Global Knowledge®

Expert Reference Series of White Papers

ESX vs. ESXi

ESX versus ESXi

John Hales, Global Knowledge Instructor, VCP, VCI, A+, Network+, MCSE, MCDBA, MCT, EMCSA

Introduction

With the recent statement from VMware that ESX 4.1 will be the last major version of ESX with the service console included, many are starting to look toward the future and the role of ESXi in their infrastructures. This white paper will review the following topics:

- A brief history of the service console and its purpose
- The differences in features between ESX and ESXi, with an emphasis on ESX and ESXi 4.1
- The architectural differences between ESX and ESXi
- Methods to manage both ESX and ESXi, including the DCUI (Direct Console User Interface)
- Modifying existing scripts for use with ESXi, including the role of PowerCLI, the VMA, and the vCLI

With VMware's announcement that we will all be moving to ESXi, knowing how to begin and manage the transition process is important.

Brief History of the Service Console and its Purpose

When ESX was first introduced (in 2001) and in version 2, Linux booted 1st, and when it finished, it loaded a script that loaded the VMkernel, and then turned the running of the server over to VMkernel with Linux running in parallel with it, though receiving most resources (including CPU and network, but not memory or disk) through the VMkernel. As of ESX 3, the VMkernel is loaded via the standard Linux **initrd** process and thus loads earlier in the boot process. In vSphere, the VMkernel loads and then loads the service console, which is actually a virtual disk (**esxconsole.vmdk**). The service console provides a simple way to load the VMkernel and also provides a place for management agents, backup agents, and system management tasks to run. On the other hand, many of the patches from VMware are not actually for the VMkernel, but rather for Linux.

The service console is based on Red Hat 7.2 for ESX 2.x, Enterprise Linux (RHEL) 3 for 3.x, and RHEL 5 for 4.0 and 4.1. Note that it is not the standard Linux kernel, but rather a customized version built for this specific purpose.

The service console, apart from running the agents and handling some of the management tasks previously described, is a convenient interface into ESX itself. This is why many hardware and software vendors, as well as many system administrators, really like the idea of a console they can log into when problems are encountered. This has been an issue with ESXi, which does not have the service console, so VMware has modified the

design of ESXi somewhat in 4.1 to allow both local and remote access to a command prompt (which is not the traditional service console, but many of the commands are the same or similar). They have also worked with the hardware and software vendors to make it easier to install (and update) various drivers and components needed, as well as building more into ESXi itself to reduce the need for those external components.

Differences in Features between ESX and ESXi

The most obvious and basic difference between the two is that ESX has a service console, whereas ESXi does not. IN ESXi 3.5 the first version of ESXi), there were many things that ESX could do that ESXi could not; ESX 4.0 has narrowed the gap, and 4.1 makes the differences almost nonexistent. For example, in 4.0, boot from SAN was supported for ESX, but not ESXi. On the other hand, ESXi can be preinstalled by hardware manufacturers (in a version known as ESXi Embedded) on an internal USB or flash device, meaning all that needs to be done is simple server setup and cabling, and the box can be managed in the existing environment within a few minutes. Note that ESXi Installable is also available, allowing administrators to download and install the OS on either local or SAN storage.

By removing the service console, ESXi is much smaller (it comes on a CD vs. a DVD for ESX in vSphere) and prevents a smaller attack surface in terms of outside hackers trying to gain entry. In addition, there is an entry-level version of ESXi, which VMware calls vSphere Hypervisor, that is completely free (though it has some significant restrictions and cannot be managed via vCenter without a license).

The question is often asked, "Should I choose ESX or ESXi?" As of 4.1, VMware recommends ESXi for all new deployments and to begin the process of migrating to ESXi now for existing deployments. Note, however, that ESX has a larger list of compatible systems than ESXi (though the gap is narrowing all of the time)

Architectural Differences between ESX and ESXi

ESX runs management agents (including third-party hardware management and backup agents), device drivers, the vCenter agent, the HA agent, etc., all in the service console. CLI commands can also be run here to setup and configure the server. It is important to know that this is a modified version of standard Linux, and, thus, many things can be run in this context that also are runnable in Linux.

ESXi does not have a service console, so the device drivers, HA and vCenter agents, syslog, SNMP, DCUI, etc., all run within the context of the VMkernel itself. Hardware management is done via CIM (Common Information Model) and thus doesn't require any agents. Some third-party agents for backup, hardware monitoring, etc., can also run in the context of the VMkernel, as long as VMware has authorized them; however, other random code, commands, drivers, etc., cannot be run within that context for the sake of system stability and security.

Note that in either case, the third-party agents, device drivers, etc., can be updated separately and independently from ESX itself.

Methods To Manage both ESX and ESXi

There are many methods that can be used to manage either ESX or ESXi. They will be discussed in this section (with the exception of the vSphere client, which is the recommended method of doing most ESX management and which is covered in the standard documentation and training materials). Which method you choose depends on your skill and ability with the method as well as how often the task will need to be done.

DCUI (Direct Console User Interface)

The primary method (and only supported method is ESXi 4.0) used to do basic configuration tasks (such as setting the host name, IP address, root password, or keyboard type) is the DCUI. It looks very much like the BIOS setup seen during a computer's boot process. It is very simple and basic by design with the idea that once basic configuration has been completed, all further configurations would be with one of the other supported methods.

Service Console

The service console allows simple to complex configuration changes via editing configuration files directly, running commands, etc. It can be accessed by pressing **<Alt> F1** from the main screen after the system has booted.

Troubleshooting Mode

Troubleshooting mode is unique to ESXi. It provides a command line interface into the VMkernel. In ESXi 3.5 and 4.0 it is unsupported and supposed to be used only at the request of and in conjunction with VMware technical support. It also requires direct access to the physical keyboard. Troubleshooting mode has been greatly enhanced in 4.1, in that it is fully supported and also that it can be accessed remotely via SSH as well as locally (though either or both of these methods can be disabled). It can be used with scripted installations as well as troubleshooting and running various scripts. However, it is still recommended that the vSphere Client or one of the scripting options (PowerCLI, the VMA, and/or vCLI) be used for most management tasks.

PowerCLI

PowerCLI is a command line interface based on Microsoft's PowerShell technology. The nice thing about PowerShell is that it is an object-oriented scripted language (as opposed to batch files) based on the .NET framework. PowerCLI makes it simple to run the same command on many objects (such as setting an option for all of the servers in your environment). PowerCLI offers several hundred cmdlets (commands for PowerCLI) that implement much of the functionality in a scripting interface that can be done via the vSphere Client. The list of cmdlets supported grows with each release of vSphere, and often grows with each update as well (such as vSphere 4.0 Update 1 or 2).

One of the great tools that VMware has (although it is in "Community Preview", aka beta) at this point is Project Onyx, which takes what you do in the vSphere Client and turns it into a set of PowerCLI commands. Microsoft has implemented similar technology with Exchange Manager for Exchange 2007 and 2010, for those familiar with that product. This is a simple way to create scripts for those who are not great script writers and/or those not familiar with PowerShell.

VMA (vSphere Management Assistant) / vCLI (vSphere Command Line Interface)

The vCLI provides a simple scripting interface for those familiar with scripting on the service console using ESX. The vCLI is based on the VMware SDK for Perl. The VMA is a preconfigured Linux server that includes the vCLI and also the SDK for Perl, and is available as a virtual appliance (VA) from VMware. The VMA includes many **vicfg-** commands that are similar to the service console version's **esxcfg-** commands.

One of the great features of the vCLI / VMA is that authentication can be against a vCenter server instead of individual ESX/ESXi servers (though it can do that as well), making it much simpler to manage security but also causing logging to be done via vCenter (for any command that makes a change to the environment, not for those commands that just display information), just as if the equivalent actions were performed in the vSphere Client connected to a vCenter server.

The VMA can also be used for centralized logging of all the ESX or ESXi servers in a single place, though this may require a little reconfiguration on the VMA to provide enough space for all of the log files. The VMA also includes two useful commands, **vi-logger** for configuring logging and **vi-fastpass** that make entering security credentials needed for commands much simpler. The VMA can also be joined to an Active Directory domain, making security an integrated, pass-through design, so names and passwords are not required for each command.

For those who are intrigued by the power and capabilities of PowerCLI but are not Windows users, VMware offers a PERL script, mcli.pl, that accepts a text file containing a list of servers to run the command against as well as a standard vCLI command to run. This is available in version 4.1 of the VMA.

Modifying Existing Scripts for Use with ESXi

Script migration may be necessary for any scripts that have been written using service console scripts (PowerCLI scripts should work fine for the most part, unless they access functionality, such as service console ports, that do not exist on ESXi).

A few things that will always require modification include:

- Any command that directly accesses a datastore (via the /vmfs/volumes directory structure)
- Any command that directly creates and/or modifies any ESX configuration file
- All references to **esxcfg-** must be replaced with **vicfg-**
- Any reference to a LINUX command won't work
- **vicfg-** commands require a target server to run against (it is always on the local server with esxcfg- commands)
- Some **vicfg-** commands have new or changed switches vs. the equivalent **esxcfg-** commands

In all cases, script functionality should be tested and debugged in a non-production environment.

Summary

To summarize, there are many things in common between ESX and ESXi – in fact, virtually everything that can be done in ESX can also be done in ESXi as of version 4.1. The primary differences relate to management of the two platforms, especially in relation to service console scripts, as ESXi doesn't have a service console and the architecture differences that require management agents to be redesigned to work properly without a service console.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge. Check out the following Global Knowledge courses:

VMware vSphere: Install, Configure, Manage [V4.1]

VMware vSphere: Troubleshooting [V4x]

VMware vSphere: Transition to ESXi [V4x]

VMware vSphere: Automation with vSphere PowerCLI [V4x]

For more information or to register, visit **www.globalknowledge.com** or call **1-866-925-7765** to speak with a sales representative. Our courses offer practical skills, exercises, and tips that you can immediately put to use. Our expert instructors draw upon their experiences to help you understand key concepts and how to apply them to your specific work situation. Choose from our more than 1,200 courses, delivered through Classrooms, e-Learning, and On-site sessions, to meet your IT, project management, and professional skills training needs.

About the Author

John Hales, VCP, VCI, is a VMware instructor at Global Knowledge, teaching all of the vSphere classes that Global Knowledge offers. John is also the author of many books, from involved technical books from Sybex to exam preparation books, to many quick reference guides from BarCharts, in addition to custom courseware for individual customers. John has various certifications, including the VMware VCP and VCI, the Microsoft MCSE, MCDBA, and MCT, the EMC EMCSA (Storage Administrator for EMC Clariion SANs), and the CompTIA A+ and Network+.

John lives with his wife and children in Sunrise, Florida.

References

VMware vSphere 4.1 Release Notes – ESX Edition: http://www.vmware.com/support/vsphere4/doc/vsp_esx41_vc41_rel_notes.html

ESX vs. ESXi: http://www.vmware.com/products/vsphere/esxi-and-esx/faqs.html

ESX Architecture: http://it20.info/2007/06/a-brief-architecture-overview-of-vmware-esx-xen-and-ms-viridian/

ESXi Architecture: http://www.vmware.com/files/pdf/vmware_esxi_architecture_wp.pdf

ESXi 3.5 and 4.0 Troubleshooting Mode: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1003677

ESXi 4.1 Troubleshooting Mode: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1017910

Project Onyx: http://communities.vmware.com/community/vmtn/vsphere/automationtools/onyx

Script to generate a list of PowerCLI cmdlets by noun: http://wannemacher.us/?p=315

PowerCLI Administration Guide: http://www.vmware.com/support/developer/windowstoolkit/wintk40/doc/viwin_admg.pdf

vCLI Documentation: http://www.vmware.com/support/developer/vcli/

Getting started with the VMA (tips & tricks): http://communities.vmware.com/docs/DOC-10878

esxcfg- vs. vicfg- commands: http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&docType=kc&externalId=1008194

VMA 4.1 Guide: http://www.vmware.com/support/developer/vima/vma41/doc/vma_41_guide.pdf