VMFS datastore on the disk during installation.
`--novmfsondisk`	Prevents a VMFS partition from being created on this disk. Must be used with `--overwritevmfs` if a VMFS partition exists on the disk.
`--systemdisk`	If you use an USB or SD device, `systemDisk` specifies local persistent disk on which to install the ESX-OSData partition. For example, **`install --firstdisk = usb --systemDisk=<diskID>`**. As a result, boot bank partitions are placed on the USB device, while the OSData partition is on the disk specified in the `systemDisk` parameter.
`--repartitionsystemdisk`	If you use an USB or SD device and the local disk that you specify with the `systemDisk` parameter is not empty or contains a datastore, you can use `repartitionSystemDisk` to make sure that the persistent disk is repartitioned before use. **Note** If a local persistent disk is not available or the disk size is less than 32GB, you see warning messages, but installation continues.
`--forceunsupportedinstall`	Blocks the installation of deprecated CPUs.

installorupgrade

Either the `install`, `upgrade`, or `installorupgrade` command is required to determine which disk to install or upgrade ESXi on.

`--disk= or --drive=`

Specifies the disk to partition. In the command `--disk=`*`diskname`*, the *diskname* can be a disk name or a full disk filesystem path in ESXi, for example:

- Disk name: `--disk=naa.6d09466044143600247aee55ca2a6405` or
- Device path: `--disk=`*`/vmfs/devices/disks/mpx.vmhba1:C0:T0:L0`*

For accepted disk name formats, see Disk Device Names.

`--firstdisk=`

`disk-type1,`

`[`*`disk-type2`*`,...]`

Note In case you system has DPUs, you also specify a PCI slot: `installorupgrade --firstdisk --overwritevmfs --dpuPciSlots=<PCIeSlotID>`

Partitions the first eligible disk found. By default, the eligible disks are set to the following order:

1. Locally attached storage (`local`)
2. Network storage (`remote`)

You can change the order of the disks by using a comma-separated list appended to the argument. If you provide a filter list, the default settings are overridden. You can combine filters to specify a particular disk, including `esx` for the first disk with ESX installed on it, model and vendor information, or the name of the VMkernel device driver. For example, to prefer a disk with the model name ST3120814A and any disk that uses the mptsas driver rather than a normal local disk, the argument is `--firstdisk=ST3120814A,mptsas,local`. You can use `localesx` for local storage that contains ESXi image or `remoteesx` for remote storage that contains ESXi image.

`--overwritevsan`

You must use the `--overwritevsan` option when you install ESXi on a disk, either SSD or HDD (magnetic), that is in a vSAN disk group. If you use this option and no vSAN partition is on the selected disk, the installation fails. When you install ESXi on a disk that is in a vSAN disk group, the result depends on the disk that you select:

- If you select an SSD, the SSD and all underlying HDDs in the same disk group is wiped.
- If you select an HDD, and the disk group size is greater than two, only the selected HDD is wiped.
- If you select an HDD disk, and the disk group size is two or less, the SSD and the selected HDD is wiped.

For more information about managing vSAN disk groups, see the *vSphere Storage* documentation.

`--overwritevmfs` Install ESXi if a VMFS partition exists on the disk, but no ESX or ESXi installation exists. Unless this option is present, the installer fails if a VMFS partition exists on the disk, but an ESX or ESXi installation is missing.

keyboard (Optional)

Sets the keyboard type for the system.

`keyboardType` Specifies the keyboard map for the selected keyboard type. *keyboardType* must be one of the following types.

- Belgian
- Brazilian
- Croatian
- Czechoslovakian
- Danish
- Estonian
- Finnish
- French
- German
- Greek
- Icelandic
- Italian
- Japanese
- Latin American
- Norwegian
- Polish
- Portuguese
- Russian
- Slovenian
- Spanish
- Swedish

- Swiss French
- Swiss German
- Turkish
- Ukrainian
- United Kingdom
- US Default
- US Dvorak

serialnum or vmserialnum (Optional)

The command is supported in ESXi version 5.1 and later. Configures licensing. If not included, ESXi installs in evaluation mode.

--esx=<license-key>	Specifies the vSphere license key to use. The format is 5 five-character groups (XXXXX-XXXXX-XXXXX-XXXXX-XXXXX).

network (Optional)

Specifies a network address for the system.

--bootproto=[dhcp|static]	Specifies whether to obtain the network settings from DHCP or set them manually.
--device=	Specifies either the MAC address of the network card or the device name, in the form `vmnicNN`, as in `vmnic0`. This option refers to the uplink device for the virtual switch.
--ip=	Sets an IP address for the machine to be installed, in the form `xxx.xxx.xxx.xxx`. Required with the `--bootproto=static` option and ignored otherwise.
--gateway=	Designates the default gateway as an IP address, in the form `xxx.xxx.xxx.xxx`. Used with the `--bootproto=static` option.
--nameserver=	Designates the primary name server as an IP address. Used with the `--bootproto=static` option. Omit this option if you do not intend to use DNS. The `--nameserver` option can accept two IP addresses. For example: `--nameserver="10.126.87.104[,10.126.87.120]"`
--netmask=	Specifies the subnet mask for the installed system, in the form `255.xxx.xxx.xxx`. Used with the `--bootproto=static` option.
--hostname=	Specifies the host name for the installed system.

--vlanid= *vlanid* Specifies which VLAN the system is on. Used with either the `--bootproto=dhcp` or `--bootproto=static` option. Set to an integer from 1 to 4096.

--addvmportgroup=(0|1) Specifies whether to add the VM Network port group, which is used by virtual machines. The default value is 1.

paranoid (Optional)

Causes warning messages to interrupt the installation. If you omit this command, warning messages are logged.

part or partition (Optional)

Creates an extra VMFS datastore on the system. Only one datastore per disk can be created. Cannot be used on the same disk as the `install` command. Only one partition can be specified per disk and it can only be a VMFS partition.

datastore name Specifies where the partition is to be mounted.

--ondisk= or --ondrive= Specifies the disk or drive where the partition is created.

--firstdisk=

disk-type1,

[*disk-type2*,...]

Note In case you system has DPUs, you also specify a PCI slot.

Partitions the first eligible disk found. By default, the eligible disks are set to the following order:

1. Locally attached storage (`local`)
2. Network storage (`remote`)

You can change the order of the disks by using a comma-separated list appended to the argument. If you provide a filter list, the default settings are overridden. You can combine filters to specify a particular disk, including `esx` for the first disk with ESX installed on it, model and vendor information, or the name of the VMkernel device driver. For example, to prefer a disk with the model name ST3120814A and any disk that uses the mptsas driver rather than a normal local disk, the argument is `--firstdisk=ST3120814A,mptsas,local`. You can use `localesx` for local storage that contains ESXi image or `remoteesx` for remote storage that contains ESXi image.

reboot (Optional)

Reboots the machine after the scripted installation is complete.

<--noeject> The CD is not ejected after the installation.

rootpw (Required)

Sets the root password for the system.

`--iscrypted` Specifies that the password is encrypted.

`password` Specifies the password value.

upgrade

Either the `install`, `upgrade`, or `installorupgrade` command is required to determine which disk to install or upgrade ESXi on.

`--disk= or --drive=` Specifies the disk to partition. In the command `--disk=`*`diskname`*, the *diskname* can be a disk name or a full disk filesystem path in ESXi, for example:

- Disk name: `--disk=naa.6d09466044143600247aee55ca2a6405` or
- Device path: `--disk=`*`/vmfs/devices/disks/mpx.vmhba1:C0:T0:L0`*

For accepted disk name formats, see Disk Device Names.

`--firstdisk=`

`disk-type1,`

`[`*`disk-type2`*`,...]`

Partitions the first eligible disk found. By default, the eligible disks are set to the following order:

1. Locally attached storage (`local`)
2. Network storage (`remote`)

You can change the order of the disks by using a comma-separated list appended to the argument. If you provide a filter list, the default settings are overridden. You can combine filters to specify a particular disk, including `esx` for the first disk with ESX installed on it, model and vendor information, or the name of the VMkernel device driver. For example, to prefer a disk with the model name ST3120814A and any disk that uses the mptsas driver rather than a normal local disk, the argument is `--firstdisk=ST3120814A,mptsas,local`. You can use `localesx` for local storage that contains ESXi image or `remoteesx` for remote storage that contains ESXi image.

%include or include (Optional)

Specifies another installation script to parse. This command is treated similarly to a multiline command, but takes only one argument.

`filename` For example: `%include part.cfg`

%pre (Optional)

Specifies a script to run before the kickstart configuration is evaluated. For example, you can use it to generate files for the kickstart file to include.

--interpreter=[python|busybox]	Specifies an interpreter to use. The default is busybox.

%post (Optional)

Runs the specified script after package installation is complete. If you specify multiple `%post` sections, they run in the order that they appear in the installation script.

--interpreter=[python|busybox]	Specifies an interpreter to use. The default is busybox.
--timeout=secs	Specifies a timeout for running the script. If the script is not finished when the timeout expires, the script is forcefully stopped.
--ignorefailure=[true|false]	If true, the installation is considered a success even if the `%post` script stops with an error.

%firstboot

Creates an `init` script that runs only during the first boot. The script has no effect on subsequent boots. If multiple `%firstboot` sections are specified, they run in the order that they appear in the kickstart file.

Note You cannot check the semantics of `%firstboot` scripts until the system is booting for the first time. A `%firstboot` script might contain potentially catastrophic errors that are not exposed until after the installation is complete.

Important The `%firstboot` script does not run, if secure boot is enabled on the ESXi host.

--interpreter=[python|busybox]	Specifies an interpreter to use. The default is busybox.

Note You cannot check the semantics of the `%firstboot` script until the system boots for the first time. If the script contains errors, they are not exposed until after the installation is complete.

Install or Upgrade ESXi from a CD or DVD by Using a Script

You can install or upgrade ESXi from a CD-ROM or DVD-ROM drive by using a script that specifies the installation or upgrade options.

You can start the installation or upgrade script by entering a boot option when you start the host. You can also create an installer ISO image that includes the installation script. With an installer ISO image, you can perform a scripted, unattended installation when you boot the resulting installer ISO image. See Create an Installer ISO Image with a Custom Installation or Upgrade Script.

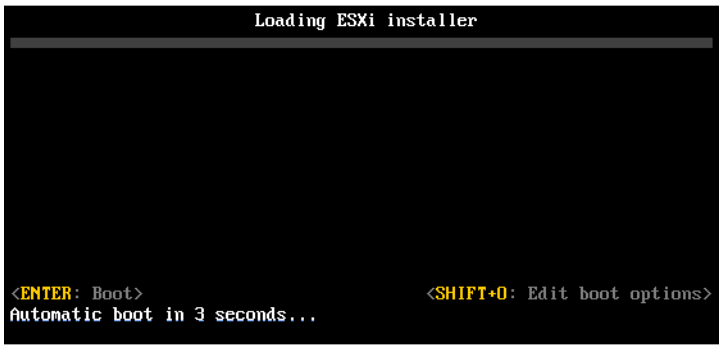
Prerequisites

Before you run the scripted installation or upgrade, verify that the following prerequisites are met:

- The system on which you are installing or upgrading meets the hardware requirements. See ESXi Hardware Requirements.
- You have the ESXi installer ISO on an installation CD or DVD . See Download and Burn the ESXi Installer ISO Image to a CD or DVD.
- The default installation or upgrade script (`ks.cfg`) or a custom installation or upgrade script is accessible to the system. See About Installation and Upgrade Scripts.
- You have selected a boot command to run the scripted installation or upgrade. See Enter Boot Options to Start an Installation or Upgrade Script. For a complete list of boot commands, see Boot Options .

Procedure

1. Boot the ESXi installer from the local CD-ROM or DVD-ROM drive.
2. When the ESXi installer window appears, press Shift+O to edit boot options.



3. Type a boot option that calls the default installation or upgrade script or an installation or upgrade script file that you created.

   The boot option has the form `ks=`.
4. Press Enter.

Results

The installation, upgrade, or migration runs, using the options that you specified.

Install or Upgrade ESXi from a USB Flash Drive by Using a Script

You can install or upgrade ESXi from a USB flash drive by using a script that specifies the installation or upgrade options.

Supported boot options are listed in Boot Options .

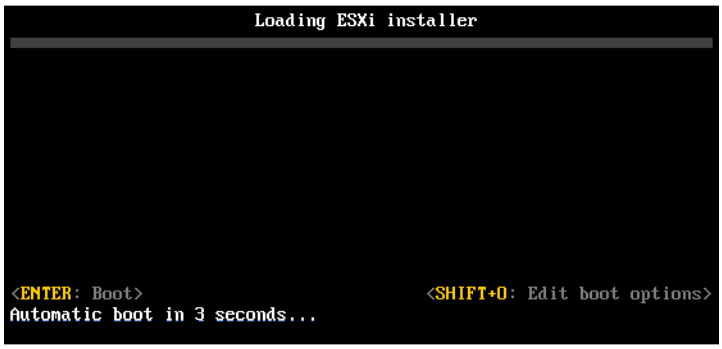
Prerequisites

Before running the scripted installation or upgrade, verify that the following prerequisites are met:

- The system that you are installing or upgrading to ESXi meets the hardware requirements for the installation or upgrade. See ESXi Hardware Requirements.
- You have the ESXi installer ISO on a bootable USB flash drive. See Format a USB Flash Drive to Boot the ESXi Installation or Upgrade.
- The default installation or upgrade script (`ks.cfg`) or a custom installation or upgrade script is accessible to the system. See About Installation and Upgrade Scripts.
- You have selected a boot option to run the scripted installation, upgrade, or migration. See Enter Boot Options to Start an Installation or Upgrade Script.

Procedure

1. Boot the ESXi installer from the USB flash drive.
2. When the ESXi installer window appears, press Shift+O to edit boot options.



3. Type a boot option that calls the default installation or upgrade script or an installation or upgrade script file that you created.

   The boot option has the form `ks=`.
4. Press Enter.

Results

The installation, upgrade, or migration runs, using the options that you specified.

Performing a Scripted Installation or Upgrade of ESXi by Network Booting the Installer

ESXi 8.0 provides many options for booting the installer over a network and using an installation or upgrade script.

- For information about setting up a network infrastructure, see Network Booting the ESXi Installer.
- For information about creating and locating an installation script, see About Installation and Upgrade Scripts.
- For specific procedures to network boot the ESXi installer and use an installation script, see one of the following topics:
  - Boot the ESXi Installer by Using Native UEFI HTTP
  - Boot the ESXi Installer by Using iPXE and HTTP
  - Boot the ESXi Installer by Using PXE and TFTP
- For information about using vSphere Auto Deploy to perform a scripted upgrade by using PXE to boot, see Chapter 4 Using vSphere Auto Deploy to Reprovision Hosts .

Disk Device Names

The `install`, `upgrade`, and `installorupgrade` installation script commands require the use of disk device names.

Table 3-9. Disk Device Names

Format	Example	Description
NAA	naa.6d09466044143600247aee55ca2a6405	SCSI INQUIRY identifier
EUI	eui.3966623838646463	SCSI INQUIRY identifier
T10	t10.SanDisk00Cruzer_Blade000000004C5300 01171118101244	SCSI INQUIRY identifier
VML	vml.00025261	Legacy VMkernel identifier
MPX	mpx.vmhba0:C0:T0:L0	Path-based identifier

For more information on storage device names, see *Storage Device Names and Identifiers* in the *vSphere Storage* documentation.

How to Boot an ESXi Host from a Network Device

Network Booting the ESXi Installer

You can use preboot execution environment (PXE) to boot an ESXi host from a network device, if your host uses legacy BIOS or UEFI. Alternatively, if your ESXi host supports native UEFI HTTP, you can use hypertext transfer protocol (HTTP) to boot the host from a network device.

ESXi is distributed in an ISO format that is used to install to flash memory or to a local hard drive. You can extract the files and boot them over a network interface.

PXE uses Dynamic Host Configuration Protocol (DHCP) and Trivial File Transfer Protocol (TFTP) to boot an operating system over a network.

PXE booting requires some network infrastructure and a machine with a PXE-capable network adapter. Most machines that can run ESXi have network adapters that can PXE boot.

Native UEFI HTTP uses DHCP and HTTP to boot over a network. UEFI HTTP boot requires a network infrastructure, UEFI firmware version on the ESXi host that includes HTTP boot feature, and a network adapter that supports UEFI networking.

Booting by using HTTP is faster and more reliable than using TFTP. This is due to the capabilities of the TCP protocol that underlies the HTTP, such as built-in streaming and lost packet recovery. If your ESXi hosts do not support native UEFI HTTP, you can use iPXE HTTP for the boot process.

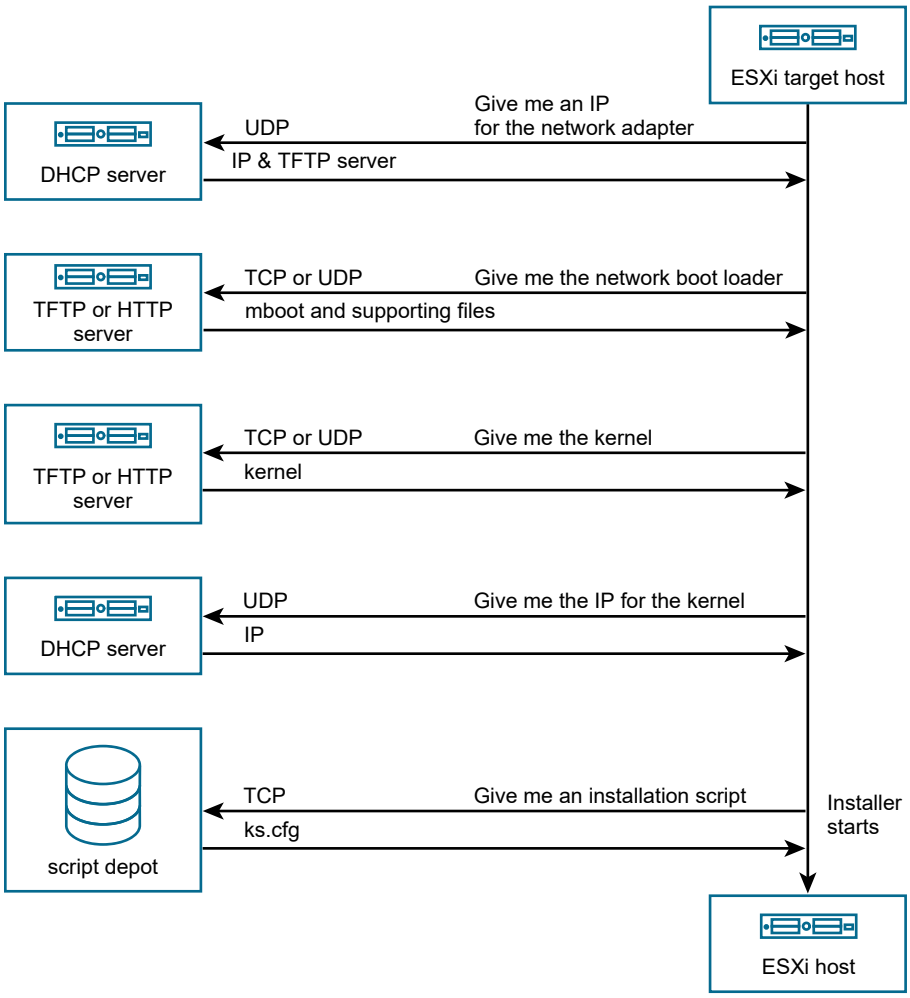
Note Network booting with legacy BIOS firmware is possible only over IPv4. Network booting with UEFI BIOS firmware is possible over IPv4 or IPv6.

Overview of the Network Boot Installation Process

You can boot an ESXi host from a network interface. The network boot process varies depending on whether the target host is using legacy BIOS or UEFI firmware, and whether the boot process uses PXE TFTP, iPXE HTTP, or UEFI HTTP.

When you boot a target host, it interacts with the different servers in the environment to get a network adapter, boot loader, kernel, IP address for the kernel, and finally an installation script. When all components are in place, installation starts, as shown in the following illustration.

Figure 3-2. Overview of PXE Boot Installation Process



The interaction between the ESXi host and other servers proceeds as follows:

1 The user boots the target ESXi host.

2 The target ESXi host makes a DHCP request.

3 The DHCP server responds with the IP information, the location of the TFTP or HTTP server, and the filename or URL of the initial network boot loader.

4 The ESXi host contacts the TFTP or HTTP server and requests the filename or URL that the DHCP server specified.

5 The TFTP or HTTP server sends the network boot loader, and the ESXi host runs it. The initial boot loader might load additional boot loader components from the server.

6 The boot loader searches for a configuration file on the TFTP or HTTP server, downloads the kernel and other ESXi components as specified in the configuration file, and boots the kernel on the ESXi host.

7 The installer runs interactively or using a kickstart script, as specified in the configuration file.

Network Boot Background Information

Understanding the network boot process can help you during troubleshooting.

TFTP Server

Trivial File Transfer Protocol (TFTP) is similar to the FTP service, and is typically used only for network booting systems or loading firmware on network devices such as routers. TFTP is available on Linux and Windows.

- Most Linux distributions include a copy of the tftp-hpa server. If you require a supported solution, purchase a supported TFTP server from your vendor of choice. You can also acquire a TFTP server from one of the packaged appliances on the VMware Marketplace.
- If your TFTP server runs on a Microsoft Windows host, use tftpd32 version 2.11 or later. See http://tftpd32.jounin.net/.

SYSLINUX and PXELINUX

If you are using PXE in a legacy BIOS environment, you must understand the different boot environments.

- SYSLINUX is an open-source boot environment for machines that run legacy BIOS firmware. The ESXi boot loader for BIOS systems, `mboot.c32`, runs as a SYSLINUX plugin. You can configure SYSLINUX to boot from several types of media, including disk, ISO image, and network. You can find the SYSLINUX package at http://www.kernel.org/pub/linux/utils/boot/syslinux/.
- PXELINUX is a SYSXLINUX configuration for booting from a TFTP server according to the PXE standard. If you use PXELINUX to boot the ESXi installer, the `pxelinux.0` binary file, `mboot.c32`, the configuration file, the kernel, and other files are transferred by TFTP.

Note VMware builds the `mboot.c32` plugin to work with SYSLINUX version 3.86 and tests PXE booting only with that version. Other versions might be incompatible. *The Open Source Disclosure Package for VMware vSphere Hypervisor* includes bug fixes for SYSLINUX version 3.86.

iPXE

iPXE is open-source software that provides an implementation of HTTP. You can use the software to perform an initial boot. For more information, see https://ipxe.org/.

VMware includes a build of iPXE as part of Auto Deploy. The source tree for this build is available in *The Open Source Disclosure Package for VMware vCenter Server*.

UEFI PXE and UEFI HTTP

Most UEFI firmware natively includes PXE support that allows booting from a TFTP server. The firmware can directly load the ESXi boot loader for UEFI systems, `mboot.efi`. Additional software such as PXELINUX is not required.

Some UEFI firmware support native UEFI HTTP boot. The feature is introduced in version 2.5 of the UEFI specification. The firmware can load the ESXi boot loader from an HTTP server, without additional software, such as iPXE.

Note Apple Macintosh products do not include PXE boot support. They include support for network booting through an Apple-specific protocol instead.

Alternative Approaches to Network Booting

Alternative approaches to network booting different software on different hosts are also possible, for example:

- Configuring the DHCP server to provide different initial boot loader filenames to different hosts depending on MAC address or other criteria. See your DCHP server's documentation.
- Approaches using iPXE as the initial bootloader with an iPXE configuration file that selects the next bootloader based on the MAC address or other criteria.

PXELINUX Configuration Files

You need a PXELINUX configuration file to boot the ESXi installer on a legacy BIOS system. The configuration file defines the menu displayed to the target ESXi host as it starts.

This section gives general information about PXELINUX configuration files.

For syntax details, see the SYSLINUX website at http://www.syslinux.org/.

Required Files

In the PXE configuration file, you must include paths to the following files:

- `mboot.c32` is the boot loader.
- `boot.cfg` is the boot loader configuration file.

See About the boot.cfg File

Filename for the PXE Configuration File

For the filename of the PXE configuration file, select one of the following options:

- `01-`*`mac_address_of_target_ESXi_host`*. For example, `01-23-45-67-89-0a-bc`
- The target ESXi host IP address in a hexadecimal notation.
- `default`

The initial boot file `pxelinux.0` tries to load a PXE configuration file in the following order:

1. It tries with the MAC address of the target ESXi host, prefixed with its ARP type code, which is 01 for Ethernet.
2. If that attempt fails, it tries with the hexadecimal notation of target ESXi system IP address.
3. Ultimately, it tries to load a file named `default`.

File Location for the PXE Configuration File

Save the file in `/tftpboot/pxelinux.cfg/` on the TFTP server.

For example, you might save the file on the TFTP server at `/tftpboot/pxelinux.cfg/01-00-21-5a-ce-40-f6`. The MAC address of the network adapter on the target ESXi host is 00-21-5a-ce-40-f6.

Boot the ESXi Installer by Using PXE and TFTP

You can use a TFTP server to PXE boot the ESXi installer. The process differs slightly depending on whether you use UEFI or boot from a legacy BIOS. Because most environments include ESXi hosts that support UEFI boot and hosts that support only legacy BIOS, this topic discusses prerequisites and steps for both types of hosts.

- For legacy BIOS machines, the procedure supports booting multiple different versions of the ESXi installer by using the same `pxelinux.0` initial boot loader for all target machines, but potentially different PXELINUX configuration files depending on the target machine's MAC address.
- For UEFI machines, the procedure supports booting multiple different versions of the ESXi installer by using the same `mboot.efi` initial boot loader for all target machines, but potentially different `boot.cfg` files depending on the target machine's MAC address.

Prerequisites

Verify that your environment meets the following prerequisites.

- ESXi installer ISO image, downloaded from the VMware Web site.
- Target host with a hardware configuration that is supported for your version of ESXi. See the *VMware Compatibility Guide*.
- Network adapter with PXE support on the target ESXi host.
- DHCP server that you can configure for PXE booting. See Sample DHCP Configurations.
- TFTP server.
- Network security policies to allow TFTP traffic (UDP port 69).
- For legacy BIOS, you can use only IPv4 networking. For UEFI PXE boot, you can use IPv4 or IPv6 networking.
- (Optional) Installation script (kickstart file).
- Use a native VLAN in most cases. If you want to specify the VLAN ID to be used with PXE booting, check that your NIC supports VLAN ID specification.

For legacy BIOS systems, obtain version 3.86 of the SYSLINUX package. For more information, see Network Boot Background Information.

Procedure

1 If your ESXi host runs legacy BIOS firmware only, obtain and configure PXELINUX.

   a Obtain SYSLINUX version 3.86, unpack it, and copy the `pxelinux.0` file to the top-level `/tftpboot` directory on your TFTP server.

   b Create a PXELINUX configuration file using the following code model.

   `ESXi-8.x.x-XXXXXX` is the name of the TFTP subdirectory that contains the ESXi installer files.

```
DEFAULT install
NOHALT 1
LABEL install
  KERNEL ESXi-8.x.x-XXXXXX/mboot.c32
  APPEND -c ESXi-8.x.x-XXXXXX/boot.cfg
  IPAPPEND 2
```

   c Save the PXELINUX file in the `/tftpboot/pxelinux.cfg` directory on your TFTP server with a filename that will determine whether all hosts boot this installer by default:

Option	Description
Same installer	Name the file `default` if you want all host to boot this ESXi installer by default.
Different installers	Name the file with the MAC address of the target host machine (01-*mac_address_of_target_ESXi_host*) if you want only a specific host to boot with this file, for example, `01-23-45-67-89-0a-bc`.

2 If your ESXi host runs UEFI firmware, copy the `efi/boot/bootx64.efi` and `efi/boot/crypto64.efi` files from the ESXi installer ISO image to the `/tftpboot` folder on your TFTP server.

3 Rename the `efi/boot/bootx64.efi` file to `mboot.efi`.

Note Newer versions of `mboot.efi` can generally boot older versions of ESXi, but older versions of `mboot.efi` might be unable to boot newer versions of ESXi. If you plan to configure different hosts to boot different versions of the ESXi installer, use the `mboot.efi` from the newest version.

4 Configure the DHCP server.

5 Create a subdirectory of your TFTP server's top-level `/tftpboot` directory and name it after the version of ESXi it will hold, for example, `/tftpboot/ESXi-8.x.x-xxxxx`.

6 Copy the contents of the ESXi installer image to the newly created directory.

7 Modify the `boot.cfg` file

   a Add the following line:

   ```
   prefix=ESXi-7.x.x-xxxxxx
   ```

   Here, `ESXi-7.x.x-xxxxxx` is the pathname of the installer files relative to the TFTP server's root directory.

   b If the filenames in the `kernel=` and `modules=` lines begin with a forward slash (/) character, delete that character.

   c If the `kernelopt=` line contains the string `cdromBoot`, remove the string only.

8 (Optional) For a scripted installation, in the `boot.cfg` file, add the `kernelopt` option to the line after the kernel command, to specify the location of the installation script.

   Use the following code as a model, where `XXX.XXX.XXX.XXX` is the IP address of the server where the installation script resides, and `esxi_ksFiles` is the directory that contains the `ks.cfg` file.

   ```
   kernelopt=ks=http://XXX.XXX.XXX.XXX/esxi_ksFiles/ks.cfg
   ```

9 If your ESXi host runs UEFI firmware, specify whether you want all UEFI hosts to boot the same installer.

Option	Description
Same installer	Copy or link the `boot.cfg` file to `/tftpboot/boot.cfg`
Different installers	a Create a subdirectory of `/tftpboot` named after the MAC address of the target host machine (01-*mac_address_of_target_ESXi_host*), for example, `01-23-45-67-89-0a-bc`. b Place a copy of (or a link to) the host's `boot.cfg` file in that directory, for example, `/tftpboot/01-23-45-67-89-0a-bc/boot.cfg`.

Boot the ESXi Installer by Using iPXE and HTTP

You can use iPXE to boot the ESXi installer from an HTTP server. The following topic discusses prerequisites and steps for ESXi hosts that support UEFI boot and hosts that support legacy BIOS only.

- For legacy BIOS machines, the procedure supports booting multiple different versions of the ESXi installer by using the same `pxelinux.0` initial boot loader for all target machines, but potentially different PXELINUX configuration files depending on the target machine's MAC address.
- For UEFI machines, the procedure supports booting multiple different versions of the ESXi installer by using the same `mboot.efi` initial boot loader for all target machines, but potentially different `boot.cfg` files depending on the target machine's MAC address.

Prerequisites

Verify that your environment has the following components:

- ESXi installer ISO image, downloaded from the VMware Web site.
- Target host with a hardware configuration that is supported for your version of ESXi. See the *VMware Compatibility Guide*.
- Network adapter with PXE support on the target ESXi host.
- DHCP server that you can configure for PXE booting. See Sample DHCP Configurations.
- TFTP server.
- Network security policies to allow TFTP traffic (UDP port 69).
- For legacy BIOS, you can use only IPv4 networking. For UEFI PXE boot, you can use IPv4 or IPv6 networking.
- (Optional) Installation script (kickstart file).
- Use a native VLAN in most cases. If you want to specify the VLAN ID to be used with PXE booting, check that your NIC supports VLAN ID specification.

Verify that your environment also meets the following prerequisites required for PXE boot using an HTTP Server:

- Verify that the HTTP server is accessible by your target ESXi hosts.
- If your ESXi host runs legacy BIOS firmware only, obtain version 3.86 of the SYSLINUX package. For more information, see Network Boot Background Information.

Procedure

1 Obtain and configure iPXE.

   a Obtain the iPXE source code.

   b On the iPXE download page, follow the build instructions, but run one of the following commands.

      - For ESXi hosts that run legacy BIOS firmware only, run `make bin/undionly.kpxe`.
      - For ESXi hosts that run UEFI firmware, run `make bin-x86_64-efi/snponly.efi`.

   c Copy the `undionly.kpxe` or `snponly.efi` file to the `/tftpboot` directory on your TFTP server.

2 If your ESXi host runs legacy BIOS firmware only, obtain and configure PXELINUX.

   a Obtain SYSLINUX version 3.86, unpack it, and copy the `pxelinux.0` file to the `/tftpboot` directory on your TFTP server.

   b Create a PXELINUX configuration file using the following code model.

   `ESXi-8.x.x-XXXXXX` is the name of the TFTP subdirectory that contains the ESXi installer files.

```
DEFAULT install
NOHALT 1
LABEL install
  KERNEL ESXi-8.x.x-XXXXXX/mboot.c32
  APPEND -c ESXi-8.x.x-XXXXXX/boot.cfg
  IPAPPEND 2
```

   c Save the PXELINUX file in the `/tftpboot/pxelinux.cfg` directory on your TFTP server.

   The filename determines whether all hosts boot this installer by default.

Option	Description
Same installer	Name the file `default` if you want all host to boot this ESXi installer by default.
Different installers	Name the file with the MAC address of the target host machine (01-*mac_address_of_target_ESXi_host*), if only a specific host must boot this file. For example, `01-23-45-67-89-0a-bc`.

3 If your ESXi host runs UEFI firmware, copy the `efi/boot/bootx64.efi` file from the ESXi installer ISO image to the `/tftpboot` folder on your TFTP server, and rename the file to `mboot.efi`.

Note Newer versions of `mboot.efi` can generally boot older versions of ESXi, but older versions of `mboot.efi` might be unable to boot newer versions of ESXi. If you plan to configure different hosts to boot different versions of the ESXi installer, use the `mboot.efi` from the newest version.

4 Configure the DHCP server.

5 Create a directory on your HTTP server with the same name as the version of ESXi it will hold. For example, `/var/www/html/ESXi-8.x.x-XXXXXX`.

6 Copy the contents of the ESXi installer image to the newly created directory.

7 Modify the `boot.cfg` file

   a Add the following line:

   ```
   prefix=http://XXX.XXX.XXX.XXX/ESXi-8.x.x-XXXXXX
   ```

   where `http://XXX.XXX.XXX.XXX/ESXi-8.x.x-XXXXXX` is the location of the installer files on the HTTP server.

   b If the filenames in the `kernel=` and `modules=` lines begin with a forward slash (/) character, delete that character.

   c If the `kernelopt=` line contains the string `cdromBoot`, remove the string only.

8 (Optional) For a scripted installation, in the `boot.cfg` file, add the `kernelopt` option to the line after the kernel command, to specify the location of the installation script.

   Use the following code as a model, where `XXX.XXX.XXX.XXX` is the IP address of the server where the installation script resides, and `esxi_ksFiles` is the directory that contains the `ks.cfg` file.

   ```
   kernelopt=ks=http://XXX.XXX.XXX.XXX/esxi_ksFiles/ks.cfg
   ```

9 If your ESXi host runs UEFI firmware, specify whether you want all UEFI hosts to boot the same installer.

Option	Description
Same installer	Copy or link the `boot.cfg` file to `/tftpboot/boot.cfg`
Different installers	a Create a subdirectory of `/tftpboot` named after the MAC address of the target host machine (01-*mac_address_of_target_ESXi_host*), for example, `01-23-45-67-89-0a-bc`. b Place a copy of (or a link to) the host's `boot.cfg` file in that directory, for example, `/tftpboot/01-23-45-67-89-0a-bc/boot.cfg`.

Boot the ESXi Installer by Using Native UEFI HTTP

You can boot the ESXi installer directly from an HTTP server, without additional software to support the process.

UEFI HTTP supports booting multiple versions of the ESXi installer. You use the same `mboot.efi` initial boot loader for all target machines, but potentially different `boot.cfg` files depending on the target machine's MAC address.

Note Do not mix IPv4 or IPv6 networking during the boot process. Use either IPv4 or IPv6 networking.

Prerequisites

Verify that your environment has the following components:

- ESXi host with UEFI firmware that supports the HTTP boot feature.
- ESXi installer ISO image, downloaded from the VMware Web site.
- Target host with a hardware configuration that is supported for your version of ESXi. See the *VMware Compatibility Guide*.
- Network adapter with UEFI networking support on the target ESXi host.
- DHCP server that you can configure for UEFI HTTP booting. See Sample DHCP Configurations
- (Optional) Installation script (kickstart file).
- Use a native VLAN in most cases. If you want to specify the VLAN ID to be used with PXE booting, check that your NIC supports VLAN ID specification.

Procedure

1. Copy the `efi/boot/bootx64.efi` file from the ESXi installer ISO image to a directory on your HTTP server and rename the file to `mboot.efi`. For example, `http://www.example.com/esxi/mboot.efi`.

   Note Newer versions of `mboot.efi` can generally boot older versions of ESXi, but older versions of `mboot.efi` might be unable to boot newer versions of ESXi. If you plan to configure different hosts to boot different versions of the ESXi installer, use the `mboot.efi` from the newest version.

2. Configure the DHCP server.
3. Create a directory on your HTTP server with the same name as the version of ESXi it will hold. For example, `http://www.example.com/esxi/ESXi-8.x.x-XXXXXX`.
4. Copy the contents of the ESXi installer image to the newly created directory.
5. Modify the `boot.cfg` file.
   a. Add the following line with the URL of the newly created directory.

   ```
   prefix=http://www.example.com/esxi/ESXi-8.x.x-XXXXXX
   ```

   b. If the filenames in the `kernel=` and `modules=` lines begin with a forward slash (/) character, delete that character.

   c. If the `kernelopt=` line contains the string `cdromBoot`, remove the string only.
6. (Optional) For a scripted installation, in the `boot.cfg` file, add the `kernelopt` option to the line after the kernel command, to specify the location of the installation script.

   For example, `kernelopt=ks=http://www.example.com/esxi_ksFiles/ks.cfg`

7 (Optional) You can use the virtual machine configuration parameters *networkBootProtocol* and *networkBootUri* to specify from where a virtual machines can boot. The setting *networkBootProtocol* specifies the boot protocol, IPv4 or IPv6. For example, `networkBootProtocol = httpv4`. The setting *networkBootUri* specifies the HTTP URL to the ESXi bootloader (bootx64.efi). For example, `networkBootUri = http://xxx.xxx.xx.x/esxi80uc1/efi/boot/bootx64.efi`.

8 Specify whether you want all UEFI hosts to boot the same installer.

Option	Description
Same installer	Add the `boot.cfg` file to the same directory as `mboot.efi`. For example, `http://www.example.com/esxi/boot.cfg`
Different installers	a Create a subdirectory of the directory that contains the `mboot.efi` file. Name the directory as the MAC address of the target host machine (01-*mac_address_of_target_ESXi_host*), for example, `01-23-45-67-89-0a-bc`. b Add the custom `boot.cfg` file in the directory. For example, `http://www.example.com/esxi/01-23-45-67-89-0a-bc/boot.cfg`.

You can use both installer types. ESXi hosts without custom `boot.cfg` file on your HTTP server, boot from the default `boot.cfg` file.

Sample DHCP Configurations

To network boot the ESXi installer, the DHCP server must send the address of the TFTP or HTTP server and the filename of the initial boot loader to the ESXi host.

When the target machine first boots, it broadcasts a packet across the network requesting information to boot itself. The DHCP server responds. The DHCP server must be able to determine whether the target machine is allowed to boot and the location of the initial boot loader binary. For PXE boot, the location is a file on a TFTP server. For UEFI HTTP boot, the location is a URL.

Caution Do not set up a second DHCP server if your network already has one. If multiple DHCP servers respond to DHCP requests, machines can obtain incorrect or conflicting IP addresses, or can fail to receive the proper boot information. Talk to a network administrator before setting up a DHCP server. For support on configuring DHCP, contact your DHCP server vendor.

There are many DHCP servers that you can use. The following examples are for a ISC DHCP server. If you are using a version of DHCP for Microsoft Windows, see the DHCP server documentation to determine how to pass the `next-server` and `filename` arguments to the target machine.

Example of Booting Using PXE and TFTP with IPv4

This example shows how to configure an ISC DHCP server to PXE boot ESXi using a TFTP server at IPv4 address xxx.xxx.xxx.xxx.

```
#
# ISC DHCP server configuration file snippet.  This is not a complete
# configuration file; see the ISC server documentation for details on
# how to configure the DHCP server.
#
allow booting;
allow bootp;
option client-system-arch code 93 = unsigned integer 16;
class "pxeclients" {
   match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";
   next-server xxx.xxx.xxx.xxx;
   if option client-system-arch = 00:07 or option client-system-arch = 00:09 {
      filename = "mboot.efi";
   } else {
      filename = "pxelinux.0";
   }
}
```

When a machine attempts to PXE boot, the DHCP server provides an IP address and the location of the `pxelinux.0` or `mboot.efi` binary file on the TFTP server.

Example of Booting Using PXE and TFTP with IPv6

This example shows how to configure an ISC DHCPv6 server to PXE boot ESXi using a TFTP server at IPv6 address xxxx:xxxx:xxxx:xxxx::xxxx.

```
#
# ISC DHCPv6 server configuration file snippet.  This is not a complete
# configuration file; see the ISC server documentation for details on
# how to configure the DHCP server.
#
allow booting;
allow bootp;
option dhcp6.bootfile-url code 59 = string;
option dhcp6.bootfile-url "tftp://[xxxx:xxxx:xxxx:xxxx::xxxx]/mboot.efi";
```

When a machine attempts to PXE boot, the DHCP server provides an IP address and the location of the `mboot.efi` binary file on the TFTP server.

Example of Booting Using iPXE and HTTP with IPv4

This example shows how to configure an ISC DHCP server to boot ESXi by loading iPXE from a TFTP server at IPv4 address `xxx.xxx.xxx.xxx`.

```
#
# ISC DHCP server configuration file snippet.  This is not a complete
# configuration file; see the ISC server documentation for details on
# how to configure the DHCP server.
```

```
#
allow booting;
allow bootp;
option client-system-arch code 93 = unsigned integer 16;
class "pxeclients" {
   match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";
   next-server xxx.xxx.xxx.xxx;
   if option client-system-arch = 00:07 or option client-system-arch = 00:09 {
      if exists user-class and option user-class = "iPXE" {
         # Instruct iPXE to load mboot.efi as secondary bootloader
         filename = "mboot.efi";
      } else {
         # Load the snponly.efi configuration of iPXE as initial bootloader
         filename = "snponly.efi";
      }
   } else {
      if exists user-class and option user-class = "iPXE" {
         # Instruct iPXE to load pxelinux as secondary bootloader
         filename = "pxelinux.0";
      } else {
         # Load the undionly configuration of iPXE as initial bootloader
         filename = "undionly.kpxe";
   }
}
```

When a machine attempts to PXE boot, the DHCP server provides an IP address and the location of the `undionly.kpxe` or `snponly.efi` binary file on the TFTP server. In the legacy BIOS case, iPXE then asks the DHCP server for the next file to load, and the server returns `pxelinux.0` as the filename. In the UEFI case, iPXE then asks the DHCP server for the next file to load, and this time the server returns `mboot.efi` as the filename. In both cases, iPXE is resident and the system has HTTP capability. As a result the system can load additional files from an HTTP server.

Example of Booting Using iPXE and HTTP with IPv6

This example shows how to configure an ISC DHCPv6 server to boot ESXi by loading iPXE from a TFTP server at IPv6 address `xxxx:xxxx:xxxx:xxxx::xxxx`.

```
#
# ISC DHCPv6 server configuration file snippet.  This is not a complete
# configuration file; see the ISC server documentation for details on
# how to configure the DHCP server.
#
allow booting;
allow bootp;

option dhcp6.bootfile-url code 59 = string;
if exists user-class and option user-class = "iPXE" {
   # Instruct iPXE to load mboot.efi as secondary bootloader
   option dhcp6.bootfile-url "tftp://[xxxx:xxxx:xxxx:xxxx::xxxx]/mboot.efi";
} else {
   # Load the snponly.efi configuration of iPXE as initial bootloader
   option dhcp6.bootfile-url "tftp://[xxxx:xxxx:xxxx:xxxx::xxxx]/snponly.efi";
}
```

When a machine attempts to PXE boot, the DHCP server provides an IP address and the location of the `snponly.efi` (iPXE) binary file on the TFTP server. iPXE then asks the DHCP server for the next file to load, and this time the server returns `mboot.efi` as the filename. iPXE is resident and the system has HTTP capability. As a result the system can load additional files from an HTTP server.

Example of Booting Using UEFI HTTP with IPv4

This example shows how to configure an ISC DHCP server to boot ESXi by using native UEFI HTTP over IPv4 from Web server `www.example.com`.

```
#
# ISC DHCP server configuration file snippet.  This is not a complete
# configuration file; see the ISC server documentation for details on
# how to configure the DHCP server.
#
allow booting;
allow bootp;
option client-system-arch code 93 = unsigned integer 16;
class "httpclients" {
   match if substring(option vendor-class-identifier, 0, 10) = "HTTPClient";
   option vendor-class-identifier "HTTPClient";

   if option client-system-arch = 00:10 {
      # x86_64 UEFI HTTP client
      filename = http://www.example.com/esxi/mboot.efi;
   }
}
```

Example of Booting Using UEFI HTTP with IPv6

This example shows how to configure an ISC DHCPv6 server to boot ESXi by using native UEFI HTTP over IPv6 from Web server `www.example.com`.

```
#
# ISC DHCPv6 server configuration file snippet.  This is not a complete
# configuration file; see the ISC server documentation for details on
# how to configure the DHCP server.
#
allow booting;
allow bootp;

option dhcp6.bootfile-url code 59 = string;
option dhcp6.user-class code 15 = { integer 16, string };
option dhcp6.vendor-class code 16 = { integer 32, integer 16, string };

if option dhcp6.client-arch-type = 00:10 {
      # x86_64 HTTP clients
      option dhcp6.vendor-class 0 10 "HTTPClient";
      option dhcp6.bootfile-url "http://www.example.com/esxi/mboot.efi";
}
```

How to Upgrade Hosts by Using ESXCLI Commands

Upgrading Hosts by Using ESXCLI Commands

By using ESXCLI, you can upgrade an ESXi 6.7 host or ESXi 7.0 host to version 8.0 and update or patch ESXi 6.7, ESXi 7.0, and ESXi 8.0 hosts.

vSphere 8.0 introduces configuration files, components, base images, and add-ons as new software deliverables that you can use to update or patch ESXi 8.0 hosts. For information about managing components, base images, and add-ons on ESXi, see ESXCLI Concepts and Examples.

To use ESXCLI commands, you must install Standalone ESXCLI. For more information about installing and using ESXCLI, see the following documents.

- Using ESXCLI
- Getting Started with ESXCLI
- ESXCLI Reference

Note If you press Ctrl+C while an `esxcli` command is running, the command-line interface exits to a new prompt without displaying a message. However, the command continues to run to completion.

For ESXi hosts deployed with vSphere Auto Deploy, the tools VIB must be part of the base booting image used for the initial Auto Deploy installation. The tools VIB cannot be added separately later.

VIBs, Image Profiles, and Software Depots

Upgrading ESXi with esxcli commands requires an understanding of VIBs, image profiles, and software depots.

The following technical terms are used throughout the vSphere documentation set in discussions of installation and upgrade tasks.

VIB

A VIB is an ESXi software package. VMware and its partners package solutions, drivers, CIM providers, and applications that extend the ESXi platform as VIBs. VIBs are available in software depots. You can use VIBs to create and customize ISO images or to upgrade ESXi hosts by installing VIBs asynchronously onto the hosts.

Image Profile

An image profile defines an ESXi image and consists of VIBs. An image profile always includes a base VIB, and might include more VIBs. You examine and define an image profile by using vSphere ESXi Image Builder.

Software Depot

A software depot is a collection of VIBs and image profiles. The software depot is a hierarchy of files and folders and can be available through an HTTP URL (online depot) or a ZIP file (offline depot). VMware and VMware partners make depots available. Companies with large VMware installations might create internal depots to provision ESXi hosts with vSphere Auto Deploy, or to export an ISO for ESXi installation.

Understanding Acceptance Levels for VIBs and Hosts

Each VIB is released with an acceptance level that cannot be changed. The host acceptance level determines which VIBs can be installed to a host.

The acceptance level applies to individual VIBs installed by using the `esxcli software vib install` and `esxcli software vib update` commands, to VIBs installed using vSphere Lifecycle Manager, and to VIBs in image profiles.

The acceptance level of all VIBs on a host must be at least as high as the host acceptance level. For example, if the host acceptance level is VMwareAccepted, you can install VIBs with acceptance levels of VMwareCertified and VMwareAccepted, but you cannot install VIBs with acceptance levels of PartnerSupported or CommunitySupported. To install a VIB with a restrictive acceptance level that is less than the acceptance level of the host, you can change the setting of the host by using the vSphere Client or by running `esxcli software acceptance` commands.

Setting host acceptance levels is a best practice that allows you to specify which VIBs can be installed on a host and used with an image profile, and the level of support you can expect for a VIB. For example, you might set a more restrictive acceptance level for hosts in a production environment than for hosts in a testing environment.

VMware supports the following acceptance levels.

VMwareCertified

The VMwareCertified acceptance level has the most stringent requirements. VIBs with this level go through thorough testing fully equivalent to VMware in-house Quality Assurance testing for the same technology. Today, only I/O Vendor Program (IOVP) program drivers are published at this level. VMware takes support calls for VIBs with this acceptance level.

VMwareAccepted

VIBs with this acceptance level go through verification testing, but the tests do not fully test every function of the software. The partner runs the tests and VMware verifies the result. Today, CIM providers and PSA plug-ins are among the VIBs published at this level. VMware directs customers with support calls for VIBs with this acceptance level to contact the partner's support organization.

PartnerSupported

VIBs with the PartnerSupported acceptance level are published by a partner that VMware trusts. The partner performs all testing. VMware does not verify the results. This level is used for a new or nonmainstream technology that partners want to enable for VMware systems. Today, driver VIB technologies such as Infiniband, ATAoE, and SSD are at this level with

nonstandard hardware drivers. VMware directs customers with support calls for VIBs with this acceptance level to contact the partner's support organization.

CommunitySupported

The CommunitySupported acceptance level is for VIBs created by individuals or companies outside of VMware partner programs. VIBs at this level have not gone through any VMware-approved testing program and are not supported by VMware Technical Support or by a VMware partner.

Table 3-10. VIB Acceptance Levels Required to Install on Hosts

Host Acceptance Level	VMwareCertified VIB	VMwareAccepted VIB	PartnerSupported VIB	CommunitySupported VIB
VMwareCertified	x			
VMwareAccepted	x	x		
PartnerSupported	x	x	x	
CommunitySupported	x	x	x	x

Match a Host Acceptance Level with an Update Acceptance Level

You can change the host acceptance level to match the acceptance level for a VIB or image profile that you want to install. The acceptance level of all VIBs on a host must be at least as high as the host acceptance level.

Use this procedure to determine the acceptance levels of the host and the VIB or image profile to install, and to change the acceptance level of the host, if necessary for the update.

When you specify a target server by using `--server=<server_name>`, the server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see Getting Started with ESXCLI, or run `esxcli --help` at the ESXCLI command prompt.

Prerequisites

Install ESXCLI. See Getting Started with ESXCLI. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Retrieve the acceptance level for the VIB or image profile.

Option	Description
List information for all VIBs	`esxcli --server=<server_name> software sources vib list --depot=<depot_URL>`
List information for a specified VIB	`esxcli --server=<server_name> software sources vib list --viburl=<vib_URL>`
List information for all image profiles	`esxcli --server=<server_name> software sources profile list --depot=<depot_URL>`
List information for a specified image profile	`esxcli --server=<server_name> software sources profile get --depot=<depot_URL> --profile=<profile_name>`

2 Retrieve the host acceptance level.

```
esxcli --server=<server_name> software acceptance get
```

3 (Optional) If the acceptance level of the VIB is more restrictive than the acceptance level of the host, change the acceptance level of the host.

```
esxcli --server=<server_name> software acceptance set --level=<acceptance_level>
```

The *acceptance_level* can be `VMwareCertified`, `VMwareAccepted`, `PartnerSupported`, or `CommunitySupported`. The values for *acceptance_level* are case-sensitive.

Note You can use the `--force` option for the `esxcli software vib` or `esxcli software profile` command to add a VIB or image profile with a lower acceptance level than the host. A warning will appear. Because your setup is no longer consistent, the warning is repeated when you install VIBs, remove VIBs, and perform certain other operations on the host.

Determine Whether an Update Requires a Host to Be in Maintenance Mode or to Be Rebooted

VIBs that you can install with a live install do not require the host to be rebooted, but might require the host to be placed in maintenance mode. Other VIBs and profiles might require the host to be rebooted after the installation or update.

When you specify a target server by using `--server=<server_name>`, the server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see Getting Started with ESXCLI, or run `esxcli --help` at the ESXCLI command prompt.

Prerequisites

Install ESXCLI. See Getting Started with ESXCLI. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Check whether the VIB or image profile that you want to install requires the host to be placed in maintenance mode or to be rebooted after the installation or update.

Run one of the following commands.

Option	Description
Check the VIB	`esxcli --server=<server_name> software sources vib get -v <absolute_path_to_vib>`
Check the VIBs in a depot	`esxcli --server=<server_name> software sources vib get --depot=<depot_name>`
Check the image profile in a depot	`esxcli --server=<server_name> software sources profile get --depot=<depot_name>`

2 Review the return values.

The return values, which are read from the VIB metadata, indicate whether the host must be in maintenance mode before installing the VIB or image profile, and whether installing the VIB or profile requires the host to be rebooted.

Note vSphere Lifecycle Manager relies on an internal ESXi software scan API to determine whether maintenance mode is required or not. When you install a VIB on a live system, if the value for `Live-Install-Allowed` is set to false, the installation result instructs vSphere Lifecycle Manager to reboot the host. When you remove a VIB from a live system, if the value for `Live-Remove-Allowed` is set to false, the removal result instructs vSphere Lifecycle Manager to reboot the host. In either case, when the remediation starts, vSphere Lifecycle Manager automatically puts the host into maintenance mode.

What to do next

If necessary, place the host in maintenance mode. See Place a Host in Maintenance Mode. If a reboot is required, and if the host belongs to a vSphere HA cluster, remove the host from the cluster or deactivate HA on the cluster before the installation or update. Also, place the host in maintenance mode to minimize boot disk activity during the upgrade.

Place a Host in Maintenance Mode

Some installation and update operations that use live install require the host to be in maintenance mode.

Maintenance mode is required when an update operation requires a reboot. However, you only put the host in maintenance mode manually when you use esxcli commands for update and upgrade operations.

To determine whether an upgrade operation requires the host to be in maintenance mode, see Determine Whether an Update Requires a Host to Be in Maintenance Mode or to Be Rebooted

Note If the host is a member of a vSAN cluster, and any virtual machine object on the host uses the "Number of failures to tolerate=0" setting in its storage policy, the host might experience unusual delays when entering maintenance mode. The delay occurs because vSAN has to evacuate this object from the host for the maintenance operation to complete successfully.

When you specify a target server by using `--server=<server_name>`, the server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see Getting Started with ESXCLI, or run `esxcli --help` at the ESXCLI command prompt.

Prerequisites

Install ESXCLI. See Getting Started with ESXCLI. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Check whether the host is in maintenance mode.

```
esxcli --server=<server_name> system maintenanceMode get
```

2 Power off each virtual machine running on the ESXi host.

Note You can list all running virtual machines and retrieve the World ID of each one by running the following command.

```
esxcli --server=<server_name> vm process list
```

Option	Command
To shut down the guest operating system and then power off the virtual machine	`esxcli --server=<server_name> vm process kill --type soft --world-id <vm_ID>`
To power off the virtual machine immediately	`esxcli --server=<server_name> vm process kill --type hard --world-id <vm_ID>`
To force the power off operation	`esxcli --server=<server_name> vm process kill --type force --world-id <vm_ID>`

Alternatively, to avoid powering off virtual machines, you can migrate them to another host. See the topic Migrating Virtual Machines in the *vCenter Server and Host Management* documentation.

3 Place the host in maintenance mode.

```
esxcli --server=<server_name> system maintenanceMode set --enable true
```

4 Verify that the host is in maintenance mode.

```
esxcli --server=<server_name> system maintenanceMode get
```

Update a Host with Individual VIBs

You can update a host with VIBs stored in a software depot that is accessible through a URL or in an offline ZIP depot.

Important If you are updating ESXi from a zip bundle in a VMware-supplied depot, either online from the VMware Web site or downloaded locally, VMware supports only the update method specified for VMware-supplied depots in the topic Upgrade or Update a Host with Image Profiles .

Note The `esxcli software vib update` and `esxcli software vib install` commands are not supported for upgrade operations. See Upgrade or Update a Host with Image Profiles .

When you specify a target server by using `--server=<server_name>`, the server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see Getting Started with ESXCLI, or run `esxcli --help` at the ESXCLI command prompt.

Prerequisites

- Install ESXCLI. See Getting Started with ESXCLI. For troubleshooting, run `esxcli` commands in the ESXi Shell.
- Determine whether the update requires the host to be in maintenance mode or to be rebooted. If necessary, place the host in maintenance mode.

  See Determine Whether an Update Requires a Host to Be in Maintenance Mode or to Be Rebooted. See Place a Host in Maintenance Mode.
- If the update requires a reboot, and if the host belongs to a vSphere HA cluster, remove the host from the cluster or deactivate HA on the cluster.

Procedure

1 Determine which VIBs are installed on the host.

```
esxcli --server=<server_name> software vib list
```

2 Find out which VIBs are available in the depot.

Option	Description
From a depot accessible by URL	`esxcli --server=<server_name> software sources vib list --depot=http://<web_server>/<depot_name>`
From a local depot ZIP file	`esxcli --server=<server_name> software sources vib list --depot=<absolute_path_to_depot_zip_file>`

You can specify a proxy server by using the `--proxy` option.

3 Update the existing VIBs to include the VIBs in the depot or install new VIBs.

Option	Description
Update VIBs from a depot accessible by URL	`esxcli --server=<server_name> software vib update --depot=http://<web_server>/<depot_name>`
Update VIBs from a local depot ZIP file	`esxcli --server=<server_name> software vib update --depot=<absolute_path_to_depot_ZIP_file>`
Install all VIBs from a ZIP file on a specified offline depot (includes both VMware VIBs and partner-supplied VIBs)	`esxcli --server=<server_name> software vib install --depot <path_to_VMware_vib_ZIP_file>\<VMware_vib_ZIP_file> --depot <path_to_partner_vib_ZIP_file>\<partner_vib_ZIP_file>`

Options for the `update` and `install` commands allow you to perform a dry run, to specify a specific VIB, to bypass the acceptance level verification, and so on. Do not bypass verification on production systems. See the *ESXCLI Reference*.

4 Verify that the VIBs are installed on your ESXi host.

```
esxcli --server=<server_name> software vib list
```

Upgrade or Update a Host with Image Profiles

You can upgrade or update a host with image profiles stored in a software depot that is accessible through a URL or in an offline ZIP depot.

You can use the `esxcli software profile update` or `esxcli software profile install` command to upgrade or update an ESXi host.

When you upgrade or update a host, the **`esxcli software profile update`** or **`esxcli software profile install`** command applies a later version (major or minor) of a full image profile onto the host. After this operation and a reboot, the host can join a vCenter Server environment of the same or later version.

The `esxcli software profile update` command brings the entire content of the ESXi host image to the same level as the corresponding upgrade method using an ISO installer. However, the ISO installer performs a pre-upgrade check for potential problems, such as insufficient memory or unsupported devices. The **`esxcli`** upgrade method only performs such checks when upgrading from ESXi 6.7 Update 1 or later to a newer version.

Note Do not use the `--dry-run` option for upgrades from ESXi 6.7.x and ESXi 7.0.x versions earlier than 7.0 Update 3i, to ESXi 8.0 and later. When the `--dry-run` option is removed, you can still use the esxcli upgrade method to upgrade from ESXi 6.7 Update 1 or later to ESXi 8.0 or later. For ESXi versions earlier than 6.7 Update 1, you must first upgrade to 6.7 Update 1 or later before you upgrade to ESXi 8.0 or later.

For more about the ESXi upgrade process and methods, see Overview of the ESXi Host Upgrade Process.

Important If you are upgrading or updating ESXi from a zip bundle in a VMware-supplied depot, either online from the VMware website or downloaded locally, VMware supports only the update command `esxcli software profile update --depot=<depot_location> --profile=<profile_name>`.

When you specify a target server by using `--server=<server_name>`, the server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see Getting Started with ESXCLI, or run `esxcli --help` at the ESXCLI command prompt.

Note Options to the `update` and `install` commands allow you to perform a dry run, to bypass acceptance level verification, to ignore hardware compatibility check warnings, and so on. The option to bypass hardware compatibility check warnings is only available for ESXi 6.7 Update 1 or later. Do not bypass verification on production systems.

For options help, type `esxcli software profile install --help` or `esxcli software profile update --help`. For the complete listing of available command-line options, see the ESXCLI Reference.

Prerequisites

- Install Standalone ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.
- Determine whether the update requires the host to be in maintenance mode or to be rebooted. If necessary, place the host in maintenance mode.

See Determine Whether an Update Requires a Host to Be in Maintenance Mode or to Be Rebooted. See Place a Host in Maintenance Mode.

Important When you use ESXCLI to update or upgrade the host, put the host manually in maintenance mode to ensure that the boot disk is not actively in use before the upgrade begins.

- If the update requires a reboot, and if the host belongs to a vSphere HA cluster, remove the host from the cluster or deactivate HA on the cluster.

Procedure

1. Determine which VIBs are installed on the host.

```
esxcli --server=<server_name> software vib list
```

2. Determine which image profiles are available in the depot.

```
esxcli --server=<server_name> software sources profile list --depot=http://<web_server>/<depot_name>
```

You can specify a proxy server by using the `--proxy` option.

3. Update the existing image profile to include the VIBs or install new VIBs.

Important The `software profile update` command updates existing VIBs with the corresponding VIBs from the specified profile, but does not affect other VIBs installed on the target server. The `software profile install` command installs the VIBs present in the depot image profile, and removes any other VIBS installed on the target server.

Option	Description
Update the image profile from a VMware-supplied ZIP bundle, in a depot, accessible online from the VMware Web site or downloaded to a local depot	`esxcli software profile update --depot=<depot_location> --profile=<profile_name>` **Important** This is the only update method that VMware supports for zip bundles supplied by VMware. VMware-supplied ZIP bundle names take the form: `VMware-ESXi-<version_number>-<build_number>-depot.zip`. The profile name for VMware-supplied zip bundles takes one of the following forms. ■ `ESXi-<version_number>-<build_number>-standard` ■ `ESXi-<version_number>-<build_number>-notools` (does not include VMware Tools)
Update the image profile from a depot accessible by URL	`esxcli --server=<server_name> software profile update --depot=http://<web_server>/<depot_name> --profile=<profile_name>`

Option	Description
Update the image profile from ZIP file stored locally on the target server	`esxcli --server=<server_name> software profile update --depot=file:///<path_to_profile_ZIP_file>/<profile_ZIP_file> --profile=<profile_name>`
Update the image profile from a ZIP file on the target server, copied into a datastore	`esxcli --server=<server_name> software profile update --depot=<datastore_name>/<profile_ZIP_file> --profile=<profile_name>`
Update the image profile from a ZIP file copied locally and applied on the target server	`esxcli --server=<server_name> software profile update --depot=/<root_dir>/<path_to_profile_ZIP_file>/<profile_ZIP_file> --profile=<profile_name>`
Install all new VIBs in a specified profile accessible by URL	`esxcli --server=<server_name> software profile install --depot=http://<web_server>/<depot_name> --profile=<profile_name>`
Install all new VIBs in a specified profile from a ZIP file stored locally on the target.	`esxcli --server=<server_name> software profile install --depot=file:///<path_to_profile_ZIP_file>/<profile_ZIP_file> --profile=<profile_name>`
Install all new VIBs from a ZIP file on the target server, copied into a datastore	`esxcli --server=<server_name> software profile install --depot=<datastore_name>/<profile_ZIP_file> --profile=<profile_name>`
Install all new VIBs from a ZIP file copied locally and applied on the target server	`esxcli --server=<server_name> software profile install --depot=/<root_dir>/<path_to_profile_ZIP_file>/<profile_ZIP_file> --profile=<profile_name>`

4 Verify that the VIBs are installed on your ESXi host.

```
esxcli --server=<server_name> software vib list
```

Update ESXi Hosts by Using Zip Files

You can update hosts with VIBs or image profiles by downloading a ZIP file of a depot.

VMware partners prepare third-party VIBs to provide management agents or asynchronously released drivers.

Important If you are updating ESXi from a zip bundle in a VMware-supplied depot, either online from the VMware Web site or downloaded locally, VMware supports only the update method specified for VMware-supplied depots in the topic Upgrade or Update a Host with Image Profiles .

The `esxcli software vib update` and `esxcli software vib install` commands are not supported for upgrade operations. See Upgrade or Update a Host with Image Profiles .

When you specify a target server by using `--server=<server_name>`, the server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see Getting Started with ESXCLI, or run `esxcli --help` at the ESXCLI command prompt.

Prerequisites

- Install ESXCLI. See Getting Started with ESXCLI. For troubleshooting, run `esxcli` commands in the ESXi Shell.
- Download the ZIP file of a depot bundle from a third-party VMware partner.
- Determine whether the update requires the host to be in maintenance mode or to be rebooted. If necessary, place the host in maintenance mode.

  See Determine Whether an Update Requires a Host to Be in Maintenance Mode or to Be Rebooted. See Place a Host in Maintenance Mode.
- If the update requires a reboot, and if the host belongs to a vSphere HA cluster, remove the host from the cluster or deactivate HA on the cluster.

Procedure

- Install the ZIP file.

```
esxcli --server=<server_name> software vib update --depot=/<path_to_vib_ZIP>/
<ZIP_file_name>.zip
```

Remove VIBs from a Host

You can uninstall third-party VIBs or VMware VIBs from your ESXi host.

VMware partners prepare third-party VIBs to provide management agents or asynchronously released drivers.

Install ESXCLI. See Getting Started with ESXCLI. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Prerequisites

- If the removal requires a reboot, and if the host belongs to a vSphere HA cluster, deactivate HA for the host.
- Determine whether the update requires the host to be in maintenance mode or to be rebooted. If necessary, place the host in maintenance mode.

  See Determine Whether an Update Requires a Host to Be in Maintenance Mode or to Be Rebooted. See Place a Host in Maintenance Mode.

  Important To ensure that the boot disk is not actively in use when you use ESXCLI to update or upgrade the host, put the host manually in maintenance mode.

- Install ESXCLI. See Getting Started with ESXCLI. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Power off each virtual machine running on the ESXi host.

Note You can list all running virtual machines and retrieve the World ID of each one by running the following command.

```
esxcli --server=<server_name> vm process list
```

Option	Command
To shut down the guest operating system and then power off the virtual machine	`esxcli --server=<server_name> vm process kill --type soft --world-id <vm_ID>`
To power off the virtual machine immediately	`esxcli --server=<server_name> vm process kill --type hard --world-id <vm_ID>`
To force the power off operation	`esxcli --server=<server_name> vm process kill --type force --world-id <vm_ID>`

Alternatively, to avoid powering off virtual machines, you can migrate them to another host. See the topic Migrating Virtual Machines in the *vCenter Server and Host Management* documentation.

2 Place the host in maintenance mode.

```
esxcli --server=<server_name> system maintenanceMode set --enable true
```

3 If necessary, shut down or migrate virtual machines.

4 Determine which VIBs are installed on the host.

```
esxcli --server=<server_name> software vib list
```

5 Remove the VIB.

```
esxcli --server=<server_name> software vib remove --vibname=<name>
```

Specify one or more VIBs to remove in one of the following forms.

- *<name>*
- *<name>*:*<version>*
- *<vendor>*:*<name>*
- *<vendor>*:*<name>*:*<version>*

For example, the command to remove a VIB specified by vendor, name and version can take the following form.

```
esxcli --server myEsxiHost software vib remove --vibname=PatchVendor:patch42:version3
```

Note The `remove` command supports several more options. See the *ESXCLI Reference*.

Adding Third-Party Extensions to Hosts with an ESXCLI Command

You can use the `esxcli software vib` command to add a third-party extension released as a VIB package to the system. When you use this command, the VIB system updates the firewall rule set and refreshes the host daemon after you reboot the system.

Otherwise, you can use a firewall configuration file to specify port rules for host services to enable for the extension. The *vSphere Security* documentation discusses how to add, apply, and refresh a firewall rule set and lists the `esxcli network firewall` commands.

Perform a Dry Run of an ESXCLI Installation or Upgrade

You can use the `--dry-run` option to preview the results of an installation or upgrade operation. A dry run of the installation or update procedure does not make any changes, but reports the VIB-level operations that will be performed if you run the command without the `--dry-run` option.

When you specify a target server by using `--server=<server_name>`, the server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see Getting Started with ESXCLI, or run `esxcli --help` at the ESXCLI command prompt.

Note Do not use the `--dry-run` option for upgrades from ESXi 6.7.x and ESXi 7.0.x versions earlier than 7.0 Update 3i, to ESXi 8.0 and later. When the `--dry-run` option is removed, you can still use the esxcli upgrade method to upgrade from ESXi 6.7 Update 1 or later to ESXi 8.0 or later. For ESXi versions earlier than 6.7 Update 1, you must first upgrade to 6.7 Update 1 or later before you upgrade to ESXi 8.0 or later.

Prerequisites

Install ESXCLI. See Getting Started with ESXCLI. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Enter the installation or upgrade command, adding the `--dry-run` option.

- `esxcli --server=<server_name> software vib install --dry-run`
- `esxcli --server=<server_name> software vib update --dry-run`

- `esxcli --server=<server_name> software profile install --dry-run`
- `esxcli --server=<server_name> software profile update --dry-run`

2 Review the output that is returned.

The output shows which VIBs will be installed or removed and whether the installation or update requires a reboot.

Display the Installed VIBs and Profiles That Will Be Active After the Next Host Reboot

You can use the `--rebooting-image` option to list the VIBs and profiles that are installed on the host and will be active after the next host reboot.

When you specify a target server by using `--server=<server_name>`, the server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see Getting Started with ESXCLI, or run `esxcli --help` at the ESXCLI command prompt.

Prerequisites

Install ESXCLI. See Getting Started with ESXCLI. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Enter one of the following commands.

Option	Description
For VIBs	`esxcli --server=<server_name> software vib list --rebooting-image`
For Profiles	`esxcli --server=<server_name> software profile get --rebooting-image`

2 Review the output that is returned.

The output displays information for the ESXi image that will become active after the next reboot. If the pending-reboot image has not been created, the output returns nothing.

Display the Image Profile and Acceptance Level of the Host

You can use the `software profile get` command to display the currently installed image profile and acceptance level for the specified host.

This command also shows details of the installed image profile history, including profile modifications.

When you specify a target server by using `--server=<server_name>`, the server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see Getting Started with ESXCLI, or run `esxcli --help` at the ESXCLI command prompt.

Prerequisites

Install ESXCLI. See Getting Started with ESXCLI. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 Enter the following command.

```
esxcli --server=<server_name> software profile get
```

2 Review the output.

After You Upgrade ESXi Hosts

To complete a host upgrade, you ensure that the host is reconnected to its managing vCenter Server system and reconfigured if necessary. You also check that the host is licensed correctly.

After you upgrade an ESXi host, take the following actions:

- View the upgrade logs. You can use the vSphere Client to export the log files.
- If a vCenter Server system manages the host, you must reconnect the host to vCenter Server by right-clicking the host in the vCenter Server inventory and selecting **Connect**.
- When the upgrade is complete, the ESXi host is in evaluation mode. The evaluation period is 60 days. You must assign a vSphere 8.0 license before the evaluation period expires. You can upgrade existing licenses or acquire new ones from My VMware. Use the vSphere Client to configure the licensing for the hosts in your environment. See the *vCenter Server and Host Management* documentation for details about managing licenses in vSphere.
- The host sdX devices might be renumbered after the upgrade. If necessary, update any scripts that reference sdX devices.
- Upgrade virtual machines on the host. See Upgrading Virtual Machines and VMware Tools.
- Set up the vSphere Authentication Proxy service. Earlier versions of the vSphere Authentication Proxy are not compatible with vSphere 8.0. See the *vSphere Security* documentation for details about configuring the vSphere Authentication Proxy service.

About ESXi Evaluation and Licensed Modes

You can use evaluation mode to explore the entire set of features for ESXi hosts. The evaluation mode provides the set of features equal to a vSphere Enterprise Plus license. Before the evaluation mode expires, you must assign to your hosts a license that supports all the features in use.

For example, in evaluation mode, you can use vSphere vMotion technology, the vSphere HA feature, the vSphere DRS feature, and other features. If you want to continue using these features, you must assign a license that supports them.

The installable version of ESXi hosts is always installed in evaluation mode. ESXi Embedded is preinstalled on an internal storage device by your hardware vendor. It might be in evaluation mode or prelicensed.

The evaluation period is 60 days and begins when you turn on the ESXi host. At any time during the 60-day evaluation period, you can convert from licensed mode to evaluation mode. The time available in the evaluation period is decreased by the time already used.

For example, suppose that you use an ESXi host in evaluation mode for 20 days and then assign a vSphere Standard Edition license key to the host. If you set the host back in evaluation mode, you can explore the entire set of features for the host for the remaining evaluation period of 40 days.

For ESXi hosts, license or evaluation period expiry leads to disconnection from vCenter Server. All powered on virtual machines continue to work, but you cannot power on virtual machines after they are powered off. You cannot change the current configuration of the features that are in use. You cannot use the features that remained unused before the license expiration.

For information about managing licensing for ESXi hosts, see the *vCenter Server and Host Management* documentation.

Licensing ESXi Hosts After Upgrade

After you upgrade to ESXi 8.0, you must apply a vSphere 8 license.

If you upgrade an ESXi host to a version that starts with the same number, you do not need to replace the existing license with a new one. For example, if you upgrade a host from ESXi 6.5 to 6.7, you can use the same license for the host.

If you upgrade an ESXi host to a version that starts with a different number, you must apply a new license. For example, if you upgrade an ESXi host from 7.x to 8.0, you need to license the host with a vSphere 8 license.

When you upgrade ESXi 6.7 or ESXi 7.0 hosts to ESXi 8.0 hosts, the hosts are in a 60-day evaluation mode period until you apply the correct vSphere 8 licenses. See About ESXi Evaluation and Licensed Modes.

You can acquire vSphere 8 licenses from My VMware. After you have vSphere 8 licenses, you must assign them to all upgraded ESXi 8.0 hosts by using the license management functionality in the vSphere Client. See the *vCenter Server and Host Management* documentation for details. If you use the scripted method to upgrade to ESXi 8.0, you can provide the license key in the kickstart (ks) file.

Run the Secure Boot Validation Script on an Upgraded ESXi Host

After you upgrade an ESXi host from an older version of ESXi that did not support UEFI secure boot, you might be able to activate secure boot. Whether you can activate secure boot depends on how you performed the upgrade and whether the upgrade replaced all the existing VIBs or left

some VIBs unchanged. You can run a validation script after you perform the upgrade to determine whether the upgraded installation supports secure boot.

For secure boot to succeed, the signature of every installed VIB must be available on the system. Older versions of ESXi do not save the signatures when installing VIBs.

- If you upgrade using ESXCLI commands, the old version of ESXi performs the installation of the new VIBs, so their signatures are not saved and secure boot is not possible.
- If you upgrade using the ISO, new VIBs do have their signatures saved. This is true also for vSphere Lifecycle Manager upgrades that use the ISO.
- If old VIBs remain on the system, the signatures of those VIBs are not available and secure boot is not possible.
  - If the system uses a third-party driver, and the VMware upgrade does not include a new version of the driver VIB, then the old VIB remains on the system after upgrade.
  - In rare cases, VMware might drop ongoing development of a specific VIB without providing a new VIB that replaces or obsoletes it, so the old VIB remains on the system after upgrade.

Note UEFI secure boot also requires an up-to-date bootloader. This script does not check for an up-to-date bootloader.

Prerequisites

- Verify that the hardware supports UEFI secure boot.
- Verify that all VIBs are signed with an acceptance level of at least PartnerSupported. If you include VIBs at the CommunitySupported level, you cannot use secure boot.

Procedure

1. Upgrade the ESXi and run the following command.

   ```
   /usr/lib/vmware/secureboot/bin/secureBoot.py -c
   ```

2. Check the output.

   The output either includes `Secure boot can be enabled` or `Secure boot CANNOT be enabled`.

Required Free Space for System Logging

If you used Auto Deploy to install your ESXi 8.0 host, or if you set up a log directory separate from the default location in a scratch directory on the VMFS volume, you might need to change your current log size and rotation settings to ensure that enough space is available for system logging .

All vSphere components use this infrastructure. The default values for log capacity in this infrastructure vary, depending on the amount of storage available and on how you have configured system logging. Hosts that are deployed with Auto Deploy store logs on a RAM disk, which means that the amount of space available for logs is small.

If your host is deployed with Auto Deploy, reconfigure your log storage in one of the following ways:

- Redirect logs over the network to a remote collector.
- Redirect logs to a NAS or NFS store.

If you redirect logs to non-default storage, such as a NAS or NFS store, you might also want to reconfigure log sizing and rotations for hosts that are installed to disk.

You do not need to reconfigure log storage for ESXi hosts that use the default configuration, which stores logs in a scratch directory on the VMFS volume. For these hosts, ESXi 8.0 configures logs to best suit your installation, and provides enough space to accommodate log messages.

Table 3-11. Recommended Minimum Size and Rotation Configuration for hostd, vpxa, and fdm Logs

Log	Maximum Log File Size	Number of Log Files to Preserve	Minimum Disk Space Required
Management Agent (hostd)	10 MB	10	100 MB
VirtualCenter Agent (vpxa)	5 MB	10	50 MB
vSphere HA agent (Fault Domain Manager, fdm)	5 MB	10	50 MB

For information about setting up and configuring syslog and a syslog server and installing vSphere Syslog Collector, see the *vCenter Server Installation and Setup* documentation.

Configure Syslog on ESXi Hosts

You can use the vSphere Client, the VMware Host Client, or the `esxcli system syslog` command to configure the syslog service.

For information about using the `esxcli system syslog` command and other ESXCLI commands, see *Getting Started with ESXCLI*. For details how to open the ESXi firewall for the port specified in each remote host specification, see #unique_75 .

Procedure

1. Browse to the host in the vSphere Client inventory.
2. Click **Configure**.
3. Under **System**, click **Advanced System Settings**.
4. Click **Edit**.
5. Filter for **`syslog`**.
6. To set up logging globally and configure various advanced settings, see ESXi Syslog Options.

7 (Optional) To overwrite the default log size and log rotation for any of the logs:

   a Click the name of the log that you want to customize.

   b Enter the number of rotations and the log size you want.

8 Click **OK**.

Results

Changes to the syslog options take effect.

Note Syslog parameter settings that you define by using the vSphere Client or VMware Host Client are effective immediately. However, most settings you define by using ESXCLI require an additional command to take effect. For more details, see ESXi Syslog Options.

ESXi Syslog Options

You can define the behavior of ESXi syslog files and transmissions by using a set of syslog options.

Apart from the base settings, such as `Syslog.global.logHost`, starting from ESXi 7.0 Update 1, a list of advanced options is available for customizations, and NIAP compliance.

Note All audit record settings, beginning with `Syslog.global.auditRecord`, take effect immediately. However, for other settings that you define by using ESXCLI, make sure to run the `esxcli system syslog reload` command to enable the changes.

Table 3-12. Legacy Syslog Options

Option	ESXCLI command	Description
`Syslog.global.logHost`	`esxcli system syslog config set --loghost=<str>`	Defines a comma-delimited list of remote hosts and specifications for message transmissions. If the `loghost=<str>` field is blank, no logs are forwarded. While no hard limit to the number of remote hosts to receive syslog messages exists, good practice is to keep the number of remote hosts to five or less. The format of a remote host specification is: `protocol://hostname|ipv4|'['ipv6']'[:port]`. The protocol must be one of TCP, UDP, or SSL. The value of a port can be any decimal number from 1 through 65535. If a port is not provided, SSL and TCP use 1514. UDP uses 514. For example: `ssl://hostName1:1514`.
`Syslog.global.defaultRotate`	`esxcli system syslog config set --default-rotate=<long>`	Maximum number of old log files to keep. You can set this number globally and for individual subloggers (see `Syslog.global.defaultSize`).
`Syslog.global.defaultSize`	`esxcli system syslog config set --default-size=<long>`	Default size of log files, in KiB. After a file reaches the default size, the syslog service creates a new file. You can set this number globally and for individual subloggers.
`Syslog.global.logDir`	`esxcli system syslog config set --logdir=<str>`	Directory where logs reside. The directory can be on mounted NFS or VMFS volumes. Only the `/scratch` directory on the local file system is persistent across reboots. Specify the directory as [*datastorename*] *path_ to_ file*, where the path is relative to the root of the volume backing the datastore. For example, the path `[storage1] /systemlogs` maps to the path `/vmfs/volumes/storage1/systemlogs`.

Table 3-12. Legacy Syslog Options (continued)

Option	ESXCLI command	Description
Syslog.global.logDirUnique	esxcli system syslog config set --logdir-unique=<bool>	Specifies the ESXi host name to be concatenated to the value of Syslog.global.logDir. It is critical that you enable this setting when multiple ESXi hosts log to a shared file system. Selecting this option creates a subdirectory with the name of the ESXi host under the directory specified by **Syslog.global.LogDir**. A unique directory is useful if the same NFS directory is used by multiple ESXi hosts.
Syslog.global.certificate.checkSSLCerts	esxcli system syslog config set --check-ssl-certs=<bool>	Enforces checking of SSL certificates when transmitting messages to remote hosts.

Table 3-13. Syslog Options Available Starting from ESXi 7.0 Update 1

Option	ESXCLI command	Description
Syslog.global.auditRecord.storageCapacity	esxcli system auditrecords local set --size=<long>	Specifies the capacity of the audit record storage directory located on the ESXi host, in MiB. You cannot decrease the capacity of the audit record storage. You can increase the capacity before or after the audit record storage is enabled (see Syslog.global.auditRecord.storageEnable).
Syslog.global.auditRecord.remoteEnable	esxcli system auditrecords remote enable	Enables sending audit records to remote hosts. Remote hosts are specified by using the Syslog.global.logHost parameter.
Syslog.global.auditRecord.storageDirectory	esxcli system auditrecords local set --directory=<dir>	Specifies the location of the audit record storage directory. You cannot change the audit record storage directory while audit record storage is enabled (see Syslog.global.auditRecord.storageEnable).
Syslog.global.auditRecord.storageEnable	esxcli system auditrecords local enable	Enables the storage of audit records on an ESXi host. If the audit record storage directory does not exist, it is created with the capacity specified by Syslog.global.auditRecord.storageCapacity.

Table 3-13. Syslog Options Available Starting from ESXi 7.0 Update 1 (continued)

Option	ESXCLI command	Description
`Syslog.global.certificate.checkCRL`	`esxcli system syslog config set --crl-check=<bool>`	Enables checking the revocation status of all the certificates in an SSL certificate chain. Enables verification of X.509 CRLs, which are not checked by default in compliance with industry conventions. A NIAP-validated configuration requires CRL checks. Due to implementation limitations, if CRL checks are enabled, then all certificates in a certificate chain must provide a CRL link. Do not enable the `crl-check` option for installations not related to certification, because of the difficulty in properly configuring an environment that uses CRL checks.
`Syslog.global.certificate.strictX509Compliance`	`esxcli system syslog config set --x509-strict=<bool>`	Enables strict compliance with X.509. Performs additional validity checks on CA root certificates during verification. These checks are generally not performed, as CA roots are inherently trusted, and might cause incompatibilities with existing, misconfigured CA roots. A NIAP-validated configuration requires even CA roots to pass validations. Do not enable the `x509-strict` option for installations not related to certification, because of the difficulty in properly configuring an environment that uses CRL checks.
`Syslog.global.droppedMsgs.fileRotate`	`esxcli system syslog config set --drop-log-rotate=<long>`	Specifies the number of old dropped message log files to keep.
`Syslog.global.droppedMsgs.fileSize`	`esxcli system syslog config set --drop-log-size=<long>`	Specifies the size of each dropped message log file before switching to a new one, in KiB.
`Syslog.global.logCheckSSLCerts`	`esxcli system syslog config set --check-ssl-certs=<bool>`	Enforces checking of SSL certificates when transmitting messages to remote hosts. **Note** Deprecated. Use `Syslog.global.certificate.checkSSLCerts` in ESXi 7.0 Update 1 and later.

Table 3-13. Syslog Options Available Starting from ESXi 7.0 Update 1 (continued)

Option	ESXCLI command	Description
`Syslog.global.logFilters`	`esxcli system syslog logfile [add | remove | set] ...`	Specifies one or more log filtering specifications. Each log filter must be separated by a double vertical bar "||". The format of a log filter is: `numLogs | ident | logRegexp`. `numLogs` sets the maximum number of log entries for the specified log messages. After reaching this number, the specified log messages are filtered and ignored. `ident` specifies one or more system components to apply the filter to the log messages that these components generate. `logRegexp` specifies a case-sensitive phrase with Python regular expression syntax to filter the log messages by their content.
`Syslog.global.logFiltersEnable`		Enables the use of log filters.
`Syslog.global.logLevel`	`esxcli system syslog config set --log-level=<str>`	Specifies the log filtering level. You must change this parameter only when troubleshooting an issue with the syslog daemon. You can use the values `debug` for the most detailed level, `info` for the default detail level, `warning` for only warnings or errors, or `error`, only for errors.
`Syslog.global.msgQueueDropMark`	`esxcli system syslog config --queue-drop-mark=<long>)`	Specifies the percent of the message queue capacity at which messages are dropped.
`Syslog.global.remoteHost.connectRetryDelay`	`esxcli system syslog config set --default-timeout=<long>`	Specifies the delay before retrying to connect to a remote host after a connection attempt fails, in seconds.

Table 3-13. Syslog Options Available Starting from ESXi 7.0 Update 1 (continued)

Option	ESXCLI command	Description
`Syslog.global.remoteHost.maxMsgLen`	`esxcli system syslog config set --remote-host-max-msg-len=<long>`	For the TCP and SSL protocols, this parameter specifies the maximum length of a syslog transmission before truncation occurs, in bytes. The default maximum length for remote host messages is 1 KiB. You can increase the maximum message length to up to 16 KiB. However, raising this value above 1 KiB does not ensure that long transmissions arrive untruncated to a syslog collector. For example, when the syslog infrastructure that issues a message is external to ESXi. RFC 5426 sets the maximum message transmission length for the UDP protocol to 480 bytes for IPV4 and 1180 bytes for IPV6.
`Syslog.global.vsanBacking`	`esxcli system syslog config set --vsan-backing=<bool>`	Allows log files and the audit record storage directory to be placed on a vSAN cluster. However, enabling this parameter might cause the ESXi host to become unresponsive.

Configure Log Filtering on ESXi Hosts

The log filtering capability lets you modify the logging policy of the syslog service that is running on an ESXi host. You can create log filters to reduce the number of repetitive entries in the ESXi logs and to denylist specific log events entirely.

Starting with vSphere 7.0 Update 2, you to add logfilters and enable logfitering by using ESXCLI.

Log filters affect all log events that are processed by the ESXi host vmsyslogd service, whether they are recorded to a log directory or to a remote syslog server.

When you create a log filter, you set a maximum number of log entries for the log messages. Log messages are generated by one or more specified system components that match a specified phrase. You must enable the log filtering capability and reload the syslog daemon to activate the log filters on the ESXi host.

Important Setting a limit to the amount of logging information restricts your ability to troubleshoot potential system failures properly. If a log rotate occurs after the maximum number of log entries is reached, you might lose all instances of a filtered message.

Prerequisites

Install ESXCLI. See *Getting Started with ESXCLI*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

1 To get to the ESXCLI system syslog config logfilter, run a command such as:

```
[root@xxx-xx-dhcp-xx-xx:~] esxcli system syslog config logfilter
```

ESXCLI commands to configure logfilters follow this pattern: `esxcli system syslog config logfilter {cmd} [cmd options]`

2 To get to the ESXCLI system syslog config logfilter, run a command such as:

```
[root@xxx-xx-dhcp-xx-xx:~] esxcli system syslog config logfilter
```

Using vSphere Auto Deploy to Reprovision Hosts

4

If a host was deployed using vSphere Auto Deploy, you can use vSphere Auto Deploy to reprovision the host with a new image profile that contains a different version of ESXi. You can use vSphere ESXi Image Builder to create and manage image profiles.

Note If you upgrade the host to use an ESXi 6.0 or later image, the vSphere Auto Deploy server provisions the ESXi host with certificates that are signed by VMCA. If you are currently using custom certificates, you can set up the host to use the custom certificates after the upgrade. See *vSphere Security*.

The vSphere Auto Deploy server is automatically upgraded if you upgrade the corresponding vCenter Server system. Starting with version 6.0, the vSphere Auto Deploy server is always on the same management node as the vCenter Server system.

This chapter includes the following topics:

Introduction to vSphere Auto Deploy

When you start a physical host that is set up for vSphere Auto Deploy, vSphere Auto Deploy uses PXE boot infrastructure in conjunction with vSphere host profiles, a desired image, or configuration on a cluster level to provision and customize that host. No state is stored on the host itself. Instead, the vSphere Auto Deploy server manages state information for each host.

State Information for ESXi Hosts

vSphere Auto Deploy stores the information for the ESXi hosts to be provisioned in different locations. Information about the location of image profiles, host profiles, or clusters that you manage by either a single image or by a configuration on a cluster level is initially specified in the rules that map machines to image profiles and host profiles.

Table 4-1. vSphere Auto Deploy Stores Information for Deployment

Information Type	Description	Source of Information
Image state	The executable software to run on an ESXi host.	Image profile, created with vSphere ESXi Image Builder or a vSphere Lifecycle Manager image.
Configuration state	The configurable settings that determine how the host is configured, for example, virtual switches and their settings, driver settings, boot parameters, and so on.	Host profile, created by using the host profile UI, or a configuration that you create when setting up a cluster that manages all ESXi host settings at a cluster level in the Inventory UI.
Dynamic state	The runtime state that is generated by the running software, for example, generated private keys or runtime databases.	Host memory, lost during reboot.
Virtual machine state	The virtual machines stored on a host and virtual machine autostart information (subsequent boots only).	Virtual machine information sent by vCenter Server to vSphere Auto Deploy must be available to supply virtual machine information to vSphere Auto Deploy.
User input	State that is based on user input, for example, an IP address that the user provides when the system starts up, cannot automatically be included in the host profile.	Host customization information, stored by vCenter Server during first boot. You can create a host profile that requires user input for certain values. When vSphere Auto Deploy applies a host profile that requires user provided information, the host is placed in maintenance mode. Use the host profile UI to check the host profile compliance, and respond to the prompt to customize the host.

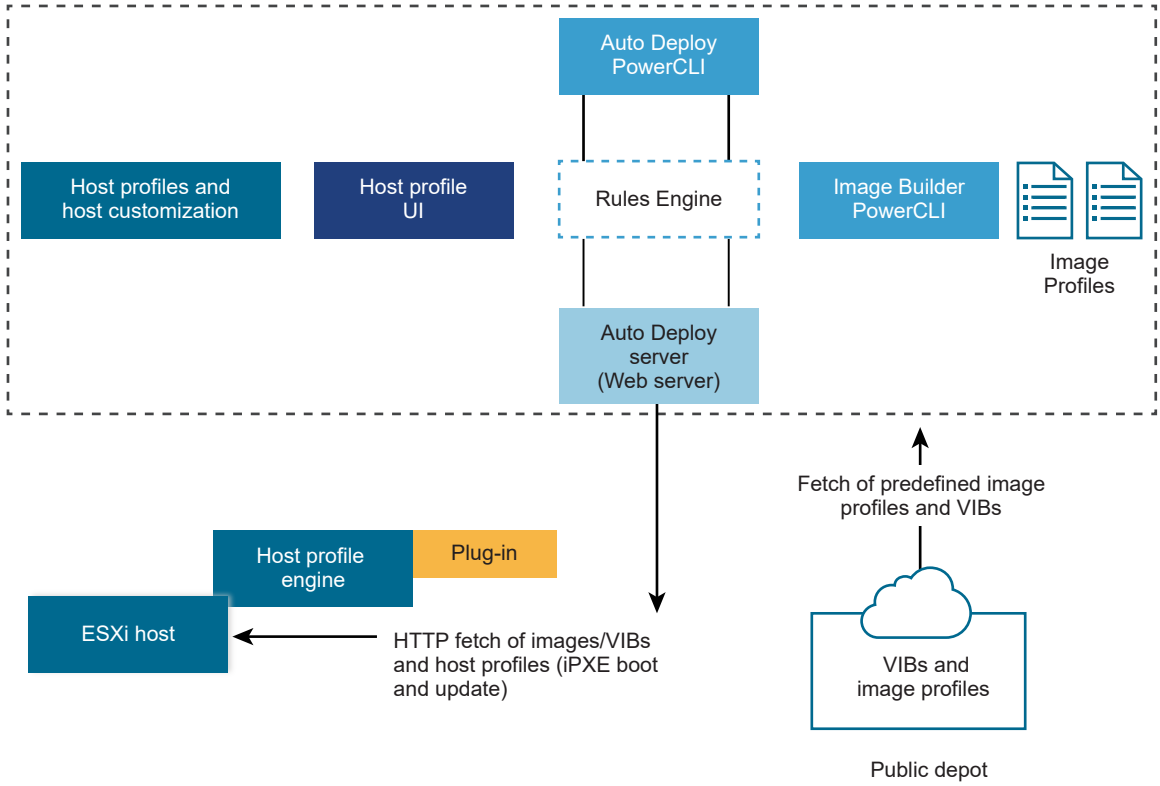
vSphere Auto Deploy Architecture

The vSphere Auto Deploy infrastructure consists of several components.

For more information, watch the video "Auto Deploy Architecture":

 (Auto Deploy Architecture)

Figure 4-1. vSphere Auto Deploy Architecture



vSphere Auto Deploy server

Serves images and host profiles to ESXi hosts.

vSphere Auto Deploy rules engine

Sends information to the vSphere Auto Deploy server which image profile and which host profile to serve to which host. Administrators use vSphere Auto Deploy to define the rules that assign image profiles and host profiles to hosts.

Apart from legacy image profiles that you create by using the VMware Image Builder and host profiles, you can also create vSphere Auto Deploy rules to deploy ESXi by using a single vSphere Lifecycle Manager image or a configuration on a cluster level.

Image profiles

Define the set of VIBs to boot ESXi hosts with.

- VMware and VMware partners make image profiles and VIBs available in public depots. Use vSphere ESXi Image Builder to examine the depot and use the vSphere Auto Deploy rules engine to specify which image profile to assign to which host.
- You use vSphere Lifecycle Manager images to apply software and firmware updates to the ESXi hosts in a cluster. Using a single image to manage all hosts in a cluster ensures cluster-wide host image homogeneity.

- With ESXi 8.0, you can set up a cluster that manages all ESXi host settings at a cluster level.
- VMware customers can create a custom image profile based on the public image profiles and VIBs in the depot and apply that image profile to the host.

Host profiles

Define machine-specific configuration such as networking or storage setup. Use the host profile UI to create host profiles. You can create a host profile for a reference host and apply that host profile to other hosts in your environment for a consistent configuration.

Note With ESXi 8.0, if you set up a cluster that manages all ESXi host settings at a cluster level, you cannot use host profiles.

Host customization

Stores information that the user provides when host profiles are applied to the host. Host customization might contain an IP address or other information that the user supplied for that host. For more information about host customizations, see the *vSphere Host Profiles* documentation.

Host customization was called answer file in earlier releases of vSphere Auto Deploy.

Auto Deploy Certificates

By default, the Auto Deploy server provisions each host with certificates that are signed by the VMware Certificate Authority (VMware CA). For more information, see Managing Certificates for ESXi Hosts.

Alternatively, if your corporate policy requires that you use custom certificates, you can set up the Auto Deploy server to provision all hosts with custom certificates that are not signed by VMware CA. The Auto Deploy server becomes a subordinate certificate authority of your third-party CA. In the Custom Certificate Authority mode, you are responsible for managing the certificates. You cannot refresh and renew certificates from the vSphere Client. In this mode, you also cannot select only a set of hosts to provision with custom certificates, and you can manually sign custom certificates only for stateful hosts. For more information, see Use Custom Certificates with Auto Deploy.

With ESXi 8.0, Auto Deploy provides a third option that allows you to generate a certificate outside vSphere and become independent of the certificate management in vCenter Server. For example, you can generate a custom certificate by using a custom script or by using a provider of domain name registry services such as Verisign. You can use custom certificates for only a set of ESXi hosts. You can provide custom certificates for stateless hosts as well. ESXi hosts are identified by the MAC address of the NIC used for network booting, or the BIOS UUID of the ESXi host. You update the VMware Endpoint Certificate Store (VECS) with the custom certificate by using PowerCLI. For more information on the new PowerCLI cmdlets, see vSphere Auto Deploy PowerCLI Cmdlet Overview. The VMware CA must trust the custom ESXi certificates so you must add the CA public certificate for the custom certificates to the TRUSTED_ROOTS store in VECS.

Auto Deploy also stores the custom certificates and when it recognizes a booting host with the respective MAC address of the NIC used for network booting, or the BIOS UUID of the ESXi host, it automatically provides the custom certificate. You do not need to stop or restart Auto Deploy or vCenter Server when you add a custom certificate to VECS, only restart the host for which you upload a custom certificate. For more information, see Use Custom Certificates with Auto Deploy.

Install and Configure vSphere Auto Deploy

Before you can start using vSphere Auto Deploy, you must prepare your environment. You start with server setup and hardware preparation. You must configure the vSphere Auto Deploy service startup type in the vCenter Server system that you plan to use for managing the hosts you provision, and install vSphere PowerCLI.

- vSphere Auto Deploy Preinstallation Checklist

  Before you can start the tasks in this vSphere Auto Deploy scenario, make sure that your environment meets the hardware and software requirements, and that you have the necessary permissions for the components included in the setup.

- Prepare Your System for vSphere Auto Deploy

  Before you can PXE boot an ESXi host with vSphere Auto Deploy, you must install prerequisite software and set up the DHCP and TFTP servers that vSphere Auto Deploy interacts with.

- Using vSphere Auto Deploy Cmdlets

  vSphere Auto Deploy cmdlets are implemented as Microsoft PowerShell cmdlets and included in vSphere PowerCLI. Users of vSphere Auto Deploy cmdlets can take advantage of all vSphere PowerCLI features.

- Set Up Bulk Licensing

  You can use the vSphere Client or ESXi Shell to specify individual license keys, or you can set up bulk licensing by using vSphere PowerCLI cmdlets. Bulk licensing works for all ESXi hosts, but is especially useful for hosts provisioned with vSphere Auto Deploy.

vSphere Auto Deploy Preinstallation Checklist

Before you can start the tasks in this vSphere Auto Deploy scenario, make sure that your environment meets the hardware and software requirements, and that you have the necessary permissions for the components included in the setup.

Table 4-2. Preinstallation Checklist

Required Software and Hardware	Details
vCenter Server	The vSphere Auto Deploy server is part of vCenter Server. You must enable and start the vSphere Auto Deploy service on the vCenter Server system. You can perform many of the setup tasks by logging in to vCenter Server. See Prepare Your System for vSphere Auto Deploy.
Storage	Storage for ESXi datastores NFS, iSCSI, or Fibre Channel, with servers and storage arrays that are configured so the servers can detect the LUNs. ■ A list of target IP addresses for NFS or iSCSI. ■ A list of target volume information for NFS or iSCSI.
Host information (for four ESXi hosts)	A list of target IP addresses for NFS or iSCSI. A list of target volume information for NFS or iSCSI. ■ Default route, net mask, and primary and secondary DNS server IP addresses. ■ IP address and net mask for the VMkernel primary management network. ■ IP address and net mask for other VMkernel networks such as storage, vSphere FT, or VMware vMotion. vSphere Auto Deploy does not overwrite existing partitions by default.
vSphere PowerCLI	See Install PowerCLI.
ESXi software depot	The location of the ESXi software depot on the Downloads page of the VMware website. You use a URL to point to the image profile stored at that location, or you download a ZIP file to work with a local depot. Do not download the ESXi image.
TFTP server	TFTP installer software such as WinAgents TFTP server.
DHCP server	The DHCP server is included in the vSphere supported Windows Server versions.
DNS server	A working DNS server. You must add entries in both Forward (A Record) and Reverse (PTR Record) Zones for each target host.

You also need information about and administrator privileges to the core servers of the environment, including the ActiveDirectory server, DNS server, DHCP server, NTP server, and so on.

You must have complete control of the broadcast domain of the subnet in which you deploy the setup. Ensure that no other DHCP, DNS, or TFTP server are on this subnet.

Prepare Your System for vSphere Auto Deploy

Before you can PXE boot an ESXi host with vSphere Auto Deploy, you must install prerequisite software and set up the DHCP and TFTP servers that vSphere Auto Deploy interacts with.

If you want to manage vSphere Auto Deploy with PowerCLI cmdlets, see *Set Up vSphere Auto Deploy and Provision Hosts with vSphere PowerCLI*.

Prerequisites

- Verify that the hosts that you plan to provision with vSphere Auto Deploy meet the hardware requirements for ESXi. See ESXi Hardware Requirements.
- Verify that the ESXi hosts have network connectivity to vCenter Server and that all port requirements are met. See *vCenter Server Upgrade*.
- Verify that you have a TFTP server and a DHCP server in your environment to send files and assign network addresses to the ESXi hosts that Auto Deploy provisions. See #unique_83 and #unique_84.
- Verify that the ESXi hosts have network connectivity to DHCP, TFTP, and vSphere Auto Deploy servers.
- If you want to use VLANs in your vSphere Auto Deploy environment, you must set up the end to end networking properly. When the host is PXE booting, the firmware driver must be set up to tag the frames with proper VLAN IDs. You must do this set up manually by making the correct changes in the UEFI/BIOS interface. You must also correctly configure the ESXi port groups with the correct VLAN IDs. Ask your network administrator how VLAN IDs are used in your environment.
- Verify that you have enough storage for the vSphere Auto Deploy repository. The vSphere Auto Deploy server uses the repository to store data it needs, including the rules and rule sets you create and the VIBs and image profiles that you specify in your rules.

  Best practice is to allocate 2 GB to have enough room for four image profiles and some extra space. Each image profile requires approximately 400 MB. Determine how much space to reserve for the vSphere Auto Deploy repository by considering how many image profiles you expect to use.
- Obtain administrative privileges to the DHCP server that manages the network segment you want to boot from. You can use a DHCP server already in your environment, or install a DHCP server. For your vSphere Auto Deploy setup, replace the `gpxelinux.0` filename with `snponly64.efi.vmw-hardwired` for UEFI or `undionly.kpxe.vmw-hardwired` for BIOS. For more information on DHCP configurations, see Sample DHCP Configurations.
- Secure your network as for any other PXE-based deployment method. vSphere Auto Deploy transfers data over SSL to prevent casual interference and snooping. However, the authenticity of the client or the vSphere Auto Deploy server is not checked during a PXE boot.
- If you want to manage vSphere Auto Deploy with PowerCLI cmdlets, verify that Microsoft .NET Framework 4.5 or 4.5.x and Windows PowerShell 3.0 or 4.0 are installed on a Windows machine. See the *vSphere PowerCLI User's Guide*.
- Set up a remote Syslog server. See the *vCenter Server and Host Management* documentation for Syslog server configuration information. Configure the first host you boot to use the remote

Syslog server and apply that host's host profile to all other target hosts. Optionally, install and use the vSphere Syslog Collector, a vCenter Server support tool that provides a unified architecture for system logging and enables network logging and combining of logs from multiple hosts.

- Install ESXi Dump Collector, set up your first host so that all core dumps are directed to ESXi Dump Collector, and apply the host profile from that host to all other hosts.
- If the hosts that you plan to provision with vSphere Auto Deploy are with legacy BIOS, verify that the vSphere Auto Deploy server has an IPv4 address. PXE booting with legacy BIOS firmware is possible only over IPv4. PXE booting with UEFI firmware is possible with either IPv4 or IPv6.

Procedure

1 Navigate to **Home > Auto Deploy**.

  By default, only the administrator role has privileges to use the vSphere Auto Deploy service.

2 On the **Auto Deploy** page, select your vCenter Server from the drop-down menu at the top.

3 Click **Enable Auto Deploy and Image Builder** to activate the service.

  If the **Image Builder** service is already active, select the **Configure** tab and click **Enable Auto Deploy Service**.

  The **Software Depot** page appears.

4 Configure the TFTP server.

  a Click the **Configure** tab.

  b Click **Download TFTP Boot Zip** to download the TFTP configuration file and unzip the file to the directory in which your TFTP server stores files.

  c (Optional) To use a proxy server, click **Add** on the *Auto Deploy Runtime Summary* pane and enter a proxy server URL in the text box.

    Using reverse proxy servers can offload the requests made to the vSphere Auto Deploy server.

5 Set up your DHCP server to point to the TFTP server on which the TFTP ZIP file is located.

  a Specify the TFTP Server's IP address in DHCP option 66, frequently called next-server.

  b Specify the boot filename, which is `snponly64.efi.vmw-hardwired` for UEFI or `undionly.kpxe.vmw-hardwired` for BIOS in the DHCP option 67, frequently called `boot-filename`.

6 Set each host you want to provision with vSphere Auto Deploy to network boot or PXE boot, following the manufacturer's instructions.

7 (Optional) If you set up your environment to use Thumbprint mode, you can use your own Certificate Authority (CA) by replacing the OpenSSL certificate `rbd-ca.crt` and the OpenSSL private key `rbd-ca.key` with your own certificate and key file.

   The files are in `/etc/vmware-rbd/ssl/`.

   By default, vCenter Server uses VMware Certificate Authority (VMCA).

Results

When you start an ESXi host that is set up for vSphere Auto Deploy, the host contacts the DHCP server and is directed to the vSphere Auto Deploy server, which provisions the host with the image profile specified in the active rule set.

What to do next

- You can change the default configuration properties of the **Auto Deploy Service**. For more information, see "Configuring vCenter Server" in the *vCenter Server and Host Management* documentation.
- You can change the default configuration properties of the **Image Builder Service**. For more information, see "Configuring vCenter Server" in the *vCenter Server and Host Management* documentation.
- Define a rule that assigns an image profile and optional host profile, host location, or script bundle to the host.
- (Optional) Configure the first host that you provision as a reference host. Use the storage, networking, and other settings you want for your target hosts to share. Create a host profile for the reference host and write a rule that assigns both the already tested image profile and the host profile to target hosts.
- (Optional) If you want to have vSphere Auto Deploy overwrite existing partitions, set up a reference host to do auto partitioning and apply the host profile of the reference host to other hosts.
- (Optional) If you have to configure host-specific information, set up the host profile of the reference host to prompt for user input. For more information about host customizations, see the *vSphere Host Profiles* documentation.

Using vSphere Auto Deploy Cmdlets

vSphere Auto Deploy cmdlets are implemented as Microsoft PowerShell cmdlets and included in vSphere PowerCLI. Users of vSphere Auto Deploy cmdlets can take advantage of all vSphere PowerCLI features.

Experienced PowerShell users can use vSphere Auto Deploy cmdlets just like other PowerShell cmdlets. If you are new to PowerShell and vSphere PowerCLI, the following tips might be helpful.

You can type cmdlets, parameters, and parameter values in the vSphere PowerCLI shell.

- Get help for any cmdlet by running `Get-Help`*`cmdlet_name`*.

- Remember that PowerShell is not case sensitive.
- Use tab completion for cmdlet names and parameter names.
- Format any variable and cmdlet output by using `Format-List` or `Format-Table`, or their short forms `fl` or `ft`. For more information, run the `Get-Help Format-List` cmdlet.

Passing Parameters by Name

You can pass in parameters by name in most cases and surround parameter values that contain spaces or special characters with double quotes.

```
Copy-DeployRule -DeployRule testrule -ReplaceItem MyNewProfile
```

Most examples in the *vCenter Server Installation and Setup* documentation pass in parameters by name.

Passing Parameters as Objects

You can pass parameters as objects if you want to perform scripting and automation. Passing in parameters as objects is useful with cmdlets that return multiple objects and with cmdlets that return a single object. Consider the following example.

1 Bind the object that encapsulates rule set compliance information for a host to a variable.

```
$tr = Test-DeployRuleSetCompliance MyEsxi42
```

2 View the `itemlist` property of the object to see the difference between what is in the rule set and what the host is currently using.

```
$tr.itemlist
```

3 Remediate the host to use the revised rule set by using the `Repair-DeployRuleSetCompliance` cmdlet with the variable.

```
Repair-DeployRuleSetCompliance $tr
```

The example remediates the host the next time you boot the host.

Set Up Bulk Licensing

You can use the vSphere Client or ESXi Shell to specify individual license keys, or you can set up bulk licensing by using vSphere PowerCLI cmdlets. Bulk licensing works for all ESXi hosts, but is especially useful for hosts provisioned with vSphere Auto Deploy.

Assigning license keys through the vSphere Client and assigning licensing by using vSphere PowerCLI cmdlets function differently.

Assign license keys with the vSphere Client

You can assign license keys to a host when you add the host to the vCenter Server system or when the host is managed by a vCenter Server system.

Assign license keys with LicenseDataManager vSphere PowerCLI

You can specify a set of license keys to be added to a set of hosts. The license keys are added to the vCenter Server database. Each time a host is added to the vCenter Server system or reconnects to it, the host is assigned a license key. A license key that is assigned through vSphere PowerCLI is treated as a default license key. When an unlicensed host is added or reconnected, it is assigned the default license key. If a host is already licensed, it keeps its license key.

The following example assigns licenses to all hosts in a data center. You can also associate licenses with hosts and clusters.

The following example is for advanced vSphere PowerCLI users who know how to use PowerShell variables.

Prerequisites

Prepare Your System for vSphere Auto Deploy.

Procedure

1 In a vSphere PowerCLI session, connect to the vCenter Server system you want to use and bind the associated license manager to a variable.

```
Connect-VIServer -Server 192.XXX.X.XX -User username -Password password
$licenseDataManager = Get-LicenseDataManager
```

2 Run a cmdlet that retrieves the data center in which the hosts for which you want to use the bulk licensing feature are located.

```
$hostContainer = Get-Datacenter -Name Datacenter-X
```

You can also run a cmdlet that retrieves a cluster to use bulk licensing for all hosts in a cluster, or retrieves a folder to use bulk licensing for all hosts in a folder.

3 Create a `LicenseData` object and a `LicenseKeyEntry` object with associated type ID and license key.

```
$licenseData = New-Object VMware.VimAutomation.License.Types.LicenseData
$licenseKeyEntry = New-Object Vmware.VimAutomation.License.Types.LicenseKeyEntry
$licenseKeyEntry.TypeId = "vmware-vsphere"
$licenseKeyEntry.LicenseKey = "XXXXX-XXXXX-XXXXX-XXXXX-XXXXX"
```

4 Associate the `LicenseKeys` attribute of the `LicenseData` object you created in step 3 with the `LicenseKeyEntry` object.

```
$licenseData.LicenseKeys += $licenseKeyEntry
```

5 Update the license data for the data center with the `LicenseData` object and verify that the license is associated with the host container.

```
$licenseDataManager.UpdateAssociatedLicenseData($hostContainer.Uid, $licenseData)
$licenseDataManager.QueryAssociatedLicenseData($hostContainer.Uid)
```

6 Provision one or more hosts with vSphere Auto Deploy and assign them to the data center or to the cluster that you assigned the license data to.

7 You can use the vSphere Client to verify that the host is successfully assigned to the default license `XXXXX-XXXXX-XXXXX-XXXXX-XXXXX`.

Results

All hosts that you assigned to the data center are now licensed automatically.

Reprovisioning Hosts

vSphere Auto Deploy supports multiple reprovisioning options. You can perform a simple reboot or reprovision with a different image profile or a different host profile.

A first boot using vSphere Auto Deploy requires that you set up your environment and add rules to the rule set. See the topic "Preparing for vSphere Auto Deploy" in the *vSphere installation and Setup* documentation.

The following reprovisioning operations are available.

- Simple reboot.
- Reboot of hosts for which the user answered questions during the boot operation.
- Reprovision with a different image profile.
- Reprovision with a different host profile.

Reprovision Hosts with Simple Reboot Operations

A simple reboot of a host that is provisioned with vSphere Auto Deploy requires only that all prerequisites are still met. The process uses the previously assigned image profile, host profile, custom script, and vCenter Server location.

Prerequisites

- Verify that the setup you performed during the first boot operation is in place.
- Verify that all associated items like are available. An item can be an image profile, host profile, custom script or vCenter Server inventory location.
- Verify that the host has the identifying information (asset tag, IP address) it had during previous boot operations.

Procedure

1 Place the host in maintenance mode.

Host Type	Action
Host is part of a DRS cluster	VMware DRS migrates virtual machines to appropriate hosts when you place the host in maintenance mode.
Host is not part of a DRS cluster	You must migrate all virtual machines to different hosts and place each host in maintenance mode.

2 Reboot the host.

Results

The host shuts down. When the host reboots, it uses the image profile that the vSphere Auto Deploy server provides. The vSphere Auto Deploy server also applies the host profile stored on the vCenter Server system.

Reprovision a Host with a New Image Profile by Using vSphere PowerCLI

You can use vSphere Auto Deploy to reprovision a host with a new image profile in a vSphere PowerCLI session by changing the rule for the host and performing a test and repair compliance operation.

Several options for reprovisioning hosts exist.

- If the VIBs that you want to use support live update, you can use an `esxcli software vib update` command. In that case, you must also update the rule set to use an image profile that includes the new VIBs.
- During testing, you can apply an image profile to an individual host with the `Apply-EsxImageProfile` cmdlet and reboot the host so the change takes effect. The `Apply-EsxImageProfile` cmdlet updates the association between the host and the image profile but does not install VIBs on the host.
- In all other cases, use this procedure.

Prerequisites

- Verify that the image profile you want to use to reprovision the host is available. Use vSphere ESXi Image Builder in a vSphere PowerCLI session. See "Using vSphere ESXi Image Builder CLI" in the *vSphere Installation and Setup* documentation.
- Verify that the setup you performed during the first boot operation is in place.

Procedure

1 At the PowerShell prompt, run the `Connect-VIServer` vSphere PowerCLI cmdlet to connect to the vCenter Server system that vSphere Auto Deploy is registered with.

```
Connect-VIServer ipv4_or_ipv6_address
```

The cmdlet might return a server certificate warning. In a production environment, make sure no server certificate warnings result. In a development environment, you can ignore the warning.

2 Determine the location of a public software depot that contains the image profile that you want to use, or define a custom image profile with vSphere ESXi Image Builder.

3 Run `Add-EsxSoftwareDepot` to add the software depot that contains the image profile to the vSphere PowerCLI session.

Depot Type	Cmdlet
Remote depot	Run **Add-EsxSoftwareDepot *depot_url***.
ZIP file	a Download the ZIP file to a local file path or create a mount point local to the vSphere PowerCLI machine. b Run **Add-EsxSoftwareDepot C:*file_path**my_offline_depot*.zip**.

4 Run `Get-EsxImageProfile` to see a list of image profiles, and decide which profile you want to use.

5 Run `Copy-DeployRule` and specify the `ReplaceItem` parameter to change the rule that assigns an image profile to hosts.

The following cmdlet replaces the current image profile that the rule assigns to the host with the *my_new_imageprofile* profile. After the cmdlet completes, `myrule` assigns the new image profile to hosts. The old version of `myrule` is renamed and hidden.

```
Copy-DeployRule myrule -ReplaceItem my_new_imageprofile
```

6 Test the rule compliance for each host that you want to deploy the image to.

a Verify that you can access the host for which you want to test rule set compliance.

```
Get-VMHost -Name ESXi_hostname
```

b Run the cmdlet that tests rule set compliance for the host, and bind the return value to a variable for later use.

```
$tr = Test-DeployRuleSetCompliance ESXi_hostname
```

c Examine the differences between the contents of the rule set and configuration of the host.

```
$tr.itemlist
```

The system returns a table of current and expected items if the host for which you want to test the new rule set compliance is compliant with the active rule set.

```
CurrentItem                          ExpectedItem
-----------                          ------------
my_old_imageprofile                        my_new_imageprofile
```

d Remediate the host to use the revised rule set the next time you boot the host.

```
Repair-DeployRuleSetCompliance $tr
```

7 Reboot the host to provision it with the new image profile.

Write a Rule and Assign a Host Profile to Hosts

vSphere Auto Deploy can assign a host profile to one or more hosts. The host profile might include information about storage configuration, network configuration, or other characteristics of the host. If you add a host to a cluster, that cluster's host profile is used.

In many cases, you assign a host to a cluster instead of specifying a host profile explicitly. The host uses the host profile of the cluster.

Prerequisites

- Install PowerCLI and all prerequisite software. For information see *vCenter Server Installation and Setup*.
- Export the host profile that you want to use.

Procedure

1 In a PowerCLI session, run the `Connect-VIServer` cmdlet to connect to the vCenter Server system that vSphere Auto Deploy is registered with.

```
Connect-VIServer ipv4_or_ipv6_address
```

The cmdlet might return a server certificate warning. In a production environment, make sure no server certificate warnings result. In a development environment, you can ignore the warning.

2 Using the vSphere Client, set up a host with the settings you want to use and create a host profile from that host.

3 Find the name of the host profile by running `Get-VMhostProfile` PowerCLI cmdlet, passing in the ESXi host from which you create a host profile.

4 At the PowerCLI prompt, define a rule in which host profiles are assigned to hosts with certain attributes, for example a range of IP addresses.

```
New-DeployRule -Name "testrule2" -Item my_host_profile -Pattern "vendor=Acme,Zven",
"ipv4=192.XXX.1.10-192.XXX.1.20"
```

The specified item is assigned to all hosts with the specified attributes. This example specifies a rule named testrule2. The rule assigns the specified host profile *my_host_profile* to all hosts with an IP address inside the specified range and with a manufacturer of Acme or Zven.

5 Add the rule to the rule set.

```
Add-DeployRule testrule2
```

By default, the working rule set becomes the active rule set, and any changes to the rule set become active when you add a rule. If you use the `NoActivate` parameter, the working rule set does not become the active rule set.

What to do next

- Assign a host already provisioned with vSphere Auto Deploy to the new host profile by performing compliance test and repair operations on those hosts. For more information, see Test and Repair Rule Compliance .
- Power on unprovisioned hosts to provision them with the host profile.

Test and Repair Rule Compliance

When you add a rule to the vSphere Auto Deploy rule set or modify one or more rules, hosts are not updated automatically. vSphere Auto Deploy applies the new rules only when you test their rule compliance and perform remediation.

Prerequisites

- Prepare your system and install the Auto Deploy Server. For more information, see Prepare Your System for vSphere Auto Deploy.
- Verify that your infrastructure includes one or more ESXi hosts provisioned with vSphere Auto Deploy, and that the host on which you installed PowerCLI can access those ESXi hosts.

Procedure

1 In a PowerCLI session, run the `Connect-VIServer` cmdlet to connect to the vCenter Server system that vSphere Auto Deploy is registered with.

```
Connect-VIServer ipv4_or_ipv6_address
```

The cmdlet might return a server certificate warning. In a production environment, make sure no server certificate warnings result. In a development environment, you can ignore the warning.

2 Use PowerCLI to check which vSphere Auto Deploy rules are currently available.

```
Get-DeployRule
```

The system returns the rules and the associated items and patterns.

3 Modify one of the available rules.

For example, you can change the image profile and the name of the rule.

```
Copy-DeployRule -DeployRule testrule -ReplaceItem MyNewProfile
```

You cannot edit a rule already added to the active rule set. Instead, you can copy the rule and replace the item or pattern you want to change.

4 Verify that you can access the host for which you want to test rule set compliance.

```
Get-VMHost -Name MyEsxi42
```

5 Run the cmdlet that tests rule set compliance for the host, and bind the return value to a variable for later use.

```
$tr = Test-DeployRuleSetCompliance MyEsxi42
```

6 Examine the differences between the contents of the rule set and configuration of the host.

```
$tr.itemlist
```

If the host for which you want to test the new rule set compliance is compliant with the active rule set, the system returns a table of current and expected items.

```
CurrentItem                          ExpectedItem
-----------                          ------------
My Profile 25                        MyNewProfile
```

7 Remediate the host to use the revised rule set the next time you boot the host.

```
Repair-DeployRuleSetCompliance $tr
```

What to do next

If the rule you changed specified the inventory location, the change takes effect when you repair compliance. For all other changes, reboot your host to have vSphere Auto Deploy apply the new rule and to achieve compliance between the rule set and the host.

Collect Logs to Troubleshoot ESXi Hosts

5

You can collect installation or upgrade log files for ESXi. If an installation or upgrade fails, checking the log files can help you identify the source of the failure.

Solution

1 Enter the `vm-support` command in the ESXi Shell or through SSH.

2 Navigate to the `/var/tmp/` directory.

3 Retrieve the log files from the `.tgz` file.