VMware ESXi Installation and Setup

Update 3

VMware vSphere 8.0
VMware ESXi 8.0

vmware by Broadcom

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

https://docs.vmware.com/

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

Contents

About VMware ESXi Installation and Setup

1

VMware ESXi Installation and Setup describes how to install and configure VMware ESXi™.

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 Installation and Setup is intended for experienced administrators who want to install and configure ESXi.

This information is written for experienced Windows or Linux system administrators who are familiar with virtual machine technology and data center operations. The information about using the Image Builder and VMware vSphere® Auto Deploy™ is written for administrators who have experience with Microsoft PowerShell and VMware vSphere® PowerCLI™.

Introduction to vSphere Installation and Setup

2

vSphere 8.0 provides various options for installation and setup that define a corresponding sequence of tasks.

The two core components of vSphere are ESXi and 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. vCenter Server lets you pool and manage the resources of multiple hosts.

You deploy the vCenter Server appliance, a preconfigured virtual machine optimized for running vCenter Server and the vCenter Server components. You can deploy the vCenter Server appliance on ESXi hosts or on vCenter Server instances.

For detailed information about the vCenter Server installation process, see *vCenter Server Installation and Setup*.

Read the following topics next:

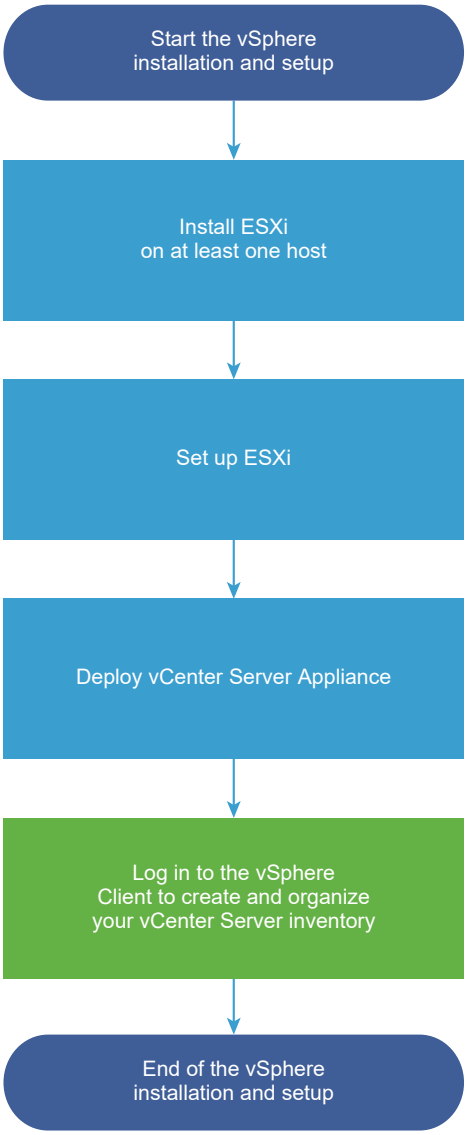
- Overview of the vSphere Installation and Setup Process
- About ESXi Evaluation and Licensed Modes

Overview of the vSphere Installation and Setup Process

vSphere is a sophisticated product with multiple components to install and set up. To ensure a successful vSphere deployment, understand the sequence of tasks required.

Installing vSphere includes the following tasks:

Figure 2-1. vSphere Installation and Setup Workflow



1 Read the vSphere release notes.

2 Install ESXi.

   a Verify that your system meets the minimum hardware requirements. See ESXi Requirements.

   b Determine the ESXi installation option to use: interactive, scripted or by using vSphere Auto Deploy.

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

   d Create a worksheet with the information you will need when you install ESXi. See Required Information for ESXi Installation.

e Install ESXi.

- Installing ESXi Interactively
- Scripted ESXi Installation

Note You can also provision ESXi hosts by using vSphere Auto Deploy, but vSphere Auto Deploy is installed together with vCenter Server. To provision ESXi hosts by using Auto Deploy, you must install vCenter Server.

3 Configure the ESXi boot and network settings, the direct console, and other settings. See Setting Up ESXi and After You Install and Set Up ESXi.

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

5 Install vCenter Server.

For detailed information, see the vCenter Server Installation and Setup guide.

About ESXi Evaluation and Licensed Modes

You can use evaluation mode to explore a set of features equal to the vSphere Enterprise Plus license.

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.

What is VMware vSphere Distributed Services Engine®

3

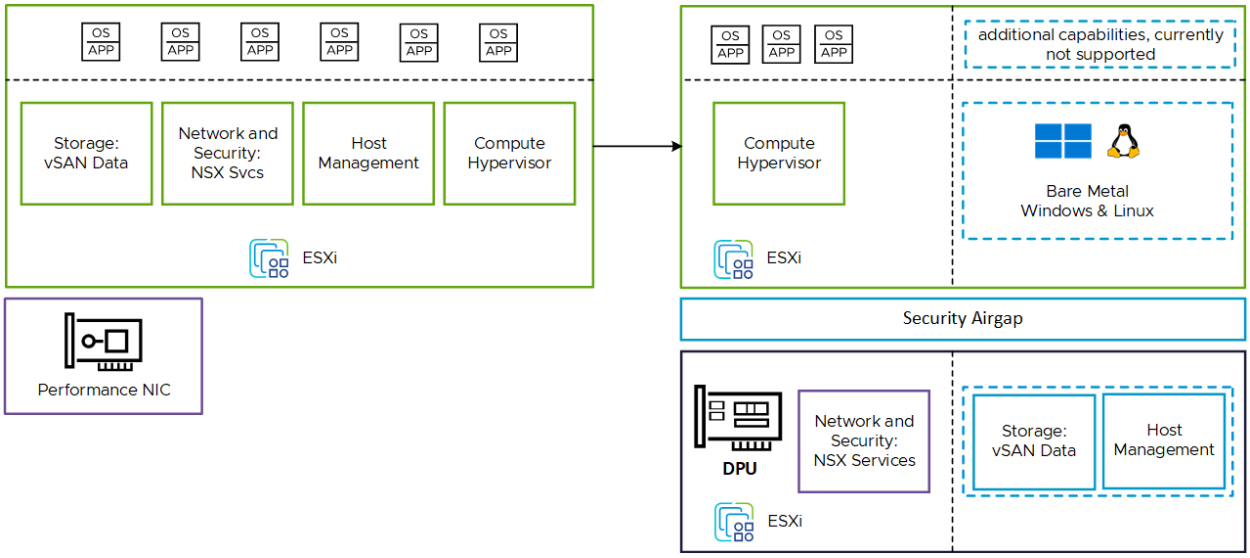
vSphere Distributed Services Engine® is a core vSphere capability that enables customers to use DPUs with vSphere and VMware Cloud Foundation.

vSphere 8.0 enables breakthrough workload performance to meet ever-increasing throughput and latency needs. With vSphere Distributed Services Engine, infrastructure services are distributed across the different compute resources available on the ESXi host, with networking functions offloaded to the DPU. Such a capability works well for modern applications, which are developed using a microservices architecture approach that seeks to break down the application into multiple independent but cooperating services. This increased complexity places new demand for the CPU. For example, processing storage requests or shuttling network traffic for these microservices leaves fewer CPU cycles for the actual workload. In this context, purpose-built accelerators such as DPUs can take on the new compute burden and help you improve the performance and efficiency of infrastructure.

With vSphere Distributed Services Engine, DPUs can accelerate the performance of your network and increase data throughput, while placing no operational burden of managing the lifecycle of DPUs, as the existing Day-0, Day-1, and Day-2 vSphere experience does not change. vSphere Distributed Services Engine is supported by DPUs from NVIDIA and AMD, and server designs from Dell, HPE, Lenovo, and Fujitsu. vSphere Distributed Services Engine is available on servers with pre-installed DPUs.

Starting with vSphere 8.0, you can offload functionality that runs on the core CPU onto the DPU to significantly improve the network and security performance. As illustrated in the Evolving vSphere Architecture diagram, DPUs can also handle additional capabilities such as storage offload and bare metal management, but these additional capabilities are currently not supported.

Figure 3-1. Evolving vSphere Architecture.



vSphere Distributed Services Engine offloads and accelerates infrastructure functions on the DPU by introducing a VMware vSphere Distributed Switch on the DPU and VMware NSX Networking and Observability, which allows you to proactively monitor, identify, and mitigate network infrastructure bottlenecks without complex network taps. The DPU becomes a new control point to scale infrastructure functions and enables security controls that are agentless and decoupled from the workload domain.

With vSphere Distributed Services Engine, you can:

- Install and update ESXi images simultaneously on the x86 server and the attached supported DPU to reduce operational overhead of DPU lifecycle management with integrated vSphere workflows. For more information, see Using vSphere Lifecycle Manager With VMware vSphere Distributed Services Engine.
- Set alarms for DPU hardware alerts and monitor performance metrics on core, memory, and network throughput from the familiar vCenter interfaces, without the need of new tools. For more information, see CPU (DPU) and Memory (DPU).
- Accelerate vSphere Distributed Switch on the DPU to improve network performance and utilize available CPU cycles to achieve higher workload consolidation per ESXi host. For more information, see What is Network Offloads Capability and Create a vSphere Distributed Switch.
- Get vSphere DRS and vSphere vMotion support for VMs running on hosts with DPUs attached to get the benefits of passthrough without sacrificing on VM portability. For more information, see Homogenous clusters for DPUs.
- Improve the security of infrastructure with zero-trust security. For more information, see vSphere Distributed Services Engine Security Best Practices.

vSphere Distributed Services Engine does not require a separate ESXi license. An internal network that is isolated from other networks, connects the DPUs with ESXi hosts. ESXi 8.0 server builds are unified images, which contain both x86 and DPU content. In your vSphere system, you see DPUs as new objects during installation and upgrade, and in networking, storage, and host profile workflows.

Read the following topics next:

- High Availability with VMware vSphere Distributed Services Engine
- Error Handling, Failover, and Rollback for VMware vSphere Distributed Services Engine

High Availability with VMware vSphere Distributed Services Engine

With ESXi 8.0 Update 3, you can opt for a VMware vSphere Distributed Services Engine installation with 2 data processing units (DPUs) to achieve high availability.

In vSphere systems with a single DPU, the device might become the single point of failure for workloads offloaded to the DPU, such as networking functions, and impact data and productivity. With ESXi 8.0 Update 3, vSphere Distributed Services Engine is also available on servers with 2 pre-installed DPUs, which provides hardware redundancy and resiliency.

You can utilize the two DPUs in Active/Standby mode to provide high availability. Such configuration provides redundancy in the event one of the DPUs fails. In the high availability configuration, both DPUs are assigned to the same NSX-backed vSphere Distributed Switch. For example, DPU-1 is attached to vmnic0 and vmnic1 of the vSphere Distributed Switch and DPU-2 is attached to vmnic2 and vmnic3 of the same vSphere Distributed Switch.

You can also utilize the two DPUs as independent devices to increase offload capacity per ESXi host. Each DPU is attached to a separate vSphere Distributed Switch and you have no failover between DPUs in such configuration.

Dual-DPU systems can use NVIDIA or Pensando devices. In ESXi 8.0 Update 3, dual-DPU systems are supported by Lenovo server designs. The DPU devices on a dual DPU server must be identical in all aspects: same vendor, same hardware version and same firmware. For a list of current vendors and server designs for VMware vSphere Distributed Services Engine, see the VMware Compatibility Guide.

Installation of VMware vSphere Distributed Services Engine with 2 DPUs

vSphere Distributed Services Engine does not require a separate ESXi license. ESXi 8.0 Update 3 server builds are unified images, which contain both x86 and DPU content, and you cannot install x86 and DPU content separately. The installation procedure on both DPUs, either interactive or scripted, also happens in parallel and you see minimal performance loss as compared to a single-DPU system.

With vSphere 8.0 Update 3, you can get a pre-installed server configuration with 2 DPUs from Dell or Lenovo, or add a second DPU to a single DPU system on the supported dual DPU servers from Dell or Lenovo.

Note In any case, you need to run a complete fresh ESXi 8.0 Update 3 installation on your system, not only on the newly added DPUs.

For more information on the installation, see Install ESXi Interactively and Installation and Upgrade Scripts Used for ESXi Installation.

Error Handling, Failover, and Rollback for VMware vSphere Distributed Services Engine

Before installing VMware vSphere Distributed Services Engine, see the error handling, failover, and rollback options.

Error Handling

An installation failure of either x86 and DPU content on an ESXi host marks the entire installation procedure as failed.

While the expectation is that the software state of DPUs remains identical at all times, in the unlikely case of an error during a lifecycle operation, such as installation or upgrade of a Component, the operation might pass on one DPU but fail on the other. Since each lifecycle operation occurs within the boundaries of each DPU, errors do not affect the state of the other DPU, but the overall result of the installation is still marked as a failure.

During interactive install, in vSphere Lifecycle Manager workflows, and when you use ESXCLI, you receive information about the DPU on which the operation failed.

After a successful installation, in case of DPU errors, the recommended action is to restart the affected ESXi host. If the DPU is still accessible from the host, the general log bundle collection is sufficient for troubleshooting. If the DPU is not accessible from the host, logging in to the DPU from a BMC, iLO, or iDRAC interface can provide troubleshooting logs.

Failover

Failover support in vSphere 8.0 Update 3 is limited to one of the DPUs becoming non-functional due to software errors within the DPU or a physical disconnect of one of the DPUs, such as cable disconnect. Failover due to Peripheral Component Interconnect (PCI) level errors is not supported.

Rollback

Rollback is a best effort mechanism to restore the system to a previous working state in case of a failure before the jumpstart phase of the ESXi boot. Rollback on both x86 servers and the attached supported DPUs is automatic in case of an error during booting. You can also opt for a manual rollback by pressing **`Shift+R`** before the bootloader starts, to return to a previous good state.

Any failure after the jumpstart phase starts does not result in a rollback.

Table 3-1. Rollback scenarios for VMware vSphere Distributed Services Engine installation

Scenario	Number of reboots required
Both DPUs boot correctly. ESXi does not boot correctly.	2
Both DPUs do not boot correctly. ESXi boots correctly.	1
One of the DPUs boots with an earlier version than the other DPU and ESXi.	2
One of the DPUs boots with an earlier version than the other DPU and ESXi does not boot correctly.	2

Installing and Setting Up ESXi

4

You can install and set up ESXi on your physical hardware so that it acts as a platform for virtual machines.

Read the following topics next:

- ESXi Requirements
- Preparing for Installing ESXi
- Customizing Installations with vSphere ESXi Image Builder
- Installing ESXi
- Setting Up ESXi
- After You Install and Set Up ESXi

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 management of partitions and third-party components, while facilitating debugging.

ESXi System Storage

The ESXi 8.0 system storage layout consists of four partitions:

Table 4-1. 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

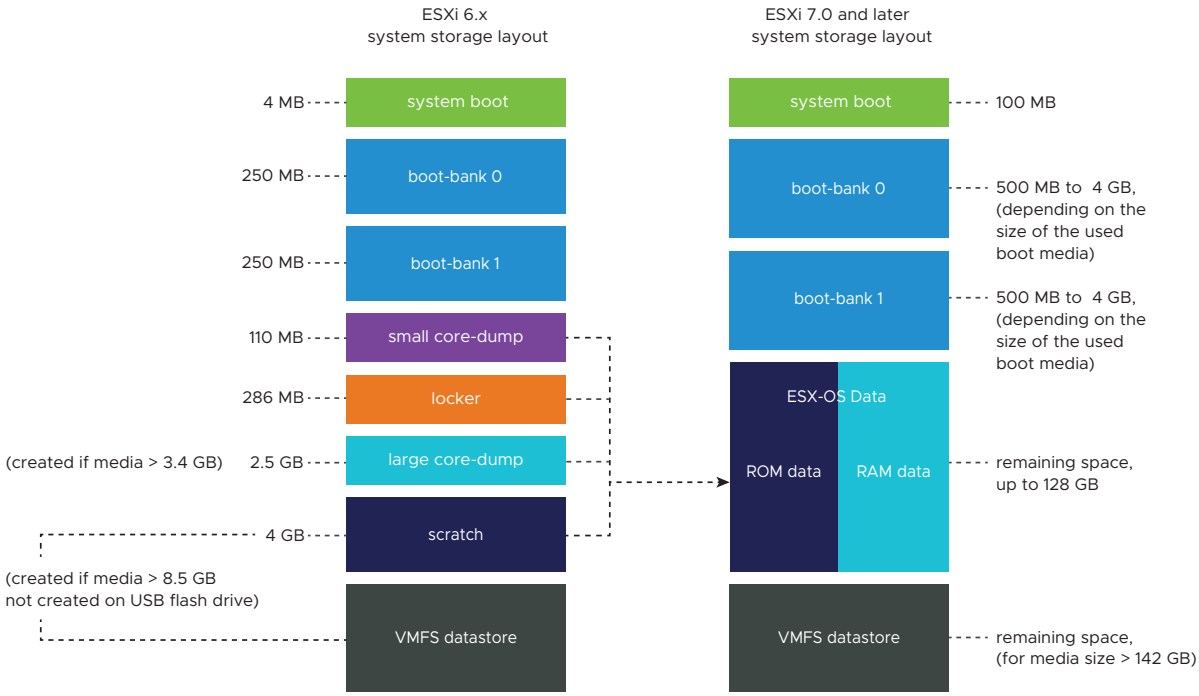
Table 4-1. ESXi system storage partitions: (continued)

Partition	Use	Type
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 4-1. 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 4-2. 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 4-3. ESXi system storage symbolic links.

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

Table 4-3. ESXi system storage symbolic links. (continued)

System Storage Volume	Symbolic Link
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. Starting with ESXi 7.0, you can activate the option `autoPartition`, which automatically formats all available empty devices with VMFS, except for legacy SD and USB devices. The default is `autoPartition=FALSE`, which formats with VMFS only boot devices with size larger than 128 GB. For more information, see VMware knowledge base article 77009.

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 system boot, boot banks, and ESX-OSData. 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 as part of the ESX-OSDATA partition.

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 upgrade process to ESXi 7.0 and later repartitions the boot device and consolidates the original core dump, locker, and scratch partitions into the ESX-OSData volume.

The scratch partition serves 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. If no scratch partition is created, you can configure one. You can also override the default configuration.

You can create the scratch partition on a remote SAN or NFS-mounted directory.

Set the Scratch Partition from the vSphere Client

If a scratch partition is not set up, you might want to configure one, especially if the host is low on memory. When a scratch partition is not present, system logs are stored in a ramdisk.

Prerequisites

The directory to use for the scratch partition must exist on the host.

Procedure

1. From the vSphere Client, connect to the vCenter Server.
2. Select the host in the inventory.
3. Click the **Configure** tab.
4. Select **System**.
5. Select **Advanced System Settings**.

   The setting **ScratchConfig.CurrentScratchLocation** shows the current location of the scratch partition.

6 In the **ScratchConfig.ConfiguredScratchLocation** text box, enter a directory path that is unique for this host.

   For example, **/vmfs/volumes/*DatastoreUUID*/*DatastoreFolder***.

7 Reboot the host for the changes to take effect.

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.
- 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.
- 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.
- 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. 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 Deprecation of legacy BIOS support in vSphere.

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.

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 SD card/USB boot device revised guidance on updated guidance for SD card or USB-based environments.
- To chose a proper SD or USB boot device, see Boot device guidance for low endurance media (vSphere and vSAN).

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 from ESXi 8.x to a version of ESXi earlier than 7.x is not possible due to the repartitioning process of the boot device. To use a version of ESXi earlier than 7.x 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. Rollback from ESXi 8.x to 7.x is possible as long as no changes to the bootbank partitions have been made and no corrupt partition is detected.

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.

To reconfigure `/scratch`, see Set the Scratch Partition from the vSphere Client.

To configure the size of ESXi system partitions, you can use the `systemMediaSize` option. For more information, see Boot option to configure the size of ESXi system partitions.

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), Fujitsu iRMC, 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 4-4. 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
Fujitsu iRMC S5	1.10P	1.7.0_60-b19
Fujitsu iRMC S6	1.06S	N/A
HP ILO	1.81, 1.92	1.6.0_22, 1.6.0_23
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 4-5. 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.
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.

Table 4-5. Recommendations for Enhanced Performance (continued)

System Element	Recommendation
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

Open and close firewall ports for each service by using either the vSphere Client or the VMware Host Client.

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 but the VMware Ports and Protocols Tool includes some other ports as well.

Required Free Space for System Logging

See the recommended minimum size and rotation configuration for hostd, vpxa, and fdm logs.

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 4-6. 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 a remote log server, see Configure Syslog on ESXi Hosts.

You can optionally install VMware vCenter Log Insight, which provides log aggregation and analytics.

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.

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
```

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. Starting in vSphere 8.0 Update 1, the default is five. Zero deactivates password history.

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

Preparing for Installing ESXi

Before you install ESXi, determine the installation option that is suitable for your environment and prepare for the installation process.

Download the ESXi Installer

You can obtain the ESXi installer software either from an OEM or from the Broadcom Support Portal.

Register on the Broadcom Support Portal. For more information, see Register for an account on the Broadcom Support Portal and Communities.

For product download instructions, see Download Broadcom products and software.

For download of offline bundle ZIP files for ESXi patches and updates, see Downloading Broadcom PTF files and solutions.

For more information, see VMware to Broadcom Support Frequently Asked Questions.

Required Information for ESXi Installation

Interactive installation prompts you for the required system information, but you must supply this information in the installation script.

In an interactive installation, the system prompts you for the required system information. In a scripted installation, you must supply this information in the installation script. For future use, note the values you use during the installation. These notes are useful if you must reinstall ESXi and reenter the values that you originally selected.

Table 4-7. Required Information for ESXi Installation

Information	Required or Optional	Default	Comments
Keyboard layout	Required	U.S. English	
VLAN ID	Optional	None	Range: 0 through 4094
IP address	Optional	DHCP	You can allow DHCP to configure the network during installation. After installation, you can change the network settings.
Subnet mask	Optional	Calculated based on the IP address	
Gateway	Optional	Based on the configured IP address and subnet mask	
Primary DNS	Optional	Based on the configured IP address and subnet mask	
Secondary DNS	Optional	None	
Host name	Required for static IP settings	None	The vSphere Client can use either the host name or the IP address to access the ESXi host.

Table 4-7. Required Information for ESXi Installation (continued)

Information	Required or Optional	Default	Comments
Install location	Required	None	Must be at least 5 GB if you install the components on a single disk.
Migrate existing ESXi settings. Preserve existing VMFS datastore.	Required if you are installing ESXi on a drive with an existing ESXi installation.	None	If you have an existing ESXi 5.x installation, the ESXi installer offers a choice between preserving or overwriting the VMFS datastore during installation
Root password	Required	None	The root password must contain between 8 and 40 characters. For information about passwords see the *vSphere Security* documentation.

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. See 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 Run 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 Installing ESXi by Using a Script 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 the Broadcom Support Portal.

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.

Customizing Installations with vSphere ESXi Image Builder

You can use VMware vSphere® ESXi™ Image Builder CLI to create ESXi installation images with a customized set of updates, patches, and drivers.

You can use vSphere ESXi Image Builder with the vSphere Client or with PowerCLI to create an ESXi installation image with a customized set of ESXi updates and patches. You can also include third-party network or storage drivers that are released between vSphere releases.

You can deploy an ESXi image created with vSphere ESXi Image Builder in either of the following ways:

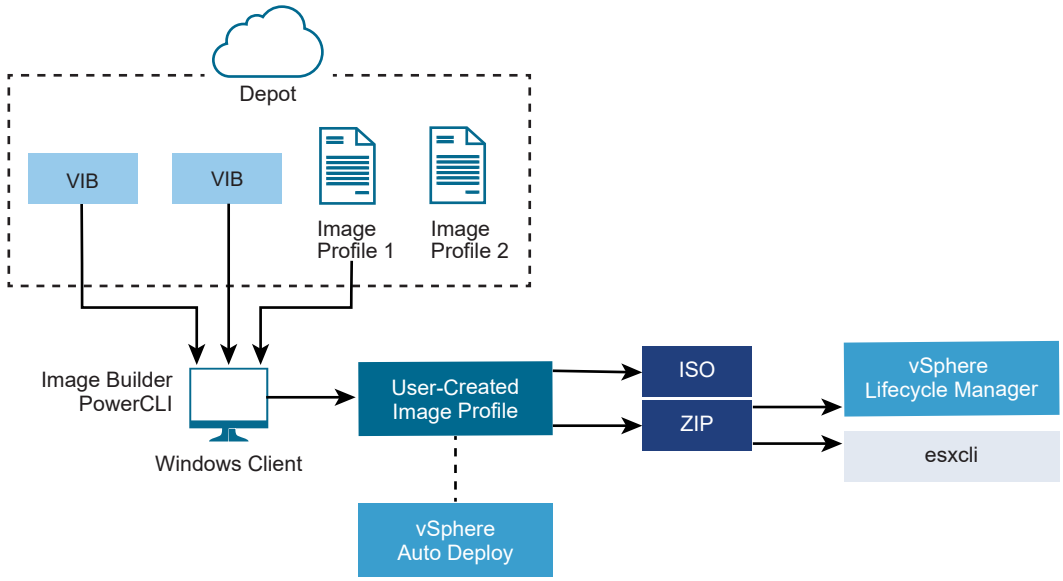
- By burning it to an installation DVD.
- Through vCenter Server, using the Auto Deploy feature.

How the vSphere ESXi Image Builder Works

Create ESXi image profiles for use by vSphere Auto Deploy, add custom third-party drivers to image profiles and export them, or perform upgrades.

With vSphere ESXi Image Builder you can create ESXi image profiles for use by vSphere Auto Deploy, add custom third-party drivers to existing image profiles and export to ISO or bundle, and perform upgrades. For basic concepts related to the way VMware vSphere software is created, packaged, and distributed, see Software Packaging Units That vSphere Lifecycle Manager Can Consume in the Managing Host and Cluster Lifecycle documentation.

Figure 4-2. Image Builder Architecture



You use vSphere ESXi Image Builder cmdlets for managing the software to deploy to your ESXi hosts in several different situations.

Table 4-8. Cases Where You Can Use vSphere ESXi Image Builder

Use Case for vSphere ESXi Image Builder	Description
Create image profiles for use by vSphere Auto Deploy	Use vSphere ESXi Image Builder to create an image profile that defines the VIBs that vSphere Auto Deploy uses to provision hosts.
Add custom third-party drivers to existing image profile and export to ISO or bundle	When you add a third-party driver or extension custom VIBs to your ESXi hosts, use vSphere ESXi Image Builder to clone the base image provided by VMware, add the custom VIBs, and export to ISO or to offline bundle ZIP file.
Perform upgrades	If you upgrade a system that includes custom extensions or drivers, you can use vSphere ESXi Image Builder to create a custom image profile that includes vSphere 8.0 compatible VIBs for the custom extensions. Export the custom image profile to an ISO or to a ZIP to upgrade your system by using vSphere Lifecycle Manager baselines.

The vSphere ESXi Image Builder cmdlets take image profiles and VIBs as input and produce various outputs.

Table 4-9. Input and Output to the vSphere ESXi Image Builder Cmdlets

Parameter	Description
Input	Image profiles and VIBs that are located in a software depot are used as input to PowerCLI cmdlets running on a Windows client.
Output	PowerCLI cmdlets create custom image profiles that can be exported to an ISO image or an offline depot ZIP file. ISO images are used for installation. The ZIP depot can be used by vSphere Lifecycle Manager or by `esxcli software` commands to update or install images. Image profiles are also used in vSphere Auto Deploy rules to customize the software to provision ESXi hosts with.

Watch the video "Using Image Builder CLI" for information about vSphere ESXi Image Builder:

 (Using Image Builder CLI)

Image Profiles

Image profiles define the set of VIBs that an ESXi installation or update process uses. Image profiles apply to ESXi hosts provisioned with vSphere Auto Deploy. You define and manipulate image profiles with vSphere ESXi Image Builder.

Image Profile Requirements

You can create a custom image profile from scratch or clone an existing profile and add or remove VIBs. A profile must meet the following requirements to be valid.

- Each image profile must have a unique name and vendor combination.
- Each image profile has an acceptance level. When you add a VIB to an image profile with an vSphere ESXi Image Builder cmdlet, Image Builder checks that the VIB matches the acceptance level defined for the profile.
- You cannot remove VIBs that are required by other VIBs.
- You cannot include two versions of the same VIB in an image profile. When you add a new version of a VIB, the new version replaces the existing version of the VIB.

Image Profile Validation

An image profile and its VIBs must meet several criteria to be valid.

- Image profiles must contain at least one base VIB and one bootable kernel module.
- If any VIB in the image profile depends on another VIB, that other VIB must also be included in the image profile. VIB creators store that information in the SoftwarePackage object's Depends property.
- VIBs must not conflict with each other. VIB creators store conflict information in the SoftwarePackage object's Conflicts property.
- Two VIBs with the same name, but two different versions, cannot coexist. When you add a new version of a VIB, the new version replaces the existing version of the VIB.
- No acceptance level validation issues exist.

When you make a change to an image profile, vSphere ESXi Image Builder checks that the change does not invalidate the profile.

Dependency Validation

When you add or remove a VIB, vSphere ESXi Image Builder checks that package dependencies are met. Each SoftwarePackage object includes a Depends property that specifies a list of other VIBs that VIB depends on. See Structure of ImageProfile, SoftwarePackage, and ImageProfileDiff Objects

Acceptance Level Validation

vSphere ESXi Image Builder performs acceptance level validation each time an image profile is created or changed. vSphere ESXi Image Builder checks the acceptance level of VIBs in the image profile against the minimum allowed acceptance level of the profile. The acceptance level of the VIB is also validated each time the signature of a VIB is validated.

VIB Validation During Export

When you export an image profile to an ISO, vSphere ESXi Image Builder validates each VIB by performing the following actions.

- Checks that no conflicts exist by checking the Conflicts property of each SoftwarePackage object.
- Performs VIB signature validation. Signature validation prevents unauthorized modification of VIB packages. The signature is a cryptographic checksum that guarantees that a VIB was produced by its author. Signature validation also happens during installation of VIBs on an ESXi host and when the vSphere Auto Deploy server uses VIBs.
- Checks that VIBs follow file path usage rules. VMware tests VMwareCertified and VMwareAccepted VIBs to guarantee those VIBs always follow file path usage rules.

Working with Acceptance Levels

Hosts, image profiles, and individual VIBs have acceptance levels. VIB acceptance levels show how the VIB was tested. Understanding what each acceptance level implies, how to change levels, and what a change implies is an important part of installation and update procedures.

Acceptance levels are set for hosts, image profiles, and individual VIBs. The default acceptance level for an ESXi image or image profile is PartnerSupported.

Host acceptance levels

The host acceptance level determines which VIBs you can install on a host. You can change a host's acceptance level with ESXCLI commands. By default, ESXi hosts have an acceptance level of PartnerSupported to allow for easy updates with PartnerSupported VIBs.

Note VMware supports hosts at the PartnerSupported acceptance level. For problems with individual VIBs with PartnerSupported acceptance level, contact your partner's support organization.

Image profile acceptance levels

The image profile acceptance level is set to the lowest VIB acceptance level in the image profile. If you want to add a VIB with a low acceptance level to an image profile, you can change the image profile acceptance level with the `Set-EsxImageProfile` cmdlet. See Set the Image Profile Acceptance Level.

The vSphere Lifecycle Manager does not display the actual acceptance level. Use vSphere ESXi Image Builder cmdlets to retrieve the acceptance level information for VIBs and image profiles.

VIB acceptance levels

A VIB's acceptance level is set when the VIB is created. Only the VIB creator can set the acceptance level.

If you attempt to provision a host with an image profile or VIB that has a lower acceptance level than the host, an error occurs. Change the acceptance level of the host to install the image profile or VIB. See Change the Host Acceptance Level. Changing the acceptance level of the host changes the support level for that host.

The acceptance level of a host, image profile, or VIB lets you determine who tested the VIB and who supports the VIB. 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.

Change the Host Acceptance Level

You can lower the host acceptance level to match the acceptance level for a VIB or image profile you want to install.

The acceptance level of each VIB on a host must be at least as high as the acceptance level of the host. For example, you cannot install a VIB with PartnerSupported acceptance level on a host with VMwareAccepted acceptance level. You must first lower the acceptance level of the host. For more information on acceptance levels, see Working with Acceptance Levels.

Warning Changing the host acceptance level to CommunitySupported affects the supportability of your host and might affect the security of your host.

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

2 View the host acceptance level.

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

3 Change the acceptance level of the host.

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

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

Note If the host has a higher acceptance level than the VIB or image profile you want to add, you can run commands in the `esxcli software vib` or `esxcli software profile` namespace with the `--force` option. When you use the `--force` option, a warning appears because you enforce a VIB or image profile with lower acceptance level than the acceptance level of the host and your setup is no longer consistent. The warning is repeated when you install VIBs, remove VIBs, or perform certain other operations on the host that has inconsistent acceptance levels.

Set the Image Profile Acceptance Level

If you want to add a VIB to an image profile, and the acceptance level of the VIB is lower than that of the image profile, you can clone the image profile with a lower acceptance level or change the image profile's acceptance level.

You can specify VMwareCertified, VMwareAccepted, PartnerSupported, or CommunitySupported as an acceptance level of an image profile. If you lower the acceptance level, the level of support for the image profile and hosts that you provision with it changes. For more information, see Working with Acceptance Levels.

Prerequisites

Install PowerCLI and all prerequisite software. See Configure vSphere ESXi Image Builder .

Procedure

1 In a PowerCLI session, run the `Add-EsxSoftwareDepot` cmdlet for each depot you want to work with.

Option	Action
Remote depot	Run `Add-EsxSoftwareDepot -DepotUrl <depot_url>`.
ZIP file	a Download the ZIP file to a local file system. b Run `Add-EsxSoftwareDepot -DepotUrl C:\<file_path>\<offline-bundle>.zip`

The cmdlet returns one or more `SoftwareDepot` objects.

2 Get the acceptance level for the image profile.

```
Get-EsxImageProfile -Name string
```

3 Set the acceptance level of the image profile.

```
Set-EsxImageProfile -Name string -AcceptanceLevel level
```

Structure of ImageProfile, SoftwarePackage, and ImageProfileDiff Objects

Knowing the structure of `ImageProfile`, `SoftwarePackage`, and `ImageProfileDiff` objects helps you manage deployment and upgrade processes.

ImageProfile Object Properties

The `ImageProfile` object, which is accessible with the `Get-EsxImageProfile` PowerCLI cmdlet, has the following properties.

Name	Type	Description
`AcceptanceLevel`	`AcceptanceLevel`	Determines which VIBs you can add to the profile. Levels are VMwareCertified, VMwareAccepted, PartnerSupported, and CommunitySupported. See Working with Acceptance Levels.
`Author`	`System.String`	The person who created the profile. 60 characters or fewer.
`CreationTime`	`System.DateTime`	The timestamp of creation time.
`Description`	`System.String`	The full text description of profile. No length limit.
`GUID`	`System.String`	Globally unique ID of the image profile.
`ModifiedTime`	`System.DateTime`	The timestamp of last modification time.
`Name`	`System.String`	The name of the image profile. 80 characters or fewer.
`ReadOnly`	`System.Boolean`	When set to `true`, the profile cannot be edited. Use `Set-EsxImageProfile -Readonly` to make your custom image profiles read-only.
`Rules`	`ImageProfileRule[]`	Any OEM hardware requirements and restrictions that the image profile might have. vSphere Auto Deploy verifies the value of this property when deploying an image profile and deploys the profile if matching hardware is available.
`Vendor`	`System.String`	The organization that publishes the profile. 40 characters or fewer.
`VibList`	`SoftwarePackage[]`	The list of VIB IDs the image contains.

SoftwarePackage Object Properties

When preparing an image profile, you can examine software packages to decide which packages are suitable for inclusion. The `SoftwarePackage` object has the following properties.

Name	Type	Description
`AcceptanceLevel`	`AcceptanceLevel`	The acceptance level of this VIB.
`Conflicts`	`SoftwareConstraint[]`	A list of VIBs that cannot be installed at the same time as this VIB. Each constraint uses the following format: `package-name[<<|<=|=|>=|>> version]`
`Depends`	`SoftwareConstraint[]`	A list of VIBs that must be installed at the same time as this VIB. Same constraint format as `Conflicts` property.
`Description`	`System.String`	The long description of the VIB.
`Guid`	`System.String`	The unique ID for the VIB.
`LiveInstallOk`	`System.Boolean`	True if live installs of this VIB are supported.
`LiveRemoveOk`	`System.Boolean`	True if live removals of this VIB are supported.
`MaintenanceMode`	`System.Boolean`	True if hosts must be in maintenance mode for installation of this VIB.
`Name`	`System.String`	The name of the VIB. Usually uniquely describes the package on a running ESXi system.
`Provides`	`SoftwareProvides`	The list of virtual packages or interfaces this VIB provides. See SoftwareProvide Object Properties.
`ReferenceURLs`	`SupportReference[]`	The list of `SupportReference` objects with in-depth support information. The `SupportReference` object has two properties, `Title` and `URL`, both of type `System.String`.
`Replaces`	`SoftwareConstraint[]`	The list of `SoftwareConstraint` objects that identify VIBs that replace this VIB or make it obsolete. VIBs automatically replace VIBs with the same name but lower versions.
`ReleaseDate`	`System.DateTime`	Date and time of VIB publication or release.
`SourceUrls`	`System.String[]`	The list of source URLs from which this VIB can be downloaded.

Name	Type	Description
StatelessReady	System.Boolean	True if the package supports host profiles or other technologies that make it suitable for use in conjunction with vSphere Auto Deploy.
Summary	System.String	A one-line summary of the VIB.
Tags	System.String[]	An array of string tags for this package defined by the vendor or publisher. Tags can be used to identify characteristics of a package.
Vendor	System.String	The VIB vendor or publisher.
Version	System.String	The VIB version.
VersionObject	Software.Version	The VersionObject property is of type SoftwareVersion. The SoftwareVersion class implements a static Compare method to compare two versions of strings. See SoftwareVersion Object Properties

ImageProfileDiff Object Properties

When you run the Compare-EsxImageProfile cmdlet, you pass in two parameters, first the reference profile, and then the comparison profile. The cmdlet returns an ImageProfileDiff object, which has the following properties.

Name	Type	Description
CompAcceptanceLevel	System.String	The acceptance level for the second profile that you passed to Compare-EsxImageProfile.
DowngradeFromRef	System.String[]	The list of VIBs in the second profile that are downgrades from VIBs in the first profile.
Equal	System.Boolean	True if the two image profiles have identical packages and acceptance levels.
OnlyInComp	System.String	The list of VIBs found only in the second profile that you passed to Compare-EsxImageProfile.
OnlyInRef	System.String[]	The list of VIBs found only in the first profile that you passed to Compare-EsxImageProfile.
PackagesEqual	System.Boolean	True if the image profiles have identical sets of VIB packages.

Name	Type	Description
RefAcceptanceLevel	System.String	The acceptance level for the first profile that you passed to Compare-EsxImageProfile.
UpgradeFromRef	System.String[]	The list of VIBs in the second profile that are upgrades from VIBs in the first profile.

SoftwareVersion Object Properties

The SoftwareVersion object lets you compare two version strings. The object includes a Comparestatic method that accepts two strings as input and returns 1 if the first version string is a higher number than the second version string. Compare returns 0 if two versions strings are equal. Compare returns –1 if the second version string is a higher number than the first string. The object has the following properties.

Name	Type	Description
Version	System.String	The part of the version before the hyphen. This part indicates the primary version.
Release	System.String	The part of the version after the hyphen. This part indicates the release version.

SoftwareConstraint Object Properties

The SoftwareConstraint object implements a MatchesProvide method. The method accepts a SoftwareProvides or SoftwarePackage object as input and returns True if the constraint matches the SoftwareProvide or the SoftwarePackage, or returns False otherwise.

The SoftwareConstraint object includes the following properties.

Name	Type	Description
Name	System.String	The name of the constraint. This name should match a corresponding SoftwareProvide Name property.
Relation	System.String	An enum, or one of the following comparison indicators: <<, <=, = >=, >>. This property can be $null if the constraint does not have a Relation and Version property.

Name	Type	Description
`Version`	`System.String`	The version to match the constraint against. This property can be $null if the constraint does not have a `Relation` and `Version` property.
`VersionObject`	`SoftwareVersion`	The version represented by a `SoftwareVersion` object.

SoftwareProvide Object Properties

The `SoftwareProvide` object includes the following properties.

Name	Type	Description
`Name`	`System.String`	The name of the provide.
`Version`	`System.String`	The version of the provide. Can be $null if the provide does not specify a version.
`Release`	`System.String`	The version of the provide as represented by a SoftwareVersion object. See SoftwareVersion Object Properties.

Configure vSphere ESXi Image Builder

Before you can run vSphere ESXi Image Builder cmdlets, you must install PowerCLI and all prerequisite software.

Prerequisites

Before you can run vSphere ESXi Image Builder cmdlets, you must install PowerCLI and all prerequisite software. The vSphere ESXi Image Builder snap-in is included with the PowerCLI installation. If you want to manage vSphere ESXi Image Builder 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 Microsoft Windows system. See thePowerCLI User's Guide.

Procedure

1. Open PowerShell on your workstation.
2. Download a version of PowerCLI later than PowerCLI 6.5R1 from the PowerCLI home page.
3. To install all PowerCLI modules, run the command: `Install-Module VMware.PowerCLI -Scope CurrentUser`. Alternatively, you can install individual PowerCLI modules by running the `Install-Module` cmdlet with the module name.If you see a warning that you are installing modules from an untrusted repository, press **y** and then press **Enter** to confirm the installation.

   You can verify that the PowerCLI module is available by using the command

```
Get-Module -Name VMware.PowerCLI* -ListAvailable.
```

What to do next

Review Using VMware.Image Builder Cmdlets . If you are new to PowerCLI, read the *PowerCLIUser's Guide*.

Use vSphere ESXi Image Builder cmdlets and other PowerCLI cmdlets and PowerShell cmdlets to manage image profiles and VIBs. Use `Get-Help` *cmdlet_name* at any time for command-line help.

Configure the vSphere ESXi Image Builder

Before you can use vSphere ESXi Image Builder with the vSphere Client, you must verify that the service is enabled and running.

Prerequisites

- 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.

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 Image Builder** to activate the service.

   The **Software Depots** tab appears.

What to do next

- Add a Software Depot.
- Import a Software Depot.
- Clone an Image Profile.
- Create an Image Profile.
- Prepare Your System for vSphere Auto Deploy.
- 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.

Using VMware.Image Builder Cmdlets

With VMware.Image Builder cmdlets, you can take advantage of all PowerCLI features.

VMware.Image Builder cmdlets are implemented as Microsoft PowerShell cmdlets and included in PowerCLI. You can take advantage of all PowerCLI features by using VMware.Image Builder cmdlets. Experienced PowerShell users can use VMware.Image Builder cmdlets just like other PowerShell cmdlets. If you are new to PowerShell and PowerCLI, follow these tips.

You can type cmdlets, parameters, and parameter values in the 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`. See `Get-Help Format-List`.
- Use wildcards for searching and filtering VIBs and image profiles. All wildcard expressions are supported.

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.

```
Add-EsxSoftwarePackage -ImageProfile profile42 -SoftwarePackage "partner package 35"
```

Passing Parameters as Objects

You can pass parameters as objects if you want to do scripting and automation. You can use the technique with cmdlets that return multiple objects or with cmdlets that return a single object.

1 Bind the output of a cmdlet that returns multiple objects to a variable.

```
$profs = Get-EsxImageProfile
```

2 When you run the cmdlet that needs the object as input, access the object by position, with the list starting with 0.

```
Add-EsxSoftwarePackage -ImageProfile $profs[4] -SoftwarePackage partner-pkg
```

The example adds the specified software package to the fifth image profile in the list returned by `Get-EsxImageProfile`.

Most of the examples in the *vCenter Server Installation and Setup* documentation pass in parameters by name. vSphere ESXi Image Builder Workflows with PowerCLI Cmdlets includes examples that pass parameters as objects.

VMware.ImageBuilder Cmdlets Overview

The VMware.Image Builder component of VMware PowerCLI provides cmdlets for managing VIBs, image profiles, and other content in software depots.

vSphere 7.0 and later introduce new ways of packaging VIBs along with legacy bulletins and patches, and software depots contain base images, vendor addons and components, along with VIBs and image profiles. VMware PowerCLI 12.0 and later provide cmdlets that work with the new content in software depots.

VMware.ImageBuilder includes the following cmdlets.

Note When you run VMware.ImageBuilder cmdlets, provide all parameters on the command line when you invoke the cmdlet. Supplying parameters in interactive mode is not recommended.

Run `Get-Help` *cmdlet_name* at the PowerCLI prompt for detailed reference information.

Table 4-10. VMware.ImageBuilder Cmdlets Used with Legacy Content in Software Depots

Cmdlet	Description
`Add-EsxSoftwareDepot`	Adds the software depot or ZIP file at the specified location to your current environment. Downloads metadata from the depot and analyzes VIBs for dependencies.
`Remove-EsxSoftwareDepot`	Disconnects from the specified software depot.
`Get-EsxSoftwareDepot`	Returns a list of software depots that are in the current environment. If you want to examine and manage image profiles and VIBs, you must first add the corresponding software depot to your environment.
`Get-EsxSoftwarePackage`	Returns a list of software package objects (VIBs). Use this cmdlet's options to filter the results.
`Get-EsxImageProfile`	Returns an array of `ImageProfile` objects from all currently added depots.
`New-EsxImageProfile`	Creates a new image profile. In most cases, creating a new profile by cloning an existing profile is recommended. See Clone an Image Profile with PowerCLI Cmdlets.
`Set-EsxImageProfile`	Modifies a local `ImageProfile` object and performs validation tests on the modified profile. The cmdlet returns the modified object but does not persist it.
`Export-EsxImageProfile`	Exports an image profile as either an ESXi ISO image for ESXi installation, or as a ZIP file.
`Compare-EsxImageProfile`	Returns an `ImageProfileDiff` structure that shows whether the two profiles have the same VIB list and acceptance level. See Working with Acceptance Levels.
`Remove-EsxImageProfile`	Removes the image profile from the software depot.
`Add-EsxSoftwarePackage`	Adds one or more new packages (VIBs) to an existing image profile.
`Remove-EsxSoftwarePackage`	Removes one or more packages (VIBs) from an image profile.
`Set-ESXImageProfileAssociation`	Associates the specified image profile with the specified ESXi system.

Table 4-11. VMware.ImageBuilder Cmdlets Used with New Content in Software Depots

Cmdlet	Description
`Get-DepotAddons`	Retrieves an array of objects that provide basic information about addons in a software depot.
`Get-DepotBaseImages`	Retrieves an array of objects that provide basic information about base images in a software depot.
`Get-DepotComponents`	Retrieves an array of objects that provide basic information about components in a software depot.
`Get-DepotInfo`	Retrieves basic information about the software depot located at the specified file path or URL address.
`Get-DepotVibs`	Retrieves an array of objects that provide basic information about VIBs in a software depot.
`New-IsoImage`	Generates an ISO image by using the specified software depot and software specification at the specified file path.
`New-PxeImage`	Generates a PXE image by using the specified software depot and software specification at the specified file path.

ESXi Image Profile Tasks

Manipulate software depots, image profiles, and VIBs by using either the VMware.Image Builder component of VMware PowerCLI or the vSphere Client.

Add a Software Depot

Add one or more software depots to the vSphere ESXi Image Builder inventory to customize image profiles.

Prerequisites

Before you can work with software depots and customize image profiles, you must add one or more software depots to the vSphere ESXi Image Builder inventory. You can add a software depot by using the vSphere Client.

Verify that the vSphere ESXi Image Builder service is enabled and running. See Configure the vSphere ESXi Image Builder.

Procedure

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

   By default, only the administrator role has privileges to use the vSphere ESXi Image Builder service.

2 On the **Software Depots** tab, click **New**.

   The **Add Software Depot** window appears.

3 Select the type of depot that you want to create.

Option	Action
Online Depot	a Enter a name of the depot in the inventory. b Enter the URL of the online depot.
Custom Depot	Enter the name of the depot in the inventory.

The VMware online software depot is located at https://hostupdate.vmware.com/software/VUM/PRODUCTION/main/vmw-depot-index.xml

4 Click **Add**.

5 (Optional) Click the **Software Packages** tab to view the contents of the selected depot and additional information about the packages.

6 (Optional) If you added an **Online depot**, you can also:

- **Check for Updates** to get the latest depot packages.
- Click **More info** to get additional depot details.

Results

The software depot is added to the list.

What to do next

- You can associate an image profile with a new vSphere Auto Deploy rule to provision ESXi hosts. See Create a Deploy Rule or Clone a Deploy Rule.
- You can associate an image profile with an ESXi host. See Add a Host to the vSphere Auto Deploy Inventory.
- Edit the Image Profile Association of a Host.
- **Remove** a custom software depot.

Import a Software Depot

If an offline depot is located on your local file system, import the ZIP file to the vSphere ESXi Image Builder inventory.

Prerequisites

If an offline depot is located on your local file system, you can import the ZIP file to the vSphere ESXi Image Builder inventory by using the vSphere Client.

Verify that the vSphere ESXi Image Builder service is enabled and running. See Configure the vSphere ESXi Image Builder.

Procedure

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

   By default, only the administrator role has privileges to use the vSphere ESXi Image Builder service.
2. On the **Software Depots** tab, click **Import** .
3. Enter the name of the software depot in the inventory.
4. Click **Browse** and select a ZIP file from the local system, that contains the software depot you want to import.
5. Click **Upload**.

What to do next

- You can associate an image profile with a new vSphere Auto Deploy rule to provision ESXi hosts. See Create a Deploy Rule or Clone a Deploy Rule.
- You can associate an image profile with an ESXi host. See Add a Host to the vSphere Auto Deploy Inventory.
- Edit the Image Profile Association of a Host.

Clone an Image Profile

Use the vSphere Client to clone image profiles.

Prerequisites

You can use the vSphere Client to clone image profiles. You can clone an image profile when you want to make small changes to the VIB list in a profile, or if you want to use hosts from different vendors and want to use the same basic profile, but want to add vendor-specific VIBs.

- Verify that the vSphere ESXi Image Builder service is enabled and running. See Configure the vSphere ESXi Image Builder.
- Add or import a software depot to the vSphere ESXi Image Builder inventory. See Add a Software Depot and Import a Software Depot.
- Verify that there is at least one custom depot in the vSphere ESXi Image Builder inventory.

Procedure

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

   By default, only the administrator role has privileges to use the vSphere ESXi Image Builder service.
2. On the **Software Depots** tab, use the drop-down menu to select the software depot that contains the image profile that you want to work with.
3. From the list of image profiles in the depot, select the image profile that you want to clone and click **Clone**.

4 Enter an image profile name, vendor, and description.

   You must enter a unique image profile name.

5 From the **Software depot** drop-down menu, select in which custom depot to add the new image profile.

6 Click **Next**.

   The **Select software packages** page appears.

7 From the drop-down menu, select an acceptance level for the image profile.

   The acceptance level of the VIBs you add to the base image must be at least as high as the level of the base image. If you add a VIB with a lower acceptance level to the image profile, you must lower the image profile acceptance level. For more information, see Working with Acceptance Levels.

8 Select the VIBs that you want to add to the image profile and deselect the ones that you want to remove, and click **Next**.

   Note The image profile must contain a bootable ESXi image to be valid.

   vSphere ESXi Image Builder verifies that the change does not invalidate the profile. Some VIBs depend on other VIBs and become invalid if you include them in an image profile separately. When you add or remove a VIB, vSphere ESXi Image Builder checks whether the package dependencies are met.

9 On the **Ready to complete** page, review the summary information for the new image profile and click **Finish**.

What to do next

- You can associate an image profile with a new vSphere Auto Deploy rule to provision ESXi hosts. See Create a Deploy Rule or Clone a Deploy Rule.
- You can associate an image profile with an ESXi host. See Add a Host to the vSphere Auto Deploy Inventory.
- Edit the Image Profile Association of a Host.

Clone an Image Profile with PowerCLI Cmdlets

Cloning a published profile is the easiest way to create a custom image profile. Cloning a profile is especially useful if you want to remove a few VIBs from a profile, or if you want to use hosts from different vendors and want to use the same basic profile, but want to add vendor-specific VIBs. VMware partners or large installations might consider creating a new profile.

Prerequisites

- Install the PowerCLI and all prerequisite software. See Configure vSphere ESXi Image Builder .
- Verify that you have access to the software depot that contains the image profile you want to clone.

Procedure

1 In a PowerCLI session, run the `Add-EsxSoftwareDepot` cmdlet for each depot you want to work with.

Option	Action
Remote depot	Run `Add-EsxSoftwareDepot -DepotUrl <depot_url>`.
ZIP file	a Download the ZIP file to a local file system. b Run `Add-EsxSoftwareDepot -DepotUrl C:\<file_path>\<offline-bundle>.zip`

The cmdlet returns one or more `SoftwareDepot` objects.

2 (Optional) Run the `Get-EsxImageProfile` cmdlet to find the name of the profile that you want to clone.

You can use filtering options with `Get-EsxImageProfile`.

3 Run the `New-EsxImageProfile` cmdlet to create the new profile and use the `-CloneProfile` parameter to specify the profile you want to clone.

```
New-EsxImageProfile -CloneProfile My_Profile -Name "Test Profile 42"
```

This example clones the profile named *My_Profile* and assigns it the name Test Profile 42. You must specify a unique combination of name and vendor for the cloned profile.

What to do next

See Examine Depot Contents for some examples of filtering.

Customize the image profile by adding or removing VIBs. See Add VIBs to an Image Profile with PowerCLI Cmdlets.

Create an Image Profile

You can create a new image profile by using the vSphere Client instead of cloning an existing one.

Prerequisites

You might consider creating a new image profile if it differs significantly from the image profiles in your inventory.

- Verify that the vSphere ESXi Image Builder service is enabled and running. See Configure the vSphere ESXi Image Builder.
- Add or import a software depot to the vSphere ESXi Image Builder inventory. See Add a Software Depot and Import a Software Depot.
- Verify that there is at least one custom depot in the vSphere ESXi Image Builder inventory.

Procedure

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

   By default, only the administrator role has privileges to use the vSphere ESXi Image Builder service.

2 From the **Software depot** drop-down menu, select in which custom depot to add the new image profile.

3 On the Image Profiles tab, click **New Image Profile**.

4 Enter an image profile name, vendor, and description.

   You must enter a unique image profile name.

5 Click **Next**.

   The **Select software packages** page appears.

6 From the drop-down menu, select an acceptance level for the image profile.

   The acceptance level of the VIBs you add to the base image must be at least as high as the level of the base image. If you add a VIB with a lower acceptance level to the image profile, you must lower the image profile acceptance level. For more information, see Working with Acceptance Levels.

7 Select the VIBs that you want to add to the image profile and deselect the ones that you want to remove, and click **Next**.

   Note The image profile must contain a bootable ESXi image to be valid.

   vSphere ESXi Image Builder verifies that the change does not invalidate the profile. Some VIBs depend on other VIBs and become invalid if you include them in an image profile separately. When you add or remove a VIB, vSphere ESXi Image Builder checks whether the package dependencies are met.

8 On the **Ready to complete** page, review the summary information for the new image profile and click **Finish**.

What to do next

- You can associate an image profile with a new vSphere Auto Deploy rule to provision ESXi hosts. See Create a Deploy Rule or Clone a Deploy Rule.
- You can associate an image profile with an ESXi host. See Add a Host to the vSphere Auto Deploy Inventory.
- Edit the Image Profile Association of a Host.
- Select and **Delete** an Image profile.
- **View Software Packages** for the selected image profile.

Create a Custom ESXi ISO Image with PowerCLI Cmdlets

With ESXi Image Builder, you can customize an ESXi image profile, but not combine content from different depots to generate an ISO image. Starting with VMware PowerCLI 12.0, you can customize ISO images by using content from multiple software depots and a custom software specification.

The `New-IsoImage` cmdlet preserves additional metadata required by the vSphere Lifecycle Manager, such as base image, addon and component. This additional metadata is not part of ISO images that you can export by using the legacy ESXi Image Builder cmdlets.

Prerequisites

Install VMware PowerCLI 12.0 or later.

Verify that you have access to the software depot that contains the software specification that you want to use.

Procedure

1 Gather the required information for the software specification that you use to create a custom ISO image.

   a Get the base image version for the required patch or upgrade by running the `Get-DepotBaseImages` cmdlet:

   ```
   PS C:\> Get-DepotBaseImages -Depot C:\VMware-ESXi-8.xxx-xxxxxxxx-depot.zip
   ```

   The command output is:

   ```
   Version                                 Vendor
   Release date
   -------                                 ------
   ------------
   8.0.0-0.0.xxxxxx                      VMware, Inc.
   01/01/20xx 00:00:00
   ```

   b Get other packages, such as OEM addons, with cmdlets used with new metadata in software depots. For example:

   ```
   PS C:\> Get-DepotAddons -Depot C:\addon-depot.zip
   ```

   The command output is:

   ```
   Name                    Version                 ID
   Vendor                  Release date
   ----                    -------                 --
   ------                  ------------
   testaddonv1             1.0.0-1                 testaddonv1:1.0.0-1     ESXLifecycle
   QE          02/20/20xx 18:28:23
   ```

You can also list all components in a sofware depot with the `Get-DepotComponents` cmdlet:

```
PS C:\> Get-DepotComponents -Depot C:\Intel-i40en_1.12.3.0-1OEM.xxxxxxx.zip
```

The command output is:

```
Name                              Version
ID                                                  Vendor
----                              -------
--                                                 ------
Intel-i40en                       1.12.3.0-1OEM.xxxxxxx   Intel-
i40en:1.12.3.0-1OEM.xxxxxxx     Intel
```

You can use any number and combination of online and offline software depots.

2 Create a software specification. For example:

```
{
    "base_image": {
        "version": "8.0.0-0.0.xxxxxxx"
    },
    "add_on": {
        "name": "testaddonv1",
        "version": "1.0.0-1"
    },
    "components": {
       "Intel-i40en": "1.12.3.0-1OEM.xxxxxxx"
   }
}
```

The software specification is a JSON file that contains information about the ESXi base image and additional packages, such as a vendor add-on.

3 Generate a custom ISO image by running the `New-IsoImage` cmdlet with the parameters `Depots`,`SoftwareSpec` and `Destination`. For example:

```
New-IsoImage -Depots “c:\temp\VMware-ESXi-8.0-xxxxxxx-depot.zip” , “c:\temp\HPE-xxxxxxx-
Jan20xx-Synergy-Addon-depot.zip” -SoftwareSpec “c:\temp\HPE-80xx-custom.JSON” -Destination
“c:\temp\HPE-80xx-custom.iso”
```

The depot(s) include the path to the zip files for the supported ESXi version and vendor add-on. The destination include the path and file name for the custom ISO file.

You can pass additional kernel options, create a live image, overwrite existing files, or check acceptance levels for individual VIBs used during the creation of the image. For more information about the `New-IsoImage` cmdlet, see https://code.vmware.com/docs/11794/cmdletreference/doc/New-IsoImage.html.

What to do next

You can import the new ISO image to the vSphere Lifecycle Manager depot, so that you can create upgrade baselines, which you use for host upgrade operations.

Create a Custom PXE Image with PowerCLI Cmdlets

Starting with VMware PowerCLI 12.0, you can create a custom PXE image by using any software depot and a custom software specification.

Prerequisites

Install VMware PowerCLI 12.0 or later.

Verify that you have access to the software depot that contains the software specification you want to use.

Procedure

1 Gather the required information for the software specification that you use to create a custom PXE image.

   a Get the base image version for the required patch or upgrade by running the `Get-DepotBaseImages` cmdlet:

   ```
   PS C:\> Get-DepotBaseImages -Depot C:\VMware-ESXi-8.xxxx-xxxxx-depot.zip
   ```

   The command output is:

   ```
   Version                                 Vendor
   Release date
   -------                                 ------
   ------------
   8.x.x.xxx.xxxxx                      VMware, Inc.
   04/29/20xx 00:00:00
   ```

   b Get other packages, such as OEM addons, with cmdlets used with new metadata in software depots. For example:

   ```
   PS C:\> Get-DepotAddons -Depot C:\addon-depot.zip
   ```

   The command output is:

   ```
   Name                    Version                 ID
   Vendor                  Release date
   ----                    -------                 --
   ------                  ------------
   testaddonv1             1.0.0-1                 testaddonv1:1.0.0-1     ESXLifecycle
   QE         02/20/20xx 18:28:23
   ```

   You can also list all components in a sofware depot with the `Get-DepotComponents` cmdlet:

   ```
   PS C:\> Get-DepotComponents -Depot C:\Intel-
   i40en_1.12.3.0-1OEM.700.1.0.15843807_18058526.zip
   ```

   The command output is:

   ```
   Name                              Version
   ID                                                  Vendor
   ```

```
----                                 --------
--                                                 ------
Intel-i40en                          1.12.3.0-1OEM.xxxxx    Intel-i40en:1.12.3.0-1OEM.xxxxx
Intel
```

You can use any number and combination of online and offline software depots.

2 Create a software specification. For example:

```
{
    "base_image": {
        "version": "8.0.xxxxx"
    },
    "add_on": {
        "name": "testaddonv1",
        "version": "1.0.0-1"
    },
    "components": {
       "Intel-i40en": "1.12.3.0-1OEM.xxxxx"
   }
}
```

The software specification is a JSON file that contains information about the ESXi base image and additional packages, such as a vendor add-on.

3 Generate a custom PXE image by running the `New-PxeImage` cmdlet with the parameters `Depots`,`SoftwareSpec` and `Destination`. For example:

```
New-PxeImage -Depots “c:\temp\VMware-ESXi-8.0xxxxx-xxxxx-depot.zip” ,
“c:\temp\HPE-8.0xxxxx-xxx-Synergy-Addon-depot.zip” -SoftwareSpec “c:\temp\HPE-xxx-
custom.JSON” -Destination “C:\pxe-image”
```

The depot(s) include the path to the zip files for the supported ESXi version and vendor add-on. The destination include the path and file name for the custom PXE file.

You can pass additional kernel options, create a live image, overwrite existing files, or check acceptance levels for individual VIBs used during the creation of the image. For more information about the `New-PxeImage` cmdlet, see https://code.vmware.com/docs/11794/cmdletreference/doc/New-PxeImage.html.

What to do next

You can use the PXE image in remediation workflows of PXE booted ESXi hosts.

Edit an Image Profile

You can edit image profiles by using the vSphere Client. You can change the name, details and VIB list of an image profile.

Prerequisites

- Verify that the vSphere ESXi Image Builder service is enabled and running. See Configure the vSphere ESXi Image Builder.

- Add or import a software depot to the vSphere ESXi Image Builder inventory. See Add a Software Depot and Import a Software Depot.
- Verify that there is at least one custom depot in the vSphere ESXi Image Builder inventory.

Procedure

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

By default, only the administrator role has privileges to use the vSphere ESXi Image Builder service.

2 On the **Software Depots** tab, use the drop-down menu to select the software depot that contains the image profile that you want to work with.

3 On the **Image Profiles** tab, select the image profile that you want to edit and click **Edit**.

The **Edit Image Profile** wizard appears.

4 (Optional) Change the name, vendor and description information of the image profile.

5 Click **Next**.

The **Select software packages** page appears.

6 From the drop-down menu, select an acceptance level for the image profile.

The acceptance level of the VIBs you add to the base image must be at least as high as the level of the base image. If you add a VIB with a lower acceptance level to the image profile, you must lower the image profile acceptance level. For more information, see Working with Acceptance Levels.

7 Select the VIBs that you want to add to the image profile and deselect the ones that you want to remove, and click **Next**.

Note The image profile must contain a bootable ESXi image to be valid.

vSphere ESXi Image Builder verifies that the change does not invalidate the profile. Some VIBs depend on other VIBs and become invalid if you include them in an image profile separately. When you add or remove a VIB, vSphere ESXi Image Builder checks whether the package dependencies are met.

8 On the **Ready to complete** page, review the summary information for the edited image profile and click **Finish**.

What to do next

- You can associate an image profile with a new vSphere Auto Deploy rule to provision ESXi hosts. See Create a Deploy Rule or Clone a Deploy Rule.
- You can associate an image profile with an ESXi host. See Add a Host to the vSphere Auto Deploy Inventory.
- Edit the Image Profile Association of a Host.

Add VIBs to an Image Profile with PowerCLI Cmdlets

You can add one or more VIBs to an image profile if that image profile is not set to read only. If the new VIB depends on other VIBs or conflicts with other VIBs in the profile, a message is displayed at the PowerShell prompt and the VIB is not added.

You can add VIBs from VMware or from VMware partners to an image profile. If you add VMware VIBs, vSphere ESXi Image Builder performs validation. If you add VIBs from two or more OEM partners simultaneously, no errors are reported but the resulting image profile might not work. Install VIBs from only one OEM vendor at a time.

If an error about acceptance level problems appears, change the acceptance level of the image profile and the acceptance level of the host. Consider carefully whether changing the host acceptance level is appropriate. VIB acceptance levels are set during VIB creation and cannot be changed.

You can add VIBs even if the resulting image profile is invalid.

Note VMware can support only environments and configurations that are proven to be stable and fully functional through rigorous and extensive testing. Use only those supported configurations. You can use custom VIBs if you lower your host acceptance level, and as a result, supportability. In that case, track the changes you made, so you can revert them if you want to remove custom VIBs and restore the host acceptance level to the default (Partner Supporter) later. See Working with Acceptance Levels.

Prerequisites

Install the PowerCLI and all prerequisite software. See Configure vSphere ESXi Image Builder

Procedure

1 In a PowerCLI session, run the `Add-EsxSoftwareDepot` cmdlet for each depot you want to work with.

Option	Action
Remote depot	Run `Add-EsxSoftwareDepot -DepotUrl <depot_url>`.
ZIP file	a Download the ZIP file to a local file system. b Run `Add-EsxSoftwareDepot -DepotUrl C:\<file_path>\<offline-bundle>.zip`

The cmdlet returns one or more `SoftwareDepot` objects.

2 Run the `Get-EsxImageProfile` cmdlet to list all image profiles in all currently visible depots.

The cmdlet returns all available profiles. You can narrow your search by using the optional arguments to filter the output.

3 Clone the profile.

```
New-EsxImageProfile -CloneProfile My_Profile -Name "Test Profile 42" -Vendor "My Vendor"
```

Image profiles published by VMware and its partners are read only. To make changes, you must clone the image profile. The `vendor` parameter is required.

4 Run the `Add-EsxSoftwarePackage` cmdlet to add a new package to one of the image profiles.

```
Add-EsxSoftwarePackage -ImageProfile My_Profile -SoftwarePackage partner-package
```

The cmdlet runs the standard validation tests on the image profile. If validation succeeds, the cmdlet returns a modified, validated image profile. If the VIB that you want to add depends on a different VIB, the cmdlet displays that information and includes the VIB that can resolve the dependency. If the acceptance level of the VIB that you want to add is lower than the image profile acceptance level, an error occurs.

Compare Image Profiles

You can compare two image profiles by using the vSphere Client to check if they have the same VIB list, version, or acceptance level.

Prerequisites

- Verify that the vSphere ESXi Image Builder service is enabled and running. See Configure the vSphere ESXi Image Builder.
- Add or import a software depot to the vSphere ESXi Image Builder inventory. See Add a Software Depot and Import a Software Depot.

Procedure

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

By default, only the administrator role has privileges to use the vSphere ESXi Image Builder service.

2 On the **Software Depots** tab, use the drop-down menu to select the software depot that contains the image profile that you want to work with.

3 On the **Image Profiles** tab, select an image profile and click **Compare To**.

The **Compare Image Profile** wizard appears.

4 Click **Change** to select a second image profile.

The **Select Image Profile** page appears.

5 Select a software depot from the drop-down menu and click on the second image profile.

6 In the **Compare Image Profile** page, select a comparison option from the **Software packages** drop-down menu.

The left side of the list displays details of the VIBs that the first chosen image profile contains. The right part of the list provides information about the second image profile. The VIBs marked as `Same` are identical in both profiles. VIBs that are present in one of the image profiles are marked as `Missing` next to the image profile that they are not present in.

Compare Image Profiles with PowerCLI Cmdlets

You can compare two image profiles by using the `Compare-EsxImageProfile` cmdlet, for example, to see if they have the same VIB list or acceptance level . Comparing image profiles or their properties is also possible by using the PowerShell comparison operators.

Prerequisites

Install the PowerCLI and all prerequisite software. See Configure vSphere ESXi Image Builder .

Procedure

1 In a PowerCLI session, run the `Add-EsxSoftwareDepot` cmdlet for each depot you want to work with.

Option	Action
Remote depot	Run `Add-EsxSoftwareDepot -DepotUrl <depot_url>`.
ZIP file	a Download the ZIP file to a local file system. b Run `Add-EsxSoftwareDepot -DepotUrl C:\<file_path>\<offline-bundle>.zip`

The cmdlet returns one or more `SoftwareDepot` objects.

2 (Optional) Run the `Get-EsxImageProfile` cmdlet to view a list of all image profiles in all available depots.

In the list, you can locate the names of the image profiles you want to compare.

3 Before comparing the image profiles, assign them to variables.

For example, you can create variables `$imageProfile1` and `$imageProfile2` to hold the names of the compared images profiles.

```
$imageProfile1
                = Get-EsxImageProfile -Name "ImageProfile1"
$imageProfile2
                = Get-EsxImageProfile -Name "ImageProfile2"
```

4 Compare the two image profiles by using the `Compare-EsxImageProfile` cmdlet or the `-eq` comparison operator, which returns a Boolean value.

- Compare the two image profiles to get a full description of the differences by using the `Compare-EsxImageProfile` cmdlet.

```
Compare-EsxImageProfile -ReferenceProfile
                    $imageProfile1 -ComparisonProfile $imageProfile2
```

- Compare the two image profiles by VIB list and acceptance level using the `-eq` comparison operator.

```
if ($imageProfile1 -eq $imageProfile2) {
   Write-host "Successfully verified that both image profiles are equal."
} else {
   Write-host "Failed to verify that the image profiles are equal."
}
```

- Compare the two image profiles by a specific property using the `-eq` comparison operator.

```
if ($imageProfile1.vendor -eq $imageProfile2.vendor) {
   Write-host "Successfully verified that both image profiles are equal."
} else {
   Write-host "Failed to verify that the image profiles are equal."
}
```

Move an Image Profile to a Different Software Depot

You can move image profiles between custom depots by using the vSphere Client and modify it without affecting the source depot's configuration.

Prerequisites

- Verify that the vSphere ESXi Image Builder service is enabled and running. See Configure the vSphere ESXi Image Builder.
- Add or import a software depot to the vSphere ESXi Image Builder inventory. See Add a Software Depot and Import a Software Depot.
- Verify that there is at least one custom depot in the vSphere ESXi Image Builder inventory.

Procedure

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

   By default, only the administrator role has privileges to use the vSphere ESXi Image Builder service.
2. On the **Software Depots** tab, use the drop-down menu to select the software depot that contains the image profile that you want to work with.
3. On the **Image Profiles** tab, select an image profile and click **Move to**.
4. From the drop-down menu, select the custom depot in which you want to move the image profile.
5. Click **OK**.

Results

The image profile is in the new software depot.

Export an Image Profile to ISO or Offline Bundle ZIP

You can export an image profile to an ISO image or a ZIP file by using the vSphere Client.

Prerequisites

You can export an image profile to an ISO image or a ZIP file by using the vSphere Client. You can use the ISO image as an ESXi installer or to upgrade hosts with vSphere Lifecycle Manager. The ZIP file contains metadata and the VIBs of the image profile. You can use it for ESXi upgrades or as an offline depot.

- Verify that the vSphere ESXi Image Builder service is enabled and running. See Configure the vSphere ESXi Image Builder.
- Add or import a software depot to the vSphere ESXi Image Builder inventory. See Add a Software Depot and Import a Software Depot.

Procedure

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

By default, only the administrator role has privileges to use the vSphere ESXi Image Builder service.

2 On the **Software Depots** tab, use the drop-down menu to select the software depot that contains the image profile that you want to work with.

3 On the **Image Profiles** tab, select the image profile that you want to export and click **Export**.

The **Export Image Profile** window appears.

4 Select the type of the exported file.

Option	Description
ISO	Exports the image profile to a bootable ISO image. If you want to create an ISO image that you can burn to a CD or DVD and use to start a stateless ESXi instance, select the **Do not include an installer on the ISO** check box.
ZIP	Exports the image profile to a ZIP file.

5 (Optional) If you want to bypass the acceptance level verification of the image profile, select **Skip acceptance level checking**.

6 Click **Ok**.

The **Download** link starts generating in the "Download Image Profiles" column of the selected image profile.

7 When the image generates successfully, click **Download** to save the exported file.

What to do next

- You can associate an image profile with a new vSphere Auto Deploy rule to provision ESXi hosts. See Create a Deploy Rule or Clone a Deploy Rule.

- You can associate an image profile with an ESXi host. See Add a Host to the vSphere Auto Deploy Inventory.
- Edit the Image Profile Association of a Host.

Export an Image Profile to an ISO or Offline Bundle ZIP with PowerCLI Cmdlets

You can export an image profile to an ISO image or a ZIP file of component files and folders. You cannot create both by running the cmdlet once. You can use the ISO image as an ESXi installer or upload the ISO into vSphere Lifecycle Manager for upgrades.

Prerequisites

Install the PowerCLI and all prerequisite software. See Configure vSphere ESXi Image Builder .

Procedure

1 In a PowerCLI session, run the `Add-EsxSoftwareDepot` cmdlet for each depot you want to work with.

Option	Action
Remote depot	Run `Add-EsxSoftwareDepot -DepotUrl <depot_url>`.
ZIP file	a Download the ZIP file to a local file system. b Run `Add-EsxSoftwareDepot -DepotUrl C:\<file_path>\<offline-bundle>.zip`

The cmdlet returns one or more `SoftwareDepot` objects.

2 Run `Export-EsxImageProfile` to export the image profile.

Export Format	Cmdlet
ISO images	`Export-EsxImageProfile` with the `-ExportToIso` parameter
Offline depot ZIP files	`Export-EsxImageProfile` with the `-ExportToBundle` parameter

Results

For the ISO image, vSphere ESXi Image Builder validates VIB signatures, adds VIB binaries to the image, and downloads the image to the specified location. For the ZIP file, vSphere ESXi Image Builder validates VIB signatures and downloads the VIB binaries to the specified location.

Example: Exporting an Image Profile

Follow these steps to export an image profile to an ISO image.

1 Add the software depot.

```
Add-EsxSoftwareDepot -DepotUrl url_or_file
```

2 View all available image profiles to find the name of the image profile to export.

```
Get-EsxImageProfile
```

3 Export the image profile.

```
Export-EsxImageProfile -ImageProfile "myprofile" -ExportToIso -FilePath iso_name
```

Follow these steps to export an image profile to a ZIP file of component files and folders.

1 Add the software depot.

```
Add-EsxSoftwareDepot -DepotUrl url_or_file
```

2 View all available image profiles to find the name of the image profile to export.

```
Get-EsxImageProfile
```

3 Export the image profile.

```
Export-EsxImageProfile -ImageProfile "myprofile" -ExportToBundle -FilePath C:\my_bundle.zip
```

What to do next

Use the ISO image in an ESXi installation or upload the ISO image into vSphere Lifecycle Manager to perform upgrades.

Use the ZIP file to upgrade an ESXi installation.

- Import the ZIP file into vSphere Lifecycle Manager for use with patch baselines.
- Download the ZIP file to an ESXi host or a datastore and run `esxcli software vib` commands to import the VIBs in the ZIP file.

See the *vSphere Upgrade* documentation.

Regenerate an Image Profile

If you use Auto Deploy to add stateful ESXi hosts to a cluster that you manage by using an image, all hosts must have the same software specification.

When an Auto Deploy rule is created and the target cluster is managed by an image, a Preboot Execution Environment (PXE) image is created based on the image specification of the cluster. The generated PXE image is cached and is not automatically updated. As a result, if you modify the image specification in the vSphere Lifecycle Manager, you must update the PXE image manually.

For information how to create a rule to add hosts to a cluster managed by an image, see *Use Auto Deploy to Add a Host to a Cluster Managed by an Image* from the *vSphere Lifecycle Manager* documentation.

Prerequisites

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 **Deploy Rules** tab, select the desired rule.

   You must select a rule, which matches ESXi hosts to a cluster that you manage by an image.
3. If the rule is active, first you must deactivate the rule.
   a. Click the **Activate/Deactive Rules** tab
   b. In the dialog box, select the rule.
   c. In the dialog box, select **Deactivate** and click **OK**.
4. Select **Reacreate Image Profile** and in the confirmation dialog box, click **Recreate**.
5. (Optional) Activate the rule again.
   a. Click the **Activate/Deactive Rules** tab
   b. In the dialog box, select the rule.
   c. In the dialog box, select **Activate** and click **OK**.

Results

The PXE image is running the latest image specification.

Preserve Image Profiles Across Sessions with PowerCLI

You can export the image profile to a ZIP file software depot and add that depot in the next session.

Prerequisites

When you create an image profile and exit the PowerCLI session, the image profile is no longer available when you start a new session. You can export the image profile to a ZIP file software depot, and add that depot in the next session.

Install the PowerCLI and all prerequisite software. See Configure vSphere ESXi Image Builder .

Procedure

1. In a PowerCLI session, create an image profile, for example by cloning an existing image profile and adding a VIB.
2. Export the image profile to a ZIP file by calling `Export-EsxImageProfile` with the `ExportToBundle` parameter.

   ```
   Export-EsxImageProfile    -ImageProfile "my_profile" -ExportToBundle -FilePath
                  "C:\isos\temp-base-plus-vib25.zip"
   ```
3. Exit the PowerCLI session.

4 When you start a new PowerCLI session, add the depot that contains your image profile to access it.

```
Add-EsxSoftwareDepot "C:\isos\temp-base-plus-vib25.zip"
```

Compare VIBs with PowerCLI Cmdlets

You can compare two VIBs or their properties by using the PowerShell comparison operators.

Prerequisites

Install the PowerCLI and all prerequisite software. See Configure vSphere ESXi Image Builder .

Procedure

1 In a PowerCLI session, run the `Add-EsxSoftwareDepot` cmdlet for each depot you want to work with.

Option	Action
Remote depot	Run `Add-EsxSoftwareDepot -DepotUrl <depot_url>`.
ZIP file	a Download the ZIP file to a local file system. b Run `Add-EsxSoftwareDepot -DepotUrl C:\<file_path>\<offline-bundle>.zip`

The cmdlet returns one or more `SoftwareDepot` objects.

2 (Optional) Run the `Get-EsxSoftwarePackage` cmdlet to view all available VIBs.

In the list, you can locate the names of the VIBs you want to compare.

3 Before comparing the VIBs, assign them to variables.

For example, you can create variables `$vib1` and `$vib2` to hold the names of the compared VIBs.

```
$vib1 = Get-EsxSoftwarePackage -Name "ReferenceVIB"
$vib2 = Get-EsxSoftwarePackage -Name "ComparisonVIB"
```

4 Use a comparison operator to compare the VIBs by contents and acceptance level or by a specific property.

- Compare the two VIBs by their contents and acceptance level.

```
if ($vib1 -eq $vib2) {
   Write-host "Successfully verified that both VIBs are equal."
} else {
   Write-host "Failed to verify that the VIBs are equal."
}
```

- Compare a specific property of the VIBs by using a comparison operator such as `-eq`, `-lt`, `-le`, `-gt`, or `-ge`.

```
if ($vib1.VersionObject -lt $vib2.VersionObject) {
   Write-host "Successfully verified that both the VIBs are equal."
} else {
   Write-host "Failed to verify that the VIBs are equal."
}
```

vSphere ESXi Image Builder Workflows with PowerCLI Cmdlets

vSphere ESXi Image Builder workflows are examples for cmdlet use and do not represent actual tasks.

vSphere ESXi Image Builder workflows are examples for cmdlet use. Workflows do not represent actual tasks, but illustrate how you might explore different ways of using a cmdlet. Administrators trying out the workflows benefit from some experience with PowerCLI, Microsoft PowerShell, or both.

Examine Depot Contents

You can examine software depots and VIBs with vSphere ESXi Image Builder cmdlets by using all kinds of wildcard expressions.

The workflow itself passes parameters by name. However, you can pass parameters as objects by accessing variables.

You can use filtering options and wildcard expressions to examine depot contents.

Prerequisites

Verify that PowerCLI and prerequisite software is installed. See Configure vSphere ESXi Image Builder .

Procedure

1 In a PowerCLI session, run the `Add-EsxSoftwareDepot` cmdlet for each depot you want to work with.

Option	Action
Remote depot	Run `Add-EsxSoftwareDepot -DepotUrl <depot_url>`.
ZIP file	a Download the ZIP file to a local file system. b Run `Add-EsxSoftwareDepot -DepotUrl C:\<file_path>\<offline-bundle>.zip`

The cmdlet returns one or more `SoftwareDepot` objects.

2 Retrieve image profiles.

You can filter by vendor, name, and acceptance level.

- `Get-EsxImageProfiles`

Returns an array of `ImageProfile` objects from all depots you added to the session.

- `Get-EsxImageProfile -Vendor "C*"`

  Returns all image profiles created by a vendor with a name that starts with the letter C.

3 Retrieve software packages by using the `Get-EsxSoftwarePackage` cmdlet.

You can filter, for example by vendor or version, and you can use the standard PowerShell wildcard characters.

- `Get-EsxSoftwarePackage -Vendor "V*"`

  Returns all software packages from a vendor with a name that starts with the letter V.

- `Get-EsxSoftwarePackage -Vendor "V*" -Name "*scsi*"`

  Returns all software packages with a name that contains the string `scsi` in it from a vendor with a name that starts with the letter V.

- `Get-EsxSoftwarePackage -Version "2.0*"`

  Returns all software packages with a version string that starts with 2.0.

4 Use `-Newest` to find the latest package.

- `Get-EsxSoftwarePackage -Vendor "V*" -Newest`

  Returns the newest package for the vendors with a name that starts with the letter V, and displays the information as a table.

- `Get-EsxSoftwarePackage -Vendor "V*" -Newest | format-list`

  Returns detailed information about each software package by using a pipeline to link the output of the request for software packages to the PowerShell `format-list` cmdlet.

5 View the list of VIBs in the image profile.

```
(Get-EsxImageProfile -Name "Robin's Profile").VibList
```

`VibList` is a property of the `ImageProfile` object.

6 Retrieve software packages released before or after a certain date by using the `CreatedBefore` or `CreatedAfter` parameter.

```
Get-EsxSoftwarePackage -CreatedAfter 7/1/2010
```

Example: Depot Content Examination Using Variables

This workflow example examines depot contents by passing in parameters as objects accessed by position in a variable, instead of passing in parameters by name. You can run the following commands in sequence from the PowerCLI prompt. Replace names with names that are appropriate in your installation.

```
Get-EsxSoftwarePackage -Vendor "V*"
Get-EsxSoftwarePackage -Vendor "V*" -Name "r*"
Get-EsxSoftwarePackage -Version "2.0*"
$ip1 = Get-EsxImageProfile -name ESX-5.0.0-123456-full
$ip1.VibList
Get-EsxSoftwarePackage -CreatedAfter 7/1/2010
```

Create Image Profiles by Cloning Workflow

Use vSphere ESXi Image Builder cmdlets to check available depots, add a depot, view image profile information, and to clone a new image profile.

Published profiles are usually read-only and cannot be modified. Even if a published profile is not read-only, cloning instead of modifying the profile is a best practice, because modifying the original profile erases the original. You cannot revert to the original, unmodified profile except by reconnecting to a depot.

A profile cloning workflow might include checking the current state of the system, adding a software depot, and cloning the profile.

Prerequisites

You can use vSphere ESXi Image Builder cmdlets to check which depots are available, to add a depot, to view image profile information, and to create a new image profile by cloning one of the available image profiles.

Verify that PowerCLI and prerequisite software is installed. See Configure vSphere ESXi Image Builder .

Procedure

1 In a PowerShell window, check whether any software depots are defined for the current session.

```
$DefaultSoftwareDepots
```

PowerShell returns the currently defined depots, or nothing if you just started PowerShell.

2 If the depot containing the profile that you want to clone does not appear in the results, add it to the current session.

Option	Action
Remote depot	Run `Add-EsxSoftwareDepot -DepotUrl depot_url`.
ZIP file	a Download the ZIP file to a local file path. b Run `Add-EsxSoftwareDepot -DepotUrl C:\file_path\offline-bundle.zip`

PowerShell adds the specified depot to your current session and lists all current depots.

3 (Optional) Check the `$DefaultSoftwareDepots` variable, which now returns the newly added depot.

4 View all available image profiles.

```
Get-EsxImageProfile
```

5 To clone an image profile, enter its name, a new name for the new profile, and a name of the vendor.

```
$ip = New-EsxImageProfile -CloneProfile base-tbd-v1 -Name "Test Profile 42" -Vendor
"Vendor20"
```

6 (Optional) View the newly created image profile, `$ip`.

PowerShell returns the information about the image profile in tabular format.

```
Name                Vendor          Last Modified          Acceptance Level
----                ------          -------------          ----------------
Test Profile 42     Vendor20        9/15/2010 5:45:43...   PartnerSupported
```

Example: Creating Image Profile by Cloning Using Variables

This workflow example repeats the steps of this workflow by passing in parameters as objects accessed by position in a variable, instead of passing in parameters by name. You can run the following cmdlets in sequence from the PowerCLI prompt.

```
$DefaultSoftwareDepots
Add-EsxSoftwareDepot -DepotUrl depot_url
$DefaultSoftwareDepots
$profs = Get-EsxImageProfile
$profs
$ip = New-EsxImageProfile -CloneProfile $profs[2] -Name "new_profile_name" -Vendor "my_vendor"
$ip
```

Create New Image Profiles Workflow

You can clone an existing image profile or create a new image profile, for which you must define dependencies and acceptance levels.

The system expects that the acceptance level of the VIBs you add to the base image is at least as high as the level of the base image. If you have to add a VIB with a lower acceptance level to the image profile, you must lower the image profile acceptance level. For more information, see Set the Image Profile Acceptance Level.

As an alternative to specifying the parameters on the command line, you can use the PowerShell prompting mechanism to specify string parameters. Prompting does not work for other parameters such as objects.

Prerequisites

In most situations, you create an image profile by cloning an existing profile. Some VMware customers or partners might need to create a new image profile. Pay careful attention to dependencies and acceptance levels if you create an image profile from scratch.

- PowerCLI and prerequisite software is installed. See Configure vSphere ESXi Image Builder .
- You have access to a depot that includes a base image and one or more VIBs. VMware and VMware partners have public depots, accessible by a URL. VMware or VMware partners can create a ZIP file that you can unzip to your local environment and access by using a file path.

Procedure

1 In a PowerCLI session, run the `Add-EsxSoftwareDepot` cmdlet for each depot you want to work with.

Option	Action
Remote depot	Run `Add-EsxSoftwareDepot -DepotUrl <depot_url>`.
ZIP file	a Download the ZIP file to a local file system. b Run `Add-EsxSoftwareDepot -DepotUrl C:\<file_path>\<offline-bundle>.zip`

The cmdlet returns one or more `SoftwareDepot` objects.

2 Run the `Get-EsxImageProfile` cmdlet to list all image profiles in all currently visible depots. You can narrow your search by using the optional arguments to filter the output.

```
Get-EsxSoftwarePackage -CreatedAfter 7/1/2010
```

3 Create a new profile, assign it a name and vendor, and add a base package.

```
New-EsxImageProfile -NewProfile -Name "Test #2" -vendor "Vendor42" -SoftwarePackage esx-base[0],esx-xlibs[0]
```

The example uses the esx-base package. In most cases, you include the esx-base package when you create a new image profile. Names that contain spaces are surrounded by quotes.

4 Use a pipeline to pass the new image profile to `format-list` for detailed information about the new package.

```
(Get-EsxImageProfile -Name "Test #2").VibList | format-list
```

Example: Creating Image Profiles from Scratch Using Variables

This command sequence repeats the steps of the workflow, but passes parameters as objects, accessed by position in a variable, instead of passing parameters by name. You can run the following commands in sequence at thePowerCLI prompt.

```
Add-EsxSoftwareDepot depoturl
$pkgs = Get-EsxSoftwarePackage -CreatedAfter 7/1/2010
$ip2 = New-EsxImageProfile -NewProfile -Name "Test #2" -vendor "Vendor42" -SoftwarePackage
$pkgs[0]
$ip2.VibList | format-list
```

Edit Image Profiles Workflow

You can create a custom image by cloning and editing an image profile by using PowerCLI.

Prerequisites

You can create a custom image by cloning and editing an image profile by using PowerCLI. You can add or remove one or more VIBs in the existing profile. If adding or removing VIBs prevents the image profile from working correctly, an error occurs.

- PowerCLI and prerequisite software is installed. See Configure vSphere ESXi Image Builder .
- You have access to a depot that includes a base image and one or more VIBs. VMware and VMware partners make public depots, accessible by a URL, available. VMware or VMware partners can create a ZIP file that you can download to your local environment and access by using a file path.

Procedure

1 In a PowerCLI session, run the `Add-EsxSoftwareDepot` cmdlet for each depot you want to work with.

Option	Action
Remote depot	Run `Add-EsxSoftwareDepot -DepotUrl <depot_url>`.
ZIP file	a Download the ZIP file to a local file system. b Run `Add-EsxSoftwareDepot -DepotUrl C:\<file_path>\<offline-bundle>.zip`

The cmdlet returns one or more `SoftwareDepot` objects.

2 Use a pipeline to pass the image profile you intend to edit to `format-list` to see detailed information.

In this example, the image profile created in Create New Image Profiles Workflow contains only the base image. A newly created image profile is not included in the depot. Instead, you access the image profile by name or by binding it to a variable.

```
Get-EsxImageProfile "Test #2" | format-list
```

PowerShell returns the information.

```
Name            : Test #2
Vendor           : Vendor42
...
VibList          : {esx-base 5.0.0.-...,}
```

3 (Optional) If you are adding a VIB with a lower acceptance level than that of the image profile, change the acceptance level of the image profile.

```
Set-EsxImageProfile -ImageProfile "Test #2" -AcceptanceLevel VMwareAccepted
```

PowerShell returns the information about the changed profile in tabular format.

```
Name           Vendor           Last Modified       Acceptance Level
----           ------           -------------       ----------------
Test #2        Vendor42         9/22/2010 12:05:... VMwareAccepted
```

4 Add a software package (VIB) to the image profile. You can add the package by name.

```
Add-EsxSoftwarePackage -ImageProfile "Test #2"
                -SoftwarePackage NewPack3
```

PowerShell returns the information about the image profile in tabular format.

```
Name           Vendor           Last Modified       Acceptance Level
----           ------           -------------       ----------------
Test #2        Vendor42         9/22/2010 12:05:... VMwareAccepted
```

Note If an error occurs when you add the software package, you might have a problem with acceptance levels, see Working with Acceptance Levels

5 View the image profile again.

```
Get-EsxImageProfile "Test #2" | format-list
```

The VIB list is updated to include the new software package and the information is displayed.

```
Name            : Test #2
Vendor           : Vendor42
...
VibList          : {esx-base 5.0.0.-..., NewPack3}
```

Example: Editing Image Profiles by Using Variables

This cmdlet sequence repeats the steps of the workflow but passes parameters as objects, accessed by position in a variable, instead of passing parameters by name. You can run the following cmdlets in sequence from the PowerCLI prompt.

```
Add-EsxSoftwareDepot -DepotUrl depot_url
$ip2 = Get-EsxImageProfile -name "Test #2"
$ip2 | format-list
Set-EsxImageProfile -ImageProfile $ip2 -AcceptanceLevel VMwareAccepted
Add-EsxImageSoftwarePackage -ImageProfile $ip2 -SoftwarePackage NewPack3
$ip2 | format-list
```

Installing ESXi

You can install ESXi interactively, with a scripted installation, or with vSphere Auto Deploy.

Installing ESXi Interactively

Use the interactive installation option for small deployments of fewer than five hosts.

In a typical interactive installation, you boot the ESXi installer and respond to the installer prompts to install ESXi to the local host disk. The installer reformats and partitions the target disk and installs the ESXi boot image. If you have not installed ESXi on the target disk before, all data on the drive is overwritten, including hardware vendor partitions, operating system partitions, and associated data.

Note To ensure that you do not lose any data, migrate the data to another machine before you install ESXi.

If you are installing ESXi on a disk that contains a previous installation of ESXi or ESX, or a VMFS datastore, the installer provides you with options for upgrading. See the *vSphere Upgrade* documentation.

Interactive ESXi Installation

You boot the server from an attached vSphere ESXi CD or DVD, or from a bootable USB device, or by PXE booting the server from a location on the network. You follow the prompts in the installation wizard to install ESXi to disk.

Install ESXi Interactively

You use the ESXi CD/DVD or a USB flash drive to install the ESXi software onto a SAS, SATA, SCSI hard drive, or USB drive.

Prerequisites

- You must have the ESXi installer ISO in one of the following locations:
  - On CD or DVD. If you do not have the installation CD/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 PXE boot the ESXi installer to run an interactive installation or a scripted installation. See Network Booting the ESXi Installer.
- Verify that the server hardware clock is set to UTC. This setting is in the system BIOS or UEFI.
- Verify that a keyboard and monitor are attached to the machine on which the ESXi software is installed. Alternatively, use a remote management application. See Using Remote Management Applications.
- Consider disconnecting your 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 ESX or ESXi installation. Do not disconnect a VMFS datastore that contains the Service Console of an existing ESX installation. These actions can affect the outcome of the installation.
- Gather the information required by the ESXi installation wizard. See Required Information for ESXi Installation.
- Verify that ESXi Embedded is not present on the host machine. ESXi Installable and ESXi Embedded cannot exist on the same host.

Procedure

1. Insert the ESXi installer CD/DVD into the CD/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.

   Note If your system has data processing units (DPUs), you can only use UEFI to install and boot ESXi on the DPUs.

   See your hardware vendor documentation for information on changing boot order.
3. On the welcome screen, press Enter to continue.

4 Accept the End User License Agreement by pressing **Enter**.

Starting with ESXi 8.0 Update 3, after the scanning for available devices completes, if your system has DPUs, you see them automatically listed with their respective PCI slots. You no longer select a slot. The DPU devices must be identical: same vendor, same hardware version and same firmware.

5 On the **Select a Disk to Install or Upgrade ESXi** screen, select the drive on which to install 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 and might be out of order. This might occur on systems where drives are continuously being added and removed.

If you select a disk that contains data, the **Confirm Disk Selection** page appears.

If you are installing on a disc with a previous ESXi or ESX installation or VMFS datastore, the installer provides several choices.

Important If you are upgrading or migrating an existing ESXi installation, see the *VMware ESXi Upgrade* documentation.

If you select a disk that is in vSAN disk group, the resulting installation depends on the type of disk and the group size:

- If you select an SSD, the SSD and all underlying HDDs in the same disk group are 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.

6 Select the keyboard type for the host.

You can change the keyboard type after installation in the direct console.

7 Enter the root password for the host.

You can change the password after installation in the direct console.

8 In the **Confirm Install** screen, if your vSphere system has DPUs, you see each listed on a separate row. Press **`F11`** to confirm the start of the installation.

Starting with 8.0 Update 3, if your systems has DPUs, you see a single progress bar for the ESXi and DPU installation, with dynamic updates to the label showing what stage of the installer is being run.

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

If you select an SD or USB device, you see a warning that prompts you to select a persistent disk to store the ESXi-OSData partition. In the **Select a Disk to store ESX OSData** screen, select a persistent storage device with minimum 32 GB available space.

9 When the installation is complete, remove the installation CD, DVD, or USB flash drive.

Starting with 8.0 Update 3, if the installation is not successful on any of the targets, you see the **Operation Failed** screen. Record the error log in the screen to help troubleshooting.

10 Press **Enter** to reboot the host.

11 Set the first boot device to be the drive on which you installed ESXi in Step 5.

For information about changing boot order, see your hardware vendor documentation.

Note UEFI systems might require additional steps to set the boot device. See Host Fails to Boot After You Install ESXi in UEFI Mode

Results

After the installation is complete, you can migrate existing VMFS data to the ESXi host.

You can boot a single machine from each ESXi image. Booting multiple devices from a single shared ESXi image is not supported.

What to do next

Set up basic administration and network configuration for ESXi. See After You Install and Set Up ESXi.

Install ESXi on a Software iSCSI Disk

When you install ESXi to a software iSCSI disk, you must configure the target iSCSI qualified name (IQN).

During system boot, the system performs a Power-On Self Test (POST), and begins booting the adapters in the order specified in the system BIOS. When the boot order comes to the iSCSI Boot Firmware Table (iBFT) adapter, the adapter attempts to connect to the target, but does not boot from it. See Prerequisites.

If the connection to the iSCSI target is successful, the iSCSI boot firmware saves the iSCSI boot configuration in the iBFT. The next adapter to boot must be the ESXi installation media, either a mounted ISO image or a physical CD-ROM.

Prerequisites

- Verify that the target IQN is configured in the iBFT BIOS target parameter setting. This setting is in the option ROM of the network interface card (NIC) to be used for the iSCSI LUN. See the vendor documentation for your system.

- Deactivate the iBFT adapter option to boot to the iSCSI target. This action is necessary to make sure that the ESXi installer boots, rather than the iSCSI target. When you start your system, follow the prompt to log in to your iBFT adapter and deactivate the option to boot to the iSCSI target. See the vendor documentation for your system and iBFT adapter. After you finish the ESXi installation, you can reenable the option to boot from the LUN you install ESXi on.

Procedure

1. Start an interactive installation from the ESXi installation CD/DVD or mounted ISO image.
2. On the Select a Disk screen, select the iSCSI target you specified in the iBFT BIOS target parameter setting.

   If the target does not appear in this menu, make sure that the TCP/IP and initiator iSCSI IQN settings are correct. Check the network Access Control List (ACL) and confirm that the adapter has adequate permissions to access the target.
3. Follow the prompts to complete the installation.
4. Reboot the host.
5. In the host BIOS settings, enter the iBFT adapter BIOS configuration, and change the adapter parameter to boot from the iSCSI target.

   See the vendor documentation for your system.

What to do next

On your iBFT adapter, reenable the option to boot to the iSCSI target, so the system will boot from the LUN you installed ESXi on.

Installing ESXi by Using a Script

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.

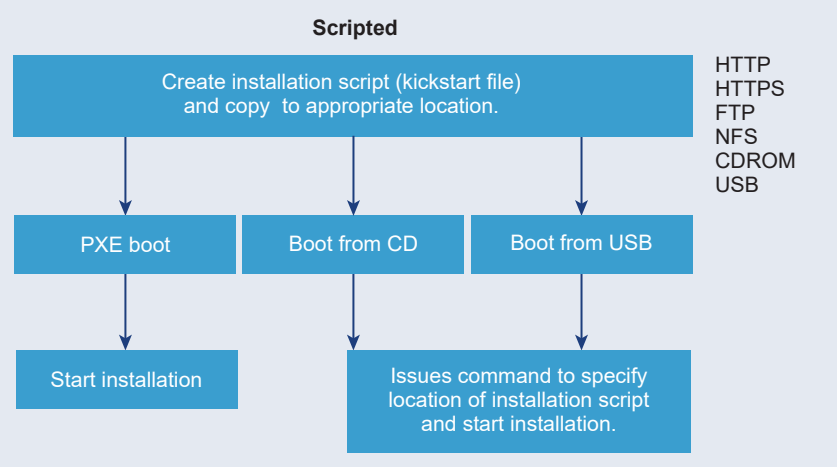
Scripted ESXi Installation

Running a script is an efficient way to deploy multiple ESXi hosts with an unattended installation.

The installation script contains the host configuration settings. You can use the script to configure multiple hosts with the same settings. See Installing or Upgrading Hosts by Using a Script.

The installation script must be stored in a location that the host can access by HTTP, HTTPS, FTP, NFS, CDROM, or USB. You can PXE boot the ESXi installer or boot it from a CD/DVD or USB drive.

Figure 4-3. Scripted Installation



Approaches for Scripted Installation

You can install ESXi on multiple machines using a single script for all of them or a separate script for each machine.

For example, because disk names vary from machine to machine, one of the settings that you might want to configure in a script is the selection for the disk to install ESXi on.

Table 4-12. Scripted Installation Choices

Option	Action
Always install on the first disk on multiple machines.	Create one script.
Install ESXi on a different disk for each machine.	Create multiple scripts.

For information about the commands required to specify the disk to install on, see Installation and Upgrade Script Commands.

Enter Boot Options to Run 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.

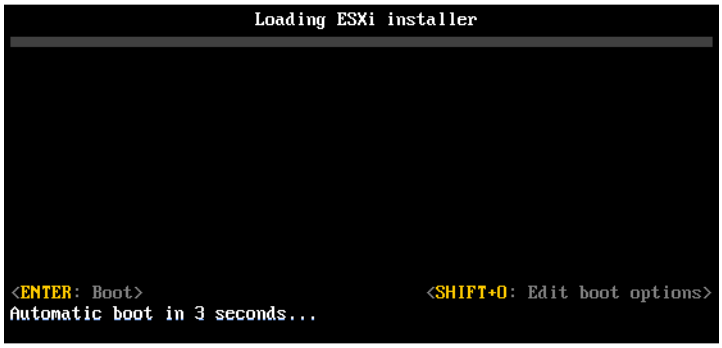
To specify the location of the installation script, set the `ks=filepath` option, where *filepath* indicates the location of your kickstart file. Otherwise, a scripted installation or upgrade cannot start. If `ks=filepath` is omitted, the text installer is run.

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 4-13. 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 4-13. 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 4-13. 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)

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

Installation and Upgrade Scripts Used for ESXi Installation

You can use a default script to perform a standard ESXi installation to the first detected disk.

During an ESXi installation, you can use a default script to perform a standard installation to the first detected disk, and the boot loader configuration file to specify the kernel, the kernel options, and the boot modules that the mboot.c32 or mboot.efi boot loader uses for the installation.

About the Default ks.cfg Installation Script

The ESXi installer includes a default installation script that performs a standard installation to the first detected disk.

The default `ks.cfg` installation script is located in the initial RAM disk at `/etc/vmware/weasel/ks.cfg`. You can specify the location of the default `ks.cfg` file with the `ks=file://etc/vmware/weasel/ks.cfg` boot option. See Enter Boot Options to Run an Installation or Upgrade Script.

When you install ESXi using the `ks.cfg` script, the default root password is `myp@ssw0rd`.

You cannot modify the default script on the installation media. After the installation, you can use the vSphere Client to log in to the vCenter Server that manages the ESXi host and modify the default settings.

With vSphere 8.0, if your system has data processing units (DPU), you use the `ks.cfg` script also to install ESXi on DPUs.

The default script contains the following commands:

```
#
# Sample scripted installation file
#

# Accept the VMware End User License Agreement
vmaccepteula

# Set the root password for the DCUI and Tech Support Mode
```

```
rootpw myp@ssw0rd

# Install on the first local disk available on machine
install --firstdisk --overwritevmfs
 In case you system has DPUs, you also specify a PCI slot:
install --firstdisk --overwritevmfs --dpupcislots=<PCIeSlotID>

# Set the network to DHCP on the first network adapter
network --bootproto=dhcp --device=vmnic0

# A sample post-install script
%post --interpreter=python --ignorefailure=true
import time
stampFile = open('/finished.stamp', mode='w')
stampFile.write( time.asctime() )
```

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 4-14. 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`*.

Table 4-14. Commands in `boot.cfg`. (continued)

Command	Description
`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 Installing ESXi by Using a Script.

To start an installation script from an interactive installation, you enter the `ks=` option manually. See Enter Boot Options to Run 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. 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 If your vSphere system is of version earlier than 8.0 Update 3 and has DPUs, you also specify a PCI slot: : `install --firstdisk --overwritevmfs --dpupcislots=<PCIeSlotID>`. For systems of version 8.0 Update 3 and later, the `dpupcislots` parameter is deprecated.

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 If your vSphere system is of version earlier than 8.0 Update 3 and has DPUs, you also specify a PCI slot: : `install --firstdisk --overwritevmfs --dpupcislots=<PCIeSlotID>`. For systems of version 8.0 Update 3 and later, the `dpupcislots` parameter is deprecated.

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 If your vSphere system is of version earlier than 8.0 Update 3 and has DPUs, you also specify a PCI slot: : `install --firstdisk --overwritevmfs --dpupcislots=<PCIeSlotID>`. For systems of version 8.0 Update 3 and later, the `dpupcislots` parameter is deprecated.

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.
--onfirstdisk= ***disk-type1,*** **[*disk-type2*,...]**	**Note** If your vSphere system is of version earlier than 8.0 Update 3 and has DPUs, you also specify a PCI slot: : `install --firstdisk --overwritevmfs --dpupcislots=<PCIeSlotID>`. For systems of version 8.0 Update 3 and later, the `dpupcislots` parameter is deprecated. 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 `--onfirstdisk=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	Specifies an interpreter to use. The default is busybox.

`=[python|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.

Disk Device Names

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

Table 4-15. 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.

Configuring External Entropy Sources During Scripted Installation

Starting with ESXi 8.0 Update 1, you can configure external entropy sources in the kickstart file for scripted installation.

You can configure ESXi in a highly secure environment to consume entropy from external entropy sources, such as a Hardware Security Module (HSM), and align with standards such as BSI Common criteria, EAL4, and NIST FIPS CMVP, by using the scripted installation method.

ESXi 8.0 Update 1 introduces an entropy daemon, `entropyd`, that creates a vAPI endpoint to provide REST API to query and send entropy data to ESXi hosts. The entropy daemon has several configurable parameters: `in-memory-cache size`, `in-storage-cache size`, `in-memory-low watermark`, and `entropy-lost-timeout`. Entropy sources, internal and external, are collected in the entropy-mixer module. The entropy daemon forwards entropy sources from the mixer to the entropy pool of the kernel.

If you do not need to select external entropy sources, you do not need to change anything in your existing scripts.

You can configure external entropy sources only for new installations. The entropy daemon uses entropy data passed during installation at first boot.

Note You cannot use any other method, such as interactive install, a cluster image, or Auto Deploy, to configure external entropy sources. You cannot enable the feature on an existing ESXi host.

Before you update the kickstart file, you must get legitimate binary entropy data from an external source such as HSM and save it as a file of size between 512 KB to 10 MB, for example, `entropy_data.dat`. You then encode the contents of the `entropy_data` file in a valid base64 format to create another file, for example `entropy_data.b64`.

In the kickstart file, you provide the following new parameters as part of the `entropy` command:

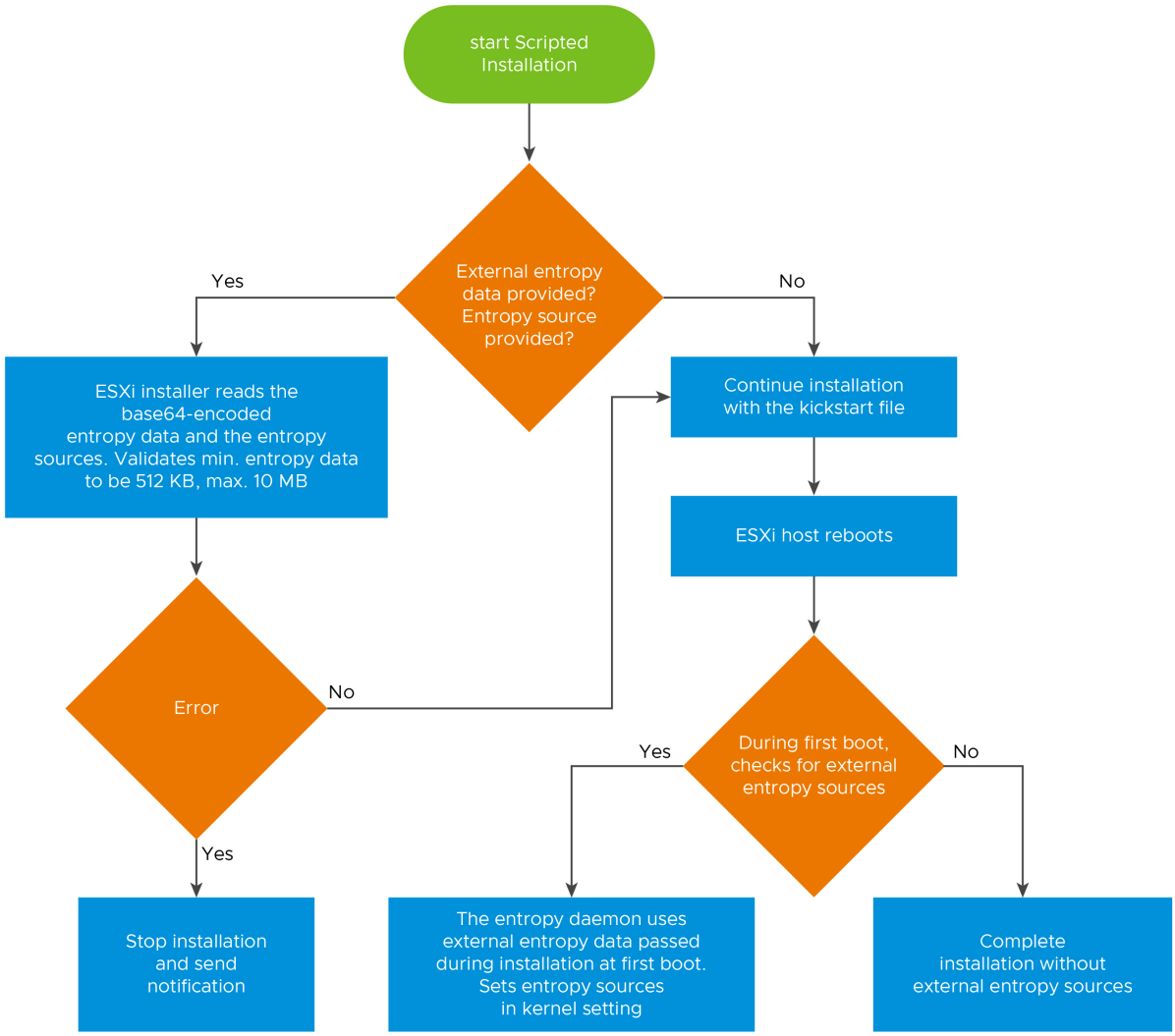
- `data` - the base64-encoded entropy data from an external source.
- `sources` - This bitmask value (`0`=**`default`**, `1`=**`interrupts`**, `2`=**`RDRAND`**, `4`=**`RDSEED`**, `8`=**`entropyd`**.) is set in the kernel settings during install time. If **`RDSEED`** is supported, the default is FIPS compliance. Otherwise, the default is all entropy sources, except **`entropyd`**. For EAL4 compliance, `8` is the only required value, but you can use other sources in the decimal range `0–15`. Source values persist as a kernel setting during install.

  Note If you configure an ESXi host with exteral entropy sources only, that is when the `sources` value is set to **8**, you must keep supplying external entropy to the host by using the entropy API. If external entropy is exhausted in the host, the host becomes unresponsive and it might require a hard reboot or re-installation to recover the host from such a situation.

A sample kickstart with the entropy parameters:

```
vmaccepteula
rootpw xxxxxxxx
entropy --sources=8 --data=xxxxxxx/xx/xxxxx/xx/xxxx...
install --firstdisk --overwritevmfs
network --bootproto=dhcp
```

Figure 4-4. Scripted installation workflow for adding external entropy sources



After installation completes, you can log in to the ESXi host and define some parameters for the entropy daemon from the shell by using the following ESXCLI commands:

ESXCLI commands

1. Get Commands

##	esxcli system entropyd get	Command Description
1	`<no argument>`	Get currently configured and default values of all `entropyd` parameters
2	`--default-values`	Get default values

2. Set Commands

##	esxcli system entropyd set	Command Description
1	`--help`	Print details of "esxcli system entropyd set" command and it's arguments

2	`--reset=all`
	`--reset=memory-cache-size`
	`--reset=memory-cache-low-watermark`
	`--reset=storage-cache-size`
	`--reset=external-entropy-lost-timeout`
		Reset a parameter or all parameters to its default values.

3	`--memory-cache-size=<value in KiB>`	Set memory cache size in KiB.
4	`--memory-cache-low-watermark=<value in %>`	Set memory cache low water mark in percentage.
5	`--storage-cache-size=<value in KiB>`	Set storage cache size in KiB.
6	`--external-entropy-lost-timeout=<value in seconds>`	Set external entropy lost timeout in seconds.

Example:

```
$ esxcli system entropyd set --external-entropy-lost-timeout=70 --memory-cache-low-
watermark=30 --memory-cache-size=612 --storage-cache-size=5096
$ esxcli system entropyd get
External Entropy Lost Timeout Seconds: 70
Memory Cache Low Watermark Percentage: 30
Memory Cache Size Kibibytes: 612
Storage Cache Size Kibibytes: 5096

$ esxcli system entropyd get --default-values
External Entropy Lost Timeout Seconds: 60
Memory Cache Low Watermark Percentage: 20
Memory Cache Size Kibibytes: 512
Storage Cache Size Kibibytes: 4096
```

During installation, the ESXi installer updates the entropy sources value in the kernel settings so that it persists in the ConfigStore after the installation. This change does not affect secure booting of ESXi hosts.

For more information, see vSphere Security and vSphere Automation SDKs Programming guides.

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.

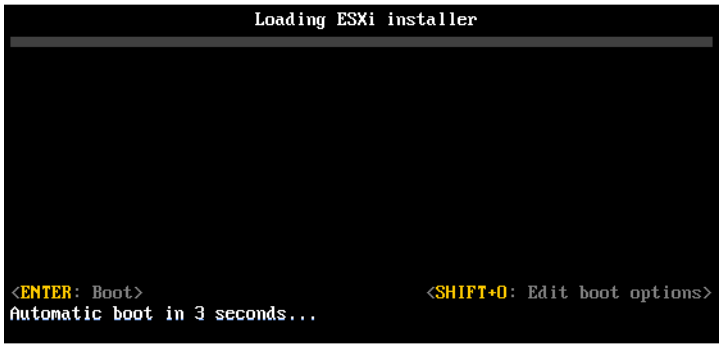
Prerequisites

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 Installing ESXi by Using a Script.
- You have selected a boot command to run the scripted installation or upgrade. See Enter Boot Options to Run 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 Stick 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 .

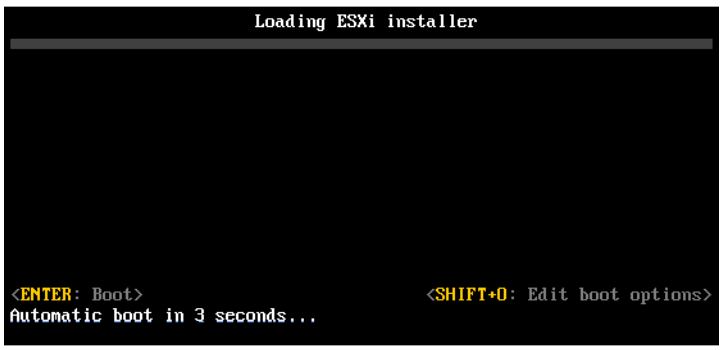
Prerequisites

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 Installing ESXi by Using a Script.
- You have selected a boot option to run the scripted installation, upgrade, or migration. See Enter Boot Options to Run 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.

Network Boot the ESXi Installer for a Scripted Installation or Upgrade

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 Installing ESXi by Using a Script.
- 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 installation by using PXE to boot, see Installing ESXi Using vSphere Auto Deploy .

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.

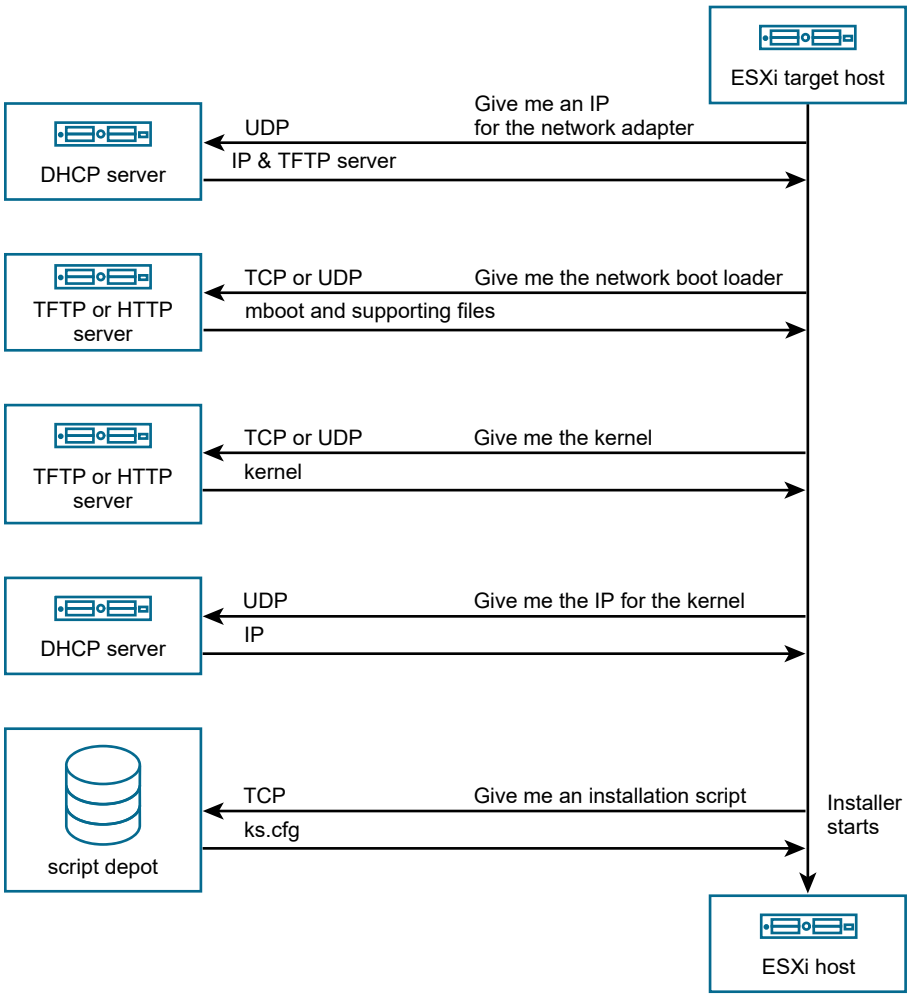
Note Network booting with legacy BIOS firmware is possible only over IPv4. Network booting with UEFI BIOS firmware is possible over IPv4 or IPv6.

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 4-5. 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.

- 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

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.

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-8.x.x-xxxxxx
   ```

   Here, `ESXi-8.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.

- 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

The prerequisites and steps depend on the support of UEFI boot or legacy BIOS only. 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.

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

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 an 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";
}
```

Installing ESXi Using vSphere Auto Deploy

vSphere Auto Deploy lets you provision hundreds of physical hosts with ESXi software.

Using Auto Deploy, experienced system administrators can manage large deployments efficiently. Hosts are network-booted from a central Auto Deploy server. Optionally, hosts are configured with a host profile of a reference host. The host profile can be set up to prompt the user for input. After boot up and configuration complete, the hosts are managed by vCenter Server just like other ESXi hosts.

Auto Deploy can also be used for stateless caching or stateful installs.

Important Auto Deploy requires a secure separation between the production network and the management or deployment networks as discussed in vSphere Auto Deploy Security Considerations. Using Auto Deploy without this separation is insecure.

Stateless caching

By default, Auto Deploy does not store ESXi configuration or state on the host disk. Instead, an image profile defines the image that the host is provisioned with, and other host attributes are managed through host profiles. A host that uses Auto Deploy for stateless caching still needs to connect to the Auto Deploy server and the vCenter Server.

Stateful installs

You can provision a host with Auto Deploy and set up the host to store the image to disk. On subsequent boots, the host boots from disk.

Understanding vSphere Auto Deploy

vSphere Auto Deploy can provision hundreds of physical hosts with ESXi software.

You can specify the image to deploy and the hosts to provision with the image. Optionally, you can specify host profiles to apply to the hosts, a vCenter Server location (datacenter, folder or cluster), and assign a script bundle for each host.

Introduction to vSphere Auto Deploy

vSphere Auto Deploy uses PXE boot infrastructure with host profiles, a desired image, or configuration on a cluster level to provision ESXi hosts.

State Information for ESXi Hosts

Note You cannot use Auto Deploy on ESXi hosts configured with DPUs as part of the vSphere Distributed Services Engine feature.

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.

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 4-16. 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.

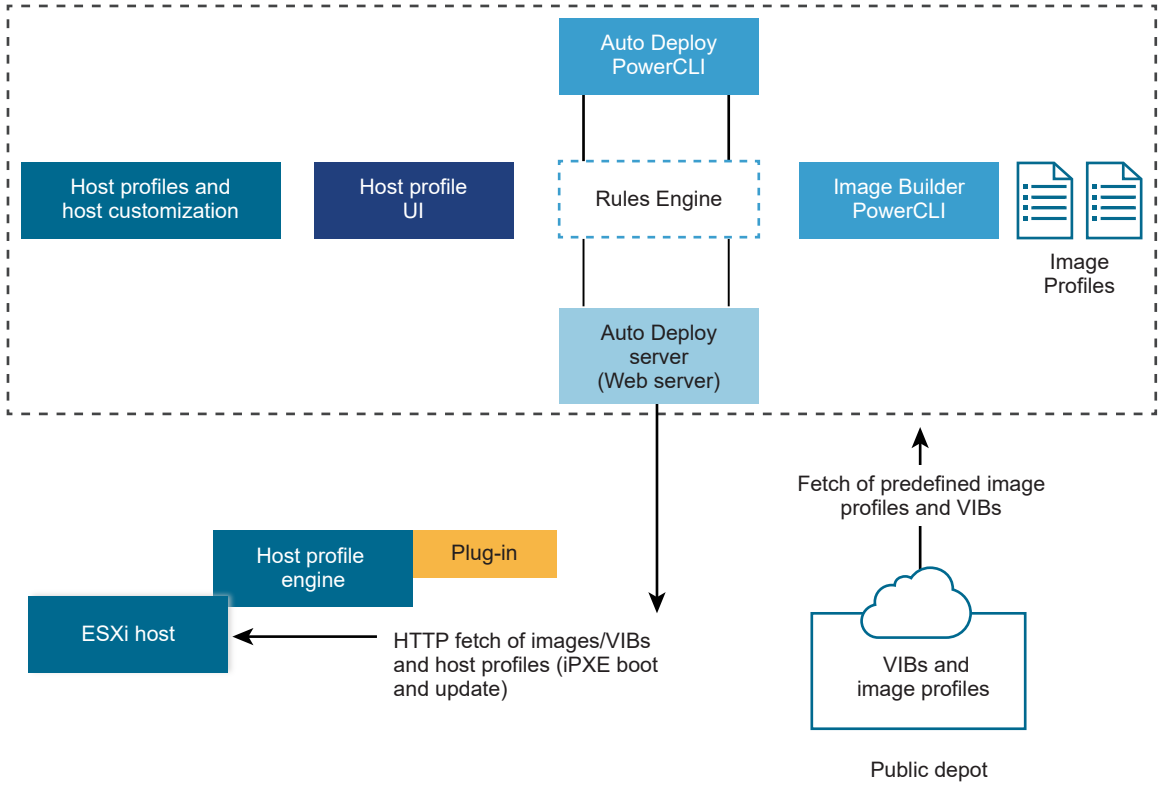
vSphere Auto Deploy Architecture

The vSphere Auto Deploy infrastructure consists of several components.

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

(Auto Deploy Architecture)

Figure 4-6. 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. For more information on vSphere Auto Deploy rules and rule sets, see Rules and Rule Sets.

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. See Customizing Installations with vSphere ESXi Image Builder.

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. For more information, see the *vSphere Host Profiles* documentation or the Setting Up a vSphere Auto Deploy Reference Host section.

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.

Rules and Rule Sets

You specify the behavior of the vSphere Auto Deploy server by using a set of rules. The vSphere Auto Deploy rules engine checks the rule set for matching host patterns to decide which items (image profile, host profile, vCenter Server location, or script object) to provision each host with.

The rules engine maps software and configuration settings to hosts based on the attributes of the host. For example, you can deploy image profiles or host profiles to two clusters of hosts by writing two rules, each matching on the network address of one cluster.

For hosts that have not yet been added to a vCenter Server system, the vSphere Auto Deploy server checks with the rules engine before serving image profiles, host profiles, and inventory location information to hosts. For hosts that are managed by a vCenter Server system, the image profile, host profile, and inventory location that vCenter Server has stored in the host object is used. If you make changes to rules, you can use the vSphere Client or vSphere Auto Deploy cmdlets in a PowerCLI session to test and repair rule compliance. When you repair rule compliance for a host, that host's image profile and host profile assignments are updated.

The rules engine includes rules and rule sets.

Rules

Rules can assign image profiles and host profiles to a set of hosts, or specify the location (folder or cluster) of a host on the target vCenter Server system. A rule can identify target hosts by boot MAC address, SMBIOS information, BIOS UUID, Vendor, Model, or fixed DHCP IP address. In most cases, rules apply to multiple hosts. You create rules by using the vSphere Client or vSphere Auto Deploy cmdlets in a PowerCLI session. After you create a rule, you must add it to a rule set. Only two rule sets, the active rule set and the working rule set, are supported. A rule can belong to both sets, the default, or only to the working rule set. After you add a rule to a rule set, you can no longer change the rule. Instead, you copy the rule and replace items or patterns in the copy. If you are managing vSphere Auto Deploy with the vSphere Client, you can edit a rule if it is in inactive state.

You can specify the following parameters in a rule.

Parameter	Description
Name	Name of the rule, specified with the `-Name` parameter.
Item	One or more items, specified with the `-Item` parameter. An item can be an image profile, a host profile, a vCenter Server inventory location (datacenter, folder, cluster) for the target host, or a custom script. You can specify multiple items separated by commas.
Pattern	The pattern specifies the host or group of hosts to which the rule applies. **vendor** Machine vendor name. **model** Machine model name. **serial** Machine serial number. **hostname** Machine hostname. **domain** Domain name. **ipv4** IPv4 address of the machine. **ipv6** IPv6 address of the machine. PXE booting with BIOS firmware is possible only with IPv4, PXE booting with UEFI firmware is possible with either IPv4 or IPv6. **mac** Boot NIC MAC address. **asset** Machine asset tag. **oemstring** OEM-specific strings in the SMBIOS. You can specify `-AllHosts` to apply the item or items to all hosts.

Active Rule Set

When a newly started host contacts the vSphere Auto Deploy server with a request for an image profile, the vSphere Auto Deploy server checks the active rule set for matching rules. The image profile, host profile, vCenter Server inventory location, and script object that are mapped by matching rules are then used to boot the host. If more than one item of the same

type is mapped by the rules, the vSphere Auto Deploy server uses the item that is first in the rule set.

Working Rule Set

The working rule set allows you to test changes to rules before making the changes active. For example, you can use vSphere Auto Deploy cmdlets for testing compliance with the working rule set. The test verifies that hosts managed by a vCenter Server system are following the rules in the working rule set. By default, cmdlets add the rule to the working rule set and activate the rules. Use the `NoActivate` parameter to add a rule only to the working rule set.

You use the following workflow with rules and rule sets.

1. Make changes to the working rule set.
2. Test the working rule set rules against a host to make sure that everything is working correctly.
3. Refine and retest the rules in the working rule set.
4. Activate the rules in the working rule set.

   If you add a rule in a PowerCLI session and do not specify the `NoActivate` parameter, all rules that are currently in the working rule set are activated. You cannot activate individual rules.

See the PowerCLI command-line help and Managing vSphere Auto Deploy with PowerCLI Cmdlets for more information on using vSphere Auto Deploy with PowerCLI cmdlets. See Managing vSphere Auto Deploy with the vSphere Client for more information on using vSphere Auto Deploy with the vSphere Client.

Install and Configure vSphere Auto Deploy

Before you can start using vSphere Auto Deploy, you must prepare your environment in several steps.

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 PowerCLI.

What to read next

- 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 PowerCLI. Users of vSphere Auto Deploy cmdlets can take advantage of all 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 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 4-17. 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.
PowerCLI	See Install PowerCLI.

Table 4-17. Preinstallation Checklist (continued)

Required Software and Hardware	Details
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 Install the TFTP Server and Prepare the DHCP Server for vSphere Auto Deploy Provisioning.
- 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 VMware vCenter Log Insight, which provides log aggregation and analytics for VMware and non-VMware products, virtual and physical, with near real-time search and analytics of log events.

- 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. See Configure ESXi Dump Collector with ESXCLI .

- 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. For Managing vSphere Auto Deploy with PowerCLI cmdlets, see the Managing vSphere Auto Deploy with PowerCLI Cmdlets section. For managing vSphere Auto Deploy with the vSphere Client, see the Managing vSphere Auto Deploy with the vSphere Client section.
- (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. See Configure a Reference Host for Auto-Partitioning .
- (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 PowerCLI. Users of vSphere Auto Deploy cmdlets can take advantage of all PowerCLI features.

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

You can type cmdlets, parameters, and parameter values in the 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 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 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 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 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 PowerCLI users who know how to use PowerShell variables.

Prerequisites

Prepare Your System for vSphere Auto Deploy.

Procedure

1 In a 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.

How vSphere Auto Deploy Works with PowerCLI

You can manage vSphere Auto Deploy with PowerCLI cmdlets to create rules that associate ESXi hosts with various objects on the vCenter target.

You can manage vSphere Auto Deploy with PowerCLI cmdlets to create rules that associate hosts with image profiles, host profiles, custom scripts and locations on the vCenter Server target. You can also update hosts by testing rule compliance and repairing compliance issues.

Managing vSphere Auto Deploy with PowerCLI Cmdlets

You can manage vSphere Auto Deploy with PowerCLI cmdlets to create rules that associate hosts with image profiles, host profiles, custom scripts and locations on the vCenter Server target. You can also update hosts by testing rule compliance and repairing compliance issues.

Overview of the vSphere Auto Deploy Process by Using PowerCLI

Getting started with vSphere Auto Deploy requires that you learn how vSphere Auto Deploy works, install the vSphere Auto Deploy server, install PowerCLI, write PowerCLI rules that provision hosts, and power on your hosts to be booted with the image profile you specify. You can customize of the image profile, host profile, and vCenter Server location.

See vSphere PowerCLI Scenario for vSphere Auto Deploy for a step-by-step exercise that helps you set up your first vSphere Auto Deploy environment.

To provision the hosts in your environment with vSphere Auto Deploy successfully, you can follow these steps.

1 Deploy vCenter Server.

   The vSphere Auto Deploy server is included.

2 Configure the vSphere Auto Deploy service startup type.

   See Prepare Your System for vSphere Auto Deploy.

3 Install PowerCLI, which includes vSphere Auto Deploy and vSphere ESXi Image Builder cmdlets.

   See Configure vSphere ESXi Image Builder , Using vSphere Auto Deploy Cmdlets, and Using VMware.Image Builder Cmdlets .

4 Find the image profile that includes the VIBs that you want to deploy to your hosts.

   - Usually, you add the depots containing the required software to your PowerCLI session, and then select an image profile from one of those depots.
   - To create a custom image profile, use vSphere ESXi Image Builder cmdlets to clone an existing image profile and add the custom VIBs to the clone. Add the custom image profile to the PowerCLI session.

   You must use vSphere ESXi Image Builder for customization only if you have to add or remove VIBs. In most cases, you can add the depot where VMware hosts the image profiles to your PowerCLI session as a URL.

5 Start a PowerCLI session and connect to the vCenter Server system that vSphere Auto Deploy is registered with.

6 Use the `New-DeployRule` PowerCLI cmdlet to write a rule that assigns the image profile to one host, to multiple hosts specified by a pattern, or to all hosts.

```
New-DeployRule -Name "testrule" -Item image-profile -AllHosts
```

   See Assign an Image Profile to Hosts.

   Note vSphere Auto Deploy is optimized for provisioning hosts that have a fixed MAC address to IP address mapping in DHCP (sometimes called DHCP reservations). If you want to use static IP addresses, you must set up the host profile to prompt for host customization. For more information, see the *vSphere Host Profiles* documentation.

7 Power on the hosts that you want to provision.

8 Set up the host you provisioned as a reference host for your host profile.

   You can specify the reference host syslog settings, firewall settings, storage, networking, and so on.

9 Set up the host you provisioned as a reference host for your host profile.

   You can specify the reference host syslog settings, firewall settings, storage, networking, and so on. See Setting Up a vSphere Auto Deploy Reference Host.

10 Create and export a host profile for the reference host.

   See the *Host Profiles* documentation.

11 To provision multiple hosts with the host profile, use the `Copy-DeployRule` cmdlet to edit the previously created rule.

You can revise the rule to assign not only an image profile but also a host profile, a vCenter Server location, and a custom script bundle.

```
Copy-DeployRule -DeployRule "testrule" -ReplaceItem
my_host_profile_from_reference_host,my_target_cluster
               -ReplacePattern  "ipv4=192.XXX.1.10-192.XXX.1.20"
```

Where *my_host_profile_from_reference_host* is the name of the reference host profile, and *my_target_cluster* is the name of the target cluster.

12 Perform the test and repair compliance operations to remediate the hosts.

See Test and Repair Rule Compliance .

13 Verify that the hosts you provisioned meet the following requirements.

- Each host is connected to the vCenter Server system.
- The hosts are not in maintenance mode.
- The hosts have no compliance failures.
- Each host with a host profile that requires user input has up-to-date host customization information.

Remediate host associations and compliance problems and reboot hosts until all hosts meet the requirements.

Read for an introduction to the boot process, differences between first and subsequent boots, and an overview of using host customization.

Using vSphere Auto Deploy Cmdlets

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

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

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

vSphere Auto Deploy PowerCLI Cmdlet Overview

You specify the rules that assign image profiles and host profiles to hosts using a set of PowerCLI cmdlets that are included in PowerCLI.

If you are new to PowerCLI, read thePowerCLI documentation and review Using vSphere Auto Deploy Cmdlets. You can get help for any command at the PowerShell prompt.

- Basic help: `Get-Help` *cmdlet_name*
- Detailed help: `Get-Help` *cmdlet_name* `-Detailed`

Note When you run vSphere Auto Deploy cmdlets, provide all parameters on the command line when you invoke the cmdlet. Supplying parameters in interactive mode is not recommended.

Table 4-18. Rule Engine PowerCLI Cmdlets

Command	Description
`Get-DeployCommand`	Returns a list of vSphere Auto Deploy cmdlets.
`New-DeployRule`	Creates a new rule with the specified items and patterns.

Table 4-18. Rule Engine PowerCLI Cmdlets (continued)

Command	Description
`Set-DeployRule`	Updates an existing rule with the specified items and patterns. You cannot update a rule that is part of a rule set.
`Get-DeployRule`	Retrieves the rules with the specified names.
`Copy-DeployRule`	Clones and updates an existing rule.
`Add-DeployRule`	Adds one or more rules to the working rule set and, by default, also to the active rule set. Use the `NoActivate` parameter to add a rule only to the working rule set.
`Remove-DeployRule`	Removes one or more rules from the working rule set and from the active rule set. Run this command with the `-Delete` parameter to completely delete the rule.
`Set-DeployRuleset`	Explicitly sets the list of rules in the working rule set.
`Get-DeployRuleset`	Retrieves the current working rule set or the current active rule set.
`Switch-ActiveDeployRuleset`	Activates a rule set so that any new requests are evaluated through the rule set.
`Get-VMHostMatchingRules`	Retrieves rules matching a pattern. For example, you can retrieve all rules that apply to a host or hosts. Use this cmdlet primarily for debugging.
`Test-DeployRulesetCompliance`	Checks whether the items associated with a specified host are in compliance with the active rule set.
`Repair-DeployRulesetCompliance`	Given the output of `Test-DeployRulesetCompliance`, this cmdlet updates the image profile, host profile, and location for each host in the vCenter Server inventory. The cmdlet might apply image profiles, apply host profiles, or move hosts to prespecified folders or clusters on the vCenter Server system.
`Apply-EsxImageProfile`	Associates the specified image profile with the specified host.
`Get-VMHostImageProfile`	Retrieves the image profile in use by a specified host. This cmdlet differs from the `Get-EsxImageProfile` cmdlet in vSphere ESXi Image Builder.
`Repair-DeployImageCache`	Use this cmdlet only if the vSphere Auto Deploy image cache is accidentally deleted.
`Get-VMHostAttributes`	Retrieves the attributes for a host that are used when the vSphere Auto Deploy server evaluates the rules.
`Get-DeployMachineIdentity`	Returns a string value that vSphere Auto Deploy uses to logically link an ESXi host in vCenter Server to a physical machine.
`Set-DeployMachineIdentity`	Logically links a host object in the vCenter Server database to a physical machine. Use this cmdlet to add hosts without specifying rules.

Table 4-18. Rule Engine PowerCLI Cmdlets (continued)

Command	Description
`Get-DeployOption`	Retrieves the vSphere Auto Deploy global configuration options. This cmdlet currently supports the `vlan-id` option, which specifies the default VLAN ID for the ESXi Management Network of a host provisioned with vSphere Auto Deploy. vSphere Auto Deploy uses the value only if the host boots without a host profile.
`Set-DeployOption`	Sets the value of a global configuration option. Currently supports the `vlan-id` option for setting the default VLAN ID for the ESXi Management Network.
`Add-ProxyServer`	Adds a proxy server to the vSphere Auto Deploy database. Run the command with the `-Address` parameter to specify the IPv4 or IPv6 address. The address can include a port number.
`List-ProxyServer`	Lists the proxy servers that are currently registered with vSphere Auto Deploy.
`Delete-ProxyServer`	Deletes one or more proxy servers from the list of proxy servers that are registered with vSphere Auto Deploy. You can run the command with the `-id` parameter from the list of proxy servers or with the`-Address` parameter by specifying the IPv4 or IPv6 address of the proxy server you want to delete.
`Add-ScriptBundle`	Adds one or more script bundles to the vSphere Auto Deploy server.
`Get-ScriptBundle`	Retrieves the list of script bundles available on the vSphere Auto Deploy server and the scripts they contain.
`Remove-ScriptBundle`	Removes a script bundle from vSphere Auto Deploy. Applicable for vSphere version 6.7 and later.
`Get-CustomCertificate`	Retrieves the custom host certificate uploaded into AutoDeploy. You must run the command with the `-HostId [MAC_Address | BIOS_UUID]` parameter. The first time you add custom certificates, you don't see any certificates returned by this cmdlet.
`List-CustomCertificates`	Retrieves information about all custom host certificates used by Auto Deploy. The list provides details for the name of the certificate, `Host ID`, and `Associated Host Name`, which reflects the name of the vCenter Server for the Auto Deploy server.

Table 4-18. Rule Engine PowerCLI Cmdlets (continued)

Command	Description
`Add-CustomCertificate`	Adds a custom certificate to the VMware Endpoint Certificate Store and associates it with an ESXi host. The certificate becomes active upon host reboot. You can use the `Get-CustomCertificate` cmdlet to retrieve the custom host certificate key. You can run the command with the `-HostId [MAC_Address | BIOS_UUID]` parameter to associate the certificate to the host, specifying a `-Key [file:///path/to/key.key]` and `-Cert [file:///path/to/cert.crt]`. Using this cmdlet requires the **AutoDeploy.Rule.Create** privilege on the root folder of vCenter Server.
`Remove-CustomCertificate`	Removes a set of custom host certificates from Auto Deploy. The certificate entries are deleted from the database and the certificate files are removed from the filestore. Hosts that have already booted with a custom certificate must be rebooted to receive a new certificate. You must provide at least one of `-Cert` or `-HostId` parameters. Using this cmdlet requires the **AutoDeploy.Rule.Create** privilege on the root folder of vCenter Server.

Assign an Image Profile to Hosts

Before you can provision a host, you must create rules that assign an image profile to each host that you want to provision by using vSphere Auto Deploy.

vSphere Auto Deploy extensibility rules enforce that VIBs at the CommunitySupported level can only contain files from certain predefined locations, such as the ESXCLI plug-in path, jumpstart plug-in path, and so on. If you add a VIB that is in a different location to an image profile, a warning results. You can override the warning by using the `force` option.

If you call the `New-DeployRule` cmdlet on an image profile that includes VIBs at the CommunitySupported level which violate the rule, set `$DeployNoSignatureCheck = $true` before adding the image profile. With that setting, the system ignores signature validation and does not perform the extensibility rules check.

Note Image profiles that include VIBs at the CommunitySupported level are not supported on production systems.

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 Determine the location of a public software depot, or define a custom image profile by using vSphere ESXi Image Builder.

3 Run `Add-EsxSoftwareDepot` to add the software depot that contains the image profile to the PowerCLI session.

Depot Type	Cmdlet
Remote depot	Run **Add-EsxSoftwareDepot *depot_url***.
ZIP file	a Download the ZIP file to a local file path. b Run **Add-EsxSoftwareDepot C:*file_path**my_offline_depot*.zip**.

4 In the depot, find the image profile that you want to use by running the `Get-EsxImageProfile` cmdlet.

By default, the ESXi depot includes one base image profile that includes VMware tools and has the string `standard` in its name, and one base image profile that does not include VMware tools.

5 Define a rule in which hosts with certain attributes, for example a range of IP addresses, are assigned to the image profile.

```
New-DeployRule -Name "testrule" -Item "My Profile25" -Pattern "vendor=Acme,Zven",
"ipv4=192.XXX.1.10-192.XXX.1.20"
```

Double quotes are required if a name contains spaces, optional otherwise. Specify `-AllHosts` instead of a pattern to apply the item to all hosts.

The cmdlet creates a rule named `testrule`. The rule assigns the image profile named My Profile25 to all hosts with a vendor of Acme or Zven that also have an IP address in the specified range.

6 Add the rule to the rule set.

```
Add-DeployRule testrule
```

By default, the rule is added to both the working rule set and the active rule set. If you use the `NoActivate` parameter, the working rule set does not become the active rule set.

Results

When the host boots from iPXE, it reports attributes of the machine to the console. Use the same format of the attributes when writing deploy rules.

```
****************************************************************
* Booting through VMware AutoDeploy...
```

```
*
* Machine attributes:
* . asset=No Asset Tag
* . domain=vmware.com
* . hostname=myhost.mycompany.com
* . ipv4=XX.XX.XXX.XXX
* . mac=XX:Xa:Xb:Xc:Xx:XX
* . model=MyVendorModel
* . oemstring=Product ID: XXXXXX-XXX
* . serial=XX XX XX XX XX XX...
* . uuid=XXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXX
* . vendor=MyVendor
****************************************************************
```

What to do next

- For hosts already provisioned with vSphere Auto Deploy, perform the compliance testing and repair operations to provision them with the new image profile. See Test and Repair Rule Compliance .
- Turn on unprovisioned hosts to provision them 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 ESXi hosts.

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

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.

- 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.

Write a Rule and Assign a Host to a Folder or Cluster

vSphere Auto Deploy can assign a host to a folder or cluster. When the host boots, vSphere Auto Deploy adds it to the specified location on the vCenter Server. Hosts assigned to a cluster inherit the cluster's host profile.

Prerequisites

- Prepare your system and install the Auto Deploy Server. For more information, see Prepare Your System for vSphere Auto Deploy.
- Verify that the folder you select is in a data center or in a cluster. You cannot assign the host to a standalone top-level folder.

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 Define a rule in which hosts with certain attributes, for example a range of IP addresses, are assigned to a folder or a cluster.

```
New-DeployRule -Name testrule3 -Item "my folder"    -Pattern "vendor=Acme,Zven",
"ipv4=192.XXX.1.10-192.XXX.1.20"
```

This example passes in the folder by name. You can instead pass in a folder, cluster, or data center object that you retrieve with the `Get-Folder`, `Get-Cluster`, or `Get-Datacenter` cmdlet.

3 Add the rule to the rule set.

```
Add-DeployRule testrule3
```

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 folder or cluster location by performing test and repair compliance operation. See Test and Repair Rule Compliance .
- Power on unprovisioned hosts to add them to the specified vCenter Server location.

Configure a Stateless System by Running a Custom Script

You can use vSphere Auto Deploy to configure one or more hosts by associating custom scripts with a vSphere Auto Deploy rule.

The scripts run in alphabetical order after the initial ESXi boot workflow of the host.

Prerequisites

- Verify that the script bundle you want to associate with a vSphere Auto Deploy rule is in `.tgz` format, with a maximum size of 10 MB, and written in Python or BusyBox ash scripting language.

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 Run the `Add-ScriptBundle` cmdlet to add the script bundle that contains the necessary scripts to the vSphere Auto Deploy inventory.

```
Add-ScriptBundle c:/temp/MyScriptBundle.tgz
```

The name of the script bundle without the `.tgz` extension is the name identifier or object of the script bundle item. You can update an existing script bundle by using the `-Update` parameter with the `Add-ScriptBundle` cmdlet.

3 (Optional) Run the `Get-ScriptBundle` cmdlet to verify that the script bundle is added to the vSphere Auto Deploy inventory.

4 Define a rule in which hosts with certain attributes, for example a range of IP addresses, are assigned to the script bundle.

```
New-DeployRule -Name "testrule4" -Item "MyScriptBundle" -Pattern "vendor=Acme,Zven",
"ipv4=192.XXX.1.10-192.XXX.1.20"
```

Double quotes are required if a name contains spaces, optional otherwise. Specify `-AllHosts` instead of a pattern to apply the item to all hosts.

You create a rule named *testrule4*. The rule assigns the script bundle named My Script Bundle to all hosts with a vendor of Acme or Zven that also have an IP address in the specified range. You can use the name identifier of the script bundle or the object returned by the `Get-ScriptBundle` cmdlet to identify the script bundle you want to associate with the rule.

5 Add the rule to the rule set.

```
Add-DeployRule testrule4
```

By default, the rule is added to both the working rule set and the active rule set. If you use the `NoActivate` parameter, the working rule set does not become the active rule set.

What to do next

- For hosts already provisioned with vSphere Auto Deploy, perform the compliance testing and repair operations to provision them with the new scripts. See Test and Repair Rule Compliance .
- Turn on unprovisioned hosts to provision them with the new scripts.

Test and Repair Rule Compliance

Test new or modified rules for compliance and repair accordingly, as changes in the vSphere Auto Deploy rule set are not updated automatically.

Prerequisites

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.

- 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.

Register a Caching Proxy Server Address with vSphere Auto Deploy

Simultaneously booting large number of stateless hosts places a significant load on the vSphere Auto Deploy server. You can load balance the requests between the vSphere Auto Deploy server and one or more proxy servers that you register with vSphere Auto Deploy.

Prerequisites

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 Register a caching proxy server addresses with vSphere Auto Deploy by running the `Add-ProxyServer` cmdlet.

```
Add-ProxyServer -Address 'https://proxy_server_ip_address:port_number'
```

You can run the cmdlet multiple times to register multiple proxy servers. The address can contain a port number.

3 (Optional) Run the `List-ProxyServer` cmdlet to verify that the caching proxy server is registered with vSphere Auto Deploy.

vSphere Auto Deploy Tasks by Using the vSphere Client

Managing vSphere Auto Deploy with the vSphere Client

You can add ESXi hosts to the vSphere Auto Deploy inventory, create, monitor, and manage rules, and host associations by using the vSphere Client.

 (Auto Deploy Enhancements in the vSphere Client)

Overview of the vSphere Auto Deploy Process by Using the vSphere Client

Getting started with vSphere Auto Deploy requires that you learn how vSphere Auto Deploy works, start the vSphere Auto Deploy and vSphere ESXi Image Builder vCenter Server services, create deploy rules that provision hosts, and power on your hosts to be booted with the image profile you specify.

The workflow for provisioning the hosts in your environment with vSphere Auto Deploy includes the following tasks:

1 Deploy vCenter Server.

   The vSphere Auto Deploy server is included.

2 Configure the vSphere Auto Deploy and vSphere ESXi Image Builder service startup types.

   See Prepare Your System for vSphere Auto Deploy and Configure the vSphere ESXi Image Builder.

3 Add or import a software depot to the vSphere Auto Deploy inventory.

   See Add a Software Depot or Import a Software Depot.

4 (Optional) If you want to create a custom image profile, clone, or create an image profile by using the vSphere Client.

   See Clone an Image Profile or Create an Image Profile.

5 Create a deploy rule that assigns the image profile to one host, to multiple hosts specified by a pattern, or to all hosts.

   See Create a Deploy Rule.

   Note vSphere Auto Deploy is optimized for provisioning hosts that have a fixed MAC address to IP address mapping in DHCP (sometimes called DHCP reservations). If you want to use static IP addresses, you must set up the host profile to prompt for host customization. For more information, see the *vSphere Host Profiles* documentation.

6 Power on the hosts that you want to provision.

7 Set up the host you provisioned as a reference host for your host profile.

   You can specify the reference host syslog settings, firewall settings, storage, networking, and so on.

8 Extract a host profile from the reference host.

   See the *Host Profiles* documentation.

9 To provision multiple hosts with the host profile, clone or edit the previously created rule by using the vSphere Client.

   See Clone a Deploy Rule or Edit a Deploy Rule.

10 Activate the new rule and deactivate the old one.

   See Activate, Deactivate, and Reorder Deploy Rules.

11 Remediate the host associations to apply the new rule to the host.

   See Remediate a Non-compliant Host.

12 Verify that the hosts you provisioned meet the following requirements.

   - Each host is connected to the vCenter Server system.
   - The hosts are not in maintenance mode.
   - The hosts have no compliance failures.
   - Each host with a host profile that requires user input has up-to-date host customization information.

   Remediate host associations and compliance problems and reboot hosts until all hosts meet the requirements.

Read for an introduction to the boot process, differences between first and subsequent boots, and an overview of using host customization.

Create a Deploy Rule

Before you provision ESXi hosts with vSphere Auto Deploy, you must create rules that assign host locations, image and host profiles to the hosts.

Prerequisites

An ESXi host can match more than one vSphere Auto Deploy rule criteria, when this is the case, the rule order is considered.

- Prepare your system and install the Auto Deploy Server. For more information, see Prepare Your System for vSphere Auto Deploy.
- If you want to include an image profile to the rule, verify that the software depot you need is added to the inventory. See Add a Software Depot or Import a Software Depot.

Procedure

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 **Deploy Rules** tab, click **New Deploy Rule**.

   The **New Deploy Rule** wizard appears.

3 On the **Name and hosts** page of the wizard, enter a name for the new rule.

4 Select to either apply the rule to all hosts in the inventory or only to hosts that match a specific pattern.

You can select one or more patterns.

For example, the rule can apply only to hosts in a vCenter Single Sign-On domain, with a specific host name, or that match a specific IPv4 range.

5 On the **Configuration** page of the wizard, you can optionally include items in the rule.

Each enabled item adds a new page to the wizard.

Option	Action
Host Location	Add the hosts that match the criteria of the rule to a specific location.
Image Profile	Assign an image profile to the hosts that match the rule criteria.
Host Profile	Assign a host profile to the hosts that match the rule criteria.
Script Bundle	Assign a script bundle to the host that match the rule criteria.

6 (Optional) On the **Select host location** page of the wizard, select a data center, folder, or cluster as host location for the hosts that match the rule.

7 (Optional) On the **Select image profile** page of the wizard, use the drop-down menu to select a software depot and choose an image profile from the list.

If you want to bypass the acceptance level verification for the image profile, select the **Skip image profile signature check** check box.

8 (Optional) On the **Select host profile** page of the wizard, select a host profile from the list.

9 (Optional) On the **Select script bundle** page of the wizard, select a script bundle from the list.

10 On the **Ready to complete** page, review the summary information for the new rule.

Results

You can view the newly created rule listed on the **Deploy Rules** tab.

What to do next

- Activate a vSphere Auto Deploy rule. See Activate, Deactivate, and Reorder Deploy Rules.
- Edit a vSphere Auto Deploy rule. See Edit an Image Profile.
- Clone a vSphere Auto Deploy rule. See Clone a Deploy Rule
- View the host location, image profile, host profile, and added script bundles. See View Host Associations.
- Remediate non-compliant hosts. See Remediate a Non-compliant Host.

- Change the image profile association of a host. See Edit the Image Profile Association of a Host.

Clone a Deploy Rule

You can use a vSphere Auto Deploy rule as a template and modify only parts of the rule instead of creating a new one.

You can clone an existing vSphere Auto Deploy rule by using the Clone Deploy Rule wizard.

Prerequisites

- Prepare your system and install the Auto Deploy Server. For more information, see Prepare Your System for vSphere Auto Deploy.
- Create a vSphere Auto Deploy rule. See Create a Deploy Rule.
- If you want to include an image profile to the rule, verify that the software depot you need is added to the inventory. See Add a Software Depot or Import a Software Depot.

Procedure

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 **Deploy Rules** tab, select a rule from the list.
3. Click **Clone**.

   The **Clone Deploy Rule** wizard appears.
4. On the **Name and hosts** page of the wizard, enter a name for the new rule.
5. Select to either apply the rule to all hosts in the inventory or only to hosts that match a specific pattern.

   You can select one or more patterns.

   For example, the rule can apply only to hosts in a vCenter Single Sign-On domain, with a specific host name, or that match a specific IPv4 range.
6. On the **Configuration** page of the wizard, you can optionally include items in the rule.

   Each enabled item adds a new page to the wizard.

Option	Action
Host Location	Add the hosts that match the criteria of the rule to a specific location.
Image Profile	Assign an image profile to the hosts that match the rule criteria.

Option	Action
Host Profile	Assign a host profile to the hosts that match the rule criteria.
Script Bundle	Assign a script bundle to the host that match the rule criteria.

7 On the **Select host location** page of the wizard, select a location for the hosts that match the rule.

Option	Action
If you want to keep the host location used in the cloned rule	Select the **Same Host location** check box.
If you want to select a new location for the selected hosts	1 Select the **Browse for Host location** check box. 2 Select a data center, folder, or cluster as host location. 3 Click **Next**.

8 On the **Select image profile** page of the wizard, select an image profile.

Option	Action
If you do not want to change the image profile	Select the **Same image profile** check box.
If you want to assign a new image profile to the selected hosts	1 Select the **Browse for Image Profile** check box. 2 Select a software depot from the drop-down menu. 3 Select an image profile from the list. 4 (Optional) If you want to bypass the acceptance level verification for the image profile, select the **Skip image profile signature check** check box.

9 On the **Select host profile** page of the wizard, select a host profile.

Option	Action
If you want to keep the host profile used in the cloned rule	Select the **Same Host profile** check box.
If you want to assign a new host profile to the selected hosts	1 Select the **Browse for Host Profile** check box. 2 Select a host profile from the list and click **Next**.

10 On the **Select script bundle** page of the wizard, select a script bundle from the list.

11 On the **Ready to complete** page, review the summary information for the new rule.

What to do next

- Activate a vSphere Auto Deploy rule. See Activate, Deactivate, and Reorder Deploy Rules.
- Edit a vSphere Auto Deploy rule. See Edit an Image Profile.

Edit a Deploy Rule

You can edit the name of an inactive Auto Deploy rule, its matching hosts, the host location, the image profile, and the host profile.

Prerequisites

- Prepare your system and install the Auto Deploy Server. For more information, see Prepare Your System for vSphere Auto Deploy.
- Create a vSphere Auto Deploy rule. See Create a Deploy Rule.

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 **Deploy Rules** tab, from the list of rules in the inventory select the rule and click **Edit**.

   The Edit Deploy Rule dialog box appears.

3. (Optional) On the **Name and hosts** page of the wizard, enter a new name for the rule.

4. Select to either apply the rule to all hosts in the inventory or only to hosts that match a specific pattern.

   You can select one or more patterns.

   For example, the rule can apply only to hosts in a vCenter Single Sign-On domain, with a specific host name, or that match a specific IPv4 range.

5. On the **Configuration** page of the wizard, you can optionally include items in the rule.

   Each enabled item adds a new page to the wizard.

Option	Action
Host Location	Add the hosts that match the criteria of the rule to a specific location.
Image Profile	Assign an image profile to the hosts that match the rule criteria.
Host Profile	Assign a host profile to the hosts that match the rule criteria.
Script Bundle	Assign a script bundle to the host that match the rule criteria.

6 On the **Select host location** page of the wizard, select a location for the hosts that match the rule.

Option	Action
If you want to keep the host location used in the cloned rule	Select the **Same Host location** check box.
If you want to select a new location for the selected hosts	1 Select the **Browse for Host location** check box. 2 Select a data center, folder, or cluster as host location. 3 Click **Next**.

7 On the **Select image profile** page of the wizard, select an image profile.

Option	Action
If you do not want to change the image profile	Select the **Same image profile** check box.
If you want to assign a new image profile to the selected hosts	1 Select the **Browse for Image Profile** check box. 2 Select a software depot from the drop-down menu. 3 Select an image profile from the list. 4 (Optional) If you want to bypass the acceptance level verification for the image profile, select the **Skip image profile signature check** check box.

8 On the **Select host profile** page of the wizard, select a host profile.

Option	Action
If you want to keep the host profile used in the cloned rule	Select the **Same Host profile** check box.
If you want to assign a new host profile to the selected hosts	1 Select the **Browse for Host Profile** check box. 2 Select a host profile from the list and click **Next**.

9 On the **Select script bundle** page of the wizard, select a script bundle from the list.

10 On the **Ready to complete** page, review the summary information for the new rule.

What to do next

- Activate a vSphere Auto Deploy rule. See Activate, Deactivate, and Reorder Deploy Rules.
- Clone a vSphere Auto Deploy rule. See Clone a Deploy Rule

Activate, Deactivate, and Reorder Deploy Rules

After you create a vSphere Auto Deploy rule, the rule is in inactive state. You must activate the rule for it to take effect.

The upper list on the **Activate and Reorder** page of the wizard displays the rules in the active rule set. The lower list displays the inactive rules.

Prerequisites

You can use the Activate and Reorder wizard to activate, deactivate, and change the order of the rules.

- Prepare your system and install the Auto Deploy Server. For more information, see Prepare Your System for vSphere Auto Deploy.
- Create a vSphere Auto Deploy rule. See Create a Deploy Rule.

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 **Deploy Rules** tab, click **Activate/Deactivate rules**.

   The Activate and Reorder wizard appears.

3. (Optional) If you want to deactivate an active rule, select the rule from the active rules list and click the **Deactivate** button.

4. From the list of inactive rules, select the rule that you want to activate and click the **Activate** button.

5. (Optional) If you want to reorder the rules in the active rule list, select a rule that you want to move up or down in the list and click **Move up** or **Move down** above the list of active rules.

   The rules are listed by priority. For example, if two or more rules apply to the same host but are set to provision the host with different host locations, image profiles, and host profiles, the rule that is highest in the list takes effect on the host.

6. (Optional) If you want to test an inactive rule before activation, click **Test rules before activation**.

   a. Select a host from the list and click **Check Compliance** to view the current status of the host and the changes that are expected after the activation of the rule.

      If the host is compliant with the rule, you do not need to remediate the host after you activate the rule.

   b. (Optional) If you want to remediate the selected hosts after the rule activation, enable the toggle button or select the **Remediate all host associations after rule activation** check box to remediate all hosts.

7. Review the list of active rules and click **OK**.

Results

On the **Deploy Rules** tab, the rule is listed as active in the Status column.

What to do next

- View the host location, image profile, host profile, and added script bundles. See View Host Associations.

- Remediate non-compliant hosts. See Remediate a Non-compliant Host.

View Host Associations

Some ESXi hosts in the vSphere Auto Deploy inventory might not be compliant with the active deploy rules and you must check for compliance.

Prerequisites

- Prepare your system and install the Auto Deploy Server. For more information, see Prepare Your System for vSphere Auto Deploy.
- Create a vSphere Auto Deploy rule. See Create a Deploy Rule.
- Activate a vSphere Auto Deploy rule. See Activate, Deactivate, and Reorder Deploy Rules.

Procedure

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

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

2 Check the host associations compliance.

The **Check Host Associations Compliance** window displays the status of the host and whether the host is compliant with the active rule set. You can view the currently assigned host location, image profile, host profile, script bundle, and the associations that will take effect after a remediation of the host.

Option	Steps
If you want to check the host associations compliance of a single host	1 On the **Deployed Hosts** tab, select an ESXi host. 2 Click **Check Host Associations Compliance**. 3 Check if the host associations are compliant with the current active rule set. 4 (Optional) If you want to remediate the host, click **Remediate**. 5 Close the **Check Host Associations Compliance** window.
If you want to check the host associations compliance of multiple hosts	1 On the **Deployed Hosts** tab, select multiple ESXi hosts. 2 Click **Check Host Associations Compliance**. 3 Confirm that you want to check the compliance of all selected hosts. 4 Review the compliance status of the hosts in the left pane. 5 (Optional) Select a host to view the compliance status details. 6 (Optional) Select a host and click **Remediate** . 7 (Optional) Select the **Remediate all host associations after rule activation** check box to remediate all hosts. 8 **Close** the **Check Host Associations Compliance** window.

What to do next

- Remediate non-compliant hosts. See Remediate a Non-compliant Host.
- Edit the image profile association of a host. See Edit the Image Profile Association of a Host.
- Edit a vSphere Auto Deploy rule. See Edit an Image Profile.

Edit the Image Profile Association of a Host

You can edit the image profile association of a single host if the host is not associated with a vSphere Auto Deploy rule.

Prerequisites

Alternatively, you might not want to change the image profile association of multiple hosts by editing a rule.

- Prepare your system and install the Auto Deploy Server. For more information, see Prepare Your System for vSphere Auto Deploy.
- Create a vSphere Auto Deploy rule. See Create a Deploy Rule.
- Activate a vSphere Auto Deploy rule. See Activate, Deactivate, and Reorder Deploy Rules.

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 **Deployed Hosts** tab, select an ESXi host.
3. Click **Edit Image Profile Association**.

   The Edit Image Profile Association dialog box appears.
4. Edit the image profile association of the host.

Option	Action
If you do not want to change the image profile	Select the **Same image profile** check box.
If you want to assign a new image profile to the selected hosts	1 Select the **Browse for Image Profile** check box. 2 Select a software depot from the drop-down menu. 3 Select an image profile from the list. 4 (Optional) If you want to bypass the acceptance level verification for the image profile, select the **Skip image profile signature check** check box.

5. Click **OK**.

Results

The new image profile is listed in the Associated Image Profile column after a refresh of the page.

What to do next

- View the host location, image profile, host profile, and added script bundles. See View Host Associations.
- If the host is associated with a rule and you want to revert to the image profile defined in the rule, remediate the host. See Remediate a Non-compliant Host.

Remediate a Non-compliant Host

Remediate the ESXi host associations when you add a rule to the vSphere Auto Deploy active rule set or make changes to one or more rules.

Prerequisites

When you add a rule to the vSphere Auto Deploy active rule set or make changes to one or more rules, hosts are not updated automatically. You must remediate the host associations to apply the new rules to the host.

- Prepare your system and install the Auto Deploy Server. For more information, see Prepare Your System for vSphere Auto Deploy.
- Create a vSphere Auto Deploy rule. See Create a Deploy Rule.
- Activate a vSphere Auto Deploy rule. See Activate, Deactivate, and Reorder Deploy Rules.
- If the remediation of a host, results in a change in its location, the host must be placed in maintenance mode.

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 **Deployed Hosts** tab, select a single or multiple ESXi hosts.
3. Click **Remediate Host Associations**.

   If you remediate a host that has an edited image profile association, the host reverts to the settings defined in the rule that it matches.

   You can monitor the progress of the remediation process in the Recent Tasks pane.

What to do next

- View the host location, image profile, host profile, and added script bundles. See View Host Associations.
- Change the image profile association of a host. See Edit the Image Profile Association of a Host.

Add a Host to the vSphere Auto Deploy Inventory

You can view the hosts that do not match any vSphere Auto Deploy rule and manually add a host to the vSphere Auto Deploy inventory.

To add a host to the current vSphere Auto Deploy inventory of deployed hosts, you can create a new rule or edit an existing rule to include a host that is not deployed with vSphere Auto Deploy and associate it with a specific host location, image profile, host profile, and script bundle. Alternatively, you can manually add a host to the inventory by assigning it a host location, image profile, host profile, and script bundle.

Prerequisites

- Prepare your system and install the Auto Deploy Server. For more information, see Prepare Your System for vSphere Auto Deploy.
- To assign an image profile to the host, add the software depot that you need to the inventory. See Add a Software Depot or Import a Software Depot.

Procedure

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 **Discovered Hosts** tab, select one or more hosts that you want to provision with a host location, image profile, and host profile.
3. Select **Add to Inventory**.

   Alternatively, click **Remove** to dismiss the selected hosts from the **Discovered Hosts** tab.

   The Add to Inventory wizard appears.
4. On the **Select host location** page of the wizard, select a data center, folder, or cluster as host location for the hosts that match the rule.
5. On the **Select image profile** page of the wizard, use the drop-down menu to select a software depot and choose an image profile from the list.

   If you want to bypass the acceptance level verification for the image profile, select the **Skip image profile signature check** check box.
6. On the **Select host profile** page of the wizard, select a host profile from the list.
7. On the **Select host profile** page of the wizard, use the **Filter** to search the host profiles list or select the **Do not include a host profile** check box to continue without adding a host profile.
8. On the **Select script bundle** page of the wizard, select a script bundle from the list.
9. On the Ready to complete page, review the selected host associations.

What to do next

- Edit a vSphere Auto Deploy rule. See Edit an Image Profile.
- Clone a vSphere Auto Deploy rule. See Clone a Deploy Rule
- View the host location, image profile, host profile, and added script bundles. See View Host Associations.
- Remediate non-compliant hosts. See Remediate a Non-compliant Host.

Add a Host to a Cluster That Uses a Single Image

Create a rule in Auto Deploy that assigns a cluster that you manage by a single image as the host location for newly added ESXi hosts.

By creating an Auto Deploy rule, where the host target location is a cluster managed by an image, you can transition stateful ESXi hosts to the cluster. Based on host identification mechanisms, the rule adds the hosts to the target cluster.

Such Auto Deploy rule is not allowed to contain an Image Profile or a Host Profile, as the image specification and configuration of hosts in the target cluster are created automatically.

Prerequisites

To add ESXi hosts to a cluster that you manage by a single image, you create a rule in Auto Deploy that assigns such a cluster as the host location for newly added hosts. Each host inherits the same image, which enables hardware compatibility checking, cluster-wide remediation, and easier upgrades.

- Prepare your system and install the Auto Deploy Server. For more information, see *Prepare Your System for vSphere Auto Deploy* in the *VMware ESXi Installation and Setup* documentation.
- Verify that each ESXi host is version 8.0 or later.
- Verify that all hosts in the cluster are stateful and have a physical storage attached.
- Verify that solutions, which are not integrated with vSphere Lifecycle Manager are not enabled for the cluster.

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 **Deploy Rules** tab, click **New Deploy Rule**.

   The **New Deploy Rule** wizard appears.
3. On the **Name and hosts** page of the wizard, enter a name for the new rule.
4. Select to either apply the rule to all hosts in the inventory or only to hosts that match a specific pattern.

   You can select one or more patterns.

   For example, the rule can apply only to hosts in a vCenter Single Sign-On domain, with a specific host name, or that match a specific IPv4 range.
5. On the **Configuration** page of the wizard, select the **Host Location** check box, and click **Next**.

   You can optionally include a script bundle to the ESXi hosts that match the rule criteria by selecting the **Script Bundle** check box.

   Note The image profile and host profile attached to the target cluster are used.

6 On the **Select host location** page of the wizard, select a cluster that uses a single image.

7 On the **Select script bundle** page of the wizard, select a script bundle from the list.

8 On the **Ready to complete** page, review the summary information for the new rule.

9 Click **Finish**.

Results

You can view the newly created rule listed on the **Deploy Rules** tab.

What to do next

- Activate a vSphere Auto Deploy rule. See Activate, Deactivate, and Reorder Deploy Rules.
- For more information on stateless caching and stateful installs, see Use vSphere Auto Deploy for Stateless Caching and Stateful Installs.

Add a Host to a Cluster That Manages ESXi Configuration at a Cluster Level

Create a rule in Auto Deploy that assigns newly added hosts to a cluster that manages ESXi configuration at a cluster level.

By creating an Auto Deploy rule, where the host target location is a cluster that manages ESXi configuration at a cluster level, you remove the need to use Host Profiles, or make any manual configurations, while keeping the flexibility to define custom settings per host or override the cluster-level settings for a group of hosts. For more information on how to set up clusters that you manage with a configuration at a cluster level, and VMware vSphere Configuration Profiles, see Using vSphere Configuration Profiles to Manage Host Configuration at a Cluster Level.

Note Once you set up a cluster that manages ESXi configuration at a cluster level, you cannot roll back to using Host Profiles or single images, and you cannot use the Quickstart option for such clusters. However, you can switch from a cluster that you manage with a single image to a cluster that you manage with a configuration at a cluster level just by selecting the **Set Up Host Settings** option under **Configure > Desired State > Host Settings**.

Prerequisites

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.

- Prepare your system and install the Auto Deploy Server. For more information, see Install and Configure vSphere Auto Deploy.
- Verify that each ESXi host is version 8.0 or later.

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 **Deploy Rules** tab, click **New Deploy Rule**.

   The **New Deploy Rule** wizard appears.

3. On the **Name and hosts** page of the wizard, enter a name for the new rule.

4. Select to either apply the rule to all hosts in the inventory or only to hosts that match a specific pattern.

   You can select one or more patterns.

   For example, the rule can apply only to hosts in a vCenter Single Sign-On domain, with a specific host name, or that match a specific IPv4 range.

5. On the **Configuration** page of the wizard, select the **Host Location** check box, and click **Next**.

   You can ignore the **Image**, **Host Settings**, and **Script Bundle** check boxes. Image specification and configuration of hosts in the target cluster, and any post-installation settings are created automatically. Even if you select any of the check boxes, the rule ignores the selections.

6. On the **Select host location** page of the wizard, select a cluster that manages ESXi configuration at a cluster level.

7. On the **Ready to complete** page, review the summary information for the new rule.

8. Click **Finish**.

Results

You can view the newly created rule listed on the **Deploy Rules** tab.

What to do next

- Activate a vSphere Auto Deploy rule. See Activate, Deactivate, and Reorder Deploy Rules.
- For more information on stateless caching and stateful installs, see Use vSphere Auto Deploy for Stateless Caching and Stateful Installs.

Working with Script Bundles

You can add a custom script for additional post-deployment host configuration. The script runs after you provision an ESXi host with Auto Deploy. For example, you can create a custom ESXi firewall rule and other configurations not available with Host Profiles.

Since vSphere 6.7 Update 1, you can add or remove a custom script by using the vSphere Client. A script bundle can include multiple scripts and must be delivered as a single compressed file with the `.tgz` extension. After uploaded to the vCenter Server, you can include the script bundle to an Auto Deploy rule.

Prerequisites

- Verify that you can run the script in the ESXi Shell.

Procedure

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

   By default, only the administrator role has privileges to use the vSphere Auto Deploy service.
2. Select the **Script Bundles** tab.
3. Click **Upload**.
4. Browse to a script bundle file and select **Upload**.

   The script is present in the **Script Bundles** list.
5. (Optional) Select a script bundle, click **Remove**, and confirm the selection.

   The script bundle is deleted from the list.

What to do next

- Activate a vSphere Auto Deploy rule. See Activate, Deactivate, and Reorder Deploy Rules.
- Edit a vSphere Auto Deploy rule. See Edit an Image Profile.
- Clone a vSphere Auto Deploy rule. See Clone a Deploy Rule
- View the host location, image profile, host profile, and added script bundles. See View Host Associations.
- Remediate non-compliant hosts. See Remediate a Non-compliant Host.
- Change the image profile association of a host. See Edit the Image Profile Association of a Host.

Download vSphere Auto Deploy Logs

You can use the vSphere Auto Deploy logging information from the vSphere Client to resolve problems that you encounter with vSphere Auto Deploy.

Prerequisites

Use the vSphere Client to log in to the vCenter Server instance that vSphere Auto Deploy is registered with.

Procedure

1. Navigate to**Home > Administration** and select **Deployment > System Configuration**.
2. Select one of the nodes for which you want to retrieve a support bundle. The support bundle holds the services logs.
3. Click **Export Support Bundle**.
4. Select only **VirtualAppliance > Auto Deploy**.

5 Click the **Export Support Bundle** button to download the log files.

Start, Stop, or Restart the vSphere Auto Deploy Service

You can start, stop, or restart the Auto Deploy service in the vCenter Server Management Interface.

To start, stop, and restart services in vCenter Server, you use the vCenter Server Management Interface.

Prerequisites

Verify that you have a root access to the vCenter Server Management Interface.

Procedure

1 Log in to the vCenter Server Management Interface, https://*IP-address-or-FQDN*:5480.

2 Click **Services**.

The **Services** pane displays a table of all installed services. You can sort them by name, startup type, health, and state.

3 Select the **Auto Deploy** service and select your action.

The available actions depend on whether the Auto Deploy service is already running or not.

- Click **Restart** to restart the service.

  Restarting the service requires confirmation and might lead to the Auto Deploy functionality becoming temporarily unavailable.

- Click **Start** to start the service.
- Click **Stop** to stop the service.

  Stopping the service requires confirmation.

Provision ESXi Hosts with vSphere Auto Deploy

Use vSphere Auto Deploy to provision or reprovision hundreds of physical hosts with ESXi software.

You can use vSphere Auto Deploy to provision hundreds of physical hosts with ESXi software either for the first time (first boot), or reboot hosts, or reprovision hosts with a different image profile, host profile, custom script, or folder, or cluster location. You can also select to provision the hosts with an image profile that does not contain VMware Tools binaries.

Provisioning ESXi Systems with vSphere Auto Deploy

vSphere Auto Deploy can provision hundreds of physical hosts with ESXi software, either for first boot, reboot, or reprovisioning.

You can provision hosts that did not previously run ESXi software (first boot), reboot hosts, or reprovision hosts with a different image profile, host profile, custom script, or folder, or cluster location. The vSphere Auto Deploy process differs depending on the state of the host and on the changes that you want to make.

vSphere Auto Deploy Boot Process

When you boot a host that you want to provision or reprovision with vSphere Auto Deploy, the vSphere Auto Deploy infrastructure supplies the image profile and, optionally, a host profile, a vCenter Server location, and script bundle for that host.

The boot process is different for hosts that have not yet been provisioned with vSphere Auto Deploy (first boot) and for hosts that have been provisioned with vSphere Auto Deploy and added to a vCenter Server system (subsequent boot).

First Boot Prerequisites

Before a first boot process, you must set up your system. Setup includes the following tasks, which are discussed in more detail in Install and Configure vSphere Auto Deploy.

- Set up a DHCP server that assigns an IP address to each host upon startup and that points the host to the TFTP server to download the iPXE boot loader from.
- 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.
- Identify an image profile to be used in one of the following ways.
  - Choose an ESXi image profile in a public depot.
  - (Optional) Create a custom image profile by using vSphere ESXi Image Builder, and place the image profile in a depot that the vSphere Auto Deploy server can access. The image profile must include a base ESXi VIB.
- (Optional) If you have a reference host in your environment, export the host profile of the reference host and define a rule that applies the host profile to one or more hosts. See Setting Up a vSphere Auto Deploy Reference Host.
- Specify rules for the deployment of the host and add the rules to the active rule set.

First Boot Overview

When a host that has not yet been provisioned with vSphere Auto Deploy boots (first boot), the host interacts with several vSphere Auto Deploy components.

1 When the administrator turns on a host, the host starts a PXE boot sequence.

  The DHCP Server assigns an IP address to the host and instructs the host to contact the TFTP server.

2 The host contacts the TFTP server and downloads the iPXE file (executable boot loader) and an iPXE configuration file.

3 iPXE starts executing.

The configuration file instructs the host to make a HTTP boot request to the vSphere Auto Deploy server. The HTTP request includes hardware and network information.

4 In response, the vSphere Auto Deploy server performs these tasks:

   a Queries the rules engine for information about the host.

   b Streams the components specified in the image profile, the optional host profile, and optional vCenter Server location information.

5 The host boots using the image profile.

   If the vSphere Auto Deploy server provided a host profile, the host profile is applied to the host.

6 vSphere Auto Deploy adds the host to thevCenter Server system that vSphere Auto Deploy is registered with.

   a If a rule specifies a target folder or cluster on the vCenter Server system, the host is placed in that folder or cluster. The target folder must be under a data center.

   b If no rule exists that specifies a vCenter Server inventory location, vSphere Auto Deploy adds the host to the first datacenter displayed in the vSphere Client UI.

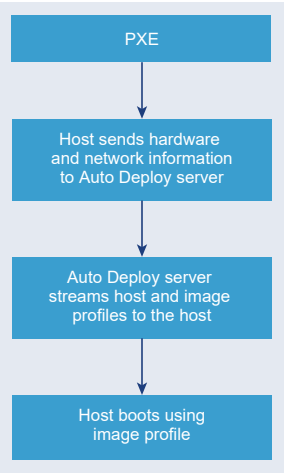
7 (Optional) If the host profile requires the user to specify certain information, such as a static IP address, the host is placed in maintenance mode when the host is added to the vCenter Server system.

   You must reapply the host profile and update the host customization to have the host exit maintenance mode. When you update the host customization, answer any questions when prompted.

8 If the host is part of a DRS cluster, virtual machines from other hosts might be migrated to the host after the host has successfully been added to the vCenter Server system.

See Provision a Host (First Boot).

Figure 4-7. vSphere Auto Deploy Installation, First Boot



Subsequent Boots Without Updates

For hosts that are provisioned with vSphere Auto Deploy and managed by vCenter Server, subsequent boots can become completely automatic.

1 The administrator reboots the host.

2 As the host boots up, vSphere Auto Deploy provisions the host with its image profile and host profile.

3 Virtual machines are brought up or migrated to the host based on the settings of the host.

- Standalone host. Virtual machines are powered on according to autostart rules defined on the host.
- DRS cluster host. Virtual machines that were successfully migrated to other hosts stay there. Virtual machines for which no host had enough resources are registered to the rebooted host.

If vCenter Server is unavailable, the host contacts the vSphere Auto Deploy server and is provisioned with an image profile. The host continues to contact the vSphere Auto Deploy server until vSphere Auto Deploy reconnects to the vCenter Server.

vSphere Auto Deploy cannot set up vSphere distributed switches if vCenter Server is unavailable, and virtual machines are assigned to hosts only if they participate in an vSphere HA cluster. Until the host is reconnected to vCenter Server and the host profile is applied, the switch cannot be created. Because the host is in maintenance mode, virtual machines cannot start. See Reprovision Hosts with Simple Reboot Operations.

Any hosts that are set up to require user input are placed in maintenance mode. See Update the Host Customization in the vSphere Client.

Subsequent Boots With Updates

You can change the image profile, host profile, vCenter Server location, or script bundle for hosts. The process includes changing rules and testing and repairing the host's rule compliance.

1 The administrator uses the `Copy-DeployRule` PowerCLI cmdlet to copy and edit one or more rules and updates the rule set. See Overview of the vSphere Auto Deploy Process by Using PowerCLI for an example.

2 The administrator runs the `Test-DeployRulesetCompliance` cmdlet to check whether each host is using the information that the current rule set specifies.

3 The host returns a PowerCLI object that encapsulates compliance information.

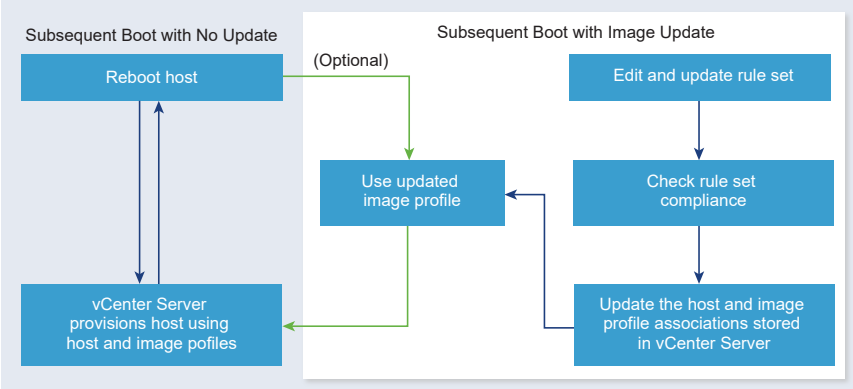
4 The administrator runs the `Repair-DeployRulesetCompliance` cmdlet to update the image profile, host profile, or vCenter Server location the vCenter Server system stores for each host.

5 When the host reboots, it uses the updated image profile, host profile, vCenter Server location, or script bundle for the host.

If the host profile is set up to request user input, the host is placed in maintenance mode. Follow the steps in Update the Host Customization in the vSphere Client.

See Test and Repair Rule Compliance .

Figure 4-8. vSphere Auto Deploy Installation, Subsequent Boots



Provisioning of Systems that Have Distributed Switches

You can configure the host profile of a vSphere Auto Deploy reference host with a distributed switch.

When you configure the distributed switch, the boot configuration parameters policy is automatically set to match the network parameters required for host connectivity after a reboot.

When vSphere Auto Deploy provisions the ESXi host with the host profile, the host goes through a two-step process.

1 The host creates a standard virtual switch with the properties specified in the boot configuration parameters field.

2 The host creates the VMkernel NICs. The VMkernel NICs allow the host to connect to vSphere Auto Deploy and to the vCenter Server system.

When the host is added to vCenter Server, vCenter Server removes the standard switch and reapplies the distributed switch to the host.

Note Do not change the boot configuration parameters to avoid problems with your distributed switch.

Provision a Host (First Boot)

Provisioning a host that has never been provisioned with vSphere Auto Deploy (first boot) differs from subsequent boot processes. You must prepare the host and fulfill all other prerequisites before you can provision the host. You can optionally define a custom image profile with vSphere ESXi Image Builder by using the vSphere Client or PowerCLI cmdlets.

Prerequisites

- Make sure your host meets the hardware requirements for ESXi hosts.

  See ESXi Hardware Requirements.

- Prepare the system for vSphere Auto Deploy. See Install and Configure vSphere Auto Deploy.
- Write rules that assign an image profile to the host and optionally assign a host profile and a vCenter Server location to the host. See Managing vSphere Auto Deploy with PowerCLI Cmdlets or Managing vSphere Auto Deploy with the vSphere Client.

  When the setup is complete, the vSphere Auto Deploy service is enabled, DHCP setup is complete, and rules for the host that you want to provision are in the active rule set.

Procedure

1. Turn on the host.

   The host contacts the DHCP server and downloads iPXE from the location the server points it to. Next, the vSphere Auto Deploy server provisions the host with the image specified by the rule engine. The vSphere Auto Deploy server might also apply a host profile to the host if one is specified in the rule set. Finally, vSphere Auto Deploy adds the host to the vCenter Server system that is specified in the rule set.
2. (Optional) If vSphere Auto Deploy applies a host profile that requires user input such as an IP address, the host is placed in maintenance mode. Reapply the host profile with the vSphere Client and provide the user input when prompted.

Results

After the first boot process, the host is running and managed by a vCenter Server system. The vCenter Server stores the host's image profile, host profile, and location information.

You can now reboot the host as needed. Each time you reboot, the host is reprovisioned by the vCenter Server system.

What to do next

Reprovision hosts as needed. See Reprovisioning Hosts.

If you want to change the image profile, host profile, custom script, or location of the host, update the rules and activate them by using the vSphere Client or perform a test and repair compliance operation in a PowerCLI session. See Rules and Rule Sets or Test and Repair Rule Compliance .

Reprovisioning Hosts

Use vSphere Auto Deploy to reprovision ESXi hosts with a different image profile or a different host profile.

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 Install and Configure vSphere Auto Deploy.

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

You can reprovisions ESXi hosts with the image profile, host profile, custom script, and vCenter Server location assigned during first boot.

Prerequisites

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.

- Verify that the setup you performed during the first boot operation is in place. See Provision a Host (First Boot).
- 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.

Use PowerCLI To Reprovision a Host

You can use vSphere Auto Deploy to reprovision a host with a new image profile in a PowerCLI session.

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 PowerCLI session. See Customizing Installations with vSphere ESXi Image Builder.
- Verify that the setup you performed during the first boot operation is in place.

Procedure

1. At the PowerShell prompt, run the `Connect-VIServer` 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 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 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.

Reprovision a Host with a New Image Profile by Using the vSphere Client

You can use vSphere Auto Deploy to reprovision a host with a new image profile with the vSphere Client by changing the rule that the host corresponds to and activating the rule.

Prerequisites

- Verify that the image profile you want to use to reprovision the host is available. See Create an Image Profile.
- Verify that the setup you performed during the first boot operation is in place.

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 **Deploy Rules** tab, from the list of rules in the inventory select the rule and click **Edit**.

The Edit Deploy Rule dialog box appears.

3 (Optional) On the **Name and hosts** page of the wizard, enter a name for the new rule.

4 (Optional) Select to either apply the rule to all hosts in the inventory or only to hosts that match a specific pattern.

You can select one or more patterns.

For example, the rule can apply only to hosts in a vCenter Single Sign-On domain, with a specific host name, or that match a specific IPv4 range.

5 On the **Configuration** page of the wizard, you can optionally include items in the rule.

Each enabled item adds a new page to the wizard.

Option	Action
Host Location	Add the hosts that match the criteria of the rule to a specific location.
Image Profile	Assign an image profile to the hosts that match the rule criteria.
Host Profile	Assign a host profile to the hosts that match the rule criteria.
Script Bundle	Assign a script bundle to the host that match the rule criteria.

6 Click **Next** to skip the Host Location selection.

7 On the **Select image profile** page of the wizard, assign an image profile to the hosts that match the rule criteria.

Option	Action
If you do not want to change the image profile	Select the **Same image profile** check box.
If you want to assign a new image profile to the selected hosts	1 Select the **Browse for Image Profile** check box. 2 Select a software depot from the drop-down menu. 3 Select an image profile from the list. 4 (Optional) If you want to bypass the acceptance level verification for the image profile, select the **Skip image profile signature check** check box.

8 Click **Next** to skip the Host profile selection.

9 On the **Ready to complete** page, review the summary information for the new image profile and click **Finish**.

10 Click **Activate/Deactivate rules**.

11 From the list of inactive rules, select the rule that you want to activate and click the **Activate** button.

12 (Optional) If you want to reorder the rules in the active rule list, select a rule that you want to move up or down in the list and click **Move up** or **Move down** above the list of active rules.

The rules are listed by priority. For example, if two or more rules apply to the same host but are set to provision the host with different host locations, image profiles, and host profiles, the rule that is highest in the list takes effect on the host.

13 (Optional) If you want to test an inactive rule before activation, click **Test rules before activation**.

a Select a host from the list and click **Check Compliance** to view the current status of the host and the changes that are expected after the activation of the rule.

If the host is compliant with the rule, you do not need to remediate the host after you activate the rule.

b (Optional) If you want to remediate the selected hosts after the rule activation, enable the toggle button or select the **Remediate all host associations after rule activation** check box to remediate all hosts.

14 Review the list of active rules and click **OK**.

15 Reboot the host to provision it with the new image profile.

Update the Host Customization in the vSphere Client

If a host required user input during a previous boot, the answers are saved with the vCenter Server. If you want to prompt the user for new information, you must remediate the host.

Prerequisites

Attach a host profile that prompts for user input to the host.

Procedure

1 Migrate all virtual machines to different hosts, and place the host into 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 Navigate to **Home > Auto Deploy**.

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

3 On the **Deployed Hosts** tab, select an ESXi host.

4 Click **Remediate Host Associations**.

You can monitor the progress of the remediation process in the Recent Tasks pane.

5 When prompted, provide the user input.

6 Direct the host to exit maintenance mode.

Results

The host customization is saved and takes effect the next time you boot the host.

Provision ESXi Host with an Image Profile Without VMware Tools

You can select to provision ESXi hosts with an image profile that does not contain VMware Tools binaries.

If the network boot time is too slow when using the standard image, or if you want to save space on the hosts, you can use an image profile that does not include VMware Tools and place the VMware Tools binaries on a shared storage.

Prerequisites

When you provision ESXi hosts with vSphere Auto Deploy, you can select to provision the hosts with an image profile that does not contain VMware Tools binaries. This image profile is smaller, has a lower memory overhead, and boots faster in a PXE-boot environment.

Download the *xxxxx*-no-tools image profile from the VMware download site.

Procedure

1 Boot an ESXi host that was not provisioned with vSphere Auto Deploy.

2 Copy the /productLocker directory from the ESXi host to a shared storage.

You can connect to an ESXi host using an SSH client, see Knowledge Base article1019852.

3 Change the *UserVars.ProductLockerLocation* variable to point to the new /productLocker directory location.

a In the vSphere Client, select the reference host and click the **Configure** tab.

b Under **System**, click **Advanced System Settings**.

c Click **Edit**.

d Filter the settings for **uservars**, and select **UserVars.ProductLockerLocation**.

e Click the current value and edit the location so it points to the shared storage.

4 Create a host profile from the reference host.

5 Create a vSphere Auto Deploy rule that assigns the *xxxxx*-no-tools image profile and host profile from the reference host to all other hosts.

6 Boot your target hosts with the rule so they pick up the product locker location from the reference host.

Use vSphere Auto Deploy for Stateless Caching and Stateful Installs

With stateless caching, you can cache the image of an ESXi host. With stateful installs, you can install hosts over the network.

The vSphere Auto Deploy stateless caching feature lets you cache the host's image. The vSphere Auto Deploy stateful installs feature lets you install hosts over the network. After the initial network boot, these hosts boot like other ESXi hosts. The stateless caching solution is primarily intended for situations when several hosts boot simultaneously. The locally cached image helps prevent a bottleneck that results if several hundreds of hosts connect to the vSphere Auto Deploy server simultaneously. After the boot operation is complete, hosts connect to vSphere Auto Deploy to complete the setup.

The stateful installs feature lets you provision hosts with the image profile over the network without having to set up the PXE boot infrastructure.

What to read next

- Introduction to Stateless Caching and Stateful Installs

  You can use the System Cache Configuration host profile to provision hosts with vSphere Auto Deploy stateless caching and stateful installs.

- Understanding Stateless Caching and Stateful Installs

  When you want to use vSphere Auto Deploy with stateless caching or stateful installs, you must set up a host profile, apply the host profile, and set the boot order.

Introduction to Stateless Caching and Stateful Installs

You can use the System Cache Configuration host profile to provision hosts with vSphere Auto Deploy stateless caching and stateful installs.

Examples of Stateless Caching and Stateful Installs

Hosts provisioned with vSphere Auto Deploy cache the image (stateless caching)

Set up and apply a host profile for stateless caching. You can cache the image on a local disk, a remote disk, or a USB drive. Continue provisioning this host with vSphere Auto Deploy. If the vSphere Auto Deploy server becomes unavailable, for example because hundreds of hosts attempt to access it simultaneously, the host boots from the cache. The host attempts to reach the vSphere Auto Deploy server after the boot operation to complete configuration.

Hosts provisioned with vSphere Auto Deploy become stateful hosts

Set up and apply a host profile for stateful installs. When you provision a host with vSphere Auto Deploy, the image is installed on the local disk, a remote disk, or a USB drive. For subsequent boots, you boot from the disk. The host no longer uses vSphere Auto Deploy.

Preparation

To successfully use stateless caching or stateful installs, decide how to configure the system and set the boot order.

Table 4-19. Preparation for Stateless Caching or Stateful Installs

Requirement or Decision	Description
Decide on VMFS partition overwrite	When you install ESXi by using the interactive installer, you are prompted whether you want to overwrite an existing VMFS datastore. The System Cache Configuration host profile provides an option to overwrite existing VMFS partitions. The option is not available if you set up the host profile to use a USB drive.
Decide whether you need a highly available environment	If you use vSphere Auto Deploy with stateless caching, you can set up a highly available vSphere Auto Deploy environment to guarantee that virtual machines are migrated on newly provisioned hosts and that the environment supports vNetwork Distributed Switch even if the vCenter Server system becomes temporarily unavailable.
Set the boot order	The boot order you specify for your hosts depends on the feature you want to use. ■ To set up vSphere Auto Deploy with stateless caching, configure your host to first attempt to boot from the network, and to then attempt to boot from disk. If the vSphere Auto Deploy server is not available, the host boots using the cache. ■ To set up vSphere Auto Deploy for stateful installs on hosts that do not currently have a bootable disk, configure your hosts to first attempt to boot from disk, and to then attempt to boot from the network. **Note** If you currently have a bootable image on the disk, configure the hosts for one-time PXE boot, and provision the host with vSphere Auto Deploy to use a host profile that specifies stateful installs.

Stateless Caching and Loss of Connectivity

If the ESXi hosts that run your virtual machines lose connectivity to the vSphere Auto Deploy server, the vCenter Server system, or both, some limitations apply the next time you reboot the host.

- If vCenter Server is available but the vSphere Auto Deploy server is unavailable, hosts do not connect to the vCenter Server system automatically. You can manually connect the hosts to the vCenter Server, or wait until the vSphere Auto Deploy server is available again.
- If both vCenter Server and vSphere Auto Deploy are unavailable, you can connect to each ESXi host by using the VMware Host Client, and add virtual machines to each host.
- If vCenter Server is not available, vSphere DRS does not work. The vSphere Auto Deploy server cannot add hosts to the vCenter Server. You can connect to each ESXi host by using the VMware Host Client, and add virtual machines to each host.
- If you make changes to your setup while connectivity is lost, the changes are lost when the connection to the vSphere Auto Deploy server is restored.

Understanding Stateless Caching and Stateful Installs

When you want to use vSphere Auto Deploy with stateless caching or stateful installs, you must set up a host profile, apply the host profile, and set the boot order.

When you apply a host profile that enables caching to a host, vSphere Auto Deploy partitions the specified disk. What happens next depends on how you set up the host profile and how you set the boot order on the host.

- vSphere Auto Deploy caches the image when you apply the host profile if **Enable stateless caching on the host** is selected in the System Cache Configuration host profile. No reboot is required. When you later reboot, the host continues to use the vSphere Auto Deploy infrastructure to retrieve its image. If the vSphere Auto Deploy server is not available, the host uses the cached image.
- vSphere Auto Deploy installs the image if **Enable stateful installs on the host** is selected in the System Cache Configuration host profile. When you reboot, the host initially boots using vSphere Auto Deploy to complete the installation. A reboot is then issued automatically, after which the host boots from disk, similar to a host that was provisioned with the installer. vSphere Auto Deploy no longer provisions the host.

You can apply the host profile from the vSphere Client, or write a vSphere Auto Deploy rule in a PowerCLI session that applies the host profile.

Using the vSphere Client to Set Up vSphere Auto Deploy for Stateless Caching or Stateful Installs

You can create a host profile on a reference host and apply that host profile to additional hosts or to a vCenter Server folder or cluster. The following workflow results.

1. You provision a host with vSphere Auto Deploy and edit that host's System Image Cache Configuration host profile.
2. You place one or more target hosts in maintenance mode, apply the host profile to each host, and instruct the host to exit maintenance mode.
3. What happens next depends on the host profile you selected.
   - If the host profile enabled stateless caching, the image is cached to disk. No reboot is required.
   - If the host profile enabled stateful installs, the image is installed. When you reboot, the host uses the installed image.

Using PowerCLI to Set Up vSphere Auto Deploy for Stateless Caching or Stateful Installs

You can create a host profile for a reference host and write a vSphere Auto Deploy rule that applies that host profile to other target hosts in a PowerCLI session. The following workflow results.

1. You provision a reference host with vSphere Auto Deploy and create a host profile to enable a form of caching.
2. You write a rule that provisions additional hosts with vSphere Auto Deploy and that applies the host profile of the reference host to those hosts.

3 vSphere Auto Deploy provisions each host with the image profile or by using the script bundle associated with the rule. The exact effect of applying the host profile depends on the host profile you selected.

- For stateful installs, vSphere Auto Deploy proceeds as follows:
  - During first boot, vSphere Auto Deploy installs the image on the host.
  - During subsequent boots, the host boots from disk. The hosts do not need a connection to the vSphere Auto Deploy server.
- For stateless caching, vSphere Auto Deploy proceeds as follows:
  - During first boot, vSphere Auto Deploy provisions the host and caches the image.
  - During subsequent boots, vSphere Auto Deploy provisions the host. If vSphere Auto Deploy is unavailable, the host boots from the cached image, however, setup can only be completed when the host can reach the vSphere Auto Deploy server.

Configure a Host Profile to Use Stateless Caching

If the vSphere Auto Deploy Server is not available, the host uses a cached image and to use stateless caching, you must configure a host profile.

Prerequisites

When a host is set up to use stateless caching, if the vSphere Auto Deploy Server is not available, the host uses a cached image. To use stateless caching, you must configure a host profile. You can apply that host profile to other hosts that you want to set up for stateless caching.

- Decide which disk to use for caching and determine whether the caching process will overwrite an existing VMFS partition.
- In production environments, protect the vCenter Server system and the vSphere Auto Deploy server by including them in a highly available environment. Having the vCenter Server in a management cluster guarantees that VDS and virtual machine migration are available. If possible, also protect other elements of your infrastructure. See Set Up Highly Available vSphere Auto Deploy Infrastructure.
- Set up your environment for vSphere Auto Deploy. See Install and Configure vSphere Auto Deploy.
- Verify that a disk with at least 4GB of free space is available. If the disk is not yet partitioned, partitioning happens when you apply the host profile.
- Set up the host to attempt a network boot first and to boot from disk if network boot fails. See your hardware vendor's documentation.
- Create a host profile. See the *Host Profiles* documentation.

Procedure

1 Navigate to **Home > Policies and Profiles > Host Profiles**.

2 Click the host profile you want to configure and select the **Configure** tab.

3 Click **Edit Host Profile**.

4 On the Edit host profile page of the wizard, select **Advanced Configuration Settings > System Image Cache Configuration > System Image Cache Configuration**.

5 In the **System Image Cache Profile Settings** drop-down menu, choose a policy option.

Option	Description
Enable stateless caching on the host	Caches the image to disk.
Enable stateless caching to a USB disk on the host	Caches the image to a USB disk attached to the host.

6 Depending on the policy option you select, you must do the following:

a If you select **Enable stateless caching on the host**:

1 Specify the information about the disk to use.

Option	Description
Arguments for first disk	When configuring a System Image Install disk, you have multiple options to define the device you want ESXi to be installed to and booted from. You can use the following arguments to define the disk for the installation: ■ **`localesx`** – The first disk detected containing a valid installation of ESXi ■ **`local`** – The first local disk detected by ESXi after boot ■ **`remoteesx`** - The first remote disk detected containing a valid installation of ESXi ■ **`sortedremoteesx`**- The first remote disk sorted by the lowest LUN ID detected containing a valid installation of ESXi ■ **`remote`** - The first remote disk detected by ESXi after boot ■ **`sortedremote`** - The first remote disk sorted by the lowest LUN ID detected by ESXi after boot ■ **`device model`** ■ **`device vendor`** ■ **`vmkernel device driver name`** You can get the values for the device model and vendor arguments by running the command `esxcli storage core device list` in a console to the ESXi host, logging in as the root. You get the vmkernel device driver name argument by running the command `esxcli storage core adapter list`. You must then identify the storage adapter to which your boot device is connected. By default, the system attempts to replace an existing ESXi installation, and then attempts to write to the local disk. You can use the **Arguments for first disk** field to specify a comma-separated list of disks to use, in order of preference. You can specify more than one disk. Use **`localesx`** for the first disk with ESX installed on it, model and vendor information, or specify the name of the vmkernel device driver. For example, to have the system first look for a disk with the model name ST3120814A, second for any disk that uses the mptsas driver, and third for the local disk, specify **`ST3120814A,mptsas,local`** as the value of this field. The first disk setting in the host profile specifies the search order for determining which disk to use for the cache. The search order is specified as a comma delimited list of values. The default setting **`localesx,local`** specifies that vSphere Auto Deploy should first look for an existing local cache disk. The cache disk is identified as a disk with an existing ESXi software image. If vSphere Auto Deploy cannot find an existing cache disk, it searches for an available local disk device. When searching for an available disk vSphere Auto Deploy uses the first empty disk that does not have an existing VMFS partition.

Option	Description
	You can use the first disk argument only to specify the search order. You cannot explicitly specify a disk. For example, you cannot specify a specific LUN on a SAN.
Check to overwrite any VMFS volumes on the selected disk	If you select this check box, the system overwrites existing VMFS volumes if not enough space is available to store the image, image profile, and host profile.
Check to ignore any SSD devices connected to the host	If you select this check box, the system ignores any existing SSD devices and does not store image profiles and host profiles on them.

1 In the **System Disk Configuration** drop-down menu, select **User must explicitly choose the policy option**.

b If you select **Enable stateless caching to a USB disk on the host**:

1 In the **System Disk Configuration** drop-down menu, select **System disk specified by user in host customization**

2 Under **Hosts and Clusters**, right-click the host and select **Host Profiles > Edit Host Customizations**. Define a disk with persistent storage in the **Value** field for the **System disk** property.

7 Click **Save** to complete the host profile configuration.

What to do next

Apply the host profile to individual hosts by using the Host Profiles feature in the vSphere Client. See the *Host Profiles* documentation. Alternatively, you can create a rule to assign the host profile to hosts with the vSphere Client or by using PowerCLI. See Write a Rule and Assign a Host Profile to Hosts.

- Create a rule that applies the host profile to all hosts that you want to provision with the settings specified in the reference host. For writing a rule in a PowerCLI session, see Write a Rule and Assign a Host Profile to Hosts.
- For hosts that are already provisioned with vSphere Auto Deploy, perform the test and repair compliance operations in a PowerCLI session, see Test and Repair Rule Compliance .
- Power on unprovisioned hosts to provision them with the new host profile.

Configure a Host Profile to Enable Stateful Installs

To set up an ESXi host provisioned with vSphere Auto Deploy to boot from disk, you must configure a host profile.

You can configure the host profile on a single host. You can also create a host profile on a reference host and apply that host profile to other hosts.

Prerequisites

- Decide which disk to use for storing the image, and determine whether the new image will overwrite an existing VMFS partition.

- Set up your environment for vSphere Auto Deploy. See Install and Configure vSphere Auto Deploy.
- Verify that a disk with at least 4GB of free space is available. If the disk is not yet partitioned, partitioning happens when you apply the host profile.
- Set up the host to boot from disk. See your hardware vendor's documentation.
- Create a host profile. See the *Host Profiles* documentation.

Procedure

1. Navigate to **Home > Policies and Profiles > Host Profiles**.
2. Click the host profile you want to configure and select the **Configure** tab.
3. Click **Edit Host Profile**.
4. On the Edit host profile page of the wizard, select **Advanced Configuration Settings > System Image Cache Configuration > System Image Cache Configuration**.
5. In the **System Image Cache Profile Settings** drop-down menu, choose a policy option.

Option	Description
Enable stateful installs on the host	Caches the image to a disk.
Enable stateful installs to a USB disk on the host	Caches the image to a USB disk attached to the host.

6 Depending on the policy option you select, you must do the following:

a If you select **Enable stateful installs on the host**:

1 Specify the information about the disk to use.

Option	Description
Arguments for first disk	When configuring a System Image Install disk, you have multiple options to define the device you want ESXi to be installed to and booted from. You can use the following arguments to define the disk for the installation: ■ **`localesx`** – The first disk detected containing a valid installation of ESXi ■ **`local`** – The first local disk detected by ESXi after boot ■ **`remoteesx`** - The first remote disk detected containing a valid installation of ESXi ■ **`sortedremoteesx`**- The first remote disk sorted by the lowest LUN ID detected containing a valid installation of ESXi ■ **`remote`** - The first remote disk detected by ESXi after boot ■ **`sortedremote`** - The first remote disk sorted by the lowest LUN ID detected by ESXi after boot ■ **`device model`** ■ **`device vendor`** ■ **`vmkernel device driver name`** You can get the values for the device model and vendor arguments by running the command `esxcli storage core device list` in a console to the ESXi host, logging in as the root. You get the vmkernel device driver name argument by running the command `esxcli storage core adapter list`. You must then identify the storage adapter to which your boot device is connected. By default, the system attempts to replace an existing ESXi installation, and then attempts to write to the local disk. You can use the **Arguments for first disk** field to specify a comma-separated list of disks to use, in order of preference. You can specify more than one disk. Use **`localesx`** for the first disk with ESX installed on it, model and vendor information, or specify the name of the vmkernel device driver. For example, to have the system first look for a disk with the model name ST3120814A, second for any disk that uses the mptsas driver, and third for the local disk, specify **`ST3120814A,mptsas,local`** as the value of this field. The first disk setting in the host profile specifies the search order for determining which disk to use for the cache. The search order is specified as a comma delimited list of values. The default setting **`localesx,local`** specifies that vSphere Auto Deploy should first look for an existing local cache disk. The cache disk is identified as a disk with an existing ESXi software image. If vSphere Auto Deploy cannot find an existing cache disk, it searches for an available local disk device. When searching for an available disk vSphere Auto Deploy uses the first empty disk that does not have an existing VMFS partition.

Option	Description
	You can use the first disk argument only to specify the search order. You cannot explicitly specify a disk. For example, you cannot specify a specific LUN on a SAN.
Check to overwrite any VMFS volumes on the selected disk	If you select this check box, the system overwrites existing VMFS volumes if not enough space is available to store the image, image profile, and host profile.
Check to ignore any SSD devices connected to the host	If you select this check box, the system ignores any existing SSD devices and does not store image profiles and host profiles on them.

1. In the **System Disk Configuration** drop-down menu, select **User must explicitly choose the policy option**.

b. If you select **Enable stateful installs to a USB disk on the host**:

1. In the **System Disk Configuration** drop-down menu, select **System disk specified by user in host customization**
2. Under **Hosts and Clusters**, right-click the host and select **Host Profiles > Edit Host Customizations**. Define a disk with persistent storage in the **Value** field for the **System disk** property.

7. Click **Save** to complete the host profile configuration.

What to do next

Apply the host profile to individual hosts by using the Host Profiles feature in the vSphere Client. See the *Host Profiles* documentation. Alternatively, you can create a rule to assign the host profile to hosts with the vSphere Client or by using PowerCLI. See Write a Rule and Assign a Host Profile to Hosts.

- Create a rule that applies the host profile to all hosts that you want to provision with the settings specified in the reference host. For writing a rule in a PowerCLI session, see Write a Rule and Assign a Host Profile to Hosts.
- For hosts that are already provisioned with vSphere Auto Deploy, perform the test and repair compliance operations in a PowerCLI session, see Test and Repair Rule Compliance .
- Power on unprovisioned hosts to provision them with the new host profile.

Setting Up a vSphere Auto Deploy Reference Host

In an environment where no state is stored on the ESXi host, a reference host helps you set up multiple hosts with the same configuration.

You configure the reference host with the logging, coredump, and other settings that you want, save the host profile, and write a rule that applies the host profile to other hosts as needed. You can configure the storage, networking, and security settings on the reference host and set up services such as syslog and NTP.

Understanding Reference Host Setup

A well-designed reference host connects to all services such as syslog, NTP, and so on. The reference host setup might also include security, storage, networking, and ESXi Dump Collector. You can apply such a host's setup to other hosts by using host profiles.

The exact setup of your reference host depends on your environment, but you might consider the following customization.

NTP Server Setup

When you collect logging information in large environments, you must make sure that log times are coordinated. Set up the reference host to use the NTP server in your environment that all hosts can share. You can specify an NTP server by running the `esxcli system ntp set` command. You can start and stop the NTP service for a host with the `esxcli system ntp set` command, or the vSphere Client.

Syslog Server Setup

All ESXi hosts run a syslog service (`vmsyslogd`), which logs messages from the VMkernel and other system components to a file. You can specify the log host and manage the log location, rotation, size, and other attributes by running the `esxcli system syslog` command or by using the vSphere Client. Setting up logging on a remote host is especially important for hosts provisioned with vSphere Auto Deploy that have no local storage.

You can optionally install VMware vCenter Log Insight, which provides log aggregation and analytics.

Core Dump Setup

You can set up your reference host to send core dumps to a shared SAN LUN, or you can install ESXi Dump Collector in your environment and configure the reference host to use ESXi Dump Collector. See Configure ESXi Dump Collector with ESXCLI . You can either install ESXi Dump Collector by using the vCenter Server installation media or use the ESXi Dump Collector that is included in vCenter Server. After setup is complete, VMkernel memory is sent to the specified network server when the system encounters a critical failure.

Security Setup

In most deployments, all hosts that you provision with vSphere Auto Deploy must have the same security settings. You can, for example, set up the firewall to allow certain services to access the ESXi system, set up the security configuration, user configuration, and user group configuration for the reference host with the vSphere Client or with ESXCLI commands. Security setup includes shared user access settings for all hosts. You can achieve unified user

access by setting up your reference host to use Active Directory. See the *vSphere Security* documentation.

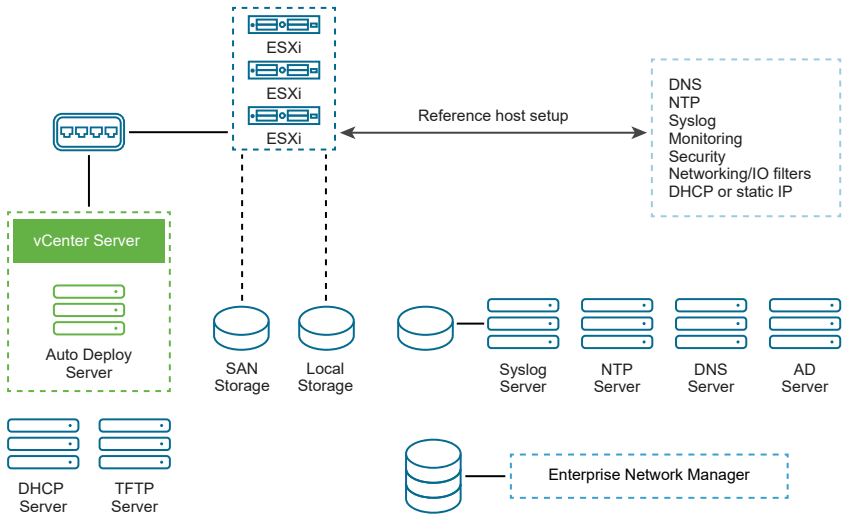
Note If you set up Active Directory by using host profiles, the passwords are not protected. Use the vSphere Authentication Service to set up Active Directory to avoid exposing the Active Directory password.

Networking and Storage Setup

If you reserve a set of networking and storage resources for use by hosts provisioned with vSphere Auto Deploy, you can set up your reference host to use those resources.

In large deployments, the reference host setup supports an Enterprise Network Manager, which collects all information coming from the different monitoring services that are running in the environment.

Figure 4-9. vSphere Auto Deploy Reference Host Setup



Options for Configuration of a vSphere Auto Deploy Reference Host explains how to perform this setup.

Watch the video "Auto Deploy Reference Hosts" for information about the reference host setup:

(vSphere Auto Deploy Reference Hosts)

Options for Configuration of a vSphere Auto Deploy Reference Host

You can configure a reference host by using the vSphere Client, ESXCLI, or host profiles.

To set up a reference host, you can use the approach that suits you best.

vSphere Client

The vSphere Client supports setup of networking, storage, security, and most other aspects of an ESXi host. Set up your environment and create a host profile from the reference host for use by vSphere Auto Deploy.

ESXCLI

You can use ESXCLI for setup of many aspects of your host. ESXCLI is suitable for configuring many of the services in the vSphere environment. Commands include `esxcli system ntp` for setting up an NTP server, `esxcli system syslog` for setting up a syslog server, `esxcli network route` for adding routes and set up the default route, and `esxcli system coredump` for configuring ESXi Dump Collector.

Host Profiles Feature

Best practice is to set up a host with vSphere Client or ESXCLI and create a host profile from that host. You can instead use the Host Profiles feature in the vSphere Client and save that host profile.

vSphere Auto Deploy applies all common settings from the host profile to all target hosts. If you set up the host profile to prompt for user input, all hosts provisioned with that host profile come up in maintenance mode. You must reapply the host profile or reset host customizations to be prompted for the host-specific information.

Configure an ESXi Dump Collector

You can configure an ESXi Dump Collector to store core dumps either by using ESXCLI commands or by configuring a reference host.

Hosts provisioned with vSphere Auto Deploy do not have a local disk on which to store core dumps. You can configure an ESXi Dump Collector to store core dumps either by using ESXCLI commands or by configuring a reference host to use the ESXi Dump Collector by using the Host Profiles feature in the vSphere Client.

Configure ESXi Dump Collector with ESXCLI

Hosts provisioned with vSphere Auto Deploy do not have a local disk to store core dumps on. You can configure ESXi Dump Collector by using ESXCLI commands and keep core dumps on a network server for use during debugging.

A core dump is the state of working memory if there is host failure. By default, a core dump is saved to the local disk. ESXi Dump Collector is especially useful for vSphere Auto Deploy, but is supported for any ESXi host. ESXi Dump Collector supports other customization, including sending core dumps to the local disk and is included with the vCenter Server management node.

Note ESXi Dump Collector is not supported to be configured on a VMkernel interface that is running on a NSX-T N-VDS switch.

If you intend to use IPv6, and if both the ESXi host and ESXi Dump Collector are on the same local link, both can use either local link scope IPv6 addresses or global scope IPv6 addresses.

If you intend to use IPv6, and if ESXi and ESXi Dump Collector are on different hosts, both require global scope IPv6 addresses. The traffic routes through the default IPv6 gateway.

Prerequisites

Install ESXCLI if you want to configure the host to use ESXi Dump Collector. In troubleshooting situations, you can use ESXCLI in the ESXi Shell instead.

Procedure

1 Set up an ESXi system to use ESXi Dump Collector by running `esxcli system coredump` in the local ESXi Shell or by using ESXCLI.

```
 esxcli system coredump network set --interface-name vmk0    --server-ip 10xx.xx.xx.xx --
server-port 6500
```

You must specify a VMkernel NIC and the IP address and optional port of the server to send the core dumps to. You can use an IPv4 address or an IPv6 address. If you configure an ESXi system that is running on a virtual machine that is using a vSphere standard switch, you must select a VMkernel port that is in promiscuous mode.

2 Enable ESXi Dump Collector.

```
esxcli system coredump network set --enable true
```

3 (Optional) Verify that ESXi Dump Collector is configured correctly.

```
esxcli system coredump network check
```

Results

The host on which you have set up ESXi Dump Collector is configured to send core dumps to the specified server by using the specified VMkernel NIC and optional port.

What to do next

- Create a rule that applies the host profile to all hosts that you want to provision with the settings specified in the reference host. For writing a rule in a PowerCLI session, see Write a Rule and Assign a Host Profile to Hosts.
- For hosts that are already provisioned with vSphere Auto Deploy, perform the test and repair compliance operations in a PowerCLI session, see Test and Repair Rule Compliance .
- Power on unprovisioned hosts to provision them with the new host profile.

Configure ESXi Dump Collector from the Host Profiles Feature in the vSphere Client

Hosts provisioned with vSphere Auto Deploy do not have a local disk to store core dumps on. You can configure a reference host to use ESXi Dump Collector by using the Host Profiles feature in the vSphere Client.

Best practice is to set up hosts to use ESXi Dump Collector with the `esxcli system coredump` command and save the host profile. For more information, see Configure ESXi Dump Collector with ESXCLI .

Prerequisites

- Verify that you have created the host profile on which you want to configure a coredump policy. For more information on how to create a host profile, see the *vSphere Host Profiles* documentation.
- Verify that at least one partition has sufficient storage capability for core dumps from multiple hosts provisioned with vSphere Auto Deploy.

Procedure

1. Navigate to **Home > Policies and Profiles > Host Profiles**.
2. Click the host profile you want to configure and select the **Configure** tab.
3. Click **Edit Host Profile**.
4. Select **Networking Configuration > Network Coredump Settings**.
5. Select the **Enabled** check box.
6. Specify the host NIC to use, the Network Coredump Server IP, and the Network Coredump Server Port.
7. Click **Save** to complete the host profile configuration.

What to do next

- Create a rule that applies the host profile to all hosts that you want to provision with the settings specified in the reference host. For writing a rule in a PowerCLI session, see Write a Rule and Assign a Host Profile to Hosts.
- For hosts that are already provisioned with vSphere Auto Deploy, perform the test and repair compliance operations in a PowerCLI session, see Test and Repair Rule Compliance .
- Power on unprovisioned hosts to provision them with the new host profile.

Configure Syslog from Host Profiles in the vSphere Client

Specify a remote syslog server by applying a host profile.

Best practice is to set up the syslog server on the reference host with the vSphere Client or the `esxcli system syslog` command and to save the host profile. You can also set up syslog from the Host Profiles feature in the vSphere Client.

Prerequisites

Hosts provisioned with vSphere Auto Deploy usually do not have sufficient local storage to save system logs. You can specify a remote syslog server for those hosts by setting up a reference host, saving the host profile, and applying that host profile to other hosts as needed.

- If you intend to use a remote syslog host, set up that host before you customize host profiles.
- Verify that you have access to the vSphere Client and the vCenter Server system.

Procedure

1. Navigate to **Home > Policies and Profiles > Host Profiles**.
2. (Optional) If no reference host exists in your environment, click **Extract Profile from Host** to create a host profile.
3. Click the host profile you want to configure and select the **Configure** tab.
4. Click **Edit Host Profile**.
5. Select **Advanced Configuration Settings > Advanced Options > Advanced configuration options**.

   You can select specific sub-profiles and edit the syslog settings.
6. (Optional) To create an advanced configuration option.
   a. Click the **Add sub-profile** icon.
   b. From the **Advanced option** drop-down list select **Configure a fixed option**.
   c. Specify *Syslog.global.loghost* as the name of the option, and your host as the value of the option.
7. Click **Save** to complete the host profile configuration.

What to do next

- Create a rule that applies the host profile to all hosts that you want to provision with the settings specified in the reference host. For writing a rule in a PowerCLI session, see Write a Rule and Assign a Host Profile to Hosts.
- For hosts that are already provisioned with vSphere Auto Deploy, perform the test and repair compliance operations in a PowerCLI session, see Test and Repair Rule Compliance .
- Power on unprovisioned hosts to provision them with the new host profile.

Enable NTP Client on a Reference Host in the vSphere Client

When you collect logging information in large environments, you must ensure that log times are coordinated.

You can set up the reference host to use the NTP server in your environment, extract the host profile and create a vSphere Auto Deploy rule to apply it to other hosts.

Procedure

1 Navigate to **Home > Hosts and Clusters**, and select an ESXi host that you want to use as a reference host.

2 Select the **Configure** tab.

3 Under **System**, select **Time Configuration** and click **Edit**.

4 Select the **Use Network Time Protocol (Enable NTP client)** radio button.

This option synchronizes the time and date of the host with an NTP server. The NTP service on the host periodically takes the time and date from the NTP server.

5 From the **NTP Service Startup Policy** drop-down menu, select **Start and stop with host**.

6 In the **NTP Servers** text box, enter the IP addresses or host names of the NTP servers that you want to use.

7 Click **OK**.

What to do next

- Extract a host profile from the reference host. See the *Host Profiles* documentation.
- Create a rule that applies the host profile to all hosts that you want to provision with the settings specified in the reference host. For writing a rule in a PowerCLI session, see Write a Rule and Assign a Host Profile to Hosts.

Configure Networking for Your vSphere Auto Deploy Host in the vSphere Client

You can set up networking for your vSphere Auto Deploy reference host and apply the host profile to all other hosts to guarantee a fully functional networking environment.

Prerequisites

Provision the host you want to use as your reference host with an ESXi image by using vSphere Auto Deploy.

Procedure

1 Navigate to **Home > Hosts and Clusters**, and select an ESXi host that you want to use as a reference host.

2 Select the **Configure** tab and navigate to **Networking**.

3 Perform the networking setup.

If you are using virtual switches and not vSphere Distributed Switch, do not add other VMkernel NICs to vSwitch0.

4 After the reference host is configured, reboot the system to verify that vmk0 is connected to the Management Network.

5 If no host profile exists for your reference host, create a host profile.

What to do next

- Create a rule that applies the host profile to all hosts that you want to provision with the settings specified in the reference host. For writing a rule in a PowerCLI session, see Write a Rule and Assign a Host Profile to Hosts.
- For hosts that are already provisioned with vSphere Auto Deploy, perform the test and repair compliance operations in a PowerCLI session, see Test and Repair Rule Compliance .
- Power on unprovisioned hosts to provision them with the new host profile.

Configure a Reference Host for Auto-Partitioning

You can create a reference host to auto-partition all host you provision with vSphere Auto Deploy.

Caution If you change the default auto-partitioning behavior, vSphere Auto Deploy overwrites existing partitions regardless of their content. If you turn on this option, ensure that no unintended data loss results.

To ensure that local SSDs remain unpartitioned during auto-partitioning, you must set the parameter **`skipPartitioningSsds=TRUE`** on the reference host.

For more information about preventing SSD formatting during auto-partitioning, see the *vSphere Storage* documentation.

Prerequisites

By default, vSphere Auto Deploy provisions hosts only if a partition is available on the host. The auto-partitioning option creates a VMFS datastore on your host's local storage. You can set up a reference host to auto-partition all hosts that you provision with vSphere Auto Deploy.

- Provision the host that you want to use as your reference host with an ESXi image by using vSphere Auto Deploy.
- Verify that you have access to avSphere Client that can connect to the vCenter Server system.

Procedure

1. Navigate to **Home > Hosts and Clusters**, and select an ESXi host that you want to use as a reference host.
2. Select the **Configure** tab.
3. Under **System**, select **Advanced System Settings** and click **Edit**.
4. Search for the `VMkernel.Boot.autoPartition` key and set the value to **true**.
5. (Optional) If you want the local SSDs to remain unpartitioned, search for the `VMkernel.Boot.skipPartitioningSsds` key and set the value to **true**.
6. Click **OK**.

7 If no host profile exists for your reference host, create a host profile.

Results

Auto-partitioning is performed when the hosts boot.

What to do next

- Use vSphere Auto Deploy to create a rule that applies the host profile of your reference host to all hosts immediately when they boot. To create a rule with the vSphere Client, see Create a Deploy Rule. For writing a rule in a PowerCLI session, see Write a Rule and Assign a Host Profile to Hosts.

Converting Stateless Hosts to Stateful Hosts

You can add physical storage to your stateless ESXi, convert them to stateful ESXi hosts, and add the hosts to a cluster that you manage by an image.

Your ESXi hosts must have physical storage attached, because a cluster managed by an image does not support stateless ESXi hosts, that use a Preboot Execution Environment (PXE) boot.

To ensure that image components are installed on a physical disk of an ESXi host during a boot process, Auto Deploy verifies that a host profile is part of the PXE boot image. The host profile must contain a "System Image Cache Profile Settings" policy with configuration set to "Enable stateful installs on the host" or "Enable stateful installs to a USB disk on the host". If the attached host profile does not contain this policy or the policy's configuration differs, then the policy is automatically configured to support a stateful installation. If a host profile is missing, a new host profile is attached to the cluster, containing only the "System Image Cache Profile Settings" policy with a configuration for a stateful installation.

Convert a Diskless ESXi Host

To add diskless ESXi hosts to a cluster managed by an image, add physical storage, convert them to stateful hosts and install an ESXi 8.0 image.

For information on stateless caching and stateful installs, see Use vSphere Auto Deploy for Stateless Caching and Stateful Installs.

Prerequisites

- Prepare your system and install the Auto Deploy Server. For more information, see *Prepare Your System for vSphere Auto Deploy* in the *VMware ESXi Installation and Setup* documentation.
- The ESXi host is booting from an Auto Deploy rule that contains an Image Profile.
- The Auto Deploy service is running.
- You have an empty USB stick or other supported storage.

Procedure

1 From the home menu, click **Policies and Profiles**, and select **Host Profiles**.

2 Extract a host profile from a running ESXi host or duplicate an existing host profile with a specified configuration and a host location.

   The existing host profile can be attached either to a host or to a cluster.

3 Right-click the new host profile, select **Edit Host Profile**, and browse to **Advanced Configuration Settings > System Image Cache Configuration > System Image Cache Configuration**.

4 In the **System Image Cache Profile Settings** drop-down menu, choose a policy option.

Option	Description
Enable stateless caching on the host	Caches the image to disk.
Enable stateless caching to a USB disk on the host	Caches the image to a USB disk attached to the host.

5 Depending on the policy option you select, you must do the following:

   a If you select **Enable stateless caching on the host**:

      1 Enter arguments for the first disk, and select the check boxes if needed.

      2 In the **System Disk Configuration** drop-down menu, select **User must explicitly choose the policy option**.

   b If you select **Enable stateless caching to a USB disk on the host**:

      1 In the **System Disk Configuration** drop-down menu, select **System disk specified by user in host customization**

      2 Under **Hosts and Clusters**, right-click the host and select **Host Profiles > Edit Host Customizations**. Define a disk with persistent storage in the **Value** field for the **System disk** property.

6 Click **Save** to complete the host profile configuration.

7 If you duplicated an existing host profile and the host profile was attached to a cluster, attach the new host profile to the cluster.

8 From the home menu, click **Auto Deploy**.

9 Deactivate the Auto Deploy rule that contains an Image Profile, and click **Edit**.

   The Edit Deploy Rule dialog box appears.

10 On the **Select image profile** page of the wizard, select an ESXi 8.0 image profile.

11 On the **Select host profile** page of the wizard, select the new host profile.

12 Activate the rule and move the rule to the rule's initial position in the ordered list.

13 On the **Deployed Hosts** tab, select a single or multiple ESXi hosts.

14 Click **Remediate Host Associations** for the ESXi host.

   You can monitor the progress of the remediation process in the Recent Tasks pane.

15 Shut down the remediated ESXi hosts.

16 After the ESXi hosts are powered off, install the selected storage as local boot disks.

17 Power on each ESXi host, enter the BIOS/UEFI setup, and change the boot order to boot first from the newly added storage and then from the network.

Since the newly added storage is empty, each ESXi host boots from the network and installs the ESXi 8.0 image that you specified earlier on the storage. After the installation, each ESXi host reboots and boots from the newly added storage.

Results

The ESXi hosts boot by default from the new storage and operate as if the ESXi 8.0 image is installed from a standard DVD.

What to do next

Add the ESXi hosts to a cluster that you manage by a single image. For more information, see *Add a Host* from the *vCenter Server and Host Management* documentation.

Convert a Stateless ESXi Host with Enabled Stateless Caching

To add your stateless ESXi hosts to a cluster that you manage by an image, convert them to stateful hosts by installing a standard 8.0 8.0 image.

For information on stateless caching and stateful installs, see Use vSphere Auto Deploy for Stateless Caching and Stateful Installs.

Prerequisites

- Prepare your system and install the Auto Deploy Server. For more information, see *Prepare Your System for vSphere Auto Deploy* in the *VMware ESXi Installation and Setup* documentation.
- The ESXi host is booting from an Auto Deploy rule that contains an Image Profile.
- The Auto Deploy service is running.

Procedure

1 From the home menu, click **Policies and Profiles**, and select **Host Profiles**.

2 Extract a host profile from a running ESXi host or duplicate an existing host profile with a specified configuration and a host location.

The existing host profile can be attached either to a host or to a cluster.

3 Right-click the new host profile, select **Edit Host Profile**, and browse to **Advanced Configuration Settings > System Image Cache Configuration > System Image Cache Configuration**.

4 Select **Enable stateful installs on the host** from the drop-down menu, enter `local` in the **Arguments for first disk** text box, and select the **Check to overwrite any VMFS volumes on the selected disk** check box.

5 If you duplicated an existing host profile and the host profile was attached to a cluster, attach the new host profile to the cluster.

6 From the home menu, click **Auto Deploy**.

7 Deactivate the Auto Deploy rule that contains an Image Profile, and click **Edit**.

The Edit Deploy Rule dialog box appears.

8 On the **Select image profile** page of the wizard, select an ESXi 8.0 image profile.

9 On the **Select host profile** page of the wizard, select the new host profile.

10 Activate the rule and move the rule to the rule's initial position in the ordered list.

11 On the **Deployed Hosts** tab, select a single or multiple ESXi hosts.

12 Click **Remediate Host Associations** for the ESXi host.

You can monitor the progress of the remediation process in the Recent Tasks pane.

13 Restart the ESXi hosts.

If an ESXi host has a legacy BIOS, you can change the boot order in the BIOS setup to first to boot from the local storage. For UEFI-based ESXi hosts the boot order changes automatically during the reboot.

Each ESXi host boots from the network and installs the ESXi 8.0 image that you specified earlier on the storage used previously for caching. After the installation, each ESXi host reboots again and boots from the local storage.

Results

The ESXi hosts boot by default from the new storage and operate as if the ESXi 8.0 image is installed from a standard DVD.

What to do next

Add the ESXi hosts to a cluster that you manage by a single image. For more information, see *Add a Host* from the *vCenter Server and Host Management* documentation.

Convert a Stateless ESXi Host with a Single VMFS Partition on the Local Disk

To add your stateless ESXi hosts to a cluster that you manage by an image, you must first convert the hosts to stateful hosts by repartitioning the VMFS partitions and installing a standard ESXi 8.0 image.

For information on stateless caching and stateful installs, see Use vSphere Auto Deploy for Stateless Caching and Stateful Installs.

Prerequisites

- Prepare your system and install the Auto Deploy Server. For more information, see *Prepare Your System for vSphere Auto Deploy* in the *VMware ESXi Installation and Setup* documentation.

- The ESXi host is booting from an Auto Deploy rule that contains an Image Profile.
- The Auto Deploy service is running.

Procedure

1. From the home menu, click **Policies and Profiles**, and select **Host Profiles**.
2. Extract a host profile from a running ESXi host or duplicate an existing host profile with a specified configuration and a host location.

   The existing host profile can be attached either to a host or to a cluster.
3. Right-click the new host profile, select **Edit Host Profile**, and browse to **Advanced Configuration Settings > System Image Cache Configuration > System Image Cache Configuration**.
4. Select **Enable stateful installs on the host** from the drop-down menu, enter `localesx` in the **Arguments for first disk** text box.
5. If you duplicated an existing host profile and the host profile was attached to a cluster, attach the new host profile to the cluster.
6. From the home menu, click **Auto Deploy**.
7. Deactivate the Auto Deploy rule that contains an Image Profile, and click **Edit**.

   The Edit Deploy Rule dialog box appears.
8. On the **Select image profile** page of the wizard, select an ESXi 8.0 image profile.
9. On the **Select host profile** page of the wizard, select the new host profile.
10. Activate the rule and move the rule to the rule's initial position in the ordered list.
11. On the **Deployed Hosts** tab, select a single or multiple ESXi hosts.
12. Click **Remediate Host Associations** for the ESXi host.

    You can monitor the progress of the remediation process in the Recent Tasks pane.
13. Restart the ESXi hosts.

    If an ESXi host has a legacy BIOS, you can change the boot order in the BIOS setup to first to boot from the local storage. For UEFI-based ESXi hosts the boot order changes automatically during the reboot.

    Each ESXi host boots from the network, repartitions the VMFS partition to a standard ESXi installation partition, and installs the ESXi 8.0 image that you specified earlier on the partition. After the installation, each ESXi host reboots again and boots from the standard ESXi partition.

Results

The ESXi hosts boot by default from the new partition and operate as if the ESXi 8.0 image is installed from a standard DVD.

What to do next

Add the ESXi hosts to a cluster that you manage by a single image. For more information, see *Add a Host* from the *vCenter Server and Host Management* documentation.

Convert a Stateless ESXi Host with a Single VMFS Partition on a Remote Disk

To add your stateless ESXi hosts to a cluster that you manage by an image, first you must convert the hosts to stateful hosts by repartitioning their remote VMFS partitions and installing a standard ESXi 8.0 image. For example, your ESXi host can boot from Fibre Channel SAN or from iSCSI SAN.

For information on stateless caching and stateful installs, see Use vSphere Auto Deploy for Stateless Caching and Stateful Installs.

For more information on booting from a remote disk, see the *Booting from iSCSI SAN* in the *vSphere Storage* documentation.

Prerequisites

- Prepare your system and install the Auto Deploy Server. For more information, see *Prepare Your System for vSphere Auto Deploy* in the *VMware ESXi Installation and Setup* documentation.
- The ESXi host is booting from an Auto Deploy rule that contains an Image Profile.
- The Auto Deploy service is running.

Procedure

1. From the home menu, click **Policies and Profiles**, and select **Host Profiles**.
2. Extract a host profile from a running ESXi host or duplicate an existing host profile with a specified configuration and a host location.

   The existing host profile can be attached either to a host or to a cluster.
3. Right-click the new host profile, select **Edit Host Profile**, and browse to **Advanced Configuration Settings > System Image Cache Configuration > System Image Cache Configuration**.
4. Select **Enable stateful installs on the host** from the drop-down menu, enter `remoteesx,remote` or `sortedremoteesx,sortedremote` in the **Arguments for first disk** text box.
5. If you duplicated an existing host profile and the host profile was attached to a cluster, attach the new host profile to the cluster.
6. From the home menu, click **Auto Deploy**.
7. Deactivate the Auto Deploy rule that contains an Image Profile, and click **Edit**.

   The Edit Deploy Rule dialog box appears.
8. On the **Select image profile** page of the wizard, select an ESXi 8.0 image profile.

9 On the **Select host profile** page of the wizard, select the new host profile.

10 Activate the rule and move the rule to the rule's initial position in the ordered list.

11 On the **Deployed Hosts** tab, select a single or multiple ESXi hosts.

12 Click **Remediate Host Associations** for the ESXi host.

You can monitor the progress of the remediation process in the Recent Tasks pane.

13 Restart the ESXi hosts.

Each ESXi host boots from the network, repartitions the VMFS partition to a standard ESXi installation partition, and installs the ESXi 8.0 image that you specified earlier on the partition. After the installation, each ESXi host reboots again and boots from the remote standard ESXi partition.

Results

The ESXi hosts boot by default from the new partition and operate as if the ESXi 8.0 image is installed from a standard DVD.

What to do next

Add the ESXi hosts to a cluster that you manage by a single image. For more information, see *Add a Host* from the *vCenter Server and Host Management* documentation.

vSphere Auto Deploy Best Practices

Set up a highly available vSphere Auto Deploy infrastructure in large production environments or when using stateless caching.

Follow best practices when installing vSphere Auto Deploy and when using vSphere Auto Deploy with other vSphere components. Set up a highly available vSphere Auto Deploy infrastructure in large production environments or when using stateless caching. Follow all security guidelines that you would follow in a PXE boot environment, and consider the recommendations in this chapter.

vSphere Auto Deploy Best Practices

Follow vSphere Auto Deploy best practices to set up networking, configure vSphere HA, and optimize your environment for vSphere Auto Deploy.

See the VMware Knowledge Base for additional best practice information.

vSphere Auto Deploy and vSphere HA Best Practices

You can improve the availability of the virtual machines running on hosts provisioned with vSphere Auto Deploy by following best practices.

Some environments configure the hosts provisioned with vSphere Auto Deploy with a distributed switch or configure virtual machines running on the hosts with Auto Start Manager. In such environments, deploy the vCenter Server system so that its availability matches the availability of the vSphere Auto Deploy server. Several approaches are possible.

- Deploy vCenter Server. The vSphere Auto Deploy server is included.

- Run vCenter Server in a vSphere HA enabled cluster and configure the virtual machine with a vSphere HA restart priority of high. Include two or more hosts in the cluster that are not managed by vSphere Auto Deploy and pin the vCenter Server virtual machine to these hosts by using a rule (vSphere HA DRS required VM to host rule). You can set up the rule and then deactivate DRS if you do not want to use DRS in the cluster. The greater the number of hosts that are not managed by vSphere Auto Deploy, the greater your resilience to host failures.

  Note This approach is not suitable if you use Auto Start Manager. Auto Start Manager is not supported in a cluster enabled for vSphere HA.

vSphere Auto Deploy Networking Best Practices

Prevent networking problems by following vSphere Auto Deploy networking best practices.

vSphere Auto Deploy and IPv6

Because vSphere Auto Deploy takes advantage of the iPXE infrastructure, if the hosts that you plan to provision with vSphere Auto Deploy are with legacy BIOS, the vSphere Auto Deploy server must have 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.

IP Address Allocation

Use DHCP reservations for address allocation. Fixed IP addresses are supported by the host customization mechanism, but providing input for each host is not recommended.

VLAN Considerations

Use vSphere Auto Deploy in environments that do not use VLANs.

If you intend to use vSphere Auto Deploy in an environment that uses VLANs, make sure that the hosts that you want to provision can reach the DHCP server. How hosts are assigned to a VLAN depends on the setup at your site. The VLAN ID might be assigned by the switch or the router, or might be set in the host's BIOS or through the host profile. Contact your network administrator to determine the steps for allowing hosts to reach the DHCP server.

vSphere Auto Deploy and VMware Tools Best Practices

When you provision hosts with vSphere Auto Deploy, you can select an image profile that includes VMware Tools, or select the smaller image associated with the image profile that does not contain VMware Tools.

You can download two image profiles from the VMware download site.

- *xxxxx*`-standard`: An image profile that includes the VMware Tools binaries, required by the guest operating system running inside a virtual machine. The image is usually named esxi-*version*-*xxxxx*-standard.
- *xxxxx*`-no-tools`: An image profile that does not include the VMware Tools binaries. This image profile is usually smaller has a lower memory overhead, and boots faster in a PXE-boot environment. This image is usually named esxi-*version*-*xxxxx*-no-tools.

You can deploy ESXi using either image profile.

- If the network boot time is of no concern, and your environment has sufficient extra memory and storage overhead, use the image that includes VMware Tools.
- If you find the network boot time too slow when using the standard image, or if you want to save some space on the hosts, you can use the image profile that does not include VMware Tools, and place the VMware Tools binaries on shared storage. See, Provision ESXi Host with an Image Profile Without VMware Tools.

vSphere Auto Deploy Load Management Best Practices

Simultaneously booting large numbers of hosts places a significant load on the vSphere Auto Deploy server. Because vSphere Auto Deploy is a Web server at its core, you can use existing Web server scaling technologies to help distribute the load. For example, one or more caching reverse proxy servers can be used with vSphere Auto Deploy. The reverse proxies serve up the static files that make up the majority of an ESXi boot image. Configure the reverse proxy to cache static content and pass all requests through to the vSphere Auto Deploy server. For more information, watch the video "Using Reverse Web Proxy Servers for vSphere Auto Deploy Scalability":

 (Using Reverse Web Proxy Servers for vSphere Auto Deploy Scalability)

Use multiple TFTP servers to point to different proxy servers. Use one TFTP server for each reverse proxy server. After that, set up the DHCP server to send different hosts to different TFTP servers.

When you boot the hosts, the DHCP server redirects them to different TFTP servers. Each TFTP server redirects hosts to a different server, either the vSphere Auto Deploy server or a reverse proxy server, significantly reducing the load on the vSphere Auto Deploy server.

After a massive power outage, bring up the hosts on a per-cluster basis. If you bring multiple clusters online simultaneously, the vSphere Auto Deploy server might experience CPU bottlenecks. All hosts might come up after a delay. The bottleneck is less severe if you set up the reverse proxy.

vSphere Auto Deploy Logging and Troubleshooting Best Practices

To resolve problems that you encounter with vSphere Auto Deploy, use the vSphere Auto Deploy logging information from the vSphere Client and set up your environment to send logging information and core dumps to remote hosts.

vSphere Auto Deploy Logs

Download the vSphere Auto Deploy logs by going to the vSphere Auto Deploy page in the vSphere Client. See, Download vSphere Auto Deploy Logs.

Setting Up Syslog

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, enables network logging, and lets you combine logs from multiple hosts.

Setting Up ESXi Dump Collector

Hosts provisioned with vSphere Auto Deploy do not have a local disk to store core dumps on. Install ESXi Dump Collector and 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. See Configure ESXi Dump Collector with ESXCLI .

Using vSphere Auto Deploy in a Production Environment

When you move from a proof of concept setup to a production environment, take care to make the environment resilient.

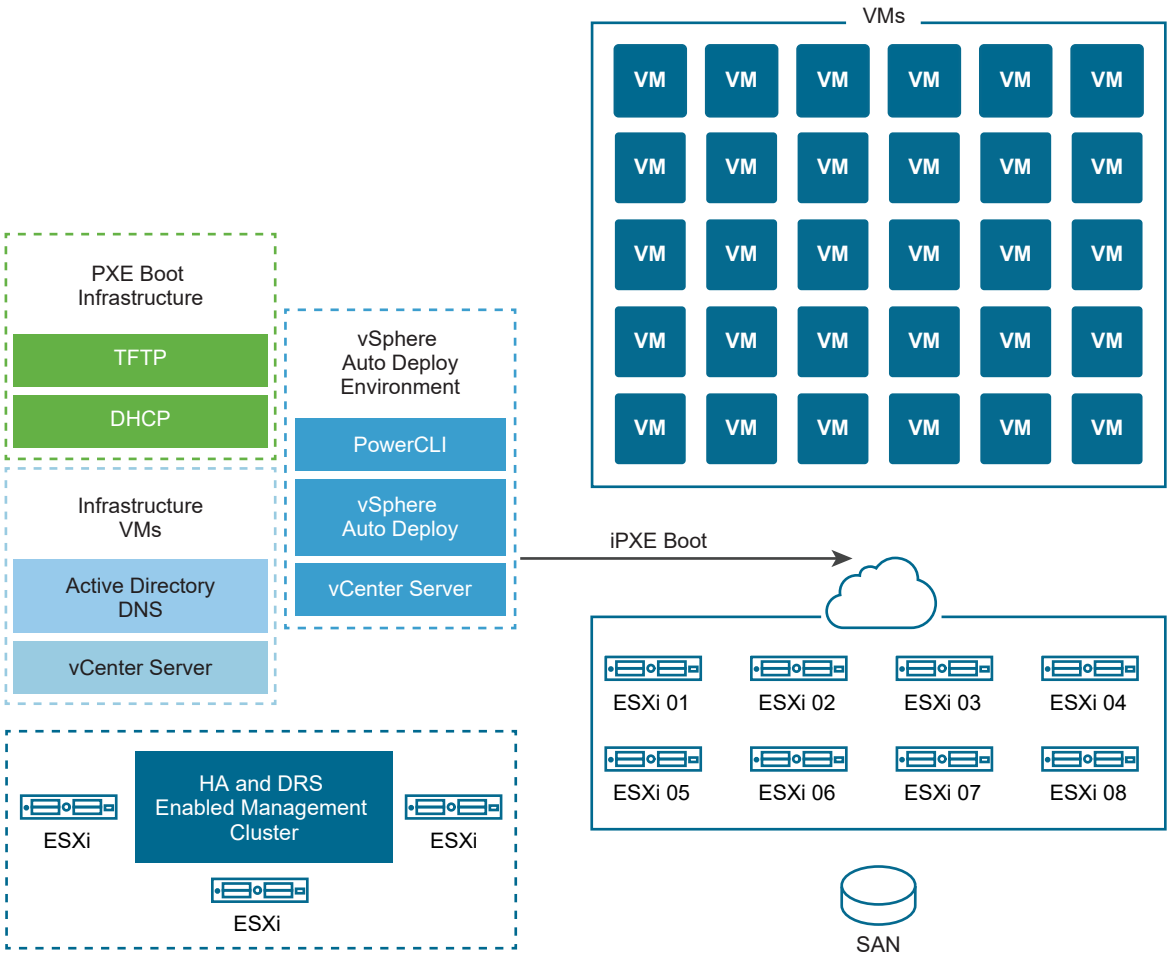
- Protect the vSphere Auto Deploy server. See vSphere Auto Deploy and vSphere HA Best Practices.
- Protect all other servers in your environment, including the DHCP server and the TFTP server.
- Follow VMware security guidelines, including those outlined in vSphere Auto Deploy Security Considerations.

Set Up Highly Available vSphere Auto Deploy Infrastructure

Highly available vSphere Auto Deploy infrastructure prevents data loss and is a prerequisite for using vSphere Auto Deploy with stateless caching.

 (Highly Available vSphere Auto Deploy Infrastructure)

Figure 4-10. Highly Available vSphere Auto Deploy Infrastructure



Prerequisites

In many production situations, a highly available vSphere Auto Deploy infrastructure is required to prevent data loss. Such infrastructure is also a prerequisite for using vSphere Auto Deploy with stateless caching.

For the management cluster, install ESXi on three hosts. Do not provision the management cluster hosts with vSphere Auto Deploy.

Watch the video "Highly Available vSphere Auto Deploy Infrastructure" for information about the implementation of a highly available vSphere Auto Deploy infrastructure:

Procedure

1 Enable vSphere HA and vSphere DRS on the management cluster.

2 Set up the following virtual machines on the management cluster.

Infrastructure Component	Description
PXE boot infrastructure	TFTP and DHCP servers.
Infrastructure VM	Active Directory, DNS, vCenter Server.
vSphere Auto Deploy environment	PowerCLI, vSphere Auto Deploy server, vCenter Server. Set up this environment on a single virtual machine or on three separate virtual machines in production systems.

The vCenter Server on the infrastructure virtual machine differs from the vCenter Server in the vSphere Auto Deploy environment.

3 Set up vSphere Auto Deploy to provision other hosts as needed.

Because the components on the management cluster are protected with vSphere HA, high availability is supported.

vSphere Auto Deploy Security Considerations

When you use vSphere Auto Deploy, mind networking security, boot image security, and potential password exposure to protect your environment.

Networking Security

Secure your network just as you secure the network 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 of the Auto Deploy server is not checked during a PXE boot.

You can greatly reduce the security risk of Auto Deploy by completely isolating the network where Auto Deploy is used.

Boot Image and Host Profile Security

The boot image that the vSphere Auto Deploy server downloads to a machine can have the following components.

- The VIB packages that the image profile consists of are always included in the boot image.
- The host profile and host customization are included in the boot image if Auto Deploy rules are set up to provision the host with a host profile or host customization.
  - The administrator (root) password and user passwords that are included with host profile and host customization are hashed with SHA-512.
  - Any other passwords associated with profiles are in the clear. If you set up Active Directory by using host profiles, the passwords are not protected.

    Use the vSphere Authentication Proxy to avoid exposing the Active Directory passwords. If you set up Active Directory using host profiles, the passwords are not protected.
- The host's public and private SSL key and certificate are included in the boot image.

Device Alias Configuration

Device aliases, also called device names, are short names associated with I/O adapters in an I/O subsystem.

For example, network uplinks have aliases such as vmnic0, vmnic1, and so on. SCSI adapter objects in the storage subsystem and graphics device objects also have aliases. A hardware device can be presented as multiple I/O adapters in the I/O subsystem. The I/O adapters can be of a different type from the underlying physical device. For example, an FCoE device is a storage I/O adapter that uses NIC hardware. Software iSCSI is a storage adapter using the network stack at the IP layer. Therefore, in the ESXi native driver model, aliases formally refer only to I/O adapters, and not to physical devices such as a PCI NIC or a PCI HBA.

Device Alias Assignment

A stateless ESXi deployment model is one where the ESXi host is not installed on hard disks, and is typically booted by using PXE. A stateful ESXi deployment model is one where the ESXi host is installed on local hard disks. Device alias assignment occurs during a stateless ESXi boot or a fresh installation of stateful ESXi. The ESXi host assigns aliases to I/O adapters in an order which is based on the underlying hardware enumeration order. The ESXi host assigns aliases first to on-board devices and then to add-in cards based on slot order. The ESXi host cannot assign aliases to absent devices or devices without supported drivers.

An uplink that uses a NIC that is built into the motherboard receives a vmnicN alias with lower number compared to an uplink of a PCI add-in card. The NIC driver might register more than one uplink. If one of the uplinks does not correspond to an enumerable hardware device, the ESXi host assigns the next available alias to the uplink after the uplink is registered with the system.

Persistence of Device Alias Configuration

After the ESXi host assigns aliases, alias configuration is persisted. The ESXi host attempts to keep the alias of each device the same regardless of the ESXi version updates, or hardware changes, such as adding or removing devices from slots.

The persistence of the alias configuration depends on the deployment model.

- In stateful systems, the alias configuration is persisted locally on the host.
- In stateless systems, if you do not manage the stateless system by using host profiles, the alias configuration is not persisted locally on the host.
- In stateful and stateless systems that you manage by using host profiles, the alias configuration is persisted in the host profile. If you apply a host profile to a stateful host, the host profile overrides any locally persisted alias configuration.

Changes in the Device Alias Configuration

The persistence of alias configuration is based on the bus addresses of devices. If the bus address of a device is altered, the persisted alias configuration becomes inapplicable and the aliases assigned to the device might change.

Changes in the device alias configuration might occur in the following cases:

- A driver upgrade might enumerate or present an I/O adapter differently to the system compared to how the I/O adapter is presented before the driver upgrade.
- A stack upgrade might result in changes to parts of a multi-module driver setup, or to the ESXi I/O stack that supports a multi-module driver.
- BIOS or device firmware upgrades might lead to incomplete port or slot information.
- Changes in the slot position of a device.

Note If you remove a device from the system, the alias configuration of the I/O adapters of the device is removed. If you add the same devices back to the system later, the I/O adapters of the device might not receive their previous aliases.

Device Alias Configuration in ESXi Clusters

Initial alias configuration is the same across a cluster of identical systems. However, even on a cluster that is considered homogenous, small differences in hardware or firmware might result in differences in the alias configuration between hosts.

Differences in the processing order during driver binding can also result in differences in alias configuration. For example, a NIC driver registers two uplinks, uplink-1 and uplink-2, for two ports of the same PCI device, where one of the ports is not hardware enumerable by the system. Timing changes in the order of registration of the uplinks might result in differences in how the ESXi hosts assign aliases to the uplinks. One ESXi host might assign the hardware-based alias to uplink-1, and another ESXi host might assign the hardware-based alias to uplink-2.

To match alias configuration across homogeneous hosts, you can use host profiles. The Device Alias Configuration host profile applies alias configuration to an ESXi host by mapping devices in the alias configuration to the ESXi host devices. The mapping operation is based on the hardware information sources that are used as the basis for initial alias assignment. For more information about information sources used for alias assignment, see the Knowledge Base article KB 2091560.

The Device Alias Configuration host profile also flags errors, for example, when a device is present in the host profile but not present on the host.

A heterogenous cluster does not have the same default alias configuration across its hosts. Due to the differences between the devices, a host profile cannot be applied cleanly.

Using ESXi Shell Commands to View Device Alias Information

On a running ESXi system, you can view information about I/O adapter aliases by running commands in an ESXi Shell.

Using ESXi Shell Commands to View Device Alias Information

Command	Description
`device alias list`	Lists all current I/O adapter aliases.
`device alias get -n<alias>`	Displays the physical device that an I/O adapter alias maps to.

Command	Description
`network nic list`	Lists aliases and general information about network devices.
`storage core adapter list`	Lists all storage I/O adapters.

Note ESXCLI commands are supported commands. Using alternative sources for displaying alias information is not recommended.

Change Device Aliases on an ESXi Host by Using Host Profiles

You can modify a device alias on ESXi hosts by editing the device alias section of the host profile attached to the hosts.

A device alias change can occur, for example, when you apply a BIOS or device firmware update. For more information about changes in the device alias configuration, see Device Alias Configuration.

I/O adapters that are based on PCI hardware usually have a logical and a PCI alias entry. Both aliases must have the same value. Some I/O adapters usually have a logical alias entry only. PCI hardware devices without I/O adapters usually have a PCI alias entry only. Modify a PCI alias entry only when you need the alias for another device.

Note Two different I/O adapters must not have the same alias, except I/O adapters that are based on PCI hardware and have a logical and a PCI alias entry.

Procedure

For information about exporting, extracting, and editing a host profile, see the *vSphere Host Profiles* documentation.

1. Export the current host profile attached to the ESXi host.
2. Extract a new host profile from the ESXi host, but do not apply the host profile.
3. Remove all entries from the device alias section of the current host profile.
4. Transfer all entries from the device alias section of the new host profile to the same section of the current host profile.

   You can modify the aliases in the entries before adding them to the current host profile.
5. Reapply the current host profile to the ESXi host.

Change Device Aliases on a Stateful ESXi Host by Using ESXCLI Commands

You can modify a device alias on a stateful ESXi host that does not have an attached host profile by running ESXCLI commands on the host.

A device alias change can occur, for example, when the slot position of a device changes. For more information about changes in the device alias configuration, see Device Alias Configuration.

Note The following ESXCLI commands might not be compatible with ESXi version 6.7 and earlier.

Procedure

1 To list the current assignment of aliases to device addresses, in the ESXi Shell, run `localcli --plugin-dir /usr/lib/vmware/esxcli/int/ deviceInternal alias list`.

   For I/O adapters that are based on PCI hardware, you usually see a logical and a PCI alias entry.

   For some I/O adapters, you usually see a logical alias entry only.

   For PCI hardware devices without I/O adapters, you usually see a PCI alias entry only.

2 To change an alias, replace the `ALIAS`, `PCI_ADDRESS`, and `LOGICAL_ADDRESS` placeholder values with actual values, and follow one of the substeps.

   a If an I/O adapter has a logical and a PCI alias, run `localcli --plugin-dir /usr/lib/vmware/esxcli/int/ deviceInternal alias store --bus-type logical --alias ALIAS --bus-address LOGICAL_ADDRESS` and `localcli --plugin-dir /usr/lib/vmware/esxcli/int/ deviceInternal alias store --bus-type pci --alias ALIAS --bus-address PCI_ADDRESS`.

      The logical and PCI aliases must have the same value.

   b If an I/O adapter has a logical alias entry only, run `localcli --plugin-dir /usr/lib/vmware/esxcli/int/ deviceInternal alias store --bus-type logical --alias ALIAS --bus-address LOGICAL_ADDRESS`.

   c If a PCI hardware device has a PCI alias entry only, run `localcli --plugin-dir /usr/lib/vmware/esxcli/int/ deviceInternal alias store --bus-type pci --alias ALIAS --bus-address PCI_ADDRESS`.

      Modify a PCI alias only when you need the alias for another device.

   Note Two different I/O adapters must not have the same alias, except I/O adapters that are based on PCI hardware and have a logical and a PCI alias entry.

3 Reboot the system.

vSphere PowerCLI Scenario for vSphere Auto Deploy

Scenario to set up and configure a working vSphere Auto Deploy environment by using PowerCLI.

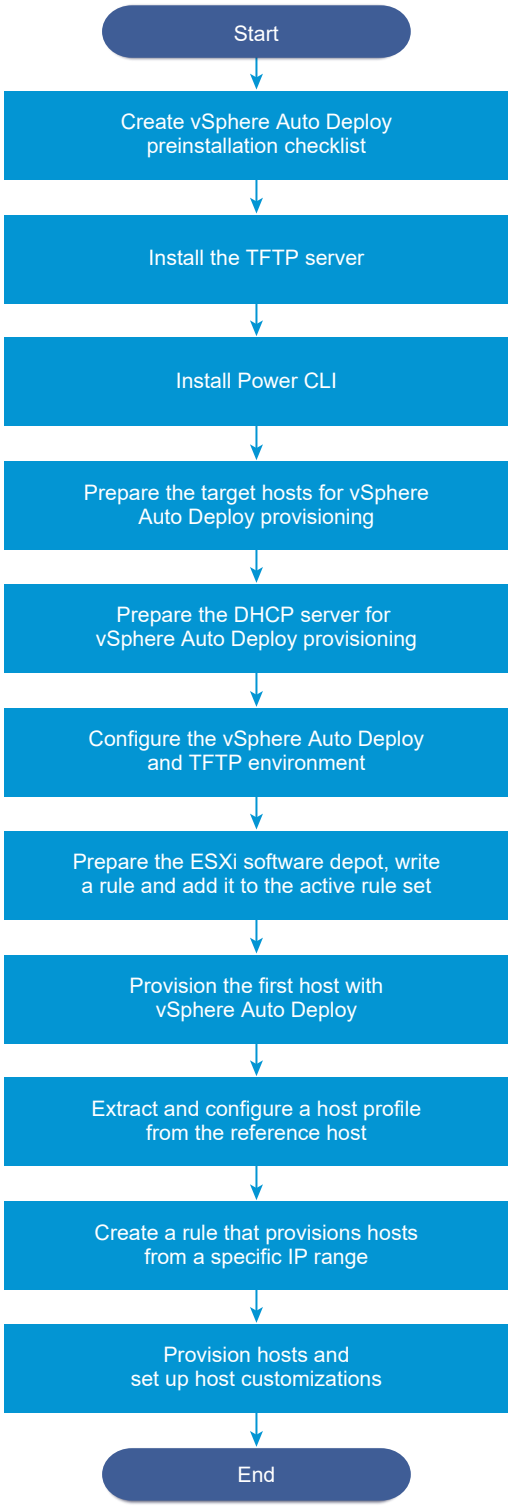
In this scenario, you are going to set up and configure a working vSphere Auto Deploy environment that includes four hosts. You will create rules and provision two of the hosts with an image profile and the other two with the same image profile and a host profile that is set up to prompt for user input. This scenario can provide you with the basis for a production environment. The task descriptions assume that you are using a flat network with no VLAN tagging between the physical hosts and the rest of your environment.

To perform the tasks in this scenario, you should have the following background knowledge and privileges.

- Experience with vSphere (vCenter Server and ESXi).
- Basic knowledge of Microsoft PowerShell and PowerCLI.
- Administrator rights to a Windows system and a vCenter Server system.

Follow the tasks in the order presented in this scenario. Some steps can be performed in a different order, but the order used here limits repeated manipulation of some components. For details on the preinstallation checklist and other prerequisites to configure vSphere Auto Deploy, see Install and Configure vSphere Auto Deploy.

Figure 4-11. vSphere Auto Deploy Setup and Hosts Provisioning Workflow



vSphere Auto Deploy takes advantage of the iPXE infrastructure and PXE booting with legacy BIOS firmware is possible only over IPv4. If the hosts that you want to provision with vSphere Auto Deploy are with legacy BIOS, the vSphere Auto Deploy server must have an IPv4 address. PXE booting with UEFI firmware is possible with either IPv4 or IPv6.

Procedure

1 Install the TFTP Server

To set up a vSphere Auto Deploy infrastructure, you must install a TFTP server in your environment. vSphere Auto Deploy relies on a TFTP server for sending a boot image to the hosts that it provisions.

2 Install PowerCLI

Before you can manage vSphere Auto Deploy with rules that you create with PowerCLI cmdlets, you must install PowerCLI.

3 Prepare the vSphere Auto Deploy Target Hosts

You must configure the BIOS settings of the four hosts and reconfirm the MAC address of the primary network device to prepare the target hosts for provisioning with vSphere Auto Deploy.

4 Prepare the DHCP Server for vSphere Auto Deploy Provisioning

When you prepare the vSphere Auto Deploy target hosts, you must set up the DHCP server in this scenario to serve each target host with an iPXE binary.

5 Configure the vSphere Auto Deploy and TFTP Environment in the vSphere Client

After you prepare the DHCP server, you must start the vSphere Auto Deploy vCenter Server service and configure the TFTP server. You must download a TFTP ZIP file from your vSphere Auto Deploy server. The customized FTP server serves the boot images that vSphere Auto Deploy provides.

6 Prepare the ESXi Software Depot and Write a Rule

After you configure the vSphere Auto Deploy infrastructure, you must add an ESXi software depot, specify an image profile, write a rule, and add it to the active rule set.

7 Provision the First Host with vSphere Auto Deploy

After creating a rule and adding it to the active rule set, you can provision the first host and check its vCenter Server location to complete verification of the image provisioning of your setup.

8 Extract and Configure a Host Profile from the Reference Host

After provisioning the first host, you can extract and configure a host profile that can be used to apply the same configuration to other target hosts. Configuration that differs for different hosts, such as a static IP address, can be managed through the host customization mechanism.

9 Create a Rule that Provisions Hosts from a Specific IP Range

After creating a host profile from a reference host, you can create a rule that applies the previously verified image profile and the host profile that you extracted to target hosts from a specific IP range.

10 Provision Hosts and Set Up Host Customizations

With the rule in place that provisions hosts using an image profile and a host profile, you can provision specific target hosts. If any host profile items are set to prompt the user for input, the host comes up in maintenance mode. You apply the host profile or check host compliance to be prompted for the information. The system associates the host customization with the host.

Install the TFTP Server

To set up a vSphere Auto Deploy infrastructure, you must install a TFTP server in your environment. vSphere Auto Deploy relies on a TFTP server for sending a boot image to the hosts that it provisions.

This task only installs the TFTP server. You later download a configuration file to the server. See Configure the vSphere Auto Deploy and TFTP Environment in the vSphere Client.

Procedure

1 Download your preferred TFTP server to a location that has network access to your vCenter Server and install the server.

2 Configure the TFTP root directory, for example `D:\TFTP_Root\`.

What to do next

Install PowerCLI, to manage vSphere Auto Deploy with PowerCLI cmdlets.

Install PowerCLI

Before you can manage vSphere Auto Deploy with rules that you create with PowerCLI cmdlets, you must install PowerCLI.

Procedure

- Use the VMware PowerCLI User's Guide to learn about PowerShell basics, PowerCLI concepts, and how to install and configure PowerCLI.

What to do next

Configure the settings of your target hosts to prepare them for provisioning with vSphere Auto Deploy.

Prepare the vSphere Auto Deploy Target Hosts

You must configure the BIOS settings of the four hosts and reconfirm the MAC address of the primary network device to prepare the target hosts for provisioning with vSphere Auto Deploy.

Prerequisites

Hosts that you want to provision with vSphere Auto Deploy must meet the requirements for ESXi.

See ESXi Hardware Requirements.

Procedure

1. Change the BIOS settings of each of the four physical hosts to force the hosts to boot from the primary network device.
2. Reconfirm the MAC address of the primary network device.

What to do next

Set up the DHCP server to serve each target host with an iPXE binary.

Prepare the DHCP Server for vSphere Auto Deploy Provisioning

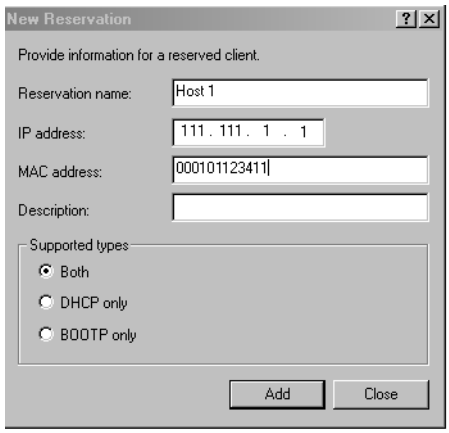
When you prepare the vSphere Auto Deploy target hosts, you must set up the DHCP server in this scenario to serve each target host with an iPXE binary.

The environment in this scenario uses Active Directory with DNS and DHCP. The DHCP server is included in the vSphere supported Windows Server versions.

Procedure

1. Log in to your DHCP Server with administrator privileges.
2. Create a DHCP scope for your IP address range.
   a. Click **Start > Settings > Control Panel > Administrative Tools** and click **DHCP**.
   b. Navigate to **DHCP > *hostname* > IPv4**.
   c. Right-click **IPv4** and select **New Scope**.
   d. On the Welcome screen, click **Next**, and specify a name and description for the scope.
   e. Specify an IP address range and click **Next**.
   f. Click **Next** until you reach the Configure DHCP Options screen and select **No, I will configure this option later**.
3. Create a DHCP reservation for each target ESXi host.
   a. In the DHCP window, navigate to **DHCP > *hostname* > IPv4 > Autodeploy Scope > Reservations**.
   b. Right-click **Reservations** and select **New Reservation**.

c In the New Reservation window, specify a name, IP address, and the MAC address for one of the hosts. Do not include the colon (:) in the MAC address.



d Repeat the process for each of the other hosts.

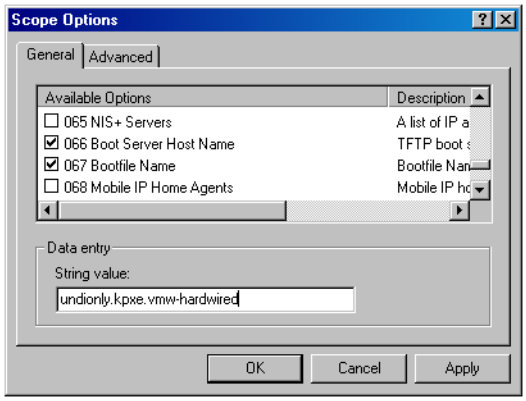
4 Set up the DHCP Server to point the hosts to the TFTP Server.

a In the DHCP window, navigate to **DHCP** > ***hostname*** > **IPv4** > **Autodeploy Scope** > **Scope Options**.

b Right click **Scope Options** and choose **Configure Options**.

c In the Scope Options window, click the **General** tab.

d Click **066 Boot Server Host Name** and enter the address of the TFTP server that you installed in the String value field below the Available Options.



e Click **067 Bootfile Name** and enter **`undionly.kpxe.vmw-hardwired`**.

The `undionly.kpxe.vmw-hardwired` iPXE binary will be used to boot the ESXi hosts.

f Click **Apply** and click **OK** to close the window.

5 In the DHCP window, right-click **DHCP** > ***hostname*** > **IPv4** > **Scope** > **Activate** and click **Activate**.

6 Do not log out from the DHCP Server if you are using Active Directory for DHCP and DNS, or log out otherwise.

What to do next

start the vCenter Server service of vSphere Auto Deploy and configure the TFTP server.

Configure the vSphere Auto Deploy and TFTP Environment in the vSphere Client

After you prepare the DHCP server, you must start the vSphere Auto Deploy vCenter Server service and configure the TFTP server. You must download a TFTP ZIP file from your vSphere Auto Deploy server. The customized FTP server serves the boot images that vSphere Auto Deploy provides.

Procedure

1. Use the vSphere Client to connect to the vCenter Server system that manages the vSphere Auto Deploy server.
2. Start the vSphere Auto Deploy service.
   a. Navigate to **Home > Auto Deploy**.
   b. On the **Auto Deploy** page, select your vCenter Server from the drop-down menu at the top.
   c. 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**.
3. In the Auto Deploy inventory, click the **Configure** tab.
4. Click the **Download TFTP Zip File**.
5. Save the file `deploy-tftp.zip` to the `TFTP_Root` directory that you created when you installed the TFTP Server, and unzip the file.

What to do next

Add a software depot to your inventory and use an image profile from the depot to create a rule for host provisioning.

Prepare the ESXi Software Depot and Write a Rule

After you configure the vSphere Auto Deploy infrastructure, you must add an ESXi software depot, specify an image profile, write a rule, and add it to the active rule set.

vSphere Auto Deploy provisions hosts with image profiles that define the set of VIBs that an ESXi installation process uses. Image profiles are stored in software depots. You must make sure that the correct image profile is available before you start provisioning hosts. When you add a software depot to a PowerCLI session, it is available only during the current session. It does not persist across sessions.

The steps in this task instruct you to run PowerCLI cmdlets. For additional information about the vSphere Auto Deploy cmdlets that you can run in a PowerCLI session, see vSphere Auto Deploy PowerCLI Cmdlet Overview.

Prerequisites

Verify that you can access the ESXi hosts that you want to provision from the system on which you run PowerCLI.

Procedure

1. Log in as an administrator to the console of the Windows system, either directly or by using RDP.

   This task assumes that you installed PowerCLI on the system on which the vCenter Server system is running.

2. 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_address
   ```

   The cmdlet might return a server certificate warning. In a production environment, make sure no server certificate issues occur. In a development environment, you can ignore the warning.

3. Enter the vCenter Server credentials.

4. Run `Add-EsxSoftwareDepot` to add the online depot to the PowerCLI session.

   ```
   Add-EsxSoftwareDepot https://hostupdate.vmware.com/software/VUM/PRODUCTION/main/vmw-depot-index.xml
   ```

   Adding the software depot is required each time you start a new PowerCLI session.

5. Validate that you successfully added the software depot by checking the contents of the depot with the `Get-EsxImageProfile` cmdlet.

   The cmdlet returns information about all image profiles in the depot.

6. Create a new rule by running the `New-DeployRule` cmdlet.

   ```
   New-DeployRule -Name "InitialBootRule" -Item ESXi-6.0.0-2494585-standard -AllHosts
   ```

   The cmdlet creates a rule that assigns the specified image profile to all hosts in the inventory.

7. Add the new rule to the active rule set to make the rule available to the vSphere Auto Deploy server.

   ```
   Add-DeployRule    -DeployRule "InitialBootRule"
   ```

What to do next

Provision your first host with vSphere Auto Deploy and verify its image provisioning.

Provision the First Host with vSphere Auto Deploy

After creating a rule and adding it to the active rule set, you can provision the first host and check its vCenter Server location to complete verification of the image provisioning of your setup.

Procedure

1. Open a console session to the physical host that you want to use as the first ESXi target host, boot the host, and look for messages that indicate a successful iPXE boot.

   During the boot process, DHCP assigns an IP address to the host. The IP address matches the name you specified earlier in the DNS server. The host contacts the vSphere Auto Deploy server and downloads the ESXi binaries from the HTTP URL indicated in the iPXE tramp file that you downloaded earlier to the TFTP_Root directory. Each instance of vSphere Auto Deploy produces a custom set of files for the TFTP Server.
2. Use the vSphere Client to connect to the vCenter Server system that manages the vSphere Auto Deploy server.
3. On the vSphere Client Home page, click **Hosts and Clusters**.
4. Verify that the newly provisioned host is now in the vCenter Server inventory at the datacenter level.

   By default, vSphere Auto Deploy adds hosts at the datacenter level when the boot process completes.

What to do next

Extract a host profile from the host and configure it to require user input.

Extract and Configure a Host Profile from the Reference Host

After provisioning the first host, you can extract and configure a host profile that can be used to apply the same configuration to other target hosts. Configuration that differs for different hosts, such as a static IP address, can be managed through the host customization mechanism.

vSphere Auto Deploy can provision each host with the same host profile. vSphere Auto Deploy can also use host customization that allows you to specify different information for different hosts. For example, if you set up a VMkernel port for vMotion or for storage, you can specify a static IP address for the port by using the host customization mechanism.

Procedure

1. Use the vSphere Client to connect to the vCenter Server system that manages the vSphere Auto Deploy server.
2. Click **Policies and Profiles** and select **Host Profiles**.
3. Click **Extract Host Profile**.
4. On the **Select host** page of the wizard, select the reference host that you configured earlier and click **Next**.

5 On the **Name and Description** page of the wizard, enter a name and description for the new profile and click **Finish**.

6 Select the host profile that you want to edit and click the **Configure** tab.

7 Click **Edit Host Profile**.

8 Select **Security and Services > Security Settings > Security > User Configuration > root**.

9 From the **Password** drop-down menu, select **User Input Password Configuration**.

10 Click **Save** to configure the host profile settings.

What to do next

Create a vSphere Auto Deploy rule to apply the host profile to other ESXi hosts.

Create a Rule that Provisions Hosts from a Specific IP Range

After creating a host profile from a reference host, you can create a rule that applies the previously verified image profile and the host profile that you extracted to target hosts from a specific IP range.

Procedure

1 Log in with administrator privileges to the console of the Windows system, either directly or by using RDP.

2 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_address
```

The cmdlet might return a server certificate warning. In a production environment, make sure no server certificate issues occur. In a development environment, you can ignore the warning.

3 Run `Add-EsxSoftwareDepot` to add the online depot to the PowerCLI session.

```
Add-EsxSoftwareDepot https://hostupdate.vmware.com/software/VUM/PRODUCTION/main/vmw-depot-index.xml
```

Adding the software depot is required each time you start a new PowerCLI session.

4 (Required) Display the rules in the active rule set by running the `Get-DeployRuleset` cmdlet.

5 Create a rule that instructs vSphere Auto Deploy to provision the set of hosts from a specified IP range with the image profile that you previously selected and the host profile that you created from the reference host.

```
New-DeployRule -name "Production01Rule" -item "image_profile",ESXiGold -Pattern "ipv4=IP_range"
```

6 Add the new rule to the active rule set.

```
Add-DeployRule    -DeployRule "Production01Rule"
```

7 Check the active rule set by running the `Get-DeployRuleset` command.

PowerCLI displays information similar to the following example.

```
Name:              Production01Rule
PatternList:       {ipv4=address_range}
ItemList:          {ESXi-version-XXXXXX-standard, Compute01, ESXiGold}
```

What to do next

Provision the hosts and set up the host customizations.

Provision Hosts and Set Up Host Customizations

With the rule in place that provisions hosts using an image profile and a host profile, you can provision specific target hosts. If any host profile items are set to prompt the user for input, the host comes up in maintenance mode. You apply the host profile or check host compliance to be prompted for the information. The system associates the host customization with the host.

Procedure

1 Boot the remaining hosts you want to provision.

vSphere Auto Deploy boots the hosts, applies the host profile, and adds the hosts to the vCenter Server inventory. The hosts remain in maintenance mode because the host profile from the reference host is set up to require user input for each host.

2 Use the vSphere Client to connect to the vCenter Server system that manages the vSphere Auto Deploy server.

3 Click **Policies and Profiles** and select **Host Profiles**.

4 Right-click the newly created host profile for Auto Deploy and click **Edit Host Customizations**.

5 Select the hosts, enter the required host customizations and click **Finish**.

Alternatively, you can also **Import Host Customizations** file.

6 Apply the host profile to each of the hosts and get the hosts out of maintenance mode. Alternatively, you can reboot each host.

When the reboot progress completes, all hosts are running with the image you specify and use the configuration in the reference host profile. The cluster shows that all hosts are fully compliant.

Results

All hosts are now configured with the shared information through the reference host profile and with the host-specific information through the host customization mechanism. The next time you

boot the hosts, they receive the complete Host Profile information, including the host-specific information, and boot up completely configured and out of Maintenance Mode.

Troubleshooting vSphere Auto Deploy

The vSphere Auto Deploy troubleshooting topics cover some situations when provisioning hosts with vSphere Auto Deploy does not work as expected.

vSphere Auto Deploy Rule Takes Long to Complete

After you run an Auto Deploy rule, it takes long to complete, and you see no progress for the tasks.

Problem

Auto Deploy automatically detects if a new version of the image that a cluster uses is available, or when the cache lifetime of files expires, and refreshes the cache by re-downloading the files from the latest version of the software depot. As a result, you might see a delay in the implementation of Auto Deploy rules.

Cause

When the cache lifetime of files cached by Auto Deploy expires, Auto Deploy automatically refreshes the cache from the software depot. Since a default software depot is usually more than 300 MB, depending on the network, the download might take long.

You can also see a delay in deploying Auto Deploy rules when converting a cluster that you manage with a single image to one that you manage with a configuration on a cluster level. If a host attempts to boot during the time Auto Deploy caches the configuration file, you might see a delay, because of the time Auto Deploy needs to create the cache.

A general vSphere infrastructure problem might also prevent Auto Deploy rules from running in a timely manner.

Solution

- Regardless of any connectivity issues or breakage, Auto Deploy keeps active sessions and persists in booting hosts until it succeeds.

vSphere Auto Deploy TFTP Timeout Error at Boot Time

A TFTP Timeout error message appears when a host provisioned with vSphere Auto Deploy boots. The text of the message depends on the BIOS.

Problem

A TFTP Timeout error message appears when a host provisioned with vSphere Auto Deploy boots. The text of the message depends on the BIOS.

Cause

The TFTP server is down or unreachable.

Solution

- Ensure that your TFTP service is running and reachable by the host that you are trying to boot.
- To view the diagnostic logs for details on the present error, see your TFTP service documentation.

vSphere Auto Deploy Host Boots with Wrong Configuration

A host is booting with a different ESXi image, host profile, or folder location than the one specified in the rules.

Problem

A host is booting with a different ESXi image profile or configuration than the image profile or configuration that the rules specify. For example, you change the rules to assign a different image profile, but the host still uses the old image profile.

Cause

After the host has been added to a vCenter Server system, the boot configuration is determined by the vCenter Server system. The vCenter Server system associates an image profile, host profile, or folder location with the host.

Solution

- Use the `Test-DeployRuleSetCompliance` and `Repair-DeployRuleSetCompliance` vSphere PowerCLI cmdlets to reevalute the rules and to associate the correct image profile, host profile, or folder location with the host.

Host Is Not Redirected to vSphere Auto Deploy Server

During boot, a host that you want to provision with vSphere Auto Deploy loads iPXE. The host is not redirected to the vSphere Auto Deploy server.

Problem

During boot, a host that you want to provision with vSphere Auto Deploy loads iPXE. The host is not redirected to the vSphere Auto Deploy server.

Cause

The `tramp` file that is included in the TFTP ZIP file has the wrong IP address for the vSphere Auto Deploy server.

Solution

- Correct the IP address of the vSphere Auto Deploy server in the `tramp` file, as explained in the *vSphere Installation and Setup* documentation.

Package Warning Message When You Assign an Image Profile to a vSphere Auto Deploy Host

When you run a vSphere PowerCLI cmdlet that assigns an image profile that is not vSphere Auto Deploy ready, a warning message appears.

Problem

When you write or modify rules to assign an image profile to one or more hosts, the following error results:

```
Warning: Image Profile <name-here> contains one or more software packages
that are not stateless-ready. You may experience problems when using this
profile with Auto Deploy.
```

Cause

Each VIB in an image profile has a `stateless-ready` flag that indicates that the VIB is meant for use with vSphere Auto Deploy. You get the error if you attempt to write a vSphere Auto Deploy rule that uses an image profile in which one or more VIBs have that flag set to FALSE.

Note You can use hosts provisioned with vSphere Auto Deploy that include VIBs that are not stateless ready without problems. However booting with an image profile that includes VIBs that are not stateless ready is treated like a fresh install. Each time you boot the host, you lose any configuration data that would otherwise be available across reboots for hosts provisioned with vSphere Auto Deploy.

Solution

1. Use vSphere ESXi Image Builder cmdlets in a vSphere PowerCLI session to view the VIBs in the image profile.
2. Remove any VIBs that are not stateless-ready.
3. Rerun the vSphere Auto Deploy cmdlet.

vSphere Auto Deploy Host with a Built-In USB Flash Drive Does Not Send Coredumps to Local Disk

If your vSphere Auto Deploy host has a built-in USB flash drive, and an error results in a coredump, the coredump is lost. Set up your system to use ESXi Dump Collector to store coredumps on a networked host.

Problem

If your vSphere Auto Deploy host has a built-in USB Flash, and if it encounters an error that results in a coredump, the coredump is not sent to the local disk.

Solution

1. Install ESXi Dump Collector on a system of your choice.

   ESXi Dump Collector is included with the vCenter Server installer.

2 Use ESXCLI to configure the host to use ESXi Dump Collector.

```
esxcli conn_options system coredump network set IP-addr,port
esxcli system coredump network set -e true
```

3 Use ESXCLI to deactivate local coredump partitions.

```
esxcli conn_options system coredump partition set -e false
```

vSphere Auto Deploy Host Reboots After Five Minutes

A vSphere Auto Deploy host boots and displays iPXE information, but reboots after five minutes.

Problem

A host to be provisioned with vSphere Auto Deploy boots from iPXE and displays iPXE information on the console. However, after five minutes, the host displays the following message to the console and reboots.

```
This host is attempting to network-boot using VMware
AutoDeploy. However, there is no ESXi image associated with this host.
Details: No rules containing an Image Profile match this
host. You can create a rule with the New-DeployRule PowerCLI cmdlet
and add it to the rule set with Add-DeployRule or Set-DeployRuleSet.
The rule should have a pattern that matches one or more of the attributes
listed below.
```

The host might also display the following details:

```
Details: This host has been added to VC, but no Image Profile
is associated with it. You can use Apply-ESXImageProfile in the
PowerCLI to associate an Image Profile with this host.
Alternatively, you can reevaluate the rules for this host with the
Test-DeployRuleSetCompliance and Repair-DeployRuleSetCompliance cmdlets.
```

The console then displays the host's machine attributes including vendor, serial number, IP address, and so on.

Cause

No image profile is currently associated with this host.

Solution

You can assign an image profile to the host by running the `Apply-EsxImageProfile` cmdlet, or by creating the following rule:

1 Run the `New-DeployRule` cmdlet to create a rule that includes a pattern that matches the host with an image profile.

2 Run the `Add-DeployRule` cmdlet to add the rule to a ruleset.

3 Run the `Test-DeployRuleSetCompliance` cmdlet and use the output of that cmdlet as the input to the `Repair-DeployRuleSetCompliance` cmdlet.

vSphere Auto Deploy Host Cannot Contact TFTP Server

The host that you provision with vSphere Auto Deploy cannot contact the TFTP server.

Problem

When you attempt to boot a host provisioned with vSphere Auto Deploy, the host performs a network boot and is assigned a DHCP address by the DHCP server, but the host cannot contact the TFTP server.

Cause

The TFTP server might have stopped running, or a firewall might block the TFTP port.

Solution

- If you installed the WinAgents TFTP server, open the WinAgents TFTP management console and verify that the service is running. If the service is running, check the Windows firewall's inbound rules to make sure the TFTP port is not blocked. Turn off the firewall temporarily to see whether the firewall is the problem.
- For all other TFTP servers, see the server documentation for debugging procedures.

vSphere Auto Deploy Host Cannot Retrieve ESXi Image from vSphere Auto Deploy Server

The host that you provision with vSphere Auto Deploy stops at the iPXE boot screen.

Problem

When you attempt to boot a host provisioned with vSphere Auto Deploy, the boot process stops at the iPXE boot screen and the status message indicates that the host is attempting to get the ESXi image from the vSphere Auto Deploy server.

Cause

The vSphere Auto Deploy service might be stopped or the vSphere Auto Deploy server might be inaccessible.

Solution

1 Log in to the system on which you installed the vSphere Auto Deploy server.

2 Check that the vSphere Auto Deploy server is running.

   a Click **Start > Settings > Control Panel > Administrative Tools**.

   b Double-click **Services** to open the Services Management panel.

   c In the Services field, look for the VMware vSphere Auto Deploy Waiter service and restart the service if it is not running.

3 Open a Web browser, enter the following URL, and check whether the vSphere Auto Deploy server is accessible.

   https://*Auto_Deploy_Server_IP_Address*:*Auto_Deploy_Server_Port*/vmw/rdb

   Note Use this address only to check whether the server is accessible.

4 If the server is not accessible, a firewall problem is likely.

   a Try setting up permissive TCP Inbound rules for the vSphere Auto Deploy server port.

   The port is 6501 unless you specified a different port during installation.

   b As a last resort, deactivate the firewall temporarily and enable it again after you verified whether it blocked the traffic. Do not deactivate the firewall on production environments.

   To deactivate the firewall, run **`netsh firewall set opmode disable`**. To enable the firewall, run **`netsh firewall set opmode enable`**.

vSphere Auto Deploy Host Does Not Get a DHCP Assigned Address

The host you provision with vSphere Auto Deploy fails to get a DHCP Address.

Problem

When you attempt to boot a host provisioned with vSphere Auto Deploy, the host performs a network boot but is not assigned a DHCP address. The vSphere Auto Deploy server cannot provision the host with the image profile.

Cause

You might have a problem with the DHCP service or with the firewall setup.

Solution

1 Check that the DHCP server service is running on the Windows system on which the DHCP server is set up to provision hosts.

   a Click **Start > Settings > Control Panel > Administrative Tools**.

   b Double-click **Services** to open the Services Management panel.

   c In the Services field, look for the DHCP server service and restart the service if it is not running.

2 If the DHCP server is running, recheck the DHCP scope and the DHCP reservations that you configured for your target hosts.

   If the DHCP scope and reservations are configured correctly, the problem most likely involves the firewall.

3 As a temporary workaround, turn off the firewall to see whether that resolves the problem.

   a Open the command prompt by clicking **Start > Program > Accessories > Command prompt**.

   b Type the following command to temporarily turn off the firewall. Do not turn off the firewall in a production environment.

   ```
   netsh firewall set opmode disable
   ```

   c Attempt to provision the host with vSphere Auto Deploy.

   d Type the following command to turn the firewall back on.

   ```
   netsh firewall set opmode enable
   ```

4 Set up rules to allow DHCP network traffic to the target hosts.

   See the firewall documentation for DHCP and for the Windows system on which the DHCP server is running for details.

vSphere Auto Deploy Host Does Not Network Boot

The host you provision with vSphere Auto Deploy comes up but does not network boot.

Problem

When you attempt to boot a host provisioned with vSphere Auto Deploy, the host does not start the network boot process.

Cause

You did not enable your host for network boot.

Solution

1 Reboot the host and follow the on-screen instructions to access the BIOS configuration.

2 In the BIOS configuration, enable Network Boot in the Boot Device configuration.

Recovering from Database Corruption on the vSphere Auto Deploy Server

In some situations, you might have a problem with the vSphere Auto Deploy database. The most efficient recovery option is to replace the existing database file with the most recent backup.

Problem

When you use vSphere Auto Deploy to provision the ESXi hosts in your environment, you might encounter a problem with the vSphere Auto Deploy database.

Important This is a rare problem. Follow all other vSphere Auto Deploy troubleshooting strategies before you replace the current database file. Rules or associations that you created since the backup you choose are lost.

Cause

This problem happens only with hosts that are provisioned with vSphere Auto Deploy.

Solution

1. Stop the vSphere Auto Deploy server service.
2. Find the vSphere Auto Deploy log by going to the vSphere Auto Deploy page in the vSphere Client.
3. Check the logs for the following message:

   ```
   DatabaseError: database disk image is malformed.
   ```

   If you see the message, replace the existing database with the most recent backup.
4. Go to the vSphere Auto Deploy data directory at `/var/lib/rbd`.

   The directory contains a file named `db`, and backup files named `db-yyy-mm-dd`.
5. Rename the current `db` file.

   VMware Support might ask for that file if you call for assistance.
6. Rename the most recent backup to `db`.
7. Restart the vSphere Auto Deploy server service.
8. If the message still appears in the log, repeat the steps to use the next recent backup until vSphere Auto Deploy works without database errors.

Setting Up ESXi

These topics provide information about using the direct console user interface and configuring defaults for ESXi.

Initial ESXi Configuration

ESXi Autoconfiguration

When you turn on the ESXi host for the first time or after resetting the configuration defaults, the host enters an autoconfiguration phase. This phase configures system network and storage devices with default settings.

By default, Dynamic Host Configuration Protocol (DHCP) configures IP, and all visible blank internal disks are formatted with the virtual machine file system (VMFS) so that virtual machines can be stored on the disks.

What to read next

Managing ESXi Remotely

You can use the VMware Host Client, the vSphere Client and vCenter Server to manage your ESXi hosts.

For instructions about downloading and installing vCenter Server and the vCenter Server components, see *vCenter Server Installation and Setup*. For information about installing the VMware Host Client, see *vSphere Single Host Management*.

About the Direct Console ESXi Interface

Use the direct console interface for initial ESXi configuration and troubleshooting.

Connect a keyboard and monitor to the host to use the direct console. After the host completes the autoconfiguration phase, the direct console appears on the monitor. You can examine the default network configuration and change any settings that are not compatible with your network environment.

Key operations available to you in the direct console include:

- Configuring hosts
- Setting up administrative access
- Troubleshooting

You can also use vSphere Client to manage the host by using vCenter Server.

Table 4-20. Navigating in the Direct Console

Action	Key
View and change the configuration	F2
Change the user interface to high-contrast mode	F4
Shut down or restart the host	F12
View the VMkernel log	Alt+F12
Switch to the shell console	Alt+F1
Switch to the direct console user interface	Alt+F2
Move the selection between fields	Arrow keys
Select a menu item	Enter
Toggle a value	Spacebar
Confirm sensitive commands, such as resetting configuration defaults	F11
Save and exit	Enter

Table 4-20. Navigating in the Direct Console (continued)

Action	Key
Exit without saving	Esc
Exit system logs	q

Configure the Keyboard Layout for the Direct Console

You can configure the layout for the keyboard that you use with the direct console.

Procedure

1. From the direct console, select **Configure Keyboard** and press Enter.
2. Select the layout to use.
3. Press the spacebar to toggle selections on and off.
4. Press Enter.

Create a Security Banner for the Direct Console

A security banner is a message that is displayed on the direct console **Welcome** screen.

Procedure

1. From the vSphere Client, connect to the vCenter Server.
2. Select the host in the inventory.
3. Click the **Configure** tab.
4. Under System, select **Advanced System Settings**.
5. Select `Annotations.WelcomeMessage`.
6. Click the **Edit** icon.
7. Enter a security message.

Results

The message is displayed on the direct console **Welcome** screen.

Redirecting the Direct Console to a Serial Port

To manage your ESXi host remotely from a serial console, you can redirect the direct console to a serial port.

vSphere supports the VT100 terminal type and the PuTTy terminal emulator to view the direct console over the serial port.

You can redirect the direct console to a serial port in several ways.

What to read next

Redirect the Direct Console to a Serial Port by Setting the Boot Options Manually

When you redirect the direct console to a serial port by setting the boot options, the change does not persist for subsequent boots.

Prerequisites

Verify that the serial port is not in use for serial logging and debugging.

Procedure

1. Start the host.
2. When the Loading VMware Hypervisor window appears, press Shift+O to edit boot options.
3. Deactivate the logPort and gdbPort on com1 and set tty2Port to com1 by entering the following boot options:

   ```
   "gdbPort=none logPort=none tty2Port=com1";
   ```

   To use com2 instead, replace `com1` with `com2`.

Results

The direct console is redirected to the serial port until you reboot the host. To redirect the direct console for subsequent boots, see Redirect the Direct Console to a Serial Port from the vSphere Client

Redirect the Direct Console to a Serial Port from the vSphere Client

You can manage the ESXi host remotely from a console that is connected to the serial port by redirecting the direct console to either of the serial ports com1 or com2. When you use the vSphere Client to redirect the direct console to a serial port, the boot option that you set persists after subsequent reboots.

Prerequisites

- Verify that you can access the host from the vSphere Client.
- Verify that the serial port is not in use for serial logging and debugging, or for ESX Shell (tty1Port).

Procedure

1. From the vSphere Client, connect to the vCenter Server.
2. Select the host in the inventory.
3. Click the **Configure** tab.
4. Under System, select **Advanced System Settings**.
5. Make sure that the **VMkernel.Boot.logPort** and **VMkernel.Boot.gdbPort** fields are not set to use the com port that you want to redirect the direct console to.

6 Set **VMkernel.Boot.tty2Port** to the serial port to redirect the direct console to: **com1** or **com2**.

7 Reboot the host.

Results

You can now manage the ESXi host remotely from a console that is connected to the serial port.

Redirect the Direct Console to a Serial Port in a Host Deployed with Auto Deploy

After you redirect the direct console to a serial port, you can make that setting part of the host profile that persists when you reprovision the host with Auto Deploy.

Prerequisites

The serial port must not already be in use for serial logging and debugging.

Procedure

1 From the , connect to the vCenter Server.

2 Select the host in the inventory.

3 Under System, select **Advanced System Settings**.

4 Make sure that the **VMkernel.Boot.logPort** and **VMkernel.Boot.gdbPort** fields are not set to use the com port that you want to redirect the direct console to.

5 Set **VMkernel.Boot.tty2Port** to the serial port to redirect the direct console to: **com1** or **com2**.

6 Click **OK**.

7 Save the host profile and attach the host to the profile. See the *vSphere Host Profiles* documentation.

Results

The setting to redirect the direct console to a serial port is stored by vCenter Server and persists when you reprovision the host with Auto Deploy.

Enable ESXi Shell and SSH Access with the Direct Console User Interface

Use the direct console user interface to enable the ESXi Shell.

Procedure

1 From the Direct Console User Interface, press F2 to access the System Customization menu.

2 Select **Troubleshooting Options** and press Enter.

3 From the Troubleshooting Mode Options menu, select a service to enable.

- Enable ESXi Shell
- Enable SSH

4 Press Enter to enable the service.

5 (Optional) Set the timeout for the ESXi Shell.

By default, timeouts for the ESXi Shell is 0 (not active).

The availability timeout setting is the number of minutes that can elapse before you must log in after the ESXi Shell is enabled. After the timeout period, if you have not logged in, the shell is deactivated.

Note If you are logged in when the timeout period elapses, your session will persist. However, the ESXi Shell is deactivated, preventing other users from logging in.

a From the Troubleshooting Mode Options menu, select **Modify ESXi Shell and SSH timeouts** and press Enter.

b Enter the availability timeout in minutes.

The availability timeout is the number of minutes that can elapse before you must log in after the ESXi Shell is enabled.

c Press Enter.

d Enter the idle timeout.

The idle timeout is the number of minutes that can elapse before the user is logged out of an idle interactive sessions. Changes to the idle timeout apply the next time a user logs in to the ESXi Shell and do not affect existing sessions.

6 Press Esc until you return to the main menu of the Direct Console User Interface.

Set the Password for the Administrator Account

You can use the direct console to set the password for the administrator account (root).

The administrative user name for the ESXi host is root. By default, the administrative password is not set.

Procedure

1 From the direct console, select **Configure Password**.

2 (Optional) If a password is already set up, type the password in the **Old Password** line and press Enter.

3 In the **New Password** line, type a new password and press Enter.

4 Retype the new password and press Enter.

Configuring the BIOS Boot Settings

If your server has multiple drives, you might need to configure the BIOS settings.

The BIOS boot configuration determines how your server boots. Generally, the CD-ROM device is listed first.

Note If you are using ESXi Embedded, the BIOS boot configuration determines whether your server boots into the ESXi boot device or another boot device. Generally, the USB flash device is listed first in the BIOS boot settings on the machine that hosts ESXi.

When you install or upgrade ESXi in UEFI mode, the installer creates an UEFI boot option named VMware ESXi and makes it the default boot option, so you do not need to change the boot order.

You can change the boot setting by configuring the boot order in the BIOS during startup or by selecting a boot device from the boot device selection menu. When you change the boot order in the BIOS, the new setting affects all subsequent reboots. When you select a boot device from the boot device selection menu, the selection affects the current boot only.

Some servers do not have a boot device selection menu, in which case you must change the boot order in the BIOS even for one-time boots, and then change it back again during a subsequent reboot.

Change the BIOS Boot Setting for ESXi

Configure the BIOS boot setting for ESXi if you want the server to boot into ESXi by default.

ESXi Installable and ESXi Embedded cannot exist on the same host.

Procedure

1 While the ESXi host is powering on, press the key required to enter your host's BIOS setup.

Depending on your server hardware, the key might be a function key or Delete. The option to enter the BIOS setup might be different for your server.

2 Select the BIOS boot setting.

Option	Description
If you are using the installable version of ESXi	Select the disk on which you installed the ESXi software and move it to the first position in the list. The host boots into ESXi.
If you are using ESXi Embedded	Select the USB flash device and move it to the first position in the list. The host starts in ESXi mode.

Configure the Boot Setting for Virtual Media

If you are using remote management software to set up ESXi, you might need to configure the boot setting for virtual media.

Virtual media is a method of connecting a remote storage media such as CD-ROM, USB mass storage, ISO image, and floppy disk to a target server that can be anywhere on the network. The target server has access to the remote media, and can read from and write to it as if it were physically connected to the server's USB port.

Prerequisites

ESXi Installable and ESXi Embedded cannot exist on the same host.

Procedure

1. Connect the media to the virtual device.

   For example, if you are using a Dell server, log in to the Dell Remote Access Controller (DRAC) or a similar remote management interface and select a physical floppy or CD-ROM drive, or provide a path to a floppy image or CD-ROM image.
2. Reboot the server.
3. While the server is powering on, enter the device selection menu.

   Depending on your server hardware, the key might be a function key or Delete.
4. Follow the instructions to select the virtual device.

Results

The server boots from the configured device once and goes back to the default boot order for subsequent boots. When you install or upgrade ESXi in UEFI mode, you do not need to change the boot order, because the system default boot order is set to VMware ESXi.

Configuring Network Settings

ESXi requires one IP address for the management network. To configure basic network settings, use the vSphere Client or the direct console.

Use the vSphere Client if you are satisfied with the IP address assigned by the DHCP server.

Use the direct console for network configuration in the following cases:

- You are not satisfied with the IP address assigned by the DHCP server.
- You are not allowed to use the IP address assigned by the DHCP server.
- ESXi does not have an IP address. This situation might occur if the autoconfiguration phase did not succeed in configuring DHCP.
- The wrong network adapter was selected during the autoconfiguration phase.

Use ESXCLI commands to configure your network settings. See esxcli network Commands.

Network Access to Your ESXi Host

The default behavior is to configure the ESXi management network using DHCP. You can override the default behavior and use static IP settings for the management network after the installation is completed.

Table 4-21. Network Configuration Scenarios Supported by ESXi

Scenario	Approach
You want to accept the DHCP-configured IP settings.	In the ESXi direct console, you can find the IP address assigned through DHCP to the ESXi management interface. You can use that IP address to connect to the host from the vSphere Client and customize settings, including changing the management IP address.
One of the following is true: ■ You do not have a DHCP server. ■ The ESXi host is not connected to a DHCP server. ■ Your connected DHCP server is not functioning properly.	During the autoconfiguration phase, the software assigns the link local IP address, which is in the subnet 169.254.x.x/16. The assigned IP address appears on the direct console. You can override the link local IP address by configuring a static IP address using the direct console.
The ESXi host is connected to a functioning DHCP server, but you do not want to use the DHCP-configured IP address.	During the autoconfiguration phase, the software assigns a DHCP-configured IP address. You can make the initial connection by using the DHCP-configured IP address. Then you can configure a static IP address. If you have physical access to the ESXi host, you can override the DHCP-configured IP address by configuring a static IP address using the direct console.
Your security deployment policies do not permit unconfigured hosts to be powered on the network.	Follow the setup procedure in Configure the Network Settings on a Host That Is Not Attached to the Network.

ESXi Networking Security Recommendations

Isolation of network traffic is essential to a secure ESXi environment. Different networks require a different access and level of isolation.

Your ESXi host uses several networks. Use appropriate security measures for each network, and isolate traffic for specific applications and functions. For example, ensure that VMware vSphere® vMotion® traffic does not travel over networks where virtual machines are located. Isolation prevents snooping. Having separate networks is also recommended for performance reasons.

- vSphere infrastructure networks are used for features such as vSphere vMotion, VMware vSphere Fault Tolerance, VMware vSAN, and storage. Isolate these networks for their specific functions. It is often not necessary to route these networks outside a single physical server rack.
- A management network isolates client traffic, command-line interface (CLI) or API traffic, and third-party software traffic from other traffic. In general, the management network is accessible only by system, network, and security administrators. To secure access to the management network, use a bastion host or a virtual private network (VPN). Strictly control access within this network.

- Virtual machine traffic can flow over one or many networks. You can enhance the isolation of virtual machines by using virtual firewall solutions that set firewall rules at the virtual network controller. These settings travel with a virtual machine as it migrates from host to host within your vSphere environment.

Choose Network Adapters for the Management Network

Traffic between an ESXi host and any external management software is transmitted through an Ethernet network adapter on the host. You can use the direct console to choose the network adapters that are used by the management network.

Examples of external management software include the vCenter Server and SNMP client. Network adapters on the host are named vmnic*N*, where N is a unique number identifying the network adapter, for example, vmnic0, vmnic1, and so forth.

During the autoconfiguration phase, the ESXi host chooses vmnic0 for management traffic. You can override the default choice by manually choosing the network adapter that carries management traffic for the host. In some cases, you might want to use a Gigabit Ethernet network adapter for your management traffic. Another way to help ensure availability is to select multiple network adapters. Using multiple network adapters enables load balancing and failover capabilities.

Procedure

1. From the direct console, select **Configure Management Network** and press Enter.
2. Select **Network Adapters** and press Enter.
3. Select a network adapter and press Enter.

Results

After the network is functional, you can use the vSphere Client to connect to the ESXi host through vCenter Server.

Set the VLAN ID

You can set the virtual LAN (VLAN) ID number of the ESXi host.

Procedure

1. From the direct console, select **Configure Management Network** and press Enter.
2. Select **VLAN** and press Enter.
3. Enter a VLAN ID number from 1 through 4094.

Configuring IP Settings for ESXi

By default, DHCP sets the IP address, subnet mask, and default gateway.

For future reference, write down the IP address.

For DHCP to work, your network environment must have a DHCP server. If DHCP is not available, the host assigns the link local IP address, which is in the subnet 169.254.x.x/16. The assigned IP address appears on the direct console. If you do not have physical monitor access to the host, you can access the direct console using a remote management application. See Using Remote Management Applications

When you have access to the direct console, you can optionally configure a static network address. The default subnet mask is 255.255.0.0.

Configure IP Settings from the Direct Console

If you have physical access to the host or remote access to the direct console, you can use the direct console to configure the IP address, subnet mask, and default gateway.

Procedure

1. Select **Configure Management Network** and press Enter.
2. Select **IP Configuration** and press Enter.
3. Select **Set static IP address and network configuration**.
4. Enter the IP address, subnet mask, and default gateway and press Enter.

Configure IP Settings from the vSphere Client

If you do not have physical access to the host, you can use the vSphere Client to configure static IP settings.

Procedure

1. Log in to the vCenter Server from the vSphere Client.
2. Select the host in the inventory.
3. On the **Configure** tab, expand **Networking**.
4. Select **VMkernel adapters**.
5. Select **vmk0 Management Network** and click the edit icon.
6. Select **IPv4 settings**.
7. Select **Use static IPv4 settings**.
8. Enter or change the static IPv4 address settings.
9. (Optional) Set static IPv6 addresses.
    a. Select **IPv6 settings**.
    b. Select **Static IPv6 addresses**.
    c. Click the add icon.
    d. Type the IPv6 address and click **OK**.
10. Click **OK**.

Configuring DNS for ESXi

You can select either manual or automatic DNS configuration of the ESXi host.

The default is automatic. For automatic DNS to work, your network environment must have a DHCP server and a DNS server.

In network environments where automatic DNS is not available or not desirable, you can configure static DNS information, including a host name, a primary name server, a secondary name server, and DNS suffixes.

Configure DNS Settings from the Direct Console

If you have physical access to the host or remote access to the direct console, you can use the direct console to configure DNS information.

Procedure

1. Select **Configure Management Network** and press Enter.
2. Select **DNS Configuration** and press Enter.
3. Select **Use the following DNS server addresses and hostname**.
4. Enter the primary server, an alternative server (optional), and the host name.

Configure DNS Suffixes

If you have physical access to the host, you can use the direct console to configure DNS information. By default, DHCP acquires the DNS suffixes.

Procedure

1. From the direct console, select **Configure Management Network**.
2. Select **Custom DNS Suffixes** and press Enter.
3. Enter new DNS suffixes.

Configure the Network Settings on a Host That Is Not Attached to the Network

Some highly secure environments do not permit unconfigured hosts on the network to be powered on. You can configure the host before you attach the host to the network.

Prerequisites

Verify that no network cables are connected to the host.

Procedure

1. Power on the host.
2. Use the direct console user interface to configure the password for the administrator account (root).
3. Use the direct console user interface to configure a static IP address.

4 Connect a network cable to the host.

5 (Optional) Use the vSphere Client to connect to a vCenter Server system.

6 (Optional) Add the host to the vCenter Server inventory.

Test the Management Network

You can use the direct console to do simple network connectivity tests.

The direct console performs the following tests.

- Pings the default gateway
- Pings the primary DNS name server
- Pings the secondary DNS nameserver
- Resolves the configured host name

Procedure

1 From the direct console, select **Test Management Network** and press Enter.

2 Press Enter to start the test.

Restart the Management Agents

The management agents synchronize VMware components and let you access the ESXi host by using the vSphere Client and vCenter Server.

The vSphere Client and vCenter Server are installed with the vSphere software. You might need to restart the management agents if remote access is interrupted. Restarting the management agents restarts all management agents and services that are installed and running in `/etc/init.d` on the ESXi host. Typically, these agents include hostd, ntpd, sfcbd, slpd, wsman, and vobd. The software also restarts the Fault Domain Manager (FDM) if installed.

Users accessing this host by using the vSphere Client and vCenter Server lose connectivity when you restart management agents.

Procedure

1 From the direct console, select **Troubleshooting Options** and press Enter.

2 Select **Restart Management Agents** and press Enter.

3 Press F11 to confirm the restart.

Results

The ESXi host restarts the management agents and services.

Restart the Management Network

Restarting the management network interface might be required to restore networking or to renew a DHCP lease.

Restarting the management network will result in a brief network outage that might temporarily affect running virtual machines.

If a renewed DHCP lease results in a new network identity (IP address or host name), remote management software will be disconnected.

Procedure

1 From the direct console, select **Restart Management Network** and press Enter.

2 Press F11 to confirm the restart.

Test Connectivity to Devices and Networks

You can use the direct console to perform some simple network connectivity tests, and specify other devices and networks.

Procedure

1 From the direct console, select **Test Management Network** and press Enter.

2 Type addresses to ping or another DNS host name to resolve.

3 Press Enter to start the test.

Restoring the Standard Switch

A vSphere Distributed Switch functions as a single virtual switch across all associated hosts.

Virtual machines can maintain a consistent network configuration as they migrate across multiple hosts. If you migrate an existing standard switch, or virtual adapter, to a Distributed Switch and the Distributed Switch becomes unnecessary or stops functioning, you can restore the standard switch to ensure that the host remains accessible.When you restore the standard switch, a new virtual adapter is created and the management network uplink that is currently connected to Distributed Switch is migrated to the new virtual switch.

You might need to restore the standard switch for the following reasons:

- The Distributed Switch is not needed or is not functioning.
- The Distributed Switch needs to be repaired to restore connectivity to vCenter Server and the hosts need to remain accessible.
- You do not want vCenter Server to manage the host. When the host is not connected to vCenter Server, most Distributed Switch features are unavailable to the host.

Prerequisites

Verify that your management network is connected to a distributed switch.

Procedure

1 From the direct console, select **Restore Standard Switch** and press Enter.

   If the host is on a standard switch, this selection is dimmed, and you cannot select it.

2 Press F11 to confirm.

Configuring System Logging

ESXi hosts run the syslog service (`vmsyslogd`) that writes messages from system components to log files and can forward messages to syslog collectors.

You can configure the amount and location of the logs. You can also create and apply log filters to modify the logging policy of an ESXi host.

When configuring remote hosts for syslog, you also need to open some specified ports in the ESXi host firewall to allow the transmission of log messages. For more information, see Opening the firewall for syslog emission to remote hosts.

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.

The syslog service receives, categorizes, and stores log messages for analyses that help you take preventive action in your environment.

Set ESXi Syslog by using the vSphere Client

You can use the vSphere Client command to configure the syslog service globally and edit various advanced settings.

Procedure

1 Browse to the ESXi 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.

Set ESXi Syslog by using the VMware Host Client

You can use the VMware Host Client to configure and edit syslog service parameters on ESXi hosts.

Procedure

1. In the VMware Host Client , under **Host**, click **Manage > System > Advanced settings**.
2. In the **Search** panel, type a syslog setting you want to define. See ESXi Syslog Options.
3. Select the setting and click **Edit option**.
4. Set the value as described in the table of parameters in ESXi Syslog Options.
5. Click **Save**.

Set ESXi Syslog by using ESXCLI

You can configure the syslog service on ESXi hosts by using the ESXCLI command: `esxcli system syslog config set <syslog option>`.

Prerequisites

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 Configuring the ESXi Firewall.

Note Using ESXCLI requires ESXi to open SSH logins, which is a security risk, and is not recommended. If you chose to use ESXCLI, make sure you use the `esxcli system syslog reload` command after setting each parameter to make sure that it takes effect.

Procedure

- Use the ESXCLI command `esxcli system syslog config set <syslog option>` to set a syslog option that you decide to enable. For example, to set the `Syslog.global.logHost` option, use the command `esxcli system syslog config set --loghost=<str>`

  After setting `Syslog.global.logHost`, ESXi hosts open and maintain connections to the syslog collectors, and the transmission of messages begins immediately. When ESXi generates a syslog message, it writes it to the appropriate log file on the ESXi host and also forwards it to all configured syslog collectors.

Fine-tune Syslog on ESXi Hosts

By using the right syslog settings, you can achieve proactive monitoring of your environment, reduce downtime and take preventive action on servers.

While setting up syslog, you need to consider several parameters that affect log file retention, syslog transmission, transmission length, error handling, and the set up of SSL certificates for secure syslog message transmission. What follows are recommendations for fine-tuning your syslog parameters. You can see a description of all available parameters at ESXi Syslog Options.

How to Specify Log File Retention

By default, log files cannot expand past a configured size. Once a log file reaches the configured size, logging is routed to a new log file and the oldest log file is deleted.

Note Best practice is to balance the rotate and size settings. Increasing the rotate setting ensures that syslog files are generated often enough to prevent any potential corruption or destruction from the other log files. Increasing the size setting reduces the time for switching to another log file. Optimal size settings are a multiple of 1024 KiB.

Use the `Syslog.global.defaultSize` setting to specify the log file maximum size in KiB, and `Syslog.global.defaultRotate` to set the maximum number of old log files to keep before rotating to a new log file. To change the log file retention parameters associated with a specific program, use the `Syslog.loggers.<progName>.rotate` and `Syslog.loggers.<progName>.size settings`, where `<progName>` is the name of the program whose parameters you want to adjust.

Manage Settings that Affect the Virtual Machine Log File

You can configure some settings that affect the virtual machine log file, `vmware.log`, either in the `vmx` file or in the `/etc/vmware/config` file. You must power off a virtual machine to edit the `vmx` file and edits take effect only on that virtual machine. If you use the `/etc/vmware/config` file, you must add the prefix *"vmx"* to the setting, for example `vmx.log.keepOld = "20"`, and edits affect all virtual machines on the ESXi host.

Table 4-22. Configurable settings for the vmware.log file

Parameter	Description	Example	Notes
`logging`	Disables all virtual machine logging.	The default value is `logging = "TRUE"` To disable virtual machine logging: `logging = "FALSE"`	Do not use this setting, because disabling virtual machine logging makes it difficult or impossible to get support for virtual machine problems. If you need to use this setting for some reason, you can only place it in the `vmx` file of a virtual machine.

`log.throttleBytesPerSec`	Controls when a log file throttles. Log file throttling occurs when writes to the `vmware.log` exceed the specified rate for a significant amount of time. This occurs when code within the VMX process, which controls a virtual machine, creates excessive log messages. The default value for this setting is 1 KB/sec.
	In case of log thottling, you see `<<< Log Throttled >>>` in the `vmware.log` file.
		`log.throttleBytesPerSec = "1500"`
		To disable log throttling, use
		`log.throttleBytesPerSec = "0xFFFFFFFF"`
			Log file thottling might obscure information necessary to diagnose problems with the affected virtual machine. If you need to disable log throttling, place the line in the example in the `vmx` file of the affected virtual machine. Remove the line after the debugging session ends.

`log.keepOld`	Controls the number of older `vmware.log` file to retain.	`log.keepOld = "20"`	Do not put the value of this setting below the default value (10). If virtual machines are frequently modified or moved, consider raising this setting to 20 or more.

`log.rotateSize`	Controls the maximum size of a `vmware.log` file in bytes.	`log.rotateSize = "2500000"`
		To disable limiting the maximum size of a `vmware.log` file, use
		`log.rotateSize = "0"`
			A value of this setting below 100,000 can cause a loss of critical log messages and affect virtual machine performance. In ESXi 7.x and earlier, the default value of this setting places no limit on the size of a `vmware.log` file. In ESXi 8.x and later, the default value of this setting is 2,048,000.

Table 4-22. Configurable settings for the vmware.log file (continued)

Parameter	Description	Example	Notes
log.fileName	Controls the name and location of virtual machine log files.	log.fileName = "myVMLog" This setting changes the name of the virtual machine log files from vmware.log to myVMlog. log.fileName = "/vmfs/volumes/vol1/myVM/myVM.log" This setting directs virtual machine log files to a directory on a different VMFS volume (vol1) by using myVM for a file name.	Do not place a log file outside the virtual machine directory to make sure that the collection of host support bundles picks up the log file, which can be critical to debug virtual machine problems.
log.fileLevel	Controls the minimum level at which messages are written to vmware.log. Every log message has a level associated with it. Levels below the specified setting are not added to a virtual machine log file. The virtual machine message log levels (from most to least restricted) are: ■ error ■ warning ■ notice ■ info (default) ■ trivia ■ debug ■ debug1 ■ debug2 ■ debug3 ■ debug4 ■ debug5 ■ debug6 ■ debug7 ■ debug8 ■ debug9 ■ debug10	log.fileLevel = "debug1"	Do not set a more restrictive level than "info" to avoid filtering out messages that are necessary for debugging virtual machine problems. Lower the level below "info" only upon request by licensed support. Restore the setting to "info" after debugging ends.
log.filter.minLogLevel.<groupName>	Controls the output of specialized debugging messages.	log.filter.minLogLevel.disklib = "debug5"	Use this setting only upon request by licensed support, who should provide one or more <groupName> parameters. Remove the setting after debugging ends.

Table 4-22. Configurable settings for the vmware.log file (continued)

Parameter	Description	Example	Notes
log.sysogID	Activates the sending of virtual machine log messages to the system logger of an ESXi host, such as the syslog.	log.syslogID = "vmx"	Use "vmx" as value for this setting to allow the ESXi syslog daemon, vmsyslogd, to send these messages to a separate log file.
log.syslogLevel	Controls the minimum level at which messages are output to the system logger of an ESXi host, such as the syslog.	log.syslogLevel = "debug"	The levels and functioning of this setting are identical to those for the log.fileLevel setting.

How to Specify Message Transmission to Remote Hosts

Optionally, you can configure ESXi to send syslog messages to one or more remote hosts, called syslog collectors, such as VMware Aria Operations for Logs (formerly VMware vRealize Log Insight and vCenter Log Insight), to collect syslog messages.

Note Best practice is that you configure each ESXi host to send syslog messages to at least one syslog collector. This helps ensure that the messages are preserved in case of a catastrophic system event and that you can process syslog messages in various ways, such as real-time categorization and analysis (for example, by type, time span, or machine), or archive messages.

Use the `Syslog.global.logHost` setting to define remote host specifications. Separate multiple remote host specifications with a comma (,). After setting `Syslog.global.logHost`, ESXi hosts open and maintain connections to the syslog collectors, and the transmission of messages begins immediately. When ESXi generates a syslog message, it writes it to the appropriate log file on the ESXi host and also forwards it to all configured syslog collectors.

In addition to syslog messages, audit messages can also be transmitted to syslog collectors for security purposes. Audit records track security-related activity on the ESXi host. For more information about audit records, see Audit Records.

Note Consult with your company security response team if and how to set audit records. Certified configurations usually require audit records to be activated.

What follows is the syntax for `Syslog.global.logHost` remote host specifications:

```
protocol://target[:port][?formatter=value[&framing=value]]
```

Parameter	Description	Notes
`protocol`	Specifies the networking protocol. Valid values are `udp`, `tcp`, and `ssl`.	The `ssl` protocol specifies that transmission of syslog messages is encrypted. The `tcp` and `udp` protocols do not encrypt the transmission. **Note** If capturing syslog messages or audit messages is critical to your system, avoid using the `udp` protocol because the networking infrastructure external to ESXi might drop UDP messages.
`target`	Specifies the remote host. You can use either an IPV4 or IPV6 address, or a host name.	When you use an IPV6 address, you must embed it in square brackets `[xxx]`, where `xxx` is the IPV6 address.
`port`	(Optional) Specifies the remote host port to use. If you use UDP or TCP, the default port is 514. If you use SSL, the default port is 1514. If you choose to use different ports from 514 or 1514, you must adjust the ESXi firewall to open the port.	For details how to open the ESXi firewall for the port specified in each remote host specification, see Configuring the ESXi Firewall.
`formatter`	Specifies how transmissions are formatted. The formatter must be RFC 3164 or RFC 5424.	RFC 3164 is the default.
`framing`	Specifies if transmissions are framed. Framing must be `non_transparent` or `octet_counting`.	The default is `non_transparent`. Transmissions in RFC 5424 format must specify `octet_counting` framing. For more information, see Protocols, Formats and Framing of ESXi Syslog Messages.

Examples of remote machine specifications:

Syslog.global.logHost string example	Notes
tcp://10.176.130.7:12345?formatter=RFC_3164	Transmits syslog messages to 10.176.130.7 using TCP/IP and port 12345. Transmission format is RFC 3164 with no framing.
tcp://10.176.130.7:12345? formatter=RFC_3164&framing=octet_counting	Transmits syslog messages to 10.176.130.7 using TCP/IP and port 12345. Transmission format is RFC 3164 and framing is `octet_counting`.
tcp://10.176.130.7:12345? formatter=RFC_5424&framing=octet_counting	Transmits syslog messages to 10.176.130.7 using TCP/IP and port 12345. Transmission format is RFC 5424 and framing is `octet_counting`.
tcp://[2001:db8:85a3:8d3:1319:8a2e:370:7348]	Transmits syslog messages to an IPV6 address using port 1514.

Syslog.global.logHost string example	Notes
tcp://[2001:db8:85a3:8d3:1319:8a2e:370:7348]:5432	Transmit syslog messages to an IPV6 address using port 54321.
udp://company.com	Transmits syslog messages to `company.com` using UDP and port 514.
udp://company.com,tcp://10.20.30.40:1050	Transmits syslog messages to two remote hosts. The first remote host uses UDP to communicate with `company.com` using port 514. The second remote host uses TCP to communicate with the IPV4 address `10.20.30.40` using port 1050.
ssl://company.com	Transmits syslog messages to `company.com` using SSL (TLS) and port 514.

Maximum Message Transmission Length

If you use UDP, the maximum syslog message transmission length is 480 bytes for IPV4 and 1180 bytes for IPV6.

For TCP or SSL, the default maximum syslog message transmission length is 1 kibibyte (KiB). You can increase this length by using the `Syslog.global.remoteHost.maxMsgLen` parameter. The maximum value is 16 KiB. Messages longer than 16 KiB are truncated.

Note If increasing the maximum transmission length is necessary, best practice is to increase the length only as much as specifically necessary.

Increasing the maximum syslog message length can cause problems if the networking and syslog infrastructure external to ESXi is unable to handle messages longer than 1 KiB.

Note Best practice is that you do not use UDP to transmit syslog messages due to the packet length constraints and the possibility that the external networking infrastructure might drop the messages.

Considerations for Certificates When Configuring SSL Transmissions to Remote Hosts

When configuring ESXi to transmit syslog messages to remote hosts using SSL, you must add an SSL certificate for each remote host to the ESXi host CA store. For more information, see Certificate Management for ESXi Hosts and Manage CA Certificates with ESXCLI.

Note Consult with your syslog collector documentation on how to configure the collector for secure receipt of syslog messages using SSL and a private key.

Additional SSL Transmission Parameters

An ESXi system complying with security certification requirements might require activating X509 CRL checks. You turn on the advanced settings `Syslog.global.certificate.strictX509Compliance` and `Syslog.global.certificate.checkCRL` by changing the default value of `false` to `true`.

Due to implementation limitations, if you activate CRL checks by using the setting `Syslog.global.certificate.checkSSLCerts`, then all certificates in a certificate chain must provide a CRL link. By default, the setting is active. You can deactivate SSL certificate checks by changing the setting to `false`, but this is not a best practice. You might need to turn off SSL certificate checks when troubleshooting communications with a remote host, but do this only for a limited time.

Where to Find Syslog Daemon Error and Status Information

The ESXi syslog daemon uses the log file `/var/run/log/vmsyslogd.log` to store status and error information, including dropped messages. If audit record transmission is active, the syslog daemon also emits audit records related to its operation, such as daemon start, stop, and error conditions, which allows you to verify that the syslog daemon runs properly.

How to Change the Default Syslog Log File Storage Area

The default syslog log file storage area is `/var/run/log`, local to each ESXi host. Use the `Syslog.global.logDir` syslog configuration variable to change the default syslog log file storage area, as long as the location resides on persistent storage. If `Syslog.global.logDir` is configured to a persistent store shared by multiple ESXi hosts to store their syslog log files, change the `Syslog.global.logDirUnique` setting to `true` to prevent mixing logs. The `Syslog.global.logDirUnique` setting makes sure that each ESXi machine gets a unique name added to the `Syslog.global.logDir` path, separating the log files from other hosts.

Syslog Message Queueing for Remote Hosts and Message Drops

Once syslog emissions start, they never stop except for ESXi reboots and failures, or a syslog reconfiguration to stop.

To avoid dropping messages, ESXi uses an in-memory queue that allows the `vmsyslogd` service to handle the following conditions for a short time:

- ESXi generates log messages at a faster rate than the `vmsyslogd` service can process and transmit
- Network connectivity between ESXi and the remote host fails

If either of these conditions continues for a long period, the capacity of the in-memory queue might not be sufficient and `vmsyslog` might stop transmitting messages and audit logs to remote hosts. No data is lost, because dropped messages are written to the `/var/run/log/vmsyslogd-dropped.log` file.

To minimize the chance that message dropping occurs, place the `vmsyslogd` log files on the fastest available storage and configure `vmsyslogd` and its syslog collectors on a network with end-to-end bandwidth greater than 1 GigE.

Note Consider an end-to-end bandwidth of 2.5 GigE or more, preferably 10 GigE, to optimize performance and prevent message drops.

If you see excessive logs from an application, file a support service request for analysis and correction.

You can see statistics about message drops in the syslog daemon log file.

You can see dropped messages at `/var/run/log/vmsyslogd-dropped.log`. This log file has retention settings specific to it, similar to those for the program-specific retention parameters. The dropped messages log file retention parameters are: `Syslog.global.droppedMsgs.fileRotate` and `Syslog.global.droppedMsgs.fileSize`.

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 Always configure persistent storage before you set any of the audit record parameters or the `Syslog.global.logDir` parameter.

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 4-23. 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][?formatter=value[&framing=value]]`. The protocol must be one of `tcp`,`udp`, or `ssl`. The value of a port can be any decimal number between 1 and 65535. If a port is not provided, UDP and TCP use 514; SSL uses 1514. For example: `ssl://hostName1:1514`. The formatter must be RFC_3164 or RFC_5424; RFC_3164 is the default. The framing must be `non_transparent` or `octet_counting`; the default is `non_transparent`. For more information, see Fine-tune Syslog on ESXi Hosts.
`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 4-23. 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 4-24. 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>	Creates an audit record storage directory and unless specified, sets /scratch/auditLog as the default location. You must not manually create an audit record storage directory and you cannot change the audit record storage directory while audit record storage is enabled (see Syslog.global.auditRecord.storageEnable).

Table 4-24. Syslog Options Available Starting from ESXi 7.0 Update 1 (continued)

Option	ESXCLI command	Description
`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`.
`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.

Table 4-24. Syslog Options Available Starting from ESXi 7.0 Update 1 (continued)

Option	ESXCLI command	Description
`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.
`Syslog.global.logFilters`	`esxcli system syslog config logfilter [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 4-24. 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. This setting does not affect the UDP protocol. RFC 5426 specifies the UDP message lengths that can be safely accepted at 480 bytes for IPV4 and 1180 bytes for IPV6. Because of these restrictions, and because the networking infrastructure can arbitrary drop UDP packets, the use of UDP for transmitting critical syslog messages is not recommended.
`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.

Protocols, Formats and Framing of ESXi Syslog Messages

Starting with ESXi 8.0,the syslog service uses three parameters to define messages and audit records - protocol, formatting, and framing.

The supported protocols are UDP, TCP, and TLS (SSL). Formatting of syslog messages is defined by either RFC 3164 or RFC 5424. Framing specifies how a message is encapsulated. Framing of encapsulated messages is defined as transparent, also called octet_counting, or non-transparent, if a message is not encapsulated. Transparent framing ensures that new lines embedded in a message do not confuse a syslog collector. Syslog messages sent by using the UDP protocol are considered transparently framed; a syslog collector is expected to understand this and accept the transmission as a single message.

RFC 3164 sets the maximum total length of a syslog message at 1024 bytes, while RFC 5424 specifies that syslog messages of length 2048 or less should be safely accepted. Modern systems generally accept messages longer than these specifications, but you need to confirm the actual maximum length with the specific syslog infrastructure and parameters of your environment.

The default maximum length for remote host messages in ESXi 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 external to ESXi has a maximum message length less than the maximum message length of ESXi.

Syslog messages that the vmsyslogd transmits consist of structured data, a property list formatted in compliance with RFC 5424, and free format, or unstructured, data.

When a message is longer than the maximum length, ESXi 8.0 mitigates the message, trying to preserve as much of the structured data as possible.

When a message is mitigated, three parameters are either added to existing structured data or structured data is created to contain these parameters: `msgModified`, `remoteHostMaxMsgLen`, and `originalLen` .

The `msgModified` parameter indicates how the mitigation impacts the message: only structured data, only unstructured data, or both.

The `remoteHostMaxMsgLen` parameter specifies the maximum message length that ESXi can handle.

The `originalLen` parameter specifies the message length before it is mitigated.

Supported options for protocols, formatting and framing of ESXi syslog messages:

Formatting	Framing	UDP	TCP	SSL	Comments
Unspecified	Unspecified	Supported RFC 5426	Supported	Supported	Formatting of messages complies to RFC 3164, only timestamps are in RFC 3339 format. Structured data is prepended to each message. Framing defaults to non-transparent with TCP or SSL (TLS) and embedded newlines in structured data might corrupt messages. With UDP, packets are framed.
Unspecified	Non_transparent	Forbidden	Supported	Supported	Formatting of messages complies to RFC 3164, only timestamps are in RFC 3339 format. Structured data is prepended to each message. Framing defaults to non-transparent with TCP or SSL (TLS) and embedded newlines in structured data might corrupt messages.

Formatting	Framing	UDP	TCP	SSL	Comments
Unspecified	Octet_counting	Forbidden	Supported RFC 6587	Supported RFC 6587	Formatting of messages complies to RFC 3164, only timestamps are in RFC 3339 format. Structured data is prepended to each message.
RFC 5424	Unspecified	Supported RFC 5426	Supported RFC 5425	Supported RFC 5424	Formatting of messages complies to RFC 5424. Framing defaults to octet-counting with TCP or SSL (TLS). With UDP, framing might not be explicitly specified.
RFC 5424	Non_transparent	Forbidden	Not Supported	Not Supported	Not supported because embedded newlines in structured data might create corrupted messages.
RFC 5424	Octet_counting	Forbidden	Supported RFC 5425	Supported RFC 5425	Formatting of messages complies to RFC 5424.
RFC 3164	Unspecified	Supported RFC 5426	Supported	Supported	Formatting of messages complies to RFC 3164, only timestamps are in RFC 3339 format. Structured data is prepended to each message. Framing defaults to non-transparent with TCP or SSL (TLS) and embedded newlines in structured data might corrupt messages. With UDP, packets are framed.
RFC 3164	Non_transparent	Forbidden	Supported	Supported	Formatting of messages complies to RFC 3164, only timestamps are in RFC 3339 format. Structured data is prepended to each message. Framing defaults to non-transparent with TCP or SSL (TLS) and embedded newlines in structured data might corrupt messages.
RFC 3164	Octet_counting	Forbidden	Supported RFC 6587	Supported RFC 6587	Formatting of messages complies to RFC 3164, only timestamps are in RFC 3339 format. Structured data is prepended to each message.

Log File Formats

Starting with ESXi 8.0, the format of log files is standardized and is expressed in Augmented Backus-Naur Form (ABNF).

In ESXi 8.0, log files are written either directly, from a single service such as VMX, or indirectly, when logs from a service are submitted to a syslog. For example, VMX always writes log messages in the `vmware.log` file of each virtual machine. To spare system resources, VMX does not submit log messages to the syslog. On the other hand, in some log files that `vmsyslogd` generates, you see messages from multiple programs, because the ESXi syslog daemon creates and manages all log files, and messages to these files, from multiple services.

Format for direct log messages:

Parameter	Value
`LOG-MSG`	HEADER SP MSG
`HEADER`	TIMESTAMP SP SEVERITY SP THREAD-NAME SP OPID
`TIMESTAMP`	FULL-DATE T FULL-TIME (Complies to RFC 5424 with the requirement for UTC/GMT formatting and resolution in milliseconds, or more granular where possible.)
`FULL-DATE`	DATE-FULLYEAR - DATE-MONTH - DATE-MDAY
`DATE-FULLYEAR`	4DIGIT
`DATE-MONTH`	2DIGIT ; 01-12
`DATE-MDAY`	2DIGIT ; 01-28, 01-29, 01-30, 01-31 based on month/year
`FULL-TIME`	TIME-HOUR : TIME-MINUTE : TIME-SECOND[TIME-SECFRAC] Z
`TIME-HOUR`	2DIGIT ; 00-23
`TIME-MINUTE`	2DIGIT ; 00-59
`TIME-SECOND`	2DIGIT ; 00-59
`TIME-SECFRAC`	'.' 1*6DIGIT
`SEVERITY`	SEVERITY-STRING SEVERITY-VALUE [LINE-MARKER]
`SEVERITY-STRING`	`Em`/ `Al` / `Cr` / `Er` / `Wa` / `No` / `In` / `Db` (The 8 severity levels specified in RFC 5424 are abbreviated as follows: ■ `Em` - Emergency ■ `Al` - Alert ■ `Cr` - Critical ■ `Er` - Error ■ `Wa` - Warning ■ `No` - Notice ■ `In` - Informational ■ `Db` - Debug

SEVERITY-VALUE	(*DIGIT) (The SEVERITY-VALUE is an optional expression of the numeric value associated with the SEVERITY-STRING. This allows levels supported by a logger to be collapsed into the 8 required strings with no loss of information (e.g. Db(5) - debug, level 5).)
LINE-MARKER	+ (The LINE-MARKER is added to each subsequent line generated from a multi-line submission. It identifies multiline submissions and prevents a log injection security attack.)
NILVALUE	- (A single threaded program might not have a thread name and NILVALUE is acceptable.)
THREAD-NAME	NILVALUE / 1*32PRINTUSASCII (The component (APP-NAME) is implied when a single program writes the file and no component field is necessary, only the thread name.)
OPID	NILVALUE / 1*128UTF-8-STRING
STRUCTURED-DATA	1*SD-ELEMENT
SD-ELEMENT	[SD-ID *(SP SD-PARAM)]
SD-PARAM	PARAM-NAME %d34 PARAM-VALUE %d34
SD-ID	SD-NAME
PARAM-NAME	SD-NAME
PARAM-VALUE	UTF-8-STRING ; characters '', '\' and ']' MUST be escaped.
SD-NAME	1*32PRINTUSASCII ; except '', SP, ']', %d34 ()
MSG	[STRUCTURED-DATA SP] UTF-8-STRING

Format for log files that the `vmsyslogd` service manages:

Parameter	Value
LOG-MSG	HEADER SP MSG
HEADER	TIMESTAMP SP SEVERITY SP APP-NAME [PROC-IDENTIFIER] :
APP-NAME	1*32PRINTUSASCII
PROC-IDENTIFIER	[*DIGITS] ; the PID associated with APP-NAME
TIMESTAMP	FULL-DATE T FULL TIME (resolution in milliseconds or more granular where possible.)
FULL-DATE	DATE-FULLYEAR - DATE-MONTH - DATE-MDAY
DATE-FULLYEAR	4DIGIT
DATE-MONTH	2DIGIT ; 01-12
DATE-MDAY	2DIGIT ; 01-28, 01-29, 01-30, 01-31 based on month/year
FULL-TIME	TIME-HOUR : TIME-MINUTE : TIME-SECOND[TIME-SECFRAC] Z
TIME-HOUR	2DIGIT ; 00-23
TIME-MINUTE	2DIGIT ; 00-59
TIME-SECOND	2DIGIT ; 00-59

TIME-SECFRAC	'.' 1*6DIGIT
SEVERITY-STRING	Em/ Al / Cr / Er / Wa / No / In / Db (The 8 severity levels specified in RFC 5424 are abbreviated as follows: ■ Em - Emergency ■ Al - Alert ■ Cr - Critical ■ Er - Error ■ Wa - Warning ■ No - Notice ■ In - Informational ■ Db - Debug
SEVERITY	SEVERITY-STRING PRI-STRING [LINE-MARKER]
PRIVAL	1*3DIGIT ; range 0 .. 191 (the MSG PRI; contains facility and severity values, ORed together)
PRI-STRING	(PRIVAL) (The PRIVAL contains the bits from the message PRI. This allows one to see the Facility of the message, as well as the severity bits themselves._
LINE-MARKER	+ (The LINE-MARKER is added to each subsequent line generated from a multi-line submission. It identifies multiline submissions and prevents a log injection security attack.)
STRUCTURED-DATA	1*SD-ELEMENT
SD-ELEMENT	[SD-ID *(SP SD-PARAM)]
SD-PARAM	PARAM-NAME %d34 PARAM-VALUE %d34
SD-ID	SD-NAME
PARAM-NAME	SD-NAME
PARAM-VALUE	UTF-8-STRING ; characters '', '\' and ']' MUST be escaped.
SD-NAME	1*32PRINTUSASCII ; except '', SP, ']', %d34 ()
MSG	[STRUCTURED-DATA SP] UTF-8-STRING

Audit Records

ESXi audit records, with facility code 13, are compliant to both RFC 3164 and 5424 formats and you find them in the structured data section. In the audit record, you also find event-based traceability information when such data is available. Audit records are stored in a special format, not a regular log file. You can access audit records locally by using the `viewAudit` program and the Virtual Infrastructure Management functionality `FetchAuditRecords`. Do not read, use, or edit an audit record storage file directly. Locally stored audit records comply with RFC 5424 transmission format where the `HOSTNAME` and `MSGID` are always `NILVALUE`.

ESXi Syslog Message Transmission Formats

ESXi 8.0 formats syslog messages in compliance with either RFC 3164 or RFC 5424.

Note The timestamps associated with RFC 3164 messages are in RFC 3339 format, an exception to the RFC 3164 specification.

The definition of the ESXi transmission formats for RFC 3164 and RFC 5424 is in Augmented Backus-Naur Form (ABNF).

RFC 3164 Transmission Message Format

Parameter	Value
SYSLOG-MSG	HEADER SP MSG
HEADER	PRI TIMESTAMP SP HOSTNAME SP APP-NAME [PROC-IDENTIFIER] ":"
PRI	"<" PRIVAL ">"
PRIVAL	1*3DIGIT ; range 0 .. 191 (the MSG PRI; contains facility and severity values, ORed together)
APP-NAME	1*32PRINTUSASCII
HOSTNAME	1*255PRINTUSASCII
PROC-IDENTIFIER	"[" *DIGITS "]" ; the PID associated with APP-NAME
TIMESTAMP	FULL-DATE "T" UTC-TIME (Never set a TIME-OFFSET)
FULL-DATE	DATE-FULLYEAR "-" DATE-MONTH "-" DATE-MDAY
DATE-FULLYEAR	4DIGIT
DATE-MONTH	2DIGIT ; 01-12
DATE-MDAY	2DIGIT ; 01-28, 01-29, 01-30, 01-31 based on month/year
UTC-TIME	TIME-HOUR ":" TIME-MINUTE ":" TIME-SECOND [TIME-SECFRAC] "Z"
TIME-HOUR	2DIGIT ; 00-23
TIME-MINUTE	2DIGIT ; 00-59
TIME-SECOND	2DIGIT ; 00-59
TIME-SECFRAC	"." 1*6DIGIT
STRUCTURED-DATA	1*SD-ELEMENT
SD-ELEMENT	"[" SD-ID *(SP SD-PARAM) "]"
SD-PARAM	PARAM-NAME "%d34 PARAM-VALUE %d34
SD-ID	SD-NAME
PARAM-NAME	SD-NAME

PARAM-VALUE	UTF-8-STRING ; characters '"', '\' and ']' MUST be escaped.
SD-NAME	1*32PRINTUSASCII ; except ', SP, ']', %d34 (")
MSG	[STRUCTURED-DATA SP] UTF-8-STRING

RFC 5424 Transmission Message Format

Parameter	Value
SYSLOG-MSG	HEADER SP STRUCTURED-DATA [SP MSG]
HEADER	PRI VERSION SP TIMESTAMP SP HOSTNAME SP APP-NAME SP PROCID SP MSGID
PRI	"<" PRIVAL ">"
PRIVAL	1*3DIGIT ; range 0 .. 191; contains facility and severity data
VERSION	NONZERO-DIGIT 0*2DIGIT
HOSTNAME	1*255PRINTUSASCII
APP-NAME	1*48PRINTUSASCII
NILVALUE	'-'
PROCID	NILVALUE | *DIGITS ; the PID associated with APP-NAME
MSGID	NILVALUE
TIMESTAMP	FULL-DATE "T" UTC-TIME
FULL-DATE	DATE-FULLYEAR "-" DATE-MONTH "-" DATE-MDAY
DATE-FULLYEAR	4DIGIT
DATE-MONTH	2DIGIT ; 01-12
DATE-MDAY	2DIGIT ; 01-28, 01-29, 01-30, 01-31 based on month/year
UTC-TIME	TIME-HOUR ":" TIME-MINUTE ":" TIME-SECOND [TIME-SECFRAC] "Z"
TIME-HOUR	2DIGIT ; 00-23
TIME-MINUTE	2DIGIT ; 00-59
TIME-SECOND	2DIGIT ; 00-59
TIME-SECFRAC	"." 1*6DIGIT
STRUCTURED-DATA	NILVALUE / 1*SD-ELEMENT
SD-ELEMENT	"[" SD-ID *(SP SD-PARAM) "]"
SD-PARAM	PARAM-NAME " %d34 PARAM-VALUE %d34
SD-ID	SD-NAME

PARAM-NAME	SD-NAME
PARAM-VALUE	UTF-8-STRING ; characters '"', '\' and ']' MUST be escaped.
SD-NAME	1*32PRINTUSASCII ; except '', SP, ']', %d34 (")
MSG	MSG-UTF8
MSG-UTF8	BOM UTF-8-STRING
BOM	%xEF.BB.BF

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.

Starting with vSphere 7.0 Update 2, you can add log filters and enable log filtering by using ESXCLI. A log filter, once established, remains in place until it is removed, even across ESXi reboots.

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.

You must enable the log filtering capability and reload the syslog daemon to activate the log filters on the ESXi host.

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

For example, to get the list of available log filters, run the following command: `[root@xxx-xx-dhcp-xx-xx:~] esxcli system syslog config logfilter list`.

Use the `set` command to activate or deactivate log filtering: `[root@xxx-xx-dhcp-xx-xx:~] esxcli system syslog config logfilter set`.

Use the `add` command to add a log filter and the `remove` command to remove a log filter.

Use the `get` command to determine if log filtering is enabled.

A log filter is specified by three components and uses the following syntax: `numLogs | ident | logRegexp`.

Parameter	Description
`numLogs`	Specifies the number of matches of the logRegexp Python regular expression that will be allowed before filtering begins.
`ident`	The `ident` string is how an application identifies itself to the syslog facility. The `logRegexp` filter must be associated with the same application. You can find the `ident` string of an application by inspecting the log files in `/var/run/log`. The third field of each log file begins with the `ident` string and ends with `[`.
`logRegexp`	Python regular expression that identifies the messages which you want to filter out.

For example, to filter out all messages from the hostd daemon that contain the word "mark" after the tenth occurrence, use the following command: `esxcli system syslog config logfilter add --filter="10|Hostd|mark"`.

To remove the log filter, use the command `esxcli system syslog config logfilter remove --filter="10|Hostd|mark"`.

For more information, see ESXi Syslog Options.

Prerequisites

You can create log filters to reduce the number of repetitive entries in the ESXi logs and to denylist specific log events entirely.

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

Set the Host Image Profile Acceptance Level

The Host Image Profile acceptance level determines which vSphere installation bundles (VIBs) are accepted for installation.

VIB signatures are checked and accepted for installation based on a combination of the VIB acceptance level and the host image profile acceptance level. VIBs are tagged with an acceptance level that depends on their signature status.

See Working with Acceptance Levels.

Prerequisites

Required privileges: **Host.Configuration.SecurityProfile** and **Host.Configuration.Firewall**

Procedure

1. From the vSphere Client, connect to the vCenter Server.
2. Select the host in the inventory.
3. Click the **Configure** tab.

4 Under System, select **Security Profile**.

5 Scroll down to Host Image Profile Acceptance Level, and click **Edit**.

6 Select the acceptance level and click **OK**.

Table 4-25. Host Image Profile Acceptance Levels

Host Image Profile Acceptance Level	Accepted Levels of VIBs
VMware Certified	VMware Certified
VMware Accepted	VMware Certified, VMware Accepted
Partner Supported	VMware Certified, VMware Accepted, Partner Supported
Community Supported	VMware Certified, VMware Accepted, Partner Supported, Community Supported

Remove All Custom Packages on ESXi

After adding custom packages, you might decide to remove them.

Prerequisites

Before you remove custom packages, shut down or migrate running virtual machines off of the ESXi host.

Procedure

1 Reboot the ESXi host.

2 In the direct console, select **Remove Custom Extensions** and press F11 to confirm.

3 Reboot the host.

Results

All custom packages are removed.

Modify ESXi Configuration Files

You can modify ESXi configuration files in the ESXi Configuration Store (ConfigStore) by using the `/bin/configstorecli` tool.

The goal of the ESXi Shell tool `configstorecli`, introduced in ESXi 7.0 Update 1, is to manage all configurations for an ESXi host centrally, instead of using different methods and a variety of configuration files.

For more information on how to manage ConfigStore, see VMware knowledge base articles 82227 and 93720.

Deactivate Support for Non-ASCII Characters on ESXi

You can deactivate support for non-ASCII characters for virtual machine files and directory names by using two methods, depending on the ESXi version.

By default, ESXi supports the use of non-ASCII characters for virtual machine file and directory names.

If you need to deactivate support for non-ASCII characters, for ESXi 7.0 Update 2 and later, see VMware knowledge base articles 82227 and 93720.

Prior to ESXi 7.0 Update 2, you can deactivate this support by modifying the `/etc/vmware/hostd/config.xml` file with the following steps:

Procedure

1. Using a text editor, open the `/etc/vmware/hostd/config.xml` file for the ESXi host.
2. Within the `<config></config>` tag, add the following code.

   ```
   <g11nSupport>false</g11nSupport>
   ```

3. Save and close the file.
4. Reboot the host.

   After you deactivate this support, you can still enter non-ASCII characters for virtual machine names. vSphere user interfaces display the virtual machine names in the non-ASCII characters, but ESXi converts the actual file and directory names to ASCII strings.

Reset the System Configuration

If you are having trouble determining the source of a problem with your ESXi host, you can reset the system configuration.

Changes in the system configuration can be related to various problems, including problems with connectivity to the network and devices. Resetting the system configuration might solve such problems. If resetting the system configuration does not solve the problem, it can still rule out configuration changes made since the initial setup as the source of the problem.

When you reset the configuration, the software overrides all your configuration changes, deletes the password for the administrator account (root), and reboots the host. Configuration changes made by your hardware vendor, such as IP address settings and license configuration, might also be deleted.

Resetting the configuration does not remove virtual machines on the ESXi host. After you reset the configuration defaults, the virtual machines are not visible, but you make them visible again by reconfiguring storage and reregistering the virtual machines.

Caution When you reset the configuration defaults, users accessing the host lose connectivity.

Prerequisites

Before resetting the configuration, back up your ESXi configuration in case you want to restore your configuration.

Procedure

1. Back up the configuration by using the `Get-VMHostFirmware` PowerCLI cmdlet.
2. From the direct console, select **Reset System Configuration** and press Enter.
3. Press F11 to confirm.

Results

The system reboots after all settings are reset to the default values.

After You Install and Set Up ESXi

After you install and set up ESXi, you can use manage hosts by various interfaces, license the hosts, and back up your configuration.

After ESXi is installed and set up, you can manage the host by using the vSphere Client and vCenter Server, license the host, and back up your ESXi configuration. You can also use the VMware Host Client to connect directly to the ESXi host and to manage it. For information about installing and using the VMware Host Client, see *vSphere Single Host Management*.

Note When you install a standalone ESXi host with TMP enabled that is not connected to a vCenter Server instance, create a backup of the ESXi configuration recovery key. To get the recovery key, run the command `esxcli system settings encryption recovery list` on the ESXi host and note it down. It is possible that the host might not be able to complete booting due to host configuration encryption related problems and in such cases, you can restore the host configuration by using the recovery key and running the configuration recovery process.

For best practices and VMware recommendations, see Best Practices for Secure ESXi Configuration.

Licensing ESXi Hosts

After you install ESXi, it has a 60-day evaluation period during which you can explore the full set of vSphere features provided with a vSphere Enterprise Plus license. You must assign the host an appropriate license before the evaluation period expires.

ESXi hosts are licensed with vSphere licenses that have per-CPU capacity. To license hosts correctly, you must assign them a vSphere license that has enough CPU capacity to cover all CPUs in the hosts. The license must support all features that the hosts are using. For example, if the hosts are connected to a vSphere Distributed Switch, you must assign a license that has the vSphere Distributed Switch feature.

You can use one of following methods to license ESXi hosts:

- License multiple hosts at a time by using the license management function in the vSphere Client. The hosts must be connected to a vCenter Server system. For more information, see *vCenter Server and Host Management*.
- Set up bulk licensing by using PowerCLI commands. Bulk licensing works for all ESXi hosts, but is especially useful for hosts provisioned with Auto Deploy. See Set Up Bulk Licensing
- License individual ESXi hosts by using a direct connection with the VMware Host Client. For information about assigning a license key to an ESXi host, see *vSphere Single Host Management*.

  For more information, see Licensing and Subscription in vSphere.

Recording the License Key of an ESXi Host

If a host becomes inaccessible or unbootable, you should have a record of its license key. You can write down the license key and tape it to the server, or put the license key in a secure location. You can access the license key from the direct console user interface or the vSphere Client.

View the License Keys of ESXi Hosts from the vSphere Client

You can view the license keys of the hosts that are connected to a vCenter Server system through the vSphere Client.

Procedure

1. In the vSphere Client, select **Administration**.
2. Under Licensing, select **Licenses**.
3. On the **Assets** tab, select **Hosts**.
4. In the License column, click a license.

Results

You view information about the license, such as its usage and license key.

Access the ESXi License Key from the Direct Console

If you have physical access to the host or remote access to the direct console, you can use the direct console to access the ESXi license key.

Procedure

- From the direct console, select **View Support Information**.

  The license key appears in the form XXXXX-XXXXX-XXXXX-XXXXX-XXXXX, labeled License Serial Number.

  Note The physical machine serial number also appears, labeled Serial Number. Do not confuse the license key with the physical machine serial number.

View System Logs

System logs provide detailed information about system operational events.

Procedure

1. From the direct console, select **View System Logs**.
2. Press a corresponding number key to view a log.

   vCenter Server Agent (vpxa) logs appear if you add the host to vCenter Server.
3. Press Enter or the spacebar to scroll through the messages.
4. Perform a regular expression search.
   a. Press the slash key (/).
   b. Type the text to find.
   c. Press Enter.

   The found text is highlighted on the screen.
5. Press q to return to the direct console.

What to do next

See also Configure Syslog on ESXi Hosts.

Troubleshooting ESXi Booting 5

The ESXi booting troubleshooting topics provide solutions to problems that you might encounter during the ESXi booting.

Read the following topics next:

- Host Stops Unexpectedly at Bootup When Sharing a Boot Disk with Another Host
- Host Fails to Boot After You Install ESXi in UEFI Mode

Host Stops Unexpectedly at Bootup When Sharing a Boot Disk with Another Host

When more than one host, either physical or virtual, boots from the same shared physical disk or LUN, they cannot use the same scratch partition.

Problem

The host stops at bootup when sharing a boot disk with another host.

Cause

More than one ESXi host can share the same physical disk or LUN. When two such hosts also have the same scratch partition configured, either of the hosts can fail at bootup.

Solution

1. Set the hosts to boot sequentially, and boot the hosts.

   This setting lets you start the hosts so that you can change the scratch partition for one of them.
2. From the vSphere Client, connect to the vCenter Server.
3. Select the host in the inventory.
4. Click the **Configure** tab.
5. Under System, select **Advanced System Settings**.
6. Select **ScratchConfig**.

   The **ScratchConfig.CurrentScratchLocation** text box shows the current location of the scratch partition.

7 In the **ScratchConfig.ConfiguredScratchLocation** text box, enter a directory path that is unique for this host.

   For example, ***/vmfs/volumes/DatastoreUUID/DatastoreFolder***.

8 Reboot the host for the changes to take effect.

Host Fails to Boot After You Install ESXi in UEFI Mode

After you install ESXi on a host machine in UEFI mode, the machine might fail to boot.

Problem

When you install or upgrade ESXi, the installer tries to create an UEFI boot option named `VMware ESXi` and make it the default boot option. When you reboot after installing ESXi the reboot might fail. This problem is accompanied by an error message similar to `No boot device available.`

Cause

- When the installer creates the UEFI boot option, a problem occurs while writing to the NVRAM on the host motherboard.
- The host firmware does not recognize the attempt to set the UEFI boot option as the first boot option, or the firmware overrides the boot order.
- Either or both BIOS and firmware are outdated.
- The boot disk has an MBR or MSDOS partition table. Due to a technical limitation, the UEFI boot option is only created for a GUID Partition Table (GPT) partition table.

Note UEFI firmware attempts to load the boot image from the EFI system partition, which is FAT based, on the disk. Booting from the EFI system partition only works if the disk is laid out by using a GPT. If the boot disk has an MBR or MSDOS partition table, a UEFI boot fails. You cannot add a boot entry for MBR. If the disk is fully consumed by ESXi, it cannot be converted to GPT, and you must boot in legacy BIOS mode.

Solution

1 While the error message is displayed on screen, open the boot options menu. Depending on your system, the boot options menu might open with a keyboard shortcut, in the BIOS menu, or in a BMC, iLO or iDRAC interface.

2 Check if a boot option `VMware ESXi` exists and try to boot from it. If the boot is successful, change the boot order and set `VMware ESXi` as the first boot option.

3 If the problem is not resolved, select an option similar to **Add boot option**.

   The wording and location of the option might vary, depending on your system.

4 Select the file `\EFI\BOOT\BOOTx64.EFI` on the disk that you installed ESXi on.

5 Change the boot order so that the host boots from the option that you added.

Decommission an ESXi Host

6

If you do not want your server to be an ESXi host, you can decommission the ESXi host machine.

Procedure

1. Remove VMFS datastores on the internal disks so that the internal disks are no longer set up to store virtual machines.
2. Change the boot setting in the BIOS so that the host no longer boots into ESXi.

   If you installed ESXi in UEFI mode, delete the boot option `VMware ESXi` or any other boot option created manually.
3. Install another operating system in its place.