SAP Solution Guide

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.

EN--

vmware®

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

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

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

Contents

1 Executive Overview

SAP creates enterprise software to manage business operations and customer relations. The company's best known software products are its Business Suite of solutions (such as SRM, CRM, SCM, PLM and ERP), its Enterprise Data Warehouse solution (SAP Business Warehouse – SAP BW), SAP Business Objects, and the Sybase and HANA data platform. For further information on SAP products, see - product categories .

The two key components of any SAP solution are the SAP NetWeaver and SAP Web Application Server.

- SAP NetWeaver is the technical foundation for many SAP applications. It is a solution stack of SAP's technology products.
- The SAP Web Application Server is the runtime environment for SAP applications on which all of the Business Suite solutions run.

Production support for SAP NetWeaver and the Business Suite of SAP solutions on VMware vSphere has existed since 2008, and this support now includes SAP Business Objects and the Sybase and HANA database platform. You can view the SAP Corporate fact sheet at http://go.sap.com/documents/2016/07/0a4e1b8c-7e7c-0010-82c7-eda71af511fa.html

VMware is the leader in virtualization and cloud infrastructure solutions. As a SAP customer you may be wondering how to realize the benefits of virtualization technology for your SAP environments. SAP and VMware have teamed up to offer a joint solution that helps speed your path to virtualization and reducing operating costs while minimizing business risks. For details on VMware offerings refer to http://www.vmware.com/products.html

This chapter includes the following topics:

Why This Guide ?

If yours is like most IT organizations today, you're under constant pressure to do more with less – even as demand for your services continues to increase. You're expected to drive down costs and operate at peak efficiency. You're also expected to do this while enabling business innovation and supporting a global user base in a secure manner where people want to access enterprise resources on whatever device they choose.

Typically, adding physical server space to meet this demand is no longer an option. This only increases hardware, operations, and maintenance costs – and the resulting complexity often impedes productivity and adversely impacts availability. This is why leading IT organizations and the businesses they serve are moving to virtualization technology to establish secure private cloud infrastructures.

Virtualized private cloud infrastructure allows you to abstract physical server resources that you can pool and from which you can automatically provision services on demand. This enables you to maximize data center capacity and ensure the highest levels of availability. The result is greater business agility, improved scalability, and faster time-to-market advantages without the risk of turning over critical corporate data and security responsibilities to a third party.

By using the right virtualization and management tools from SAP® and VMware, you can dramatically increase transparency in your *SAP landscape* and simplify SAP administration with a *single pane of glass*. This book provides you with the tools you need to understand the VMware virtualization infrastructure and management tools and how to use them in the context of your SAP landscape.

Target Audience

This document is primarily written for people familiar with SAP products but unfamiliar with VMware's infrastructure and management products, Software Defined Data Center (SDDC), and how to leverage said products to enable automated provisioning of and management of SAP systems.

Understanding the Benefits of Virtualizing SAP Deployments with VMware

Software defined data center (SDDC) refers to a data center where all infrastructure is virtualized and delivered as a service. Control of the data center is fully automated by software, meaning hardware configuration is maintained through intelligent software systems. This is in contrast to traditional data centers where the infrastructure is typically defined by hardware and devices. SDDC are considered as the next step in the evolution of virtualization and cloud computing. It provides a solution to support numerous enterprise applications like SAP and also other cloud computing services.

Customers can transform and virtualize their entire *SAP landscapes*, from transactional to analytic workloads, and take the next step toward the *Software Defined Data Center (SDDC)*. Running SAP on VMware platform provides the simplicity, efficiency and agility customers demand for their most mission critical enterprise environments.

You get the following benefits by virtualizing SAP environments on VMware:

- High Availability – VMware vSphere High Availability capabilities with automated fault tolerance enable SAP managers to deliver exceptional uptime and business continuity for their mission critical environments
- Automated Provisioning - You can do on-demand deployment of applications in minutes with automated provisioning and template cloning. This ensures consistency and scalability across SAP environments
- Live Migration of workloads - You can migrate live running workloads from one certified host to another with zero downtime and zero data loss with VMware vSphere vMotion
- Reduce Capital and Operational Expenses - Your *CapEx* and *OpEx* expenditures reduce along with better utilization of existing infrastructure and resources along with automated provisioning and management. This enables to lower the *Total Cost of Ownership* (TCO)
- Support Services - SAP and VMware have established a world class support infrastructure along with dedicated teams focused on customers. You are thus assured of dedicated and excellent customer support that is even more significant in the context of *Business Critical Application* (BCA)

- Professional and Consulting Services - In addition to the above, VMware Professional Services and SAP Consulting together provide a full range of services from assessment to implementation and optimization to help customers transform their SAP environments.

> **Note** For further information on supported SAP applications, supported guest operating systems by SAP, supported databases, FAQs and support information, refer to https://wiki.scn.sap.com/wiki/display/VIRTUALIZATION/SAP+on+VMware+vSphere

To summarize, running software applications in a virtualized VMware environment is now a mainstream best practice that allows organizations to save time, money, energy, and space. Because you can use physical servers more efficiently, you can quickly scale your server infrastructure to meet business demands without the added investment of purchasing additional hardware. Fewer machines results in lower up-front investment costs and energy usage. Virtualized environments are also easier to manage as SAP software upgrades and other changes are managed centrally, with less risk. Virtualizing business-critical SAP software in a VMware environment also improves the efficiency, flexibility, and availability of applications, enabling you to transform service delivery while significantly reducing operational costs.

What do Customers Say About the VMware SAP Solution?

Following are some testimonials from a partner and a customer highlighting some benefits of VMware SAP Solution:

- IBM - Virtualized SAP HANA Deployment - http://bcove.me/xcw1wrxw
- EMC - Data Center Transformation with SAP HANA and VMware vSphere — http://bcove.me/fwkwi7s6

Overview of the VMware SAP Solution Guide

Here is a brief overview of the contents of this document:

1 Executive Overview (this chapter) — Discuss why this guide, who are target audience are, benefits of virtualizing the *SAP landscape* with VMware vCloud Suite, testimonials from partner/customer on how they perceive of the benefits of virtualizing the SAP Landscape.

2 Virtualization Overview – Introduction to key virtualization concepts, overview of the different components that can be virtualized like memory, CPU. Finally you look at other aspects like VMware vSphere High Availability, VMware vSphere Fault Tolerance and VMware vSphere Site Recovery Manager (SRM).

> **Note** If you are already familiar with virtualization concepts, feel free to skip over this chapter.

3 Overview of VMware vCould Suite - Define a vCloud Suite, its components in the context of virtualizing the SAP Landscape, review of an example architecture of a virtualized SAP Landscape with vCloud Suite and finally a discussion of the SAP Architecture on a virtualized VMware infrastructure.

4 Virtualize SAP HANA - Introduction to SAP HANA, its deployment types and benefits, customer testimonials on their perception of the benefits of deploying SAP HANA on a virtualized VMware infrastructure, brief overview of SAP HANA architecture and its features when deployed on VMware vSphere. Finally there is a discussion of SAP HANA deployment on VMware vSphere 6.0 for production environments.

5 VMware Adapter for SAP Landscape Management - What is the VMware Adapter for SAP Landscape Management and its role in managing the SAP landscape. Reference architecture gives an overview of the components of the VLA execution environment and their relationship to one another. The Install and Configure section references the *VMware Adapter for SAP Landscape Management Installation, Configuration, and Administration Guide for VI Administrators* document. You can go over this document to understand the installation, configuration and administration aspects of managing the VMware Adapter for SAP Landscape Management

6 SAP Solution with VMware Products - Discussion of SAP solution with VMware vSphere High Availability, VMware vSphere Fault Tolerance, VMware vSphere Site Recovery Manager (SRM) and on virtual SAN.

7 SAP on VMware Best Practices - Useful guidelines, general recommendations and deployment best practices for SAP HANA on VMware vSphere for the computing environment, networking and storage.

8 Appendix - Provides a handy quick reference list of useful terms relevant to VMware virtualization technology.

Virtualization Overview 2

This chapter discusses several key virtualization concepts and terminology. Virtualization can help you shift your IT focus from just managing boxes to improving the services that you provide to the organization. Using this technology you can not only partition on machine into several virtual resources but also aggregate multiple physical resources into a single virtual resource that can be provisioned and managed efficiently. Virtualization can really help you save money, energy, time and greatly simplify desktop management.

For those of you that are not familiar with virtualization technology you begin by comparing a non-virtualized infrastructure with a virtualized infrastructure. You then focus on specific components that can be virtualized like memory, CPU, storage and network. Finally you cover topics like VMware vSphere High Availability, VMware vSphere Fault Tolerance and disaster recovery with *Site Recovery Manager (SRM) that are required and even more significant in the context of SAP landscape.*

Note If you are already familiar to VMware Virtualization concepts, feel free to skip this chapter.

This chapter includes the following topics:

Virtualization - Before and After

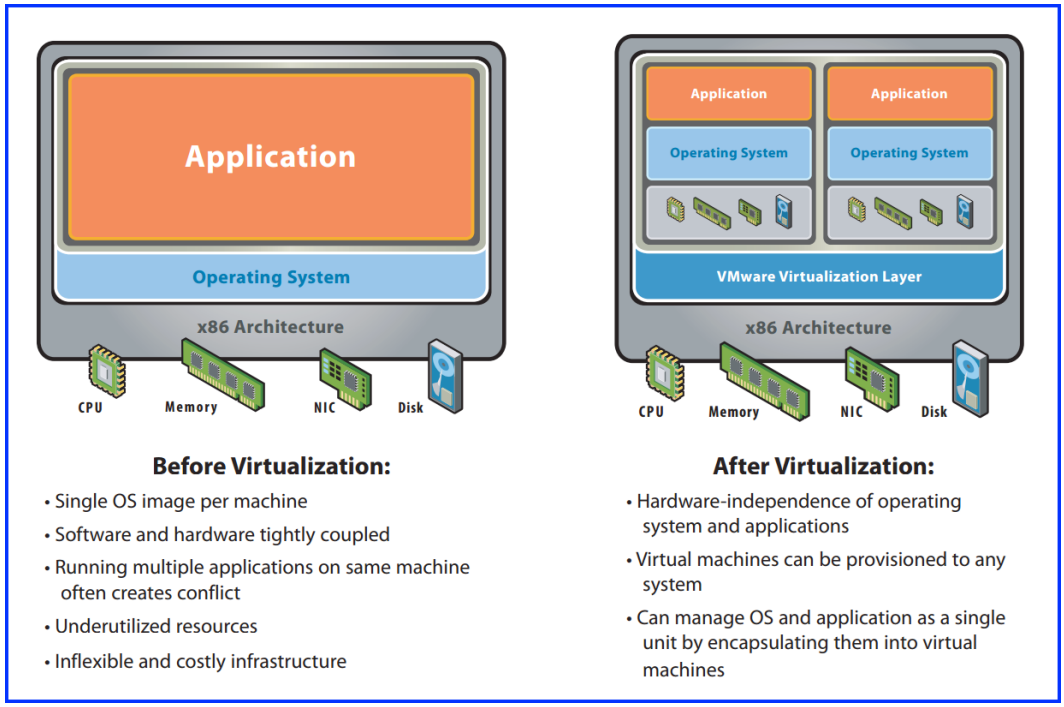
Broadly, virtualization describes the separation of a resource, or request for a service from the underlying physical delivery of that service. For example, virtual memory. Similarly, virtualization techniques can be applied to other IT infrastructure layers – including CPU, storage, Networking and Applications. You will discuss more about this in the subsequent sections of this chapter.

How do you bring enterprise-level technology to organizations that cannot afford the large capital required to pay for the hardware, software licenses, setup and continual maintenance of an actual data center infrastructure. Virtual infrastructure enables you to utilize physical server resources to host logical or virtual servers and networking hardware in order to optimize resources and drive costs down by hosting multiple virtual servers in a single host server. It is also possible to aggregate multiple physical resources into a single virtual resource that can be provisioned and managed efficiently. Benefits of virtual infrastructure include scalability, flexibility, security, load balancing and easy backup and recovery. It is possible to do the following:

- Partitioning a physical server into smaller virtual servers. Each virtual machine can interact independently with other devices, applications, data and users as though it were a separate physical resource.

- Amalgamation of multiple network storage devices into a single logical storage unit
- Using software to allow system hardware to run multiple instances of different operating systems concurrently, allowing you to run different applications requiring different operating systems on one computer system
- Using network resources through a logical segmentation of a single physical network
- Allowing multiple applications to bring down their own set of configurations on-demand and execute in a way that they see their own respective settings.
- Separates applications from teh hardware and the operating system, putting them in a container that can be relocated without disrupting other systems
- Enables a centralized server to deliver and manage individualized desktops remotely. This gives users a full client experience but lets IT staff provision, manage, upgrade and patch them virtually instead of physically.

Figure 2-1. Virtualization - Before and After



Following are some of the benefits of virtualization technology:

- Isolation of virtual machines and the hardware-independence that results from the virtualization process
- Virtual machines are highly portable, and can be moved or copied to any industry-standard (x86-based) hardware platform, regardless of the make or model
- Several system resources can be virtualized and managed, including CPUs, main memory, network, storage and I/O. It facilitates inter-partition resource management capability
- Companies often run just one application per server because they don't want to risk the possibility that one application will crash and bring down another on the same machine. Estimates indicate that most x86 servers are running at an average of only 10 to 15 percent of total capacity. With virtualization, you can turn a single purpose server into a multi-tasking one, and turn multiple servers into a computing pool that can adapt more flexibly to changing workloads.
- Businesses spend a lot of money powering unused server capacity. Virtualization reduces the number of physical servers, reducing the energy required to power and cool them.

- With fewer servers, you can spend less time on the manual tasks required for server maintenance. On the flip side, pooling many storage devices into a single virtual storage device, you can perform tasks such as backup, archiving and recovery more easily and more quickly. It's also much faster to deploy a virtual machine than it is to deploy a new physical server.
- Managing, securing and upgrading desktops and notebooks can be a hassle. You can manage user desktops centrally, making it easier to keep desktops updated and secure.
- Enables IT managers to be more responsive in supporting new business initiatives with increased flexibility in adapting to organizational changes. All this amidst a climate of IT budget constraints and more stringent regulatory requirements

Virtualization Concepts

Understanding key virtualization concepts and terminology is essential for you to be able to appreciate the real benefits of virtualizing an entire SAP Landscape using VMware vCloud Suite that is discussed in the subsequent chapters of this document. This section introduces you to the following topics —

- Common terms and concepts associated with virtualization.
- Types of virtualization and their benefits
- Differentiate between full virtualization, para-virtualization and partial virtualization
- Discuss advances in x86 hardware that have enabled increased server virtualization adoption
- Define IaaS, PaaS, SaaS, private cloud, public cloud and hybrid cloud and understand the uses of each
- Define terms like *Software Defined Data Center*, *Software Defined Networking* and *Software Defined Storage* and understand the advantages of each.

Understanding General Virtualization

Wikipedia defines virtualization as follows (see `http://en.wikipedia.org/wiki/Virtualization`) :

Virtualization (from wikipedia) — In computing, *virtualization* is simulating a hardware platform, operating system (OS), storage device, or networking resources.

For purposes of this course, this definition is not general enough. You need a more general definition, a good explanation of the definition, and a high-level understanding of how virtualization is implemented. Here is the definition we will use in this class (and throughout all classes in this program):

Virtualization (course definition) — In computing, *virtualization* is the simulation of hardware functionality in software, with or without assistance from hardware. The simulation decouples the virtualized object from the physical object.

The key differences between this and Wikipedia's definition are:

- This definition indicates the simulation is done primarily via software, with possibly assistance from hardware
- This definition does not limit itself to categories, such as hardware platform, storage, or networking
- This definition includes the concept of decoupling the virtual object from a physical object

These differences are important as virtualization continues to evolve. It has already evolved beyond the hardware / storage / networking categories listed in the Wikipedia definition.

Without getting ahead of ourselves, here are some examples of virtualization with which you may already be familiar:

- The `xterm`, `Terminal`, and `cmd` programs on Linux, Mac OS X, and Microsoft Windows platforms, respectively, are software that emulate a physical terminal (for example the old DEC VT100, WYSE60, etc.). That is, terminal emulation programs provide *virtual terminals*.
- Operating systems used to be distributed on DVDs, CD-ROMs, and even floppy disks, before that. Today, most operating systems are packaged into ISO files, which people download from the manufacturer (for example: Ubuntu Linux, SUSE Linux, or VMware). These ISO files are images of a DVD or CD-ROM. That is, they are a software emulation of the physical disk. Some servers no longer come with physical DVD or CD-ROM drives. Such systems typically provide a feature that allows you provide network access to the ISO image, which the server then treats as a *virtual CD-ROM.*
- Some mobile phones allow you to place two SIM cards in them. The phone creates a presence on the cell network associated with each SIM. This allows a single physical phone to act as two *virtual phones*.
- For decades, CPUs have supported *virtual memory*, especially *demand-paged virtual memory* (in 32 and 64-bit CPUs), which is used by modern operating systems. *Demand-paged virtual memory* schemes allow OSes to page (or swap) data out of RAM onto a *backing store* (typically a hard disk) once RAM is full. The memory written out to disk is typically the least recently used (and therefore least likely to be needed again). The OS will re-load the data from the backing store into RAM when it is needed next. Systems that have insufficient RAM for their loads end up doing a lot of paging. Since backing store is slower than RAM, this often has a significant (negative) impact on performance of the system.

These are all examples of *virtualization*, and are even computer / technology related. However, when people talk about *virtualization* in the realm of computers today, they are typically talking about virtualizing complete servers (called *server virtualization*), disks (called *storage virtualization*), networking (called *networking virtualization*), desktop operating environments (a special case of server virtualization called *desktop virtualization*) , or specific applications (called *application virtualization*).

The rest of this chapter discusses these types of virtualization, factors driving its adoption and how the technology is being used to implement *the cloud*, *cloud-based services*, *software defined data center*, *software defined storage* and *software defined networking*.

Understanding Compute and Server Virtualization: *Host Systems* , *Hypervisors* and *Virtual Machines* / *Guests*

Defining Compute Virtualization

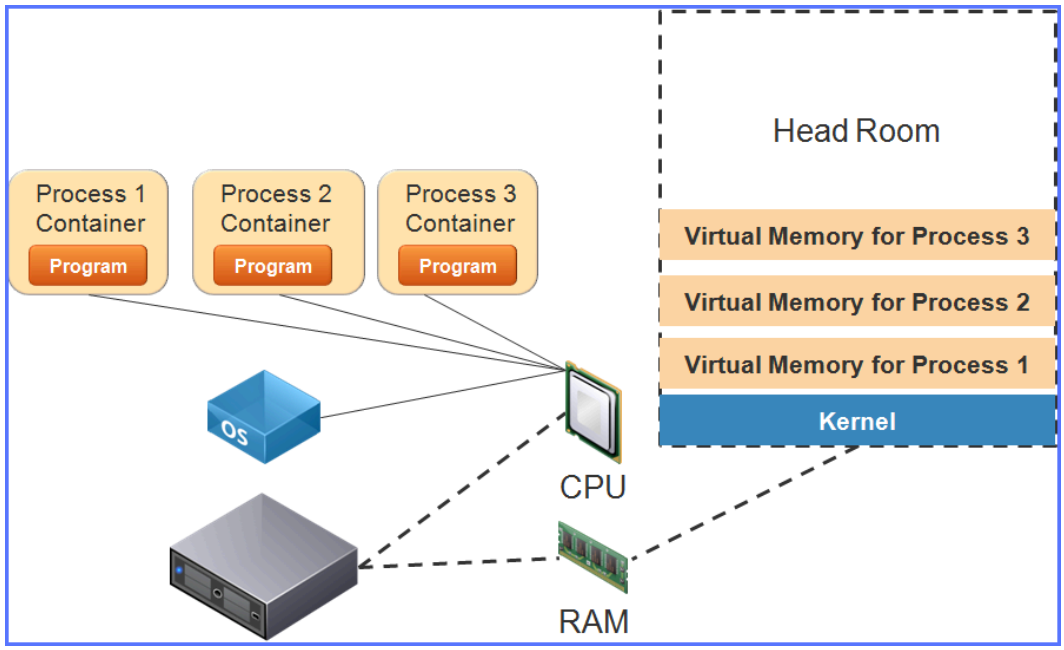
Before defining *server virtualization*, it is useful to review the function of multi-tasking operating systems (OSes), for example Unix / Linux / Windows. Even the most basic program today requires at least a CPU to execute its instructions and RAM to store its data. Multi-tasking OSes allow multiple programs to appear to run at the same time by virtualizing a computer's CPU and RAM, keeping track of the state of each running program in a set of data structures generally called a *process*. This leads to the following definition:

Process — A set of kernel data structures used to keep track of the state of an instance of a running program. State information includes (but is not limited to): The CPU(s) program counter (PC); The CPU stack pointer (SP) and other registers; User credentials; Virtual memory information.

Thus, when an OS starts an instance of a program running, it creates a *process* to keep track of it. When it's time to run a process, the *kernel* loads the process's page tables into the CPU's memory management unit (MMU), sets the CPU's other registers to a well-defined initial state, and then jumps to code in the process's program. This is called a *context switch*.

The process continues to run until one of two things happens: A) The program running in the process voluntarily gives up the CPU (for example on `exit()` or waiting for I/O); B) The OS preempts the process, allowing some other process a chance to run (for example because a higher-priority process has become runnable or because the process has used its time-slice). At that point, the OS saves the state of the CPU, including the MMU, and restores the state of or starts a new process. Given this description, one could assert that a multi-tasking OS *virtualizes* the CPU and RAM for processes. That is, the OS's process management, including context switching, creates virtual CPUs with Virtual Memory in which processes run. This is called *compute virtualization*. The following figure provides a simplified representation of compute virtualization:

Figure 2-2. Compute Virtualization



Defining Server Virtualization

Server virtualization is an extension of compute virtualization, but for multitasking instances of operating systems. That is, it virtualizes an entire server (or any class of computer), including emulation of I/O devices. To do this, server virtualization implementations must virtualize the entire computing platform – CPU, RAM, I/O bus, I/O devices such as network interface cards (NICs), Host Bus Adapters (HBAs), disks, video cards, etc. – so that each OS believes it is running on its own computer, (just as a multi-tasking OS makes each program believe it is running on its own CPU and in its own RAM). Where a traditional OS uses a *process* structure to virtualize RAM and CPU, a server virtualization implementation uses a *virtual machine* (abbreviated *VM*) structure to virtualize an entire computing platform. A server virtualization implementation is called a *hypervisor*. This yields the following preliminary definition:

Hypervisor (preliminary definition) An operating system whose kernel multi-tasks VMs

The compute platform on which a hypervisor runs can vary, including servers, desktops, laptops, and others. Thus, said platform is generically referred to as a *host system* or simply *host*.

Note There is another virtualization technology called *partitioning*. Partitioning implementations are typically *application virtualization*, or are sometimes called *OS virtualization*. In no case are they *server virtualization*. This has a separate section on *application virtualization*, but does not discuss *OS virtualization* beyond this note.

In many ways, hypervisors are just like other operating systems. For example:

- They have a *kernel* with memory management, a scheduler, role-based access control (RBAC) enforcement, etc.
- Their kernel coordinates access to the host system's physical resources that are shared by multiple VMs, including RAM, I/O devices, etc.
- They (typically) ship with utilities that manage the environment, such as viewing the contents of the filesystem, showing running operating systems, etc.

Despite the similarities, because hypervisors multi-task whole operating systems, they have significant differences. Consider the difference between a process and a virtual machine object (which is much more complicated):

- Both structures need to keep state. But a VM has to keep state of an entire virtualized host system, not just registers for CPU cores and page tables for RAM.
- A hypervisor must make the OS running in its VMs believe that it has access to a physical HBA with disks and/or CD/DVD drives, typically a video screen with monitor, and typically some number of NICS, the number and type of which may vary from VM to VM. Thus, a VM object must include data that tracks of the virtual hardware that's assigned to each VM.
- Hypervisors typically provide a shim layer between the virtual devices presented to a VM and the devices physically present in the host system. For example, a hypervisor may tell a VM it has one LSI SCSI HBA, when the host system actually has several SCSI HBAs from a different vendor. When the VM does an IO to a (virtual) disk attached to its (virtual) HBA, the hypervisor has to translate the actions of the device driver in the VM to actions against the physical HBA in the host system.

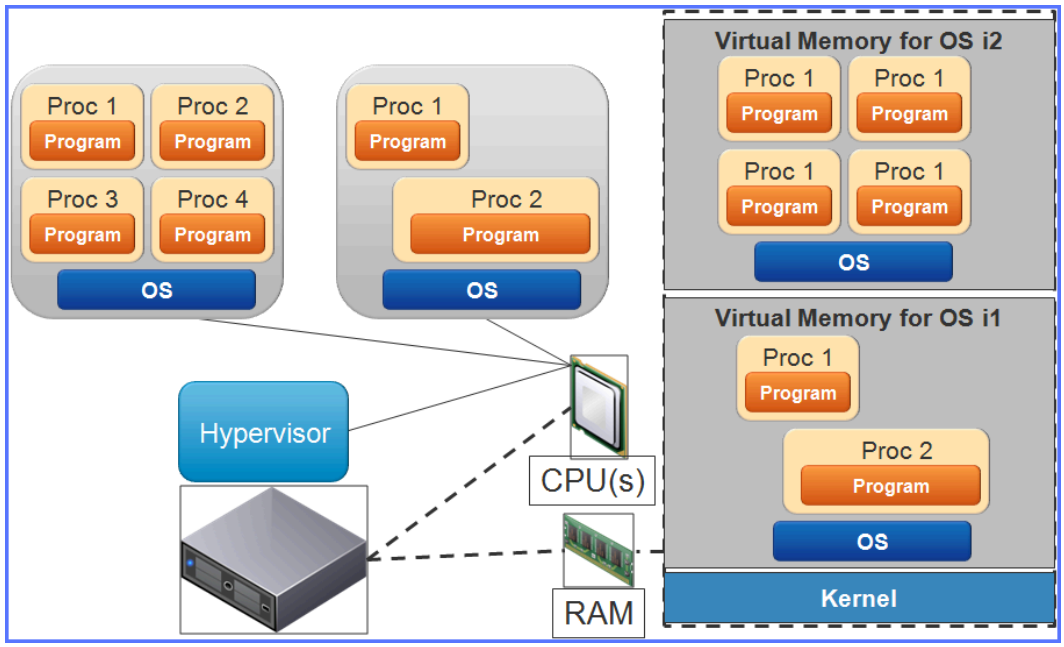
Note How hypervisors map virtual and physical hardware is very implementation-dependent. The better a hypervisor does this, the faster a VM can run.

This discussion yields the following definitions:

Server virtualization	The software emulation of an entire computer, possibly with hardware assist, using a hypervisor to manage the host system hardware and virtual machines to represent each of the virtualized computers, to enable multiple instances of OSes to run on the same host system at the same time. Server virtualization separates the server environment from the physical server hardware, via a *virtualization layer* (the hypervisor and VMs).
Hypervisor (complete definition)	Software that context-switches between virtual machines, allowing multiple VMs to run on the same host system at the same time.
Virtual Machine	Is an object with structure and methods used to virtualize an instance of a physical computer, including a computer's CPU(s), RAM and memory management system (MMU), I/O devices - including (NICs), Host Bus Adapters (HBAs), Video adapters, etc. The object must include data to maintain the state of said OS (and by inference the OSes properties) so that the hypervisor can context-switch between virtual machines on the host.
Host System	The computing platform on which a hypervisor runs.
Guest	Another name for VM. This term is an offshoot of the term *host system*, and the idea that the VMs are guests of the host.

The following figure provides a simplified representation of server virtualization as discussed thus far:

Figure 2-3. Server Virtualization



Notice that, compared to compute virtualization, server virtualization is more complex. Essentially, there is a new layer added between the hardware and the OSes (in the VMs) – the hypervisor kernel. This extra layer requires extra RAM usage to keep track of virtualization house-keeping, for example the VM structures themselves, and extra CPU cycles for context-switching VMs.

The total cost of server virtualization depends on several factors, including the type of virtualization used (full, para, or virtual), and the type of hypervisor used: Type I or Type II. These factors, and others, are discussed throughout this chapter.

Also notice that the server's RAM utilization is shown as higher than in the preceding picture. While this is not necessarily true, it is typically true. One of the advantages of server virtualization is decreased headroom in (waste of) RAM.

Important Check the licensing of operating systems before placing them in a VM. Some vendors that make hardware and OSes only allow you to run their OS on their Hardware. These restrictions may dictate which hardware you purchase for your host system. For example, Apple requires that you run Mac OS X, even virtualized, on Apple hardware (e.g. Macbook Pro, Mac Mini, etc.). Microsoft, (whose business model depends on OEM relations), has no such requirement. Therefore, if you want a system that can run Mac OS X, as well as other OSes, you must (currently) have an Apple host system. Put another way, to run both Windows and Mac applications, you have to purchase a Mac.

Understanding Virtual Devices in VMs (vHBAs, vNICs, etc.)

The preceding figure presents a simplified representation of server virtualization. It focuses on the hypervisor multi-tasking and managing RAM for VMs. However a hypervisor must virtualize more than CPUs and RAM for guest OSes to do useful work in a VM. It must virtualize devices, such as disks, host bus adapters (HBAs), CD-ROMs, NICs, etc. To differentiate between a virtual and physical device, this course (and the industry) precedes the device type with a "p" or "v", respectively, for example "pNIC" for physical NIC and vNIC for virtual NIC.

Hypervisors must provide a method for mapping virtualized devices to physical devices. For example, consider a VM that is assigned the following:

- One SCSI HBA
- One SCSI disk attached to the one SCSI HBA

- One IDE HBA
- One Floppy and one CD-ROM drive, both attached to the IDE HBA
- One NIC

Here are some points about the virtualization and mapping of the hardware for this VM:

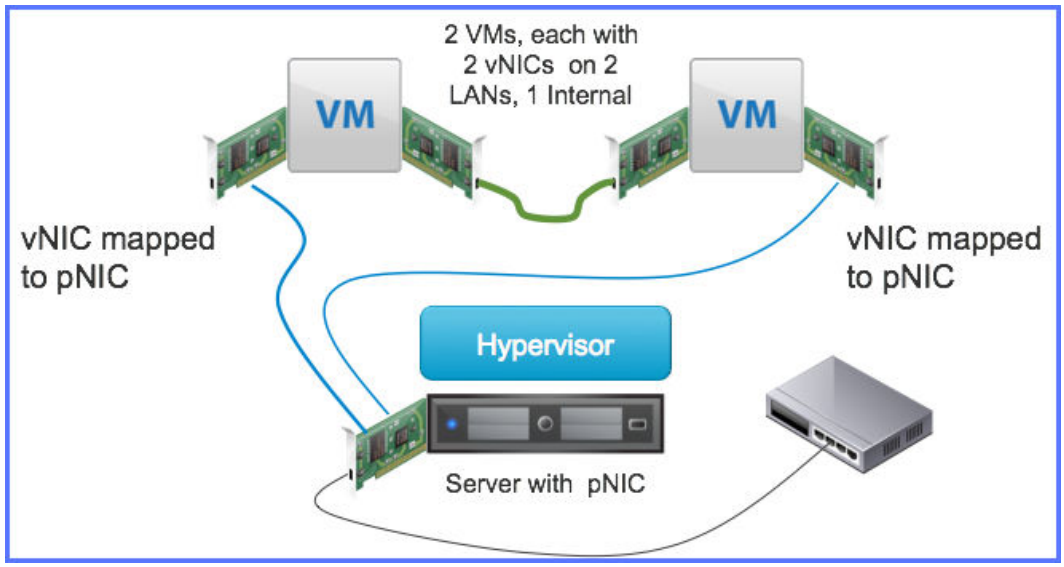
- The virtual hardware does not have to be the same type/make/model as that present in the host system. For example, you may configure the BusLogic vHBA when the physical HBA is from LSI.
- The number of virtual devices assigned to a VM need not match the number present in the host system. For example, consider a host system with one physical NIC (pNIC), with two VMs each having two virtual NICs (vNICs). In each VM, one vNIC may get mapped to the pNIC, while the other is mapped to a virtual network shared only by the VMs (that is, whose packets never touch a physical NIC). In contrast, consider that a host system may have 8 pNICs, with VMs having from 1 to 3 vNICs each.

Note Hypervisors rarely map vNICs directly to pNICs. Instead, hypervisors typically provide a *virtual switch* (*vSwitch*) object, to which vNICs and pNICs are attached. The vSwitch concept is discussed in detail in several places later in this course.

- The virtual disks (*vDisks*) are implemented as files in a filesystem. The types of filesystems supported depend on the hypervisor but typically include filesystems attached to local disks within a host system, NFS, and iSCSI LUNs treated as a flat file. When a guest OS writes to a vDisk, the hypervisor translates that to a write to the file on the local store, NFS filesystem, or iSCSI LUN that houses the filesystem containing the vDisk.
- Some devices need not be backed by physical hardware. For example, a virtual CD-ROM device can be backed by an ISO file. In these cases, the virtual devices are often faster than the physical devices. In the CD-ROM example, consider that access to the vCD-ROM happens at the speed of the disk on which the ISO sits, compared to physical CD-ROMs that rotate and transfer data at much slower speeds.

The following figure illustrates the example of VMs with more vNICs than pNICs that exist in the host system:

Figure 2-4. Virtual vs Physical Devices



There are additional important concepts to understand regarding how hypervisors emulate hardware for their VMs:

- As shown in the preceding figure, it is possible for multiple VMs to need access to the same physical device, for example two vHBAs accessing the same pHBA. Hypervisors must manage access to the physical devices. How it does this is implementation dependent. In general, it involves having the hypervisor intercept all I/O requests to the device, managing a global queue of the requests, and giving results for a given operation back to the guest OS that sourced a given request.

  Note Systems that use an *IOMMU* make sharing physical devices much more efficient. For details of IOMMUs for x86 systems, see "Understanding AMD-V and Intel VT-x," on page 25

- Some hypervisors have features that allow a VM to claim exclusive access of a physical device, bypassing the hypervisor's intercept of the I/O requests, making the I/Os faster. Devices accessed though such a feature are exclusively owned by the VM. That is, no two VMs can get direct access to the same physical device since the hypervisor is no longer involved in arbitrating the requests. The VMware DirectPath I/O is an example of this capability.

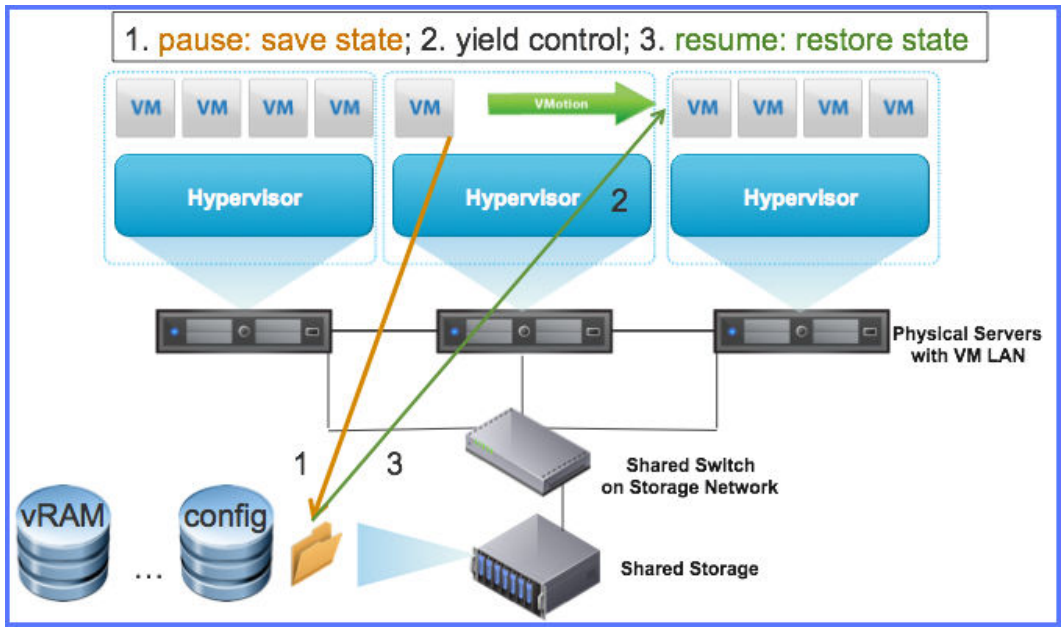
Understanding VMs as Files in Folders: Pausing, Resuming, VM Migration, and Clusters

The configuration of the VM, including an image of its virtual RAM and disks is stored in disk files, typically in a single folder named for the VM. This enables many useful features for a VM:

- A VM can be *paused* by saving its running state, including its vRAM, to files in the VM's folder. The VM can then be *resumed* by restoring state from the saved files.
- A VM can be migrated from one hypervisor to another simply by copying its folder from one hypervisor to another. VMs should be powered off or paused before they are moved.
- If the folder is located on a shared storage device, then the folder need not even be copied. Rather, the hypervisor running the VM must pause it, then relinquish control of the VM to the other hypervisor, which then resumes it. This is considerably faster than copying the folder between host systems. In VMware's vSphere hypervisor product, this capability is called *VMotion*®.
- Cluster can be created (relatively) easily by creating a pool of hypervisors that share storage and that have some device to implement high availability (HA) and/or load balancing between the hypervisors.

The following figure illustrates these concepts:

Figure 2-5. Clustering, Pausing, Resuming, Migrating VMs, and VMs as Folders with Files



This figure shows three host systems running VMs. The middle host has a VM that is being moved to the host on the right. The VM's folder does not need to be copied from the middle to the right host since it is located on a storage device shared between all host systems.

Trivia The first use of virtualization in computers is widely attributed to IBM, when they introduced the CP/CMS OS to run on their System/360 series in 1968. That operating system didn't just virtualize CPUs, but the entire "machine" (also called a "mainframe" in those days, or what would typically be called a "server" today).

Understanding Snapshots

Most modern hypervisors allow administrators to create a picture of the VM at a given point in time, to which administrators can then revert the VM at some point in the future. This picture is called a *snapshot*. Snapshots may include just the configuration of the VM and the current data in its disks, or they can also include a current image of the vRAM and vCPU states. Most hypervisors can snapshots without vRAM / vCPU quickly. Snapshots with vRAM / vCPU take longer, increasing in time with the amount of vRAM and the number of vCPUs.

Administrators can use shapshots for several purposes:

- Saving a VMs state before upgrading or installing new software
- Creating a stable image of the VM which backup software can use while the VM continues to run

Snapshots usually cost performance for reads because of the way they are implemented. Thus, while administrators can theoretically create an endless chain of snapshots, in practice they typically only have one or two snapshots on a VM at a given point in time.

Defining Types of Hypervisors: I, II, and Nested

Figure 2-3 (in the preceding section) showed a hypervisor running immediately on top of a physical computer, or *bare metal*. While this is one way that hypervisors can run, there are (currently) two other ways that hypervisors can run as well.

Type I or Bare Metal Hypervisors

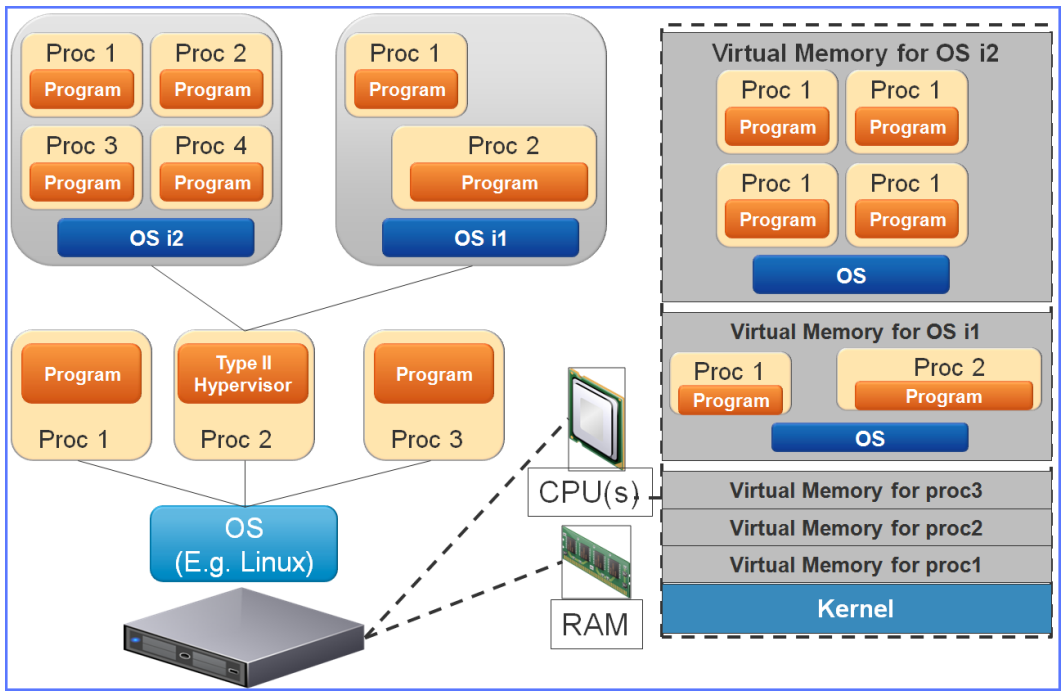
Hypervisors that can run on bare metal are called *Type I* and sometimes *bare metal* hypervisors. These hypervisors must have a kernel that understands how to manage all of the host systems hardware directly, including scheduling time on CPUs, managing page tables, having device drivers for all I/O devices such as NICs, HBAs, disks / DVD-ROM and CD-ROM attached to HBAs, and video cards. They typically support virtual machine I/O by intercepting I/O requests to their virtual hardware and queuing the request to the physical hardware. Where the type of the make/model of the virtual device differs from the physical device, the hypervisor must perform appropriate translations as well.

Compared to *type II* and *nested* hypervisors, (discussed in the next two sub-sections), all other things being equal, bare metal hypervisors are more efficient because they interact directly with the hardware.

Type II Hypervisors

Hypervisors can run on top of other operating systems, typically as user-level processes with some extensions in the kernel (including generic kernel modules and/or device drivers). The operating system on which the Type II hypervisor runs is called the *host OS*. The following figure illustrates this idea:

Figure 2-6. Type II Hypervisor



As the figure shows, Type II hypervisors add a layer of complexity because the host OS is between the hardware and the hypervisor. In this scenario, the Type II hypervisor also has to share host resources (for example RAM and CPU cycles) with the other processes running on the host OS. For these reasons alone, guests running in Type II hypervisors typically have lower performance than do guests running in Type I hypervisors.

Reasons to use Type II hypervisors instead of Type I include (but are not limited to):

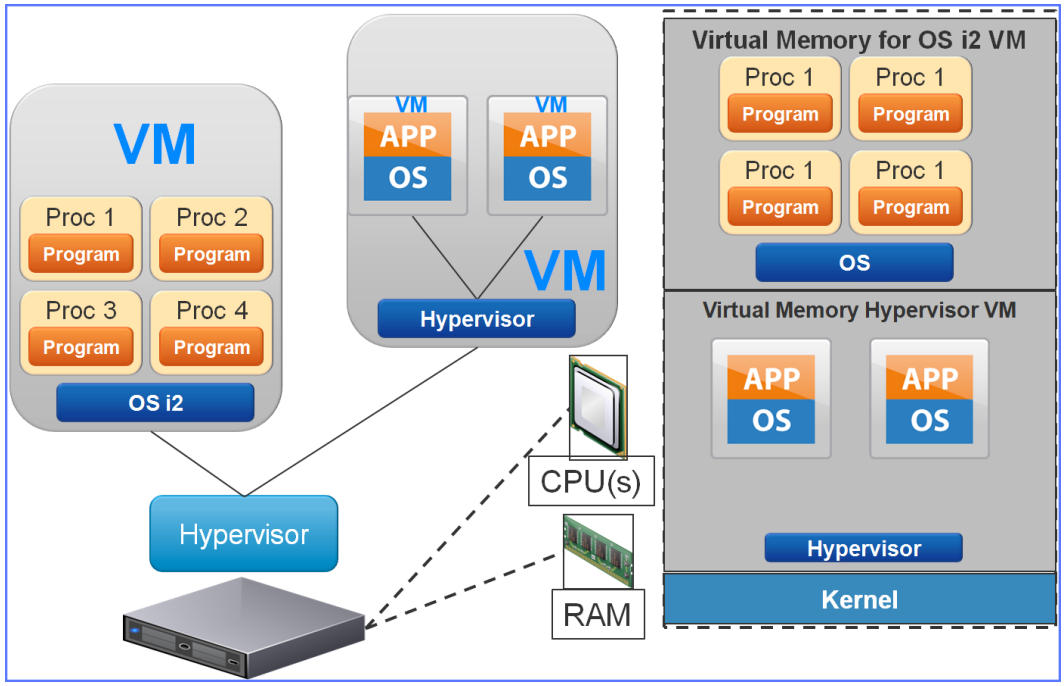
- A Type I Hypervisor may not be available for the host system. But, if the host can run Linux, Windows, or Mac OS X, then there are Type II hypervisors available.
- For desktop and laptop systems, Type II hypervisors allow people to use the native OS that came with their laptop, plus run other OSes that did not. For example, a Mac user can run a Windows and Linux VM on top of their native Mac OS X.

- For systems with low to medium performance demand, it allows people to migrate into virtualization without having to purchase new hardware dedicated to running VMs. For example, a person running a Windows system may wish to test software that runs on Linux, and can do so by placing a Type II hypervisor on their Windows system, then creating a Linux VM to run the new software.
- It allows tech-support personnel to have a single system that runs software in multiple OSes without having to have multiple machines.

Nested Hypervisors

Some hypervisors allow you to run other hypervisors as VMs. This is called *nested hypervisors* and is illustrated in the following figure:

Figure 2-7. Nested Hypervisors



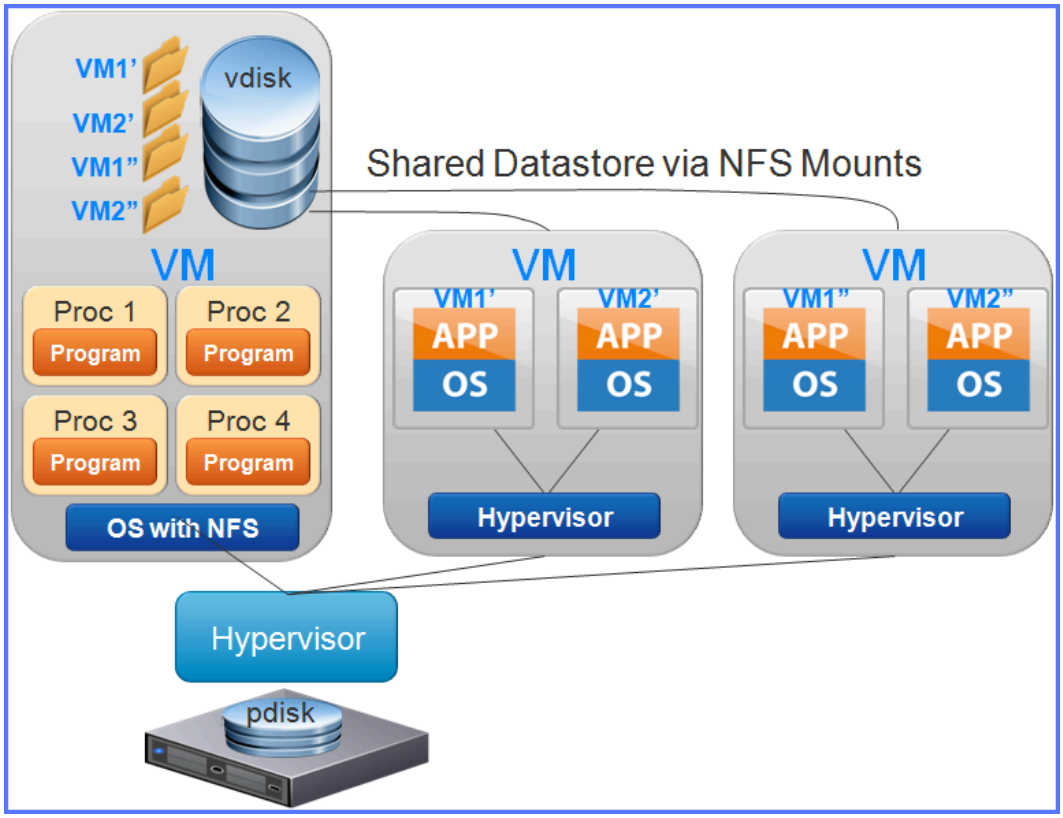
Hypervisor nested on top of a bare metal hypervisor

This figure shows a bare metal hypervisor with two VMs, the right-most is running another hypervisor with two VMs of its own. The VMs in the nested hypervisor are nested in a similar way to VMs running in a Type II hypervisor, and experience similar performance degradation. Notice that the virtual memory required by the nested hypervisor VM must include the virtual memory of the VMs it runs, where this may or may not be true for Type II hypervisors. Further, while this figure only shows one level of nesting, theoretically, one can nest hypervisors to any level. However, since each layer brings another level of performance degradation, there is a limit to how deeply you can nest hypervisors and still receive acceptable performance.

Reasons to nest hypervisors include:

- It allows you test a new version of a hypervisor without requiring new hardware or effecting an existing production environment
- You can create multi-tenancy on host systems at the hypervisor level
- You can simulate clusters of hypervisors without the expense of having multiple hosts systems. You can even provide shared storage for the cluster using a VM in the base hypervisor running an OS that acts as an NFS server as shown in the following figure:

Figure 2-8. Virtual Cluster with Nested Hypervisors



The datastore for the three VMs controlled by the bare metal hypervisor is the local pDisk. The left-most VM has a virtual disk which is managed by its guest OS, which exports some part of the vDisk as an NFS share. The datastore for the two nested hypervisors is the NFS share. Thus, the VMs in the nested hypervisors (vm1', vm2', vm1'' and vm2'') are all stored on the NFS share. This allows the nested hypervisors to be clustered and for their VMs to be moved quickly between them.

Customers are unlikely to use nested hypervisors for production environments, except for multi-tenancy solutions. However, they are very useful for developing and testing solutions, especially those that must perform correctly in a clustered environment.

Caution Hypervisors must read each packet sent to their NIC(s) so that they can pass them onto the appropriate VM. For a nested hypervisor to achieve this functionality, it must put its "NIC(s)" (really vNICs of the host hypervisor) into *promiscuous mode*. This may allow the nested hypervisor to see packets of peer VMs and / or pier nested hypervisors, creating security concerns. Datacenter and network administrators must configure their physical and virtual networking infrastructure to allow the nested hypervisor to put their vNICs in promiscuous mode without allowing them to see their peer's packets.

Defining Types of Hardware Virtualization: Full, Partial, Para, and Hardware-Assisted

Recall this course's definition of virtualization:

> In computing, *virtualization* is the simulation of hardware functionality in software, with or without assistance from hardware. The simulation decouples the virtualized object from the physical object.

Hardware virtualization is a type of virtualization, specifically for virtualizing any computer platform. *Server virtualization* is, in turn, a special case of *hardware virtualization*, where the computer platform being virtualized is a general purpose computer (contrasted, for example, with a virtualized phone, controller, etc.). That said, not all virtualization schemes are equal.

Full Virtualization

In a *full virtualization* environment, all aspects of the underlying hardware are virtualized. The guest software (OS and applications) are unaware that they are running in a VM, and thus need no modification. This is ideal for migrating operating systems, especially legacy OSes, and their applications into a virtualized environment.

Partial Virtualization

In a *partial virtualization* environment, certain features of the underlying hardware are not simulated. The feature is simply missing. If the guest software requires those features, it will either crash or (worse) fail silently, unless the guest software is modified to account for the missing features. If the missing features are obscure, this may not be an issue.

Para Virtualization

In a *para virtualized* environment, the hypervisor offers APIs to guest OSes for either of two purposes:

- To fill in the gaps of features not offered in a partial virtualization hypervisor. In this case guest OSes must be modified to call these APIs in order to have a complete set of features. Guests running on a partial virtualization with para-virtualization APIs that do not use the these APIs may crash or fail silently when they try to use the features absent in the partial virtualized hypervisor.
- To offer alternative methods for invoking features implemented in a full virtualization hypervisor, typically taking advantage of virtualization features available in underlying hardware that implement the feature faster than in software.

In either case, the guest OS must be modified in order to use these features. Said guest OS is said to be made *virtualization aware*. That is, it knows that it is running in a virtualized environment (contrasted with unmodified software running in a full virtualized environment).

Hybrid Solutions

Most modern hypervisors do not fall cleanly into any one of these types. Instead, they provide full virtualization, with para virtualization APIs. Thus, guests that have been modified to use para virtualization can run faster, while guests that have not been modified still run correctly.

Understanding Benefits of Server Virtualization

There are numerous benefits to server virtualization, including (but not limited to):

- *Server consolidation* — Customers can decrease the number of physical servers they need to accomplish the same work. This is possible for several reasons, including:
  - Where applications required a dedicated machine (for example because of TCP port conflicts, shared memory issues, etc.), the applications can each run on a dedicated VM, eliminating the conflicts.
  - Most servers have *head room* (extra CPU cycles, RAM and disk space), which was purchased for worst-case scenarios and to allow for growth. The peak-demand headroom can be shared provided that VMs running the applications do not peak at the same time. The growth head-room can certainly be shared across applications.
- Improved manageability — Most virtualization solutions include a consolidated management solution (such as VMware's vSphere Web Client). Some solutions allow management of just VMs within a single host system while others allow consolidated management across hosts, clusters, even data centers. Further, most virtualization solutions provide one or more APIs to automate management of the

virtualized environment. For example, there are solutions that allow you to create a *template* of a VM, then *clone* the template to create a new instance of the template in seconds. There are APIs that can invoke the cloning operation many times, much faster than any human could click a mouse to take the same action.

- Reduced operational expenses (*OPEX*) — Fewer physical servers, with VMs:
  - Require fewer administrators, since there are fewer things to install or repair
  - Means lower power usage for the computers, and for lower cooling
  - Means fewer networking and storage networking connections, enabling use of smaller switches, in turn lowering power and cooling costs
  - Means reduced space requirements in data centers
- Reduced capital expenditures (*CAPEX*) — Fewer servers means lower capital expenditures on hardware, though the hardware that is purchased does typically require more computing resources than servers not running VMs.

By induction, each of the above can be simplified to the following: The benefit of server virtualization is that it saves money. It does it in many ways. But it all simplifies to saving money. What organizations choose to do with those savings varies widely: Some organizations spend the same amount on the IT infrastructure, but just get better / faster hardware; Some organizations pass the savings on to consumers, making their products less expensive, making their organization more competitive; Some organizations redistribute the savings into other areas of their organization such as R&D, again increasing their competitive edge.

Understanding Factors Driving Server Virtualization

Virtualization has been around since the late 1960s. Yet, it has only really started to become widely adopted by typical IT administrators in the last decade or so. What changed? What made virtualization a valid choice now (or even a decade ago) when it wasn't seen as a valid choice before? There is a long list of reasons, but the major reasons include the following:

- Hypervisors on commodity hardware — VMware (and others) have made hypervisors that work on commodity, x86 (Intel and AMD) hardware. Prior to this, hypervisors (with acceptable performance) were only available on proprietary hardware such as the IBM Power Systems™ and System z (mainframe) servers, Unisys ClearPath mainframes, etc. These were relatively expensive, compared with commodity x86 hardware.
- Inexpensive connectivity — The explosion of inexpensive Internet connections makes it possible to locate servers in remote data centers and still receive an acceptable, if not superior, user experience accessing applications running on those servers. These datacenters allow for economies of scale, aggregating the cost of infrastructure such as top-of-rack/end-of-rack switches, routers, air conditioning, etc.
- Advances in web application technology — Much of the software to manage virtualized environments, create new services in virtualized environments, and even running applications, now happens through web browsers. For example: do you have a Gmail, Hotmail, or Yahoo email account?; Have you used Google Docs? This has driven huge demand for these applications and services, such that virtualization has become the answer to automating spin-up of new (virtual) machines on which to run these applications and services.
- Advances in commodity hardware — Beyond 64-bit servers, which allow for massive amounts of RAM and faster CPUs, both AMD and Intel have added new features to their CPUs, including virtualization-specific instructions, to assist in server virtualization. Further, there have been analogous advances in the bus PCI bus features, such as I/O MMUs, and memory bus technology such as ccNUMA. The following sub-section discusses these hardware advances in greater detail.

- Energy prices and *Elasticity*— Generally, the price of electricity is going up all over the world. In addition to server consolidation, server virtualization allows organizations to spin up new VMs to handle increased demand, and to spin them down again on decreased demand. If a server is left with no VMs to run, the server itself can be powered off, and back on when needed. The ability to turn on and off servers and VMs in response to demand is called *elasticity* in the virtualization.

Taken individually, each of the factors is significant. Combined together, they have a multiplying effect that has allowed adoption of server virtualization to accelerate exponentially.

Understanding Hardware Advances Accelerating Virtualization

Recall this course's definition of general virtualization:

Virtualization — In computing, *virtualization* is the simulation of hardware functionality in software, with or without assistance from hardware. The simulation decouples the virtualized object from the physical object.

Note that the end of the sentence says *with or without assistance from hardware.* The x86 CPU family has had hardware virtualization assistance since the 80286, which provided a *virtual 8086 mode* that allowed the 80286-based systems to run multiple 8086-based programs at once. Intel created the *Task State Segment* (*TSS*), which is a hardware structure analogous to an operating system's *process* structure, but was also intended to keep track of virtualized 8086 instances.

This same support exists even in the newest x86 CPUs. With the growing acceptance of x86-based hypervisors, people are almost entirely interested in running 32 and 64-bit VMs. Unfortunately, the x86 (pre AMD V and Intel VT extensions) did not facilitate easy virtualization of 32 and 64-bit guests. The hypervisors had to resort to many tricks, such as shadow page tables and trapping many operations back to the hypervisor, to make virtualization work on these CPUs. These tricks extracted a heavy performance penalty for virtualized guests, sometimes as much as 40%.

However, over the past few years, AMD and Intel have added features to their CPUs that assist virtualization of 32 and 64-bit VMs. Further, various industry standards groups have extended the PCI* bus protocols allowing bus/bridge implementations to also assist in virtualization. AMD and Intel have added these standards into their chipsets and I/O device manufacturers have added them into their devices.

Essentially, each of these advances provide a new feature for para virtualization. That said, some of the features make the running of guests more efficient without requiring guest to use para virtualization APIs. The more of these advances a hypervisor can leverage, the faster the hypervisor and its associated VMs run. While there are many such hardware advances, the following sub-sections discuss a few of the key advances.

Understanding x86 Privilege and Virtualization

Historically, CPUs only had two levels of permission: Supervisor and User. The kernel ran in Supervisor mode, and user processes in User mode. Some instructions were restricted to supervisor mode, such as changing page tables. Attempting to execute an instruction requiring Supervisor privilege from User mode resulted in a *trap*. All traps were handled in Supervisor mode. System calls were thus often just invoked by forcing a certain trap.

At some point, hardware vendors determined that this dichotomy was (or would be) insufficient. Instead they created *rings* of privilege, typically numbered from 0, with 0 being the most privileged level.

The x86 architecture defines 4 levels of privilege, 0 to 3. Each instruction has a privilege level associated with it. Programs running in ring X can execute instructions requiring permission >=X (for example, code running at ring 2 can execute instructions requiring ring 2 or 3). Thus, the lower the ring number, the greater the privilege. Some instructions that are available in multiple privilege levels behave differently, depending on the privilege of the code executing the instruction. Most operating systems running on x86 systems set up the kernel to run at ring 0 and all user processes at ring 3 (user mode). They did not use rings 1 or 2.

TRIVIA The GE 6180, developed in the 1960s, had one of the first hardware implementations of privilege rings. It had 8 rings of privilege.

With the advent of x86-based hypervisors (without AMD V and Intel VT technology - see "Understanding AMD-V and Intel VT-x," on page 25) the hypervisors need to control the shared resources (RAM, I/O devices, etc.). To do this, they need to trap every access by guest VMs to shared resources. However, if the guest OS was running at ring 0, the page fault and I/O accesses were valid. The question was, "How do you get the guest OS to trap on these operations?"

The (perhaps) obvious solution is to run the hypervisor kernel at ring 0, and run the VM's kernels at ring 1. For example, when a kernel in a VM tried to access the page table, the instructions would generate a trap, invoking the hypervisor, which would then handle the page mapping with a global view of memory. Then the hypervisor would have to maintain a *shadow page table*, (a software copy of the page tables seen in the guest OS) for each VM. This was both slow (many instructions to perform the trap, create the shadow page table entry) and expensive (lots of RAM used just to keep track of the shadow page tables).

NOTE This technique is known as *trap and emulate.* In the x86, it didn't work for all operations. Some additional tricks were needed.

However, the x86 presents some stumbling blocks to such a straight-forward solution, including the following:

- Misbehaving instructions — Some instructions, such as `POPF` (pop top of stack into the `FLAGS` register), fail silently when run in ring 1 or higher. That is, the instruction does not cause a trap, it simply acts as a `NO-OP` when not run at ring 0. Instead, full virtualization hypervisors must perform *binary translation* of such instructions to other instructions that have the same effect. This can be time consuming, even with optimizations.
- Hidden data structures — When the x86 loads a hardware task, it does not have registers that point to the task state in memory. Rather, it copies the hardware task state into the CPU. The memory from which the hardware task was loaded can then be overwritten. Further, there is no way to retrieve the task state from the CPU once it is loaded. To resolve this issue (and others), hypervisors can use *shadow data structures*, copies of the data structures needed by the CPU, maintained by the hypervisor. Maintaining these shadow structures also incurs some cost in performance.

Since both of the solutions to these problems effect performance, Intel and AMD have each added features to their CPUs that provide hardware assistance for these issues, known generically as *AMD-V* and *Intel VT*.

Understanding AMD-V and Intel VT-x

In 2004, AMD announced its *AMD-V* (virtualization) technology, which were 64-bit CPUs with extended features specifically designed to help said CPUs run VMs faster. The features have had many internal and external names: *Pacifica*, *Secure Virtual Machine (SVM)*, and finally just *AMD V*. Intel followed suit in late 2005, calling their technology *Intel-VT*. The Intel features were initially code-named *Vanderpool* and have also been known by other names. Both extensions have a similar base set of functionality, though the implementation differs substantially. The base functionality includes:

- *Second-Level Address Translation* (*SLAT*) — Intel calls their version of this feature *Extended Page Tables* (*EPT*). AMD calls their version of this feature *Rapid Virtualization Indexing* (*RVI*), but it's also known as *Nested Page Tables* (*NPT*). At a high level, this feature allows direct mapping of guest-virtual to host-physical addresses. It further allows most guest OS page faults to be handled in the guest, provided the

hypervisor has made a guest-physical to host-physical mapping. These two different mappings require two levels of page tables with two separate translation look-aside buffers (TLBs). This feature obviates the need for a hypervisor to keep shadow page tables for each guest. Various benchmarks have indicated that guests can realize up to 40% performance improvement running on SLAT-aware hypervisors running on SLAT-enabled hardware. The course ESXi System Fundamentals discusses this feature and how VMware's ESXi uses it in great detail.

- Hardware VM structures and World Modes — AMD and Intel created hardware structures to keep track of VMs that are analogous to the Task State Segment (TSS) in the 80286 and later x86-based CPUs (AMD calls their structures *Virtual Machine Context Blocks* or *VMCB*s. Intel calls their structures *Virtual Machine Control Structure* or *VMCS*). In general, the hypervisor now loads the context of the VM into this structure, sets some bits (discussed below) then invokes the (new) `VMRUN` instruction. This instruction causes the CPU to both switch context to the VM and to switch modes from Hypervisor Mode to VM Mode. Together, the mode switch and context switch are called a *World Switch*.

  In addition to traditional VM context, like register values and page tables, these hardware context structures include a list of which actions, when performed by a guest, will cause the CPU to exit VM mode and return control to the hypervisor. VMs with guest OSes that have no para virtualization have more items causing a world switch than those that are para virtualized.

  When a guest exits VM mode, the reason for exiting (what did the guest do to cause the exit?) is placed in the hardware structure so that the hypervisor knows what action to take on the VM's behalf.

- Additional instructions — In addition to the `VMRUN` instruction discussed in the preceding bullet, these extensions introduce additional instructions not present in the standard x86 instruction set, including (but not limited to) `VMCALL`, `VMCLEAR`, `VMLAUNCH`, `VMREAD` and `VMWRITE`. See each manufactures' product documentation for a complete list of these instructions and their function.

Both AMD and Intel have offered further extensions to the base AMD-V Intel VT extensions. The current list of extensions include:

- AMD's *Vi* and Intel's *VT-d* — Both of these extensions allow guests to perform I/O without hypervisor intervention by using an *input / output memory management unit* (*IOMMU*). Briefly, IOMMUs are like normal MMUs, but they provide virtualized-to-host-physical address translation capabilities. That is, the IOMMU gets loaded with a set of contiguous virtual I/O addresses for a VM and maps them to host-physical addresses. When a driver does I/O through the IOMMU, for example doing DMA from virtual I/O address 0x1000-0x1FFF to a disk: 1) The IOMMU translates the virtual I/O address to a host-physical address; 2) It DMAs the data from the host-physical addresses to the disk. The IOMMU can be loaded with different mappings for each guest, preventing one guest from writing into the memory of another. This unimpeded I/O is not possible with guests using standard MMUs.

- Intel VT-c — At a high level, this feature allows Intel NICs to provide separate transmit and receive (Tx/Rx) queues for each VM (there is a limit), obviating the need for the hypervisor's virtual switch code to do simple packet send/receive operations to the VM. This feature also has implications for implementing *quality of service* (*QOS*) features on the NIC instead of in the hypervisor's vSwitch code.

Note the PCI-SIG (special interest group) body's *Single Root I/O Virtualization* (*SR-IOV*) standard specifies a standard method for multiple VMs to access a physical device concurrently. This standard relies on devices implementing virtual PCI functions, i.e. AMD Vi and Intel's VT-d / VT-c features.

The common theme with all of these features is that they allow guests to run faster by obviating the need for hypervisor intervention in some cases and by offloading some operations from the CPU to I/O devices entirely. The speed increases in guests have accelerated adoption of virtualization.

Understanding Virtual Appliances, vApps, OVFs, OVAs, Templates, and Clones

Software vendors have started creating turn-key software solutions that run in virtualized environments. That is, instead of requiring customers to: 1) Create a new VM; 2) Install and configure an OS in the VM; 3) Install and configure their software on the OS; ... the customer deploys one package, called a*virtual appliance* or *virtual application* (*vApp*) that includes the pre-configured OS and their software, ready to run. For those unfamiliar with *virtual appliance* or *virtual application*, they are defined as follows:

Virtual Appliance — A software bundle that includes an installed and configured operating system, plus installed application software that may need final configuration. The operating system is typically customized to contain a kernel and only those libraries, tools, and drivers necessary to support the application software. The bundle is packaged as either a *open virtualization format* (*OVF*) file (defined later in this section), with requisite support files, or as a single *open virtualization archive* (*OVA*) file (defined later in this section), suitable for deployment to a hypervisor. The result of said deployment is a single VM running the bundled OS and applications. During deployment, the hypervisor management tool may ask questions to complete the configuration of the OS (such as IP parameters) and application software. Some virtual appliances ask these questions after the first boot.

Examples of virtual appliances include the following:

Note The examples below are taken from a combination of looking at the VMware Virtual Appliance Market Place and internet search engine results. Where there are multiple vendors for a given example, they are listed in alphabetical order. Neither a vendors appearance, or non appearance in the list, nor the order in which they appear are meant to convey any endorsement or lack thereof, by VMware, Inc.

- Anti-spam, email, and web filtering appliances — Many security related vendors including, but not limited to, and in alphabetical order, Barracuda Networks, Blue Coat, Dell (Sonic Wall), Symantec and Trend Micro.
- Layer 4 load balancing appliances — These appliances take in layer 4 traffic and use configured policies to distribute the traffic to one of many possible target servers, which in turn may be running in VMs. Policies can be as simple as round-robin or may take into account load on each of the target servers. For example, large organizations typically place layer-4 load balances in front of an array of web servers. Several companies make Layer 4 load balancers, including Aloha Networks, Barracuda Networks and F5 Networks.
- VMware's vCenter Server Appliances (VCSA) — This includes a tailored version of Linux plus the vCenter Server application software such as the Single Sign On Server (SSO), Inventory Services server, vSphere Web Client server, etc. For more information on the VCSA, see GUID-62DDC2BE-828B-4DF7-9C44-C3EFAB8BC0E5#GUID-62DDC2BE-828B-4DF7-9C44-C3EFAB8BC0E5/SECTION_BAFEE40666E94303BC1129A59B8E10E0.

Open Virtualization Format (OVF) — A standardized file format for packaging and distributing virtual appliances and virtual applications to be run in VMs on hypervisors. An *OVF Package* is a folder with a collection of files that includes a (human readable, XML-formatted) *OVF file* that contains meta-data that describes one or more VMs to deploy on a hypervisor. The format was first defined by the Distributed Management Task Force (DMTF) and has subsequently been ratified by the American National Standards Institute (ANSI) and the International Standards Organization (ISO).

There are several *sections* of meta-data, each with its own XML tag, for each virtual system. Each section describes some attributes of the VM or vApp. For example, there is a section that defines the virtual hardware to deploy in each VM including vCPUs, vRAM, vDisks, and vNICs. There is another section that maps *.vmdk* files in an OVF Package to vDisks. There are many core sections, each with many sub-sections and values that you can include in the file.

Most of the sections referenced above are used by hypervisors and virtualization management infrastructure to configure new VMs and populate their vDisks during deployment. The `<ProductSection>` of an OVF file can contain definitions of one or more *OVF properties*, which are key-value pairs typically used by developers to hold variables to be passed to the VMs after the VM is deployed, when the guest OS boots. When you deploy the OVF, you can provide values for the properties. For example, the VMware GUI management tools provide a form with each OVF property name and description, and a field for you to supply the value.

When an OVF file includes OVF properties for a VM, its hardware description must also include an `ovf:transport` attribute that specifies how the guest OS receives said properties. One option is to have the management infrastructure bundle the properties into an `ISO` file and attach said ISO to a vCD-ROM of the VM. Regardless of the transport, the software in the VM (either the guest OS or application software) is responsible for retrieving, interpreting, and implementing the OVF properties. They are not interpreted by the hypervisor or management infrastructure.

Note The OVF definition includes a security mechanism that can be used to ensure the integrity of the OVF bits.

Open Virtualization Archive (OVA)

An OVA is a file that contains an archive of an OVF package folder, using the `tar` file format. This makes deployment of VMs simpler than from OVF packages, because you only download and stage one object, the OVA file, instead of an OVF, virtual disk files, certificate files, etc.

Virtual Application (vApp)

A packaging of virtual appliances into an OVF package / OVA, where the OVF file contains multiple *virtual system* sections, deployment of which causes the creation of multiple VMs on the hypervisor. When multiple *virtual system* sections are present in an OVF file, they must be wrapped in a section titled *`VirtualSystemCollection`*. This section can also contain a set of OVF properties. If said section is present, those properties are global to all VMs described by the OVF.

Note A Virtual Appliance is a special case of a vApp with only one VM.

There are several reasons why vApps may use multiple VMs to implement their application, including:

- The application software in the vApp requires operating systems that are tuned very differently to get good performance. For example, the vApp may include a Linux OS optimized for running PostgreSQL, another Linux OS optimized for running the Apache Web Server with Tomcat.
- The vApp includes software packages that require different operating systems. For example, the vApp may include a web browser running on Linux, and an Active Directory server running on the latest version of Windows Server.

- The vApp includes software packages that require separate OSes due to conflicts. For example, consider an email-filtering vApp that employs several Mail Transport Agents (daemons that speak SMTP) that listen on port 25, and the port is not configurable. To solve these problems, run each app in its own VM. So, to resolve the conflicting MTA's issue, run each MTA in separate VMs' guest OSes.
- To allow increased performance in a cluster. If a vApp involves several services, if all of them are concentrated into a single VM, the work cannot be distributed to multiple hosts systems. If the services are deployed into separate VMs, the VMs can be distributed into multiple host systems in a cluster, allowing for increased overall performance.

While virtual appliances and virtual applications allow (relatively) rapid deployment of solutions, they typically require additional human interaction during deployment, for example to set the new VM's name. As an alternative, VM *templates* allow administrators to fully automate deployment of virtual appliances / virtual applications. While the term *template* is highly overloaded in virtualization, in this context, the term has the following meaning:

Template (for VMs) — A *gold master image* of a VM that administrators (or automation tools) use to deploy new VMs. The template can contain place holders for customizations that are to be performed when instantiating a new VM from a template.

Clone (VM) — A nearly-identical copy of an existing VM. Various hypervisors may allow customization of the new VM from the source VM. Each virtual infrastructure scheme must have some method of uniquely identifying each VM, even between clones.

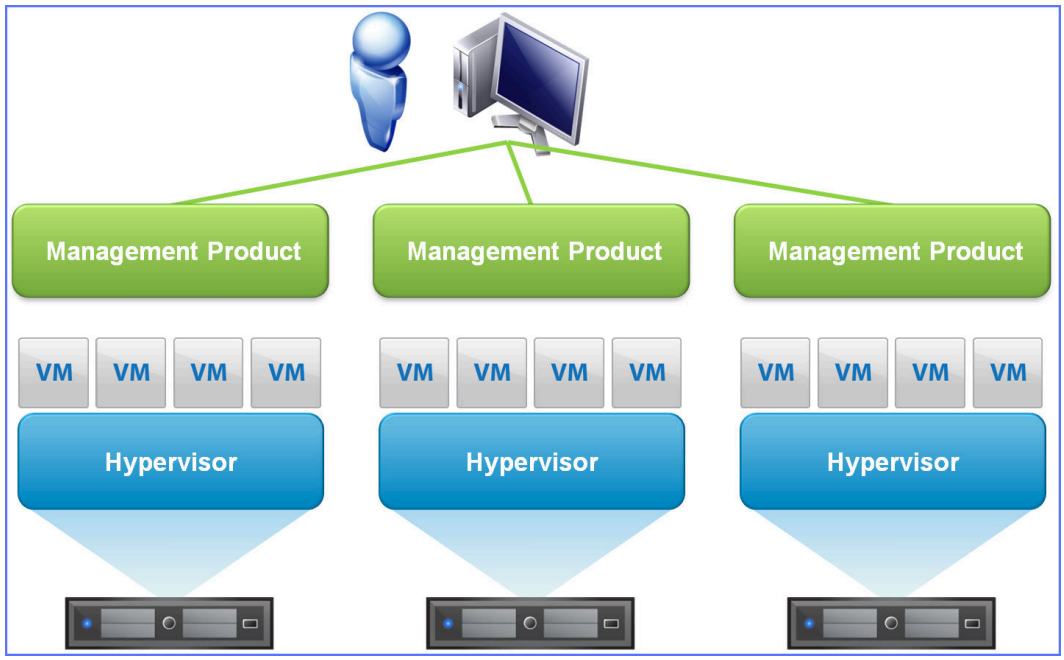
> **Note** VMware historically used a *UUID* to uniquely identify VM. While VMs still have UUID's, VMware's current vCenter Server technology assigns a unique *Managed Object Reference* (*MOR*) to each VM for identification.

Note Not all hypervisors, or their management tools, allow creation / deployment of VMs from templates.

Understanding Management Solutions for Virtualized Environments

Each hypervisor product must provide a method for managing it, including creating / deleting / starting / stopping VMs and allocating resources – for example vNICs, vDisks, vRAM, vCPUs, etc. – to VMs. Some products use a web application model with administrators using a web browser to perform administrative actions. Some products use a proprietary client / server model with proprietary protocols. Some products have a hybrid approach. Administrators authenticate to each hypervisor to manage it, as shown in the following figure:

Figure 2-9. Embedded Hypervisor Management



Environments with multiple hypervisors compel, or even require, a unified management solution that allows administrators to control the environment from a single interface. For example, moving a VM between two hypervisors *can* be done by managing individual hypervisors by:

1 Pausing a VM on the hypervisor in which it is running

2 Removing the VM from the inventory of the hypervisor on which it was running, without deleting the VM's disk files

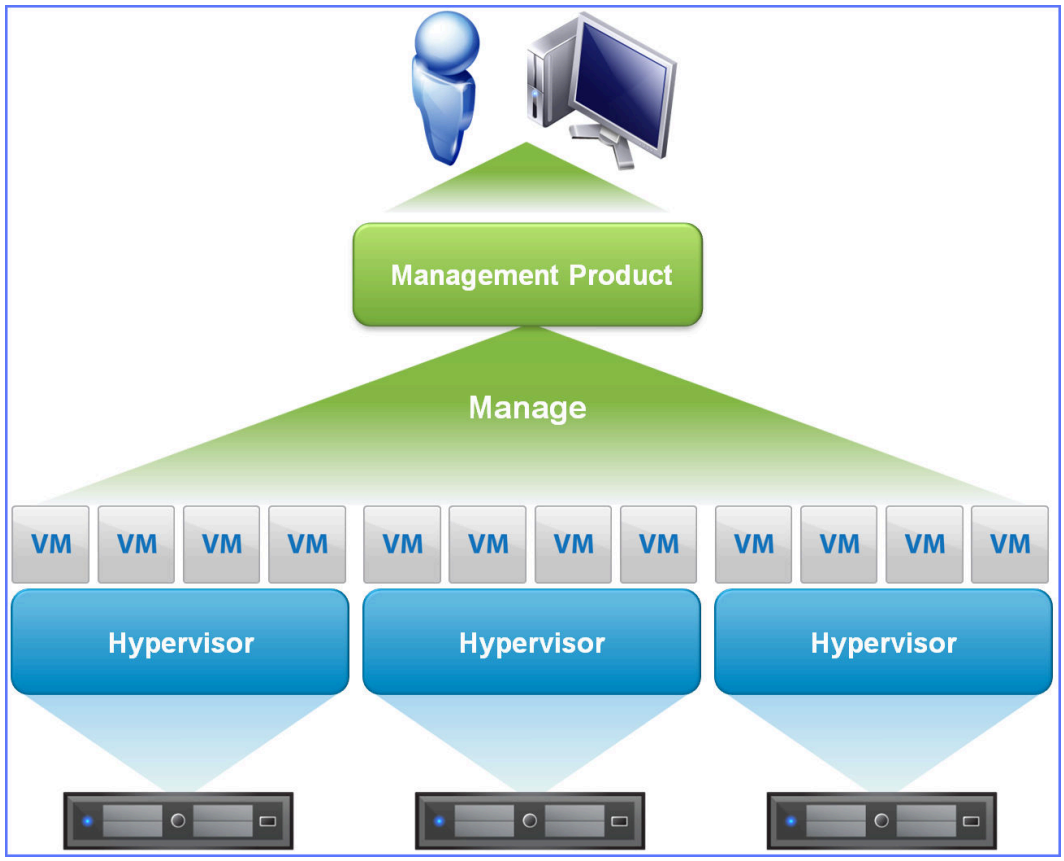
3 Adding the VM to the inventory of the new hypervisor on which it will run, using the VM's disk files

4 Resuming the VM on the new hypervisor

This procedure is tedious. While it can be scripted, the script would have to be able to authenticate to each hypervisor separately. Further, automating the invocation of the scripts to implement high availability (HA) and / or load balancing is non-trivial.

Vendors of commercial hypervisors offer additional products to manage virtual environments using their products. These add-on products are often required to enable certain features, for example creating a cluster, configuring HA and load balancing parameters within the cluster, creating distributed virtual switches (see "Understanding Virtual Distributed Switches (vDS) and Their Benefits," on page 45), etc. – basically anything that requires coordination between hypervisors. These management products may run in a VM in a hypervisor, or run on a bare-metal system. Example of such management products include VMware's vCenter Server and vCenter Server Appliance (VCSA), Microsoft's Virtual Machine Manager (part of their System Center product), and Citrix's Xen Center. These products are referred to as *single pane of glass solutions* because of their ability to manage multiple hypervisors from a single management instance, as shown by the following figure:

Figure 2-10. Centralized Virtualization Management Solution



3 hypervisors managed from a single pane of glass

In addition to products that provide interactive interfaces for administrators to manage a virtualized environment, virtualization vendors also offer other products that allow automated management of and extensive visibility into virtualized environments, for example the vCenter Orchestrator, vCenter Operations, vCloud Automation Center (vCAC - pronounced either "v-cake" or "v-cack" rhymes with "tack"), etc.

Understanding Desktop Virtualization

Wikipedia provides the following definition of *Desktop Virtualization* (see `en.wikipedia.org/wiki/Desktop_virtualization`):

Desktop Virtualization (according to Wikipedia) Software technology that separates the desktop environment and associated application software from the physical client device that is used to access it.

What does this mean? In general, it means that desktop virtualization is software that allows users to access a desktop work environment, for example `Gnome` / `Explorer` / `Finder`, from devices that may or may not be running Linux / Windows / OS X, respectively.

In practice, there are several interpretations of what constitutes *desktop virtualization*. Each interpretation has its own implementation. The next two sub-sections of this course discuss two interpretations of what constitutes *desktop virtualization*:

- *Remote desktops / terminal services / mirrored-host* solutions
- *Host-based virtual machines / virtual desktop infrastructure* (*VDI*) implementations

The discussions in the next two subsections relies on the following two definitions:

Thin Client — A *thin client* is a platform with low computing resources, such as a small amount of RAM, relatively slow CPU, little or no disk (or perhaps a small amount of NVRAM in place of disk) and a minimalist operating system. Thin clients typically consume small amounts of electricity and cost significantly less than fat clients. They typically include a decent graphics display.

Fat Client — A *fat client* is a platform with the computing resources, including RAM, disk, graphics processing unit (GPU) and reasonably powerful CPU as necessary to run a desktop OS and associated applications. Fat clients typically require significantly more power to run and cost significantly more money to purchase than do thin clients.

Note the external definitions of desktop virtualization in the subsequent sub-sections are taken from articles by Margaret Rouse and Jack Madden and presented in at the following website: `http://searchvirtualdesktop.techtarget.com/definition/desktop-virtualization`).

Defining Remote Desktops / Terminal Services / Mirrored-Host Solutions and their Benefits

One interpretation of *desktop virtualization* is where people can use either a thin or a fat client and run a client program to logon to a desktop environment provided on either a laptop, desktop, or server-class system. The "desktop" environment each user gets is either: unique to each user; or shared by all users:

- Unique to each user — The system (typically a server running a Windows Server OS or Linux) is shared by multiple users. Each connection / authentication generates a new desktop "shell" (for example `Explorer` on Windows, `Finder` on OS X, or `Gnome Desktop` on Linux).

  For example, running `Microsoft Remote Desktop Connection` on OS X allows users to access Windows desktops from an OS X desktop. There are similar applications from various vendors for accessing Windows desktops from Apple i-devices and Android-based devices. Such programs rely on Microsoft's *Remote Desktop Protocol* (*RDP*), which allows remote presentation of graphics, text, and even sound and whose specifications are published by Microsoft.

- Shared by all users — The desktop environment is *mirrored* when accessed by remote users - that is the desktop "shell" is shared between all users accessing the system.

  An example of this is running `VNC` on a Windows system allows people to access `Gnome` and `Finder` desktops running on Linux and OS X systems, respectively. VNC can actually be used to and from many different environments, not just those listed here, because VNC uses its own protocol. As long as there is a *VNC server* implementation available for a given OS, a VNC client can connect to it. VNC clients exist for Windows, Linux, OS X, Apple IOS, Android OS, and other platforms.

Note While the OSes in these implementations can be running in VMs, it is not necessary to the definition of the type of virtualized desktop.

One advantage of this type of desktop virtualization is that it does not require investment in additional server infrastructure. Rather, it just provides access to existing desktops via protocols such as VNC and RDP.

Defining Host-based Virtual Machines / Virtual Desktop Infrastructure (VDI) and its Benefits

The VDI implementation of desktop virtualization has users connect (using either a thin or fat client) to a VM that is running a desktop OS. The VMs are typically running on server-class hardware with under control of Type I hypervisors. Most configurations allocate VM's vCPU and vRAM from a *pool* of resources running under control of a cluster of hypervisors, providing high availability as well as load balancing.

The user may logon to the same VM each time (known as a *persistent desktop*), or get assigned a random VM that starts in the same initial state at each session (known as a *non-persistent desktop*). Persistent desktops allow the client to disconnect at any point, and pick up where they left off.

These implementations require server virtualization infrastructure to run the VMs, which can be costly. However, that cost buys significant advantages, including (but not limited to):

- Better protocols — The protocols used to implement VDI are typically more efficient than VNC, RDP, or similar protocols used to implement Shared Hosted solutions. This optimizes network utilization and the user experience.
- Centralized administration — Since all of the VMs for the desktops are located on a centralized set of host systems, the VMs themselves can be centrally managed, including upgrades, malware scans, additions, deletions, cloning, etc.
- High availability — The clustering of the hypervisors and storage provides for highly-available desktops. Barring a complete cluster failure, there is at least one host on which to run the Desktop VMs.

These host-based systems

Understanding Common Benefits of Desktop Virtualization

Both of the implementations of desktop virtualization discussed in this course allow (NOTE: not *require*, but *allow*) people to use thin clients to access their desktops. For mobile people, this is an added bonus when you consider the weight of carrying a tablet device vs the weight of a full laptop (fat client) and also their respective battery life.

Note It is important to point out (though it might be obvious) that these virtual desktop implementations are completely reliant on network infrastructure. The clients must be able to reach the desktops (whether on VMs or bare metal) in order for the solution to work. Reliability becomes even more important that bandwidth.

Understanding why Client/Desktop-based Virtual Machines Do Not Constitute Desktop Virtualization

The Rouse and Madden definition of desktop virtualization (see at `http://searchvirtualdesktop.techtarget.com/definition/desktop-virtualization`) included another type of desktop virtualization: Client-based Virtual Machines. Paraphrased, this is where a fat client runs a Type II hypervisor in addition to their desktop OS. For purposes of this class and the VMware Certified Developer certifications, this is not considered a form of desktop virtualization. Rather, this is just server virtualization running on a desktop (or even laptop) class machine with a Type II hypervisor.

Understanding Application Virtualization

Wikipedia offers the following definition for *Application Virtualization* (see `http://en.wikipedia.org/wiki/Application_virtualization`):

Application Virtualization	Is software technology that *encapsulates* application software from the underlying operating system on which it is executed.

Note The emphasis on encapsulates was added for this course.

What does this mean? It meas that an application is bundled up with all of the dynamic linked libraries (DLLs) / shared objects (SOs) on which it depends and configuration and data files it has after installation such that it can be instantiated as a running application at will. The virtualized application still depends on the underlying operating system.

Notice that this is an entirely different kind of virtualization from desktop and server virtualization. Those virtualization techniques involved decoupling an OS or Desktop from its underlying display or server hardware. Instead of a *decoupling* technique used in those technologies, application virtualization uses *encapsulation*. The only extent to which application virtualization may involve decoupling is that the OS on which the application depends may be emulated as is the case with Linux-based Wine and (to a lesser extent) Windows XP emulation mode in Windows 7.

However, there is a similarity between application virtualization and server / desktop virtualization. Each instance of the application runs in its own *container* or *sandbox*, much like a guest OS runs in its own VM. For example, when an virtualized Windows application changes the registry, it is only changing the registry in its own container. Not all aspects of the application are contained, though. For example, the filesystem in which the application runs is the standard filesystem of the host OS.

Some examples of application virtualization include:

- Docker (see `www.docker.io`) — Extends a common container format called `Linux Containers` (`LXC`) to rapidly deploy Linux-based applications within said containers (without using VMs)
- Wine (see `www.winehq.org`) — Allows people to run certain Windows applications on Linux and other Unix-like platforms
- Cygwin — Provides a Unix-like environment on Windows platforms. You can run Widows applications from a Unix-like CLI, as well as other Unix-like commands. For more information, see `en.wikipedia.org/wiki/Cygwin`.
- VMware ThinApp — Provides containers for running legacy Windows applications (for example IE 6 running on Windows XP) on other platforms

The next section discusses some of the benefits provided by this isolation and application virtualization in general.

Defining Benefits of Application Virtualization

There are several benefits to application virtualization, (many of these are listed in the Wikipedia site for application virtualization), including:

- Instances can be spun up quickly, without the need to install it for each new user. In a datacenter with thousands of users, this can be a huge win.
- It may allow otherwise incompatible applications to run simultaneously, if the contention is for resources isolated in the application's container, for example competing changes to the registry or requiring different versions of the same DLL or SO
- An instance of a virtualized application requires less resources than a VM required for a virtualized desktop running the non-virtualized application
- It may enable application portability and longevity for legacy applications. Both of these benefits come from providing an OS interface that supports the system calls required by the application and its libraries.
- In some cases, there is increased security. For example, an application with malware, or just incorrect code, that corrupts the registry on a Windows system will have its effects limited to the registry in its container.

There are limits to application virtualization. For example, applications that require access to physical hardware (for example drivers that use hardware-based copy protection) typically cannot be virtualized.

Understanding Storage Virtualization

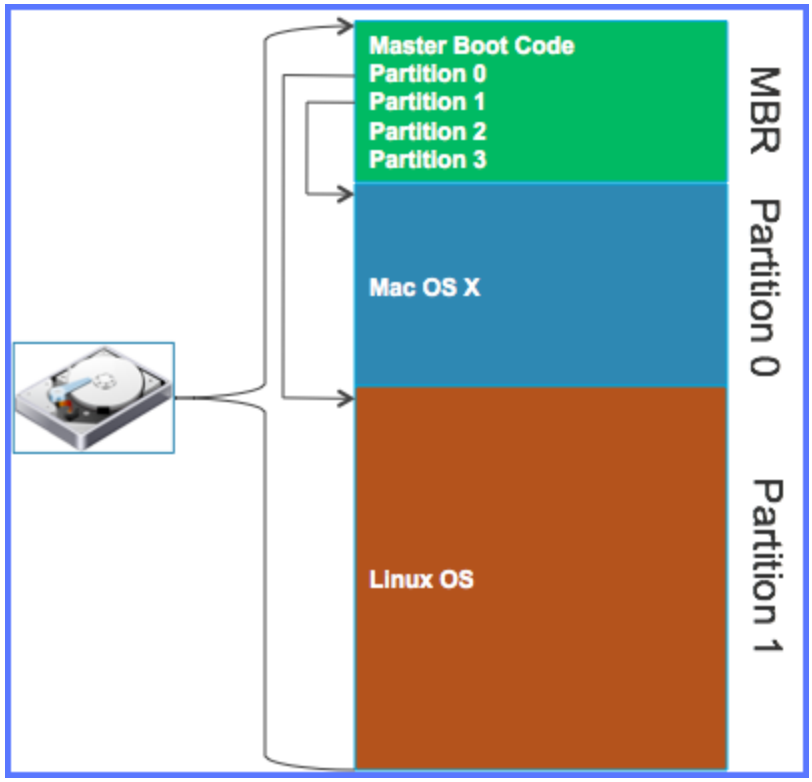
Storage virtualization is software that abstracts the location of a file from is physical storage device (disk, etc.). It has existed for decades in various forms. Two of the most notable are disk partitions and RAID arrays:

Disk partition — A *disk partition* is a specific subdivision of a disk. Historically, disks on x86-based systems recognized the first block on a physical disk as the *Master Boot Record* (*MBR*), which contained both boot code and the *partition table*.

RAID array — A technology used to aggregate several "inexpensive" disks into a single virtual disk. The resulting group of disks is called a *RAID array*

Initially, it is easiest to think of the partitioning of physical disks, such as illustrated in the following figures:

Figure 2-11. Single Physical Disk with Master Boot Record and Two Partitions



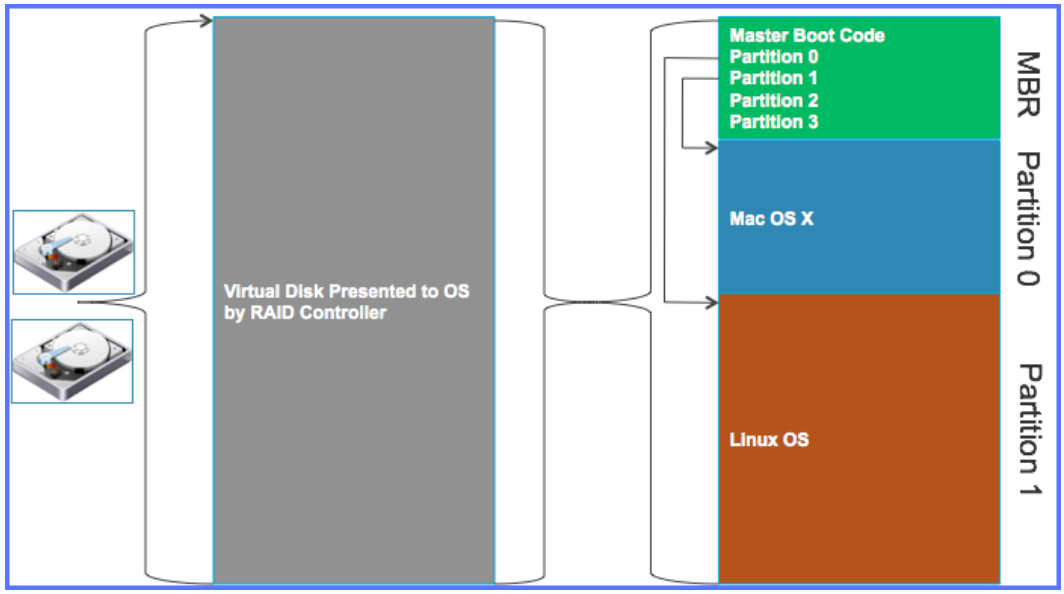
This figure shows a single physical disk with a master boot record (MBR) and two partitions. The MBR may contain a program that allows the user to decide at runtime which partition to boot from. GRUB and Bootcamp are examples of such software. Each partition is a *virtual disk* because, once the boot loader boots the OS in the partition (from the figure, the choices are Linux or Mac OS X), the OS only sees their partition as their disk. That is, the booted OS only uses their partition as their *disk*. They don't write into the other partitions, or the MBR.

Note The exception to OSes writing into the MBR or seeing other partitions are special system tools that are written to access these objects via BIOS calls. Examples include the Windows and Linux `FDISK` utilities, and Mac OS X's `Disk Utility`.

The MBR partitioning scheme has been used on x86 computers since the original IBM PC was introduced. It is limited to four (4) partitions, each of which can be up to 2TB in size. There is a new standard, called the *GUID Partition Table* (*GPT*) that allows up to 2^64 blocks / partition and (theoretically) an unlimited number of partitions. GPT-based disks can only be read by the newer operating systems that support them. For more information, see `http://en.wikipedia.org/wiki/GUID_Partition_Table`).

The partitioning software and the OS get their view of the physical disk from the BIOS, which in turn gets its view of the disk from the disk controller. It is possible (and common) for host systems to have a RAID controller, with multiple disks configured into a RAID array (for example a 2-disk mirror), and for the RAID controller to "lie" to the BIOS, saying it only has one disk. The following figure depicts such a scenario:

Figure 2-12. Internal Disk Mirroring



Note For several decades now, even the physical disks technically use virtualization. Their internal logic detects bad blocks and maps them out, replacing them with spares. This mapping is invisible to the host bus adapter (HBA), though S.M.A.R.T. disks (those that use the Self-Monitoring, Analysis and Reporting Technology) make this and other information available for query through the HBA.

The above two scenarios focus on virtual disks for *internal storage*. Modern servers can use several types of storage, defined as follows:

Internal Storage — Disks attached to an HBA on the system board's (typically PCI) bus, all of which resides within the host system's enclosure.

Network Attached Storage (NAS) — Disks residing in a chassis external to any host system, accessed using IP-based protocols, for example iSCSI, network file system (NFS) or server message block (SMB - closely related to Microsoft's Common Internet File System - CIFS). The storage is accessible by all systems (clients or servers) that have been granted access to the NAS device and use the same network protocols.

Direct Attached Storage (DAS)

Disks residing in one or more chassis internal or external to a host chassis, typically attached to one host systems via a proprietary HBA. Some DAS products with external chassis support connections to multiple hosts with multi-initiator SCSI to create a JBOD (Just a Bunch Of Disks) cluster. Besides SCSI, common DAS HBAs and drive protocols include SATA, eSATA, and SAS.

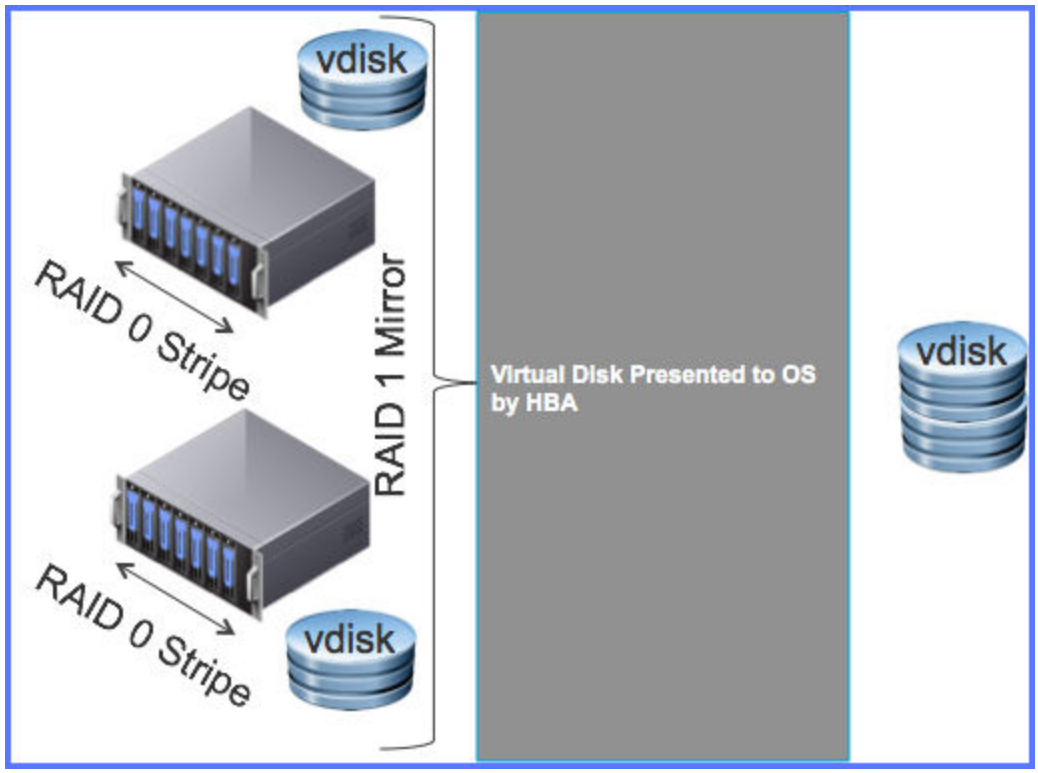
Storage Area Network (SAN)

Disks residing in one or more chassis external to a host system, attached to one or more host systems (typically) via a Fibre Channel HBA (FC HBA) and Fibre Channel switches. The presence of Fibre Channel switches eliminates the need for a crystal network between hosts and storage cabinets. The FC HBAs typically present a SCSI interface to the host system. The Fibre Channel switches are sometimes referred to as *Fabric Extenders* (described and defined separately).

Layering Storage Virtualization

External storage solutions typically allow administrators to nest or layer their virtual disks (typically called *volumes*). For example, suppose a datacenter contains two storage chassis full of disks. The storage solution may allow an administrator to create a volume that spans all the physical disks within each chassis, resulting in two virtual disks / volumes. The solution may then allow the administrator to create a volume that is a mirror between the two volumes within the chassis, resulting in a single volume from the two. The solution vendor may or may not use RAID technology within the chassis or between them, but the result is the same:

Figure 2-13. Layering volumes



The possibilities for laying volumes within and between external storage is endless. However, remember that each layer of virtualization typically adds overhead that lowers I/O performance.

Disk / Volume Presentation

The various storage types, except HBAs for IDE and SATA disks, typically presented their virtual disks to a host system as SCSI disks with a SCSI ID and LUN (logical unit number). However, NAS devices must use a networking storage protocol to present their volumes. Common networking storage protocols include:

- Network filesystem protocols — There have been several network file-sharing protocols over the years, of which NFS and SMB currently dominate. These protocols make folders on *server* computers available for integration into the folder tree on *client* computers. The folders on the servers made available to the clients are sometimes called *shares*, regardless of the protocol used.

  On Unix-derived systems (Linux / Mac) the remote filesystems are *mounted* onto a folder in the file tree. On Windows systems SMB-based shares appear in the folder `\\server\share-path`. Windows users also have the option of mapping a shared folder to a drive letter, turning the share into a *network drive*.

  Its important to note that both NFS and SMB protocols are focused on *file sharing*. That is, the client APIs for these protocols focus on opening / closing / reading / writing / locking / unlocking files, or regions of files. They implement filesystems. This is in contrast with the next two protocols that are *block-oriented* protocols.

- iSCSI protocol — The SCSI protocol has been around for decades. It is a block-oriented protocol that sits under filesystems to read/write/lock specific blocks on a disk. At a high level, the iSCSI protocol is simply the SCSI protocol encapsulated in IP packets (its really more than that, but for purpose of this discussion, its sufficient). Administrators of iSCSI capable storage solutions allocate space on one of the virtual disks and assign it a iSCSI LUN (logical unit number). A client computer with an iSCSI device driver can then access the iSCSI LUN as a network disk, create a filesystem on it, mount the filesystem, etc.

  It is important to note that SCSI (and by extension iSCSI and FC SCSI) protocols allow something called *multi-initiator mode*. When SCSI devices (HBAs and targets) operate in this mode, multiple HBAs can send commands (read/write/seek/etc.) to target devices (typically disks), which then respond to the HBA that sent the command. This can extend to HBAs in different host systems.

  That said, it is important that the management of the space on the disk be coordinated. This coordination must be done at the filesystem level in the OS running on the host system (or VM). For example, two VMware ESXi systems can share a SCSI / iSCSI / FC SCSI LUN with a VMFS filesystem on it because the VMFS code coordinates access to the shared device. However, two Linux systems cannot share an iSCSI LUN with an EXT3 filesystem on it because Linux does not coordinate management of EXT3 filesystems. That is, each Linux OS is oblivious to the actions of the other sharing the filesystem space.

Virtual Disks for VMs (VMDKs)

Another form of storage virtualization is the *virtual disk*(s) assigned to each VM. VMware calls these *vmdks* because the files that house them all end with `.vmdk`.

Recall for the discussion of Server Virtualization that each VM's configuration is (typically) contained with a VM-specific folder, and that configuration includes one or more virtual disks, which typically) consists of a meta-data file plus one or more flat files within that folder. Some hypervisors that support the *snapshot* feature keep one additional flat-file, called a *delta disk*, per snapshot on the VMDK.

Shared Storage and Server Virtualization

The most important reason virtual storage is important to server virtualization is that external storage can be shared between hosts. This supports efficient clustering. Remember that many virtualization vendors allow migration of VMs amongst a collection of hypervisors. This can be done much more quickly with VMs whose files (virtual disks, memory snapshots, etc.) reside on shared storage than it can with VMs whose files reside on local storage.

Most higher-end storage solutions enable sophisticated features including:

- Non-disruptive data migration — This type of feature enables administrators to move virtual disks, or files on virtual disks, between storage chassis without the hypervisor pausing VMs that are using the virtual disk. This is useful when upgrading storage hardware. It can also be used to locate / relocate storage so that the virtual disks with the highest performance requirements reside on the hardware with the highest performance. Customers that have storage solutions that do not offer this feature may be able to use a similar feature in their server virtualization solution, for example VMware's vMotion.
- Replication — For disaster preparation and recovery, businesses need to have their data replicated in a location separate from their main data center. Ideally, there is enough physical distance between the two sites that a single catastrophic event (such as a typhoon / hurricane, earthquake, or large-scale power outage) will not effect both sites. Reliably replicating data over such long distances without impacting performance in the primary data center often requires special techniques, for example *asynchronous mirroring* instead of *synchronous mirroring*. Some storage vendors provide this capability in their solutions. Some virtualization vendors offer their own solutions. Some virtualization vendors offer solutions that integrate with products from storage vendors.

Defining Benefits of Storage Virtualization

Today, storage is almost always virtualized because the benefits are vast, including:

- Reliability — RAID 1, 5, and 6 all provide reliability, allowing a virtual disk to lose a single (RAID 1 and 5) or multiple (RAID 6) physical disks and continue to function. Further, after the failed disk is replaced, the RAID software can rebuild the virtual disk with the data that was on the failed disk. Even servers used as bare metal hypervisors with shared storage typically have dual internal hard disks configured in a RAID 1 mirror.
- Performance — Mirrors provide faster read performance than non-mirrored storage. This is because the reads can be spread across the devices. Other levels of RAID (except RAID 0) actually cost performance, with RAID 5 and 6 having the biggest performance hits, respectively.

  Note RAID technologies can be combined in certain ways to combine performance and reliability. For example RAID 10 involves mirroring two sets of stripped disks. That is, a RAID 1 mirror over RAID 0 stripes.

  That said, shared storage solutions may provide better performance than local storage solutions. With data spread across several physical devices, each transferring at 6Gb/s (with SAS or SATA III), and efficient caching, I/O speeds become bound by the connections between the host systems and the storage chassis. Today, common interconnect choices include: 10GigE, 100GigE, Fibre Channel - ranging from 1Gb/s to 16 Gb/s, and FCOE - uses the Fibre Channel protocol over a (modified) Ethernet fabric. Each technology has its own set of benefits and issues.
- Shared Storage for Clusters — As mentioned in the preceding section, the concept that a VM has virtual disks allows for migration of the VMs by moving their virtual configuration, including their virtual disks, from one host system to another. This enables load balancing in a cluster.

It is important to understand that the different RAID levels have different performance and reliability tradeoffs. Administrators (should) choose the configuration that best suits their needs.

Understanding Network Virtualization

Network Virtualization is the faithful representation of physical networks in software (and sometimes firmware). This course has already defined and repeatedly referred to the idea of virtualized NICs (vNICs). Other network components that can be virtualized include: switches (vSwitches), routers (vRouters), and firewalls (vFirewalls). Most of these concepts emerged with the advent of server virtualization.

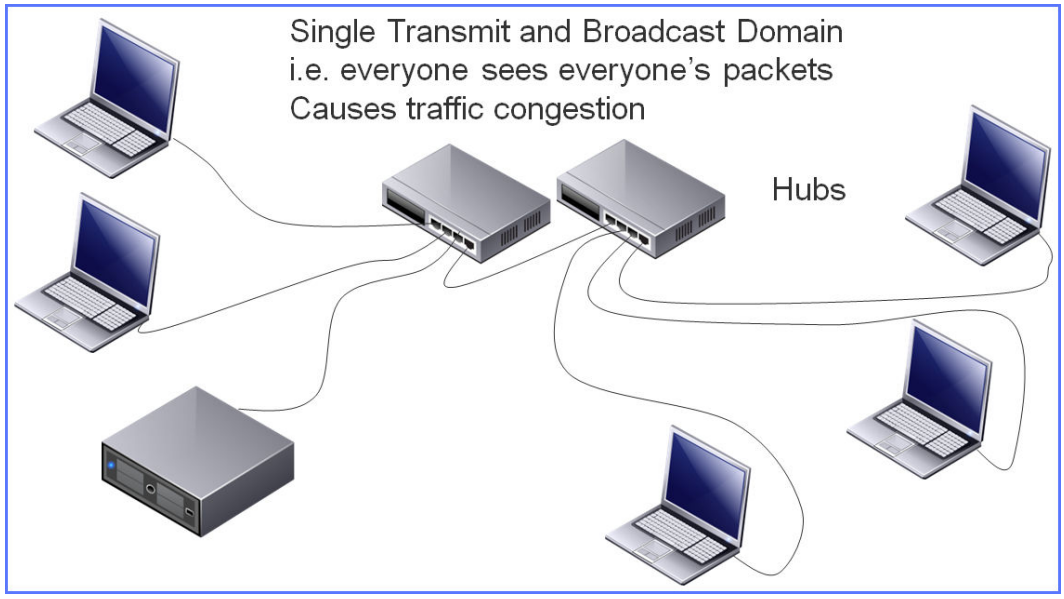
Outside of server virtualization, network virtualization has been occurring for decades, with the most prominent example being *virtual local area networks* (*VLANs*). There have been other network virtualization technologies outside of server virtualization, though, including (but not limited to): layer 2 and layer 3 *virtual private networks* (*VPNs*), virtual routing and forwarding (*VRFs*), *virtual firewalls*, and *VXLANs*.

This section, and its sub-sections, discusses each of these network virtualization technologies because they are vital to understanding *software defined networking* (*SDN*), which uses many of these technologies, and is the hottest thing both in networking and in virtualization today.

Basics of Ethernet, Hubs and Switches and Their Issues

The most basic Ethernet network contains just a few network nodes and hubs linked together, as shown in the following figure:

Figure 2-14. Basic Ethernet LAN with Hubs



While this is functional when traffic requirements are relatively low, the use of hubs is non-optimal as follows:

- Remember that Ethernet uses *carrier sense, multiple access, with collision detection* (*CSMACD*) protocol for media access. That is, before any node sends a packet: A) It listens to see if another node is transmitting, if so, it waits, if not, it starts transmitting; B) As it transmits, it listens for its own packet on the wire, which is seen by all other nodes on the wire; C) If the packet is garbled, it is (typically) because of a collision (another node transmitted at the same time), in which case the sender waits some amount of time and tries again. .
- Because hubs repeat traffic received on any port to all other ports, all nodes on all hubs see each other's unicast and broadcast packets. That is all nodes in this scenario are said to be in the same *transmit domain* and *broadcast domain*
- As traffic increases, collisions increase, causing more delays in packet transmission and more wasted bandwidth.

Note Examples of broadcast packets include ARP requests and gratuitous ARP packets.

The simplest solution to this is to replace the network hubs with network switches. The resulting network looks the same as the preceding one, except the hubs are now switches. Switches differ from hubs as follows:

- When a node connects to the network, the switch learns its MAC address.

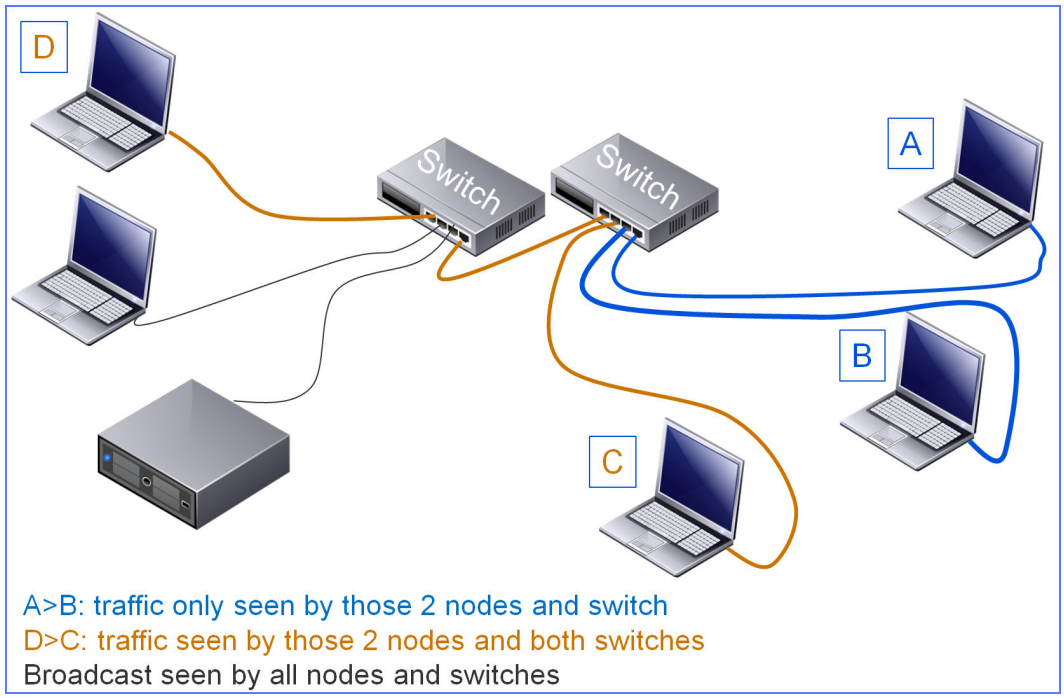
- The switch builds a table of which MAC addresses are connected to which ports.
- When a node sends a unicast packet to another node, the switch: A) Stores the packet; B) Looks in its table to determine port to send it out; C) Sends the packet to the appropriate port. By not sending it to all ports (as a hub would), only the source and destination see the packet.

Switches have the following effect on collision domains:

- For unicast packets:
  - Increases the number of collision domains, one per node / switch-port pair
  - Decreases the size of each collision domain to just the node and the switch. The possibility of a collision between the node and switch-port is eliminated entirely if the switch-port and node's Ethernet interfaces are put into *full-duplex* mode.
- For broadcast packets: There is no difference

The following figure annotates the previous figure to illustrate the isolation of collision domains:

Figure 2-15. Basic LAN with Switches



This figure illustrates that unicast packets between nodes A and B are only be seen by those nodes and their common switch. Similarly, unicast packets between nodes C and D are only seen by those nodes and both switches (since they are connected to different switches). However, a broadcast packet (for example, from the server node) will be seen by all nodes, A-D and the unlabeled node, plus both switches.

For simplicity, this section has thus far avoided discussing the handling of multicast packets. How multicast packets are handled depends on the type of switch one has and configuration options. Briefly, there are two kinds of switches:

- *Unmanaged* — These devices cannot be configured in any way. You just power them up and plug in your nodes. They implement the default protocol behaviors. Thus, they are typically less expensive than managed switches from the same manufacturer with the same number of ports and speeds.

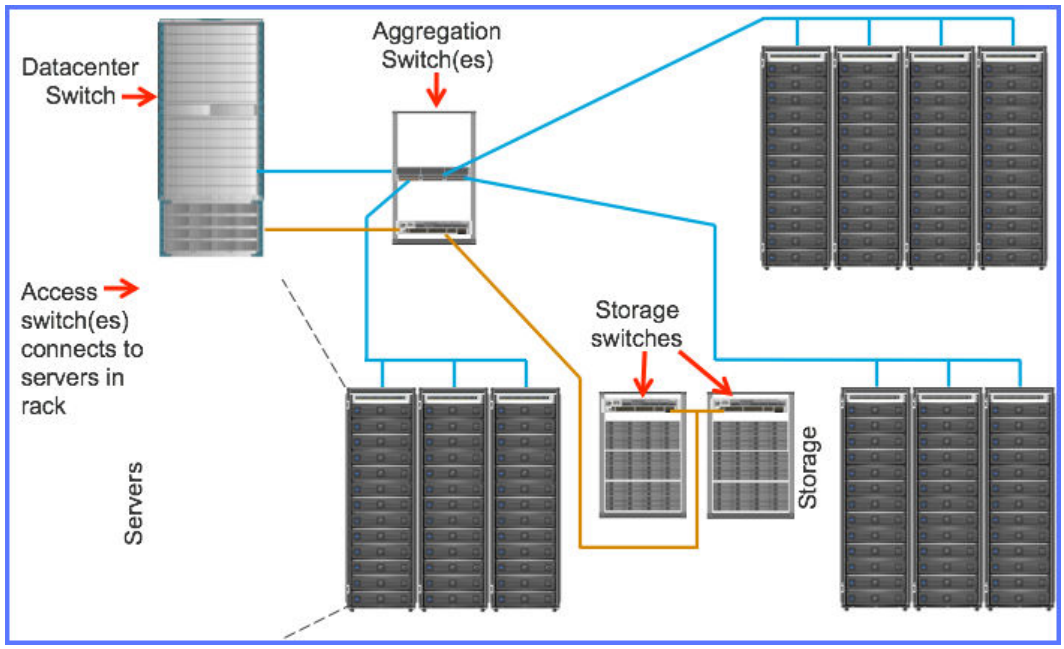
  On unmanaged switches, multicast packets have the same impact as broadcast packets. In other words, unmanaged switches send multicast packets to all their ports, including the uplink port. Unmanaged switches connected to the uplink ports cascade the multicast packets to other switches. If the cascaded switches are also unmanaged, they too have to broadcast multicast packets, etc. This can generate alot of traffic where little traffic is actually needed.

- *Managed* — These devices are configurable via a management UI that is typically either a CLI or web pages. How you access the management UI varies by make and model. The key is that you can configure things like changing the MTU on ports (perhaps enabling *jumbo frames*), pinning the duplex on a port instead of having it auto-negotiate, and configure various switch protocols such as *spanning tree* and *inter-gateway multicast protocol* (*IGMP*).

  Managed switches with the IGMP enabled only send multicast packets to ports with nodes that are members of the multicast group specified in the destination field of the Ethernet packet.

As networks grow, network administrators typically create hierarchies of switches with *access switches* connecting to workstations and servers, then those switches connecting to *aggregation switches*. In datacenters with rows of racks, the aggregation switches in turn connect to *datacenter switches* such as the Cisco Nexus 7000, Juniper EX8200 or Brocade VDX products (all of which, perhaps not coincidentally, are *unified fabric switches* – mixing Fibre Channel and Ethernet in the same switch or *Fibre Channel over Ethernet* (*FCOE*)). The following figure illustrates this hierarchy:

Figure 2-16. Switch Hierarchy



This figure shows racks of servers and storage. Each rack has a switch at the top that connects to the devices within the rack. Switches located here are known as *top of rack* or *TOR* switches because of their location. Most datacenters use *access* switches as their TOR devices, as *access* switches, by definition, provide access to endpoints, such as servers. The figure uses blue lines to indicate Ethernet connections and orange to indicate Fibre connections.

TOR switches allow network administrators to just run uplink/downlink connections between the racks. The uplink/downlink at the end of each row is then connected to an another switch, typically an *aggregation* switch, which is designed to aggregate packets across a large number of other switches. Large data centers historically have *aggregation* switches for both TCP/IP and fibre channel networks. (the entire network of a given media: Ethernet or Fibre, is also commonly referred to as a *fabric*). For simplicity, the preceding figure shows only one aggregation switch for each of TCP/IP and Fibre Channel. The number of aggregation switches is driven by the maximum number of ports, the number of TOR switch connections needed, and the maximum throughput the switch is capable of generating.

The figure also shows the aggregation switches connecting to the datacenter switch, which can connect and switch both Fibre and Ethernet packets. Datacenters typically have at least two such switches for high availability. Modern DC switches have throughput on the order of 100 terabits / second.

Issues for Switches in Large Datacenters

Extrapolate the preceding figure to a very large network with thousands of nodes and tens, or even hundreds, of switches. This leads to two issues:

- If all the switches were in the same broadcast domain, performance would become unacceptable. The solution to this problem is to divide the single layer 2 network into multiple logical layer 2 networks using VLANs (see "Understanding VLANs, Their Benefits and Issues," on page 46).
- Administering each switch, whether for software updates, QOS or ACL changes, etc. becomes cumbersome. Even if all access-level switches are of the same type and require the same configuration (not likely), the complexity of keeping all switches updated and configured properly grows almost exponentially with the number of switches. Many switch vendors offer management solutions that simplify this complexity, though they don't eliminate it.

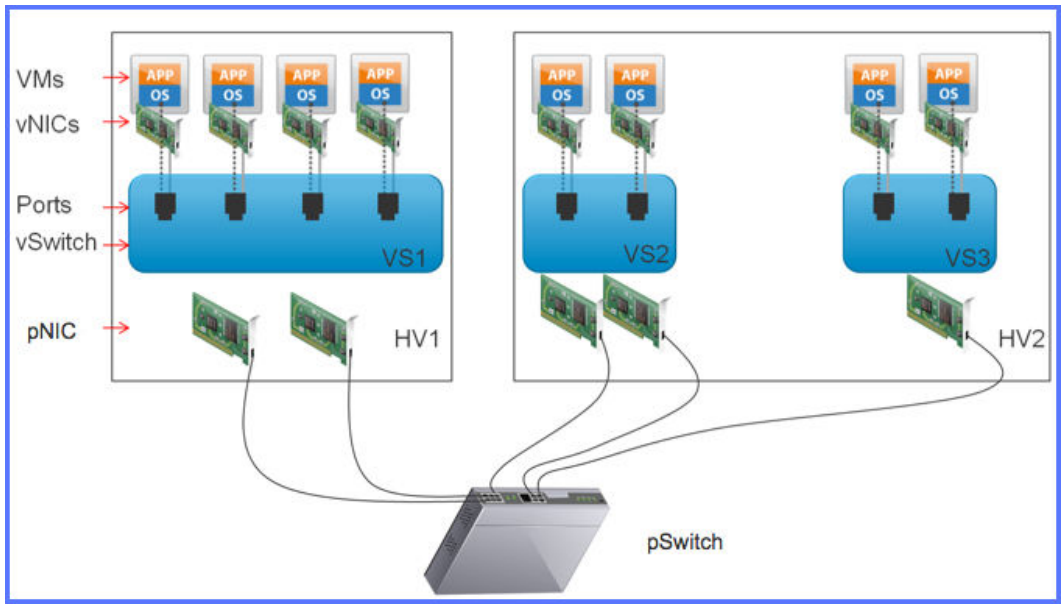
Understanding Virtual Switches (vSwitches), NIC Teaming, Their Benefits and Issues

In networking, a *virtual switch* (*vSwitch*) is software that emulates the behavior of a physical network switch (pSwitch). They are typically implemented as part of a hypervisor and have the following properties:

- They have some number of *virtual ports* analogous to physical ports. Each virtual port is logically connected to a virtual NIC (*vNIC*) of a VM.
- They may allow administrators to organize the ports into port groups, with certain properties applied to the port group, such as VLAN tagging (see "Understanding VLANs, Their Benefits and Issues," on page 46). These *virtual port groups* are analogous to port groups on pSwitches.
- They typically have some number of physical NICs (*pNICs*) assigned to them
- They typically cannot be connected directly together. That is, there is no concept of uplink / downlink ports for vSwitches.

The following figure illustrates example uses of vSwitches:

Figure 2-17. Example vSwitch Usage in Hypervisors



This figure shows two hypervisors, HV1 and HV2. HV1 has one vSwitch, VS1, with four ports, each with a VM's vNIC attached, and two pNICs. VS2, like VS1, has two pNICs but only two ports with VM's attached. VS3 also has 2 VMs attached to its two ports, but only one pNIC. The traffic for both VMs attached to VS3 must go through the same pNIC. If that pNIC fails, both of the VMs attached to VS3 become unreachable to the VMs attached to other vSwitches, though they can continue to talk to each other.

When multiple pNICs are attached to the same vSwitch, such as in VS1 and VS2, VMware calls them a *team* or *NIC team*. The pNICS are said to be doing *NIC teaming*. NIC Teaming is a form of *link aggregation* and implements the IEEE 802.1ax protocol. In this case the administrator must configure the properties of the NIC team, including:

- Whether to use them for fail-over only, or load-balancing. Load balancing implies HA as losing a NIC still allows traffic to flow in a degraded mode through the remaining NICs.
- If load-balancing, the hashing algorithm to use to determine which pNIC receives the outbound traffic.
- If fail-over only, the administrator can also configure the following:
  - The method used to detect failure
  - Whether to do fail-back
  - Whether to notify the pSwitch to which the pNIC is connected on failure / recovery of a teamed pNIC

Benefits of vSwitches

There are many benefits to vSwitches

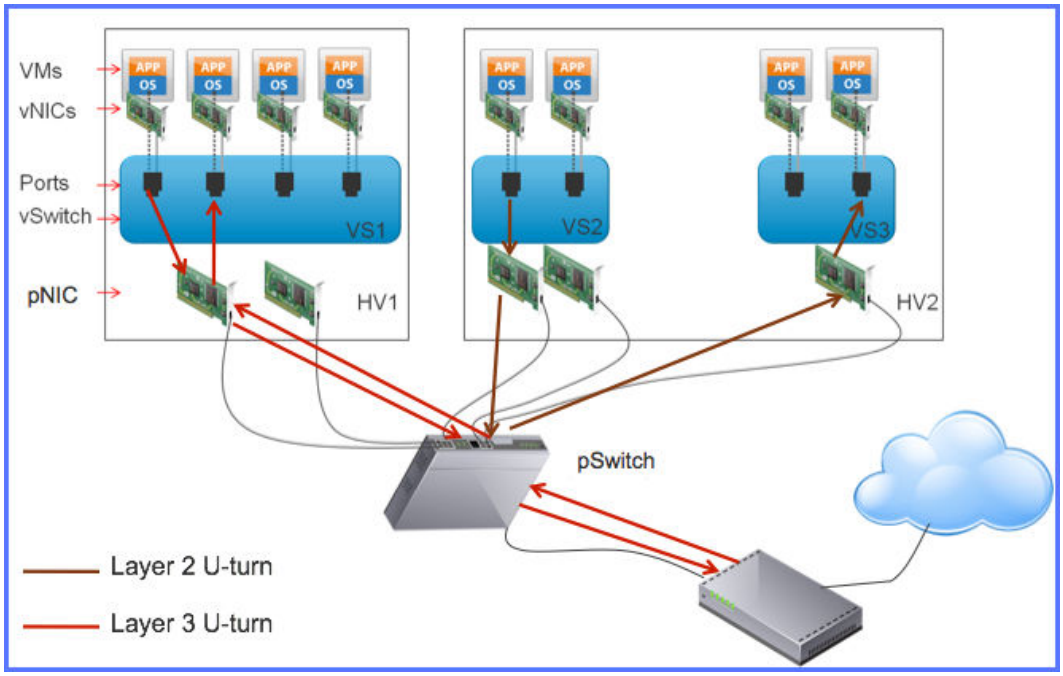
- They allow N:m mapping of VMs' vNICs and pNICs
- They provide high throughput for VM-to-VM traffic for VMs attached to the same vSwitch
- NIC-teaming provides HA and scalability for traffic headed to the pSwitch

Issues with vSwitches

While vSwitches have many benefits, they have some issues, including

- The base vSwitches included in most vendor's Type I hypervisors have few management features. The vSwitches in most Type II hypervisors are analogous to unmanaged switches. As a virtualized environment grows, network administrators need vSwitches to have the many of the same management features that are available on higher-end managed switches.
- The VMs located on a given host system are determined both by organizational design and by the host system's maximum resources. It is reasonable to expect that virtualized environments may require several host systems to process demand for their services. It is as problematic to configure a number of independent vSwitches as it is to configure a number of independent pSwitches, even if they all require the same configuration.
- Without layer2 or layer3 intelligence built into the hypervisor to do inter-vSwitch uplink/downlink or layer 3 routing, the packets from VMs may need to go out to a physical switch or router only to get turned around and headed back to the same host system or even vSwitch, respectively. Generically, this term is called *u-turning* or *hair-pinning* traffic, as shown in the following figure:

Figure 2-18. Hair-pinning Traffic



There can be several causes for U-turning traffic. In the preceding figure, consider that the two left-most VMs in VS1 have vNICs in separate IP subnets. Given the resources shown, the only way for the packets to get routed between the two VMs is to the router. This is a long path to take for packets that would ideally just get passed directly from one vNIC to the other. Similarly, consider that the VMs in HV2 are all on the same IP subnet (and same VLAN). Given the resources shown, the only way for packets to go from one vSwitch to another is via the pSwitch.

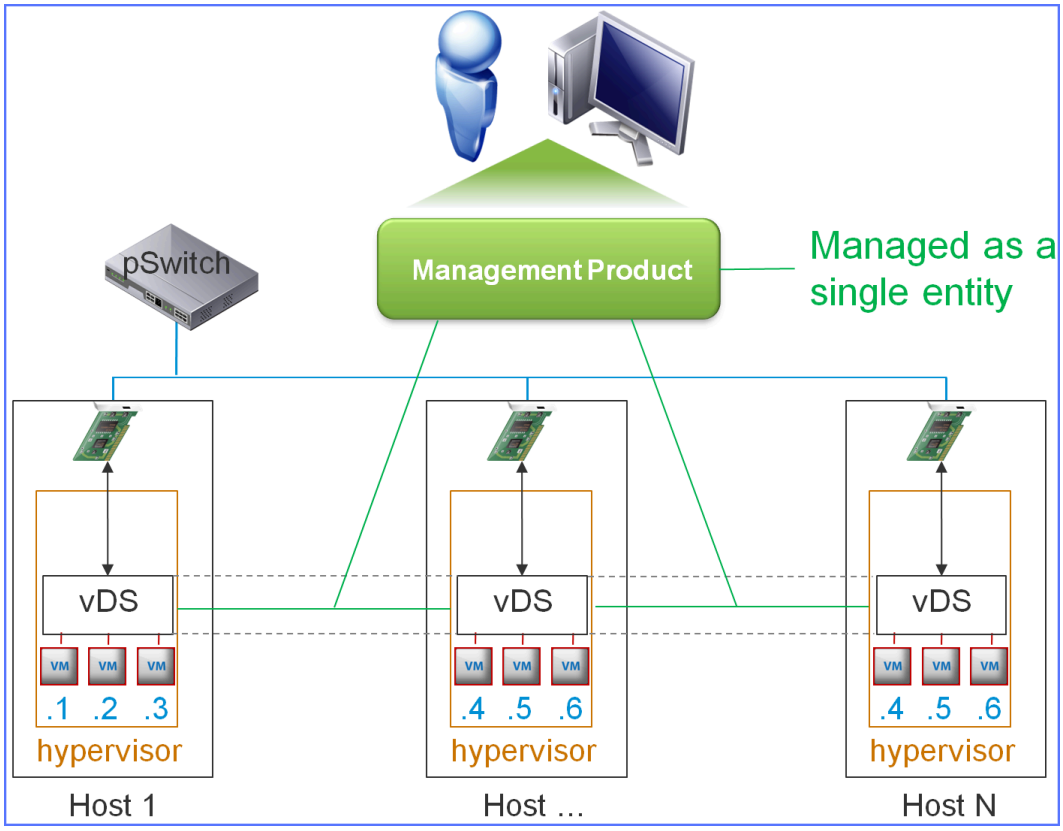
Note In the latter case, one solution to requiring the packets to go through the pSwitch is to create a VM with vNICs in both vSwitches (VS2 and VS3), where the vNICs are bridged in the guest OS, creating a virtual uplink/downlink. There are other solutions as well.

Understanding Virtual Distributed Switches (vDS) and Their Benefits

Virtual distributed switches (vDSs) are designed to overcome two of the issues with standard vSwitches:

- Limited management features — vDSs contain most, if not all, of the features found in managed pSwitches, including the ability to configure QOS policies, ACLs, etc.
- Complex management — vDSs have presence that spans multiple hypervisors. That is, an administrator can create a single vDS, then *distribute* it to multiple hypervisors, as illustrated in the following figure:

Figure 2-19. vDS Across Hypervisors, Managed as a Single Entity



This figure shows 3 hypervisors with a vDS instance in each, management console connected to them showing they are managed as a single entity

When an administrator configures the vDS, the configuration change is implemented wherever that vDS has presence. For example, in the preceding figure, if an administrator added an ACL to the vDS, the management tool communicates that change to the hypervisors on hosts 1 through N. There is no practical limit to the number of hypervisors to which administrators can distribute a vDS.

While all hypervisors include basic vSwitches, most vendors require an advanced license to enable vDS capabilities in their products. Further, most hypervisor vendors only provide vDS management capability through their management products.

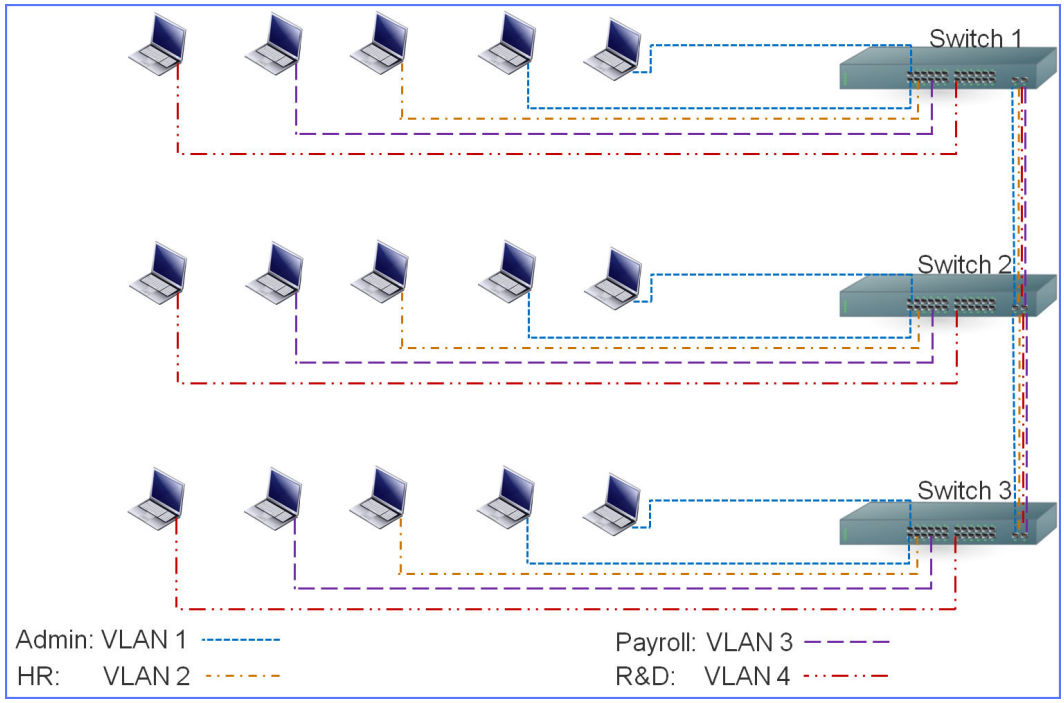
The capabilities of vDS implementations are continually growing.

Understanding VLANs, Their Benefits and Issues

The term *VLAN* means *virtual LAN*. VLANs partition a LAN into smaller, virtual LANs, each of which acts as its own Ethernet segment and thus has its own broadcast domain. Within any given LAN, the VLAN protocol allows up to 4096 VLANs, each of which are identified by an integer called the *VLAN ID*. With VLANs in use, both unicast and broadcast layer 2 addresses (and thus layer 2 packets) are scoped to a VLAN ID. That is, nodes within VLAN X can only communicate directly with other nodes within VLAN X, including other switches and routers. Typically, each VLAN has an IP subnet associated with it. These two previous points mean that trans-VLAN traffic requires a layer-3 router.

VLANs can only exist on networks with managed switches. This is because VLANs are created by having network administrators assign each port (or port-group) a VLAN ID. By default, on a managed switch, all ports belong to VLAN ID 0. Typically, network administrators use VLANs to partition the network by organizational functionality. The following figure illustrates VLAN these and other concepts:

Figure 2-20. Basic LAN with VLANs



This figure shows three switches, each of which is configured with four VLANs (indicated by different colors and line-dash patterns) with IDs 1 through 4. The administrator decided to assign VLAN1 to nodes in the ADMIN department, VLAN2 to nodes in the HR department, VLAN3 to nodes in the Payroll department, and VLAN4 to nodes in the R&D department. In this scenario, the nodes in the Admin group on each switch can communicate without sending traffic on the switch interconnect ports.

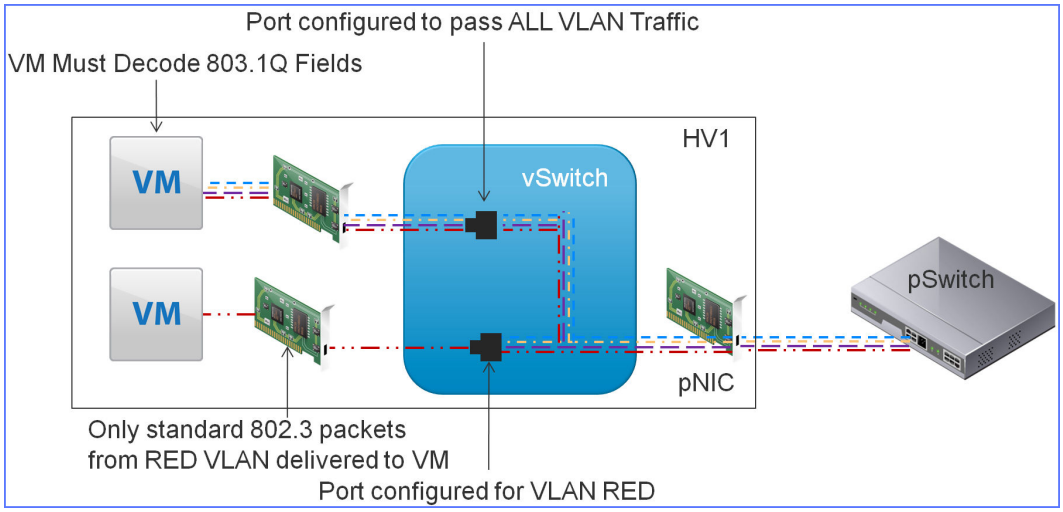
As soon as you configure multiple VLANs on a switch, on packet ingress (receipt of an standard 802.3 Ethernet packet), the switch automatically inserts an IEEE 802.1Q field into the Ethernet packet, which includes the VLAN tag associated with the switch port. This is known as *VLAN tagging*. The exception to this is if the port belongs to the *native VLAN* (whose VLAN ID is configurable). On packet egress, for non-native-VLAN packets, the switch automatically removes the IEEE 802.1Q field, which results in the destination node receiving a standard 802.3 Ethernet packet.

However, there are times when the packet should remain in 802.1Q format on egress. This is known as *port trunking*. Administrators must configure which ports do port(s) or port group(s) do port trunking. Said ports are called thus *trunk ports*. Examples of reasons to create trunk ports include:

- Inter-switch packets on the link ports — As shown in the preceding figure, the traffic traversing uplink / downlink ports must keep their VLAN ID so that the other switches know to which VLAN the packets belong.
- Ports connecting to a hypervisor with VMs in multiple VLANs — If a hypervisor is running VMs that belong in disparate VLANs, they must receive packets that include the VLAN tag. In that case, the VLAN tag can be removed in one of two places:
  - The vSwitch (see "Understanding Virtual Switches (vSwitches), NIC Teaming, Their Benefits and Issues," on page 43) — If the administrator configures the vSwitch port or port-group with a VLAN tag, the hypervisor's vSwitch code removes the 802.1Q field and presents the vNIC with a standard 802.3 Ethernet packet.
  - In the guest OS — If the administrator does not configure a VLAN tag for the port / port group that receives 802.1Q traffic, then the hypervisor's vSwitch code passes the packet to the vNIC of the guest OS unchanged,. This means that the guest OS must understand 802.1Q packets and decode them itself.

The following figure illustrates these two scenarios:

Figure 2-21. Port Trunking for Hypervisors



This figure shows the following:

- A pSwitch with port trunking configured on the port that connects to a pNIC in hypervisor HV1. Thus, all VLAN traffic is passed into the vSwitch as 802.1Q packets.
- The lower port on the vSwitch is configured on VLAN RED. Thus, the vSwitch only passes traffic on the RED VLAN to its vNIC, passing the packets as 802.3 (standard Ethernet) packets. The VM receives and processes standard Ethernet packets.
- The upper port on the vSwitch is configured to pass all VLAN packets. Thus, the ports vNIC and associated VM receive 802.1Q packets. The VM must decode the 802.1Q packets itself.

Benefits of VLANs

The preceding section mentioned some of the benefits of VLANs as part of their definition. But to recap, the benefits of VLANs include:

- Limiting the broadcast domain of layer 2 traffic
- Limiting the failure domain of layer 2 traffic, for example broadcast storms
- Transparency in management tools
- Traffic organized by functionality instead of by geography
- Can be used in multi-tenant data centers to group tenant traffic (with a limit of 4096 VLAN IDs)

Issues with VLANs

The preceding section also mentioned some drawback of VLANs (though they may not have been presented as such), including:

- The number of VLANs is limited to 4096 for a particular layer 2 switch domain. Multi-tenant data centers typically use one VLAN / tenant, with some tenants even requiring multiple VLANs. In reasonably large multi-tenant datacenters, this is not enough.
- Routing between VLANs requires layer 3 switch or router
- Deploying VLANs requires configuration of each switch (and possibly routers) over which the traffic flows. In a cloud datacenter, this may require coordination between the three administrative groups: network, cloud, and tenant. Alternatively, the datacenter can use the *vlan trunk protocol* (*VTP*) to deploy VLANs in an automated manner, most datacenters typically disable this protocol. (For more information on VTP, see `http://en.wikipedia.org/wiki/VLAN_Trunking_Protocol`).

- Its not possible to extend a VLAN across a layer 3 device without either special layer 3 device technology, such as *Overlay Transport Virtualization* (*OTV* see `http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DCI/whitepaper/DCI_1.html`) or VxLAN (see "Understanding VxLANs, Their Benefits and Issues," on page 49)

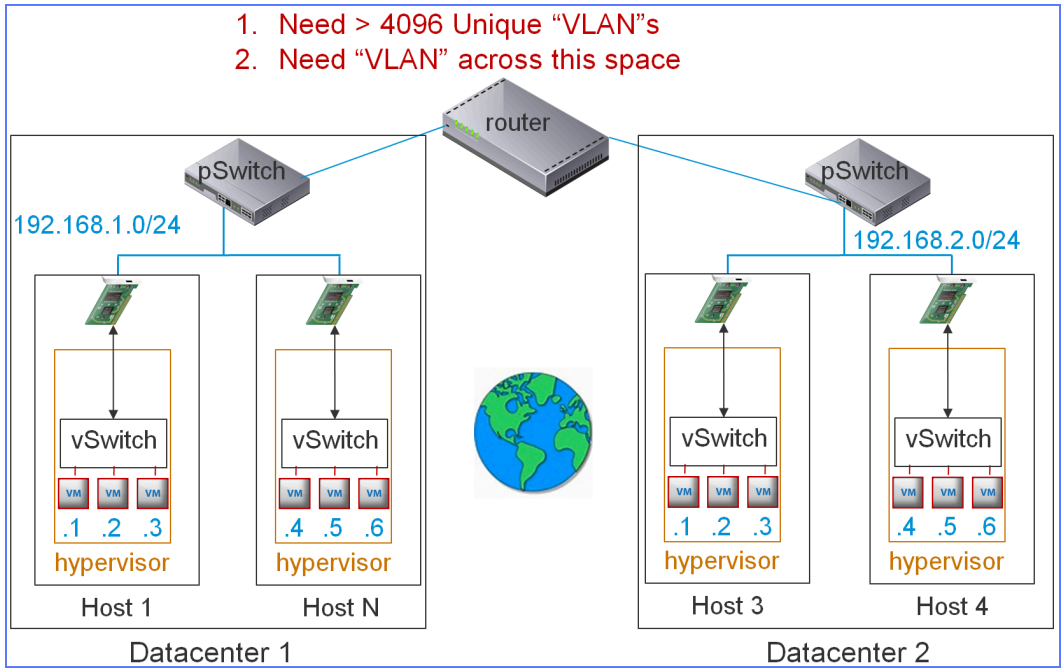
Understanding VxLANs, Their Benefits and Issues

A Brief Description of VxLAN and the Problems it Solves

VxLAN, an abbreviation for Virtual Extensible LAN, is a technology, including a network protocol, designed to overcome the limits of VLANs (see "Understanding VLANs, Their Benefits and Issues," on page 46), including a limited number of VLAN IDs (4096) and an inability to efficiently bridge geographies.

Consider the following figure:

Figure 2-22. Problems solved by VxLAN, with Sample Network



two data centers with their own IP subnets and no way to extend VLANs between them

This figure shows two datacenters, each of which (for simplicity) has a single IP subnet, which could be associated with a VLAN. In multi-tenant environments, imagine many VLANs on in each datacenter, each VLAN having its own IP subnet. As discussed in the VLAN section, there can only be 4096 VLANs in a datacenter. Further, there is no way, with the native VLAN technology, to have a VLAN in one datacenter extent to the other datacenter (excepting solutions such as layer-2 VPN (L2VPN), which is beyond the scope of this course).

The VxLAN protocol overcomes both the 4096 limit and the geography limitations. It does this by creating a virtual layer 2 network segment and associated layer 3 (IP) subnet on top of routed sub-nets. The resulting virtual network is generically called an *overlay network* and in VxLAN terminology is called a *virtual wire*.

How VxLAN Works

Suppose you wanted to create a single VxLAN that spanned the entire network depicted in the preceding figure. At a high level, to do this:

1 Determine which hosts have VMs that will exist in the new overlay network. In this scenario it would be all four hosts shown.

2 Determine the IP address of the overlay network. For this scenario, use 10.10.1.0/24.

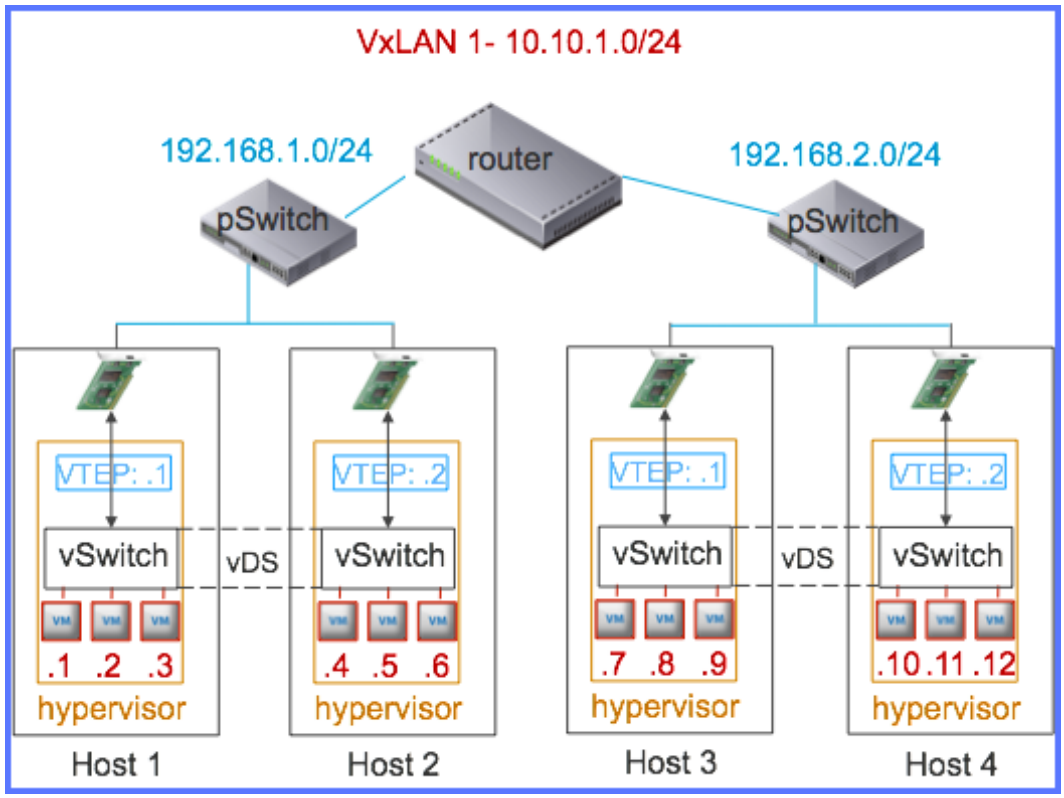
3 Install VxLAN software on each host containing VMs you wish to participate in the overlay network. At a minimum, the VxLAN software must install an ARP proxy and a*VxLAN Tunnel End Point* (*VTEP*) module. This module encapsulates packets between the local subnets and the overlay network. Each VTEP gets an IP address on the local subnet. For example, Host 1 could have 192.168.1.1, Host N: 192.168.1.2, Host 3: 192.168.2.1, Host 4: 192.168.2.2.

   The software installation must also include some management software for creating, deleting, and otherwise managing instances of virtual wires / overlay networks, and for managing which VMs have presence in which overlay networks.

4 Use the management tool to create the virtual wire and place the desired VMs on the resulting overlay network.

The following figure illustrates the results of these steps (with Host N re-labeled Host 2):

Figure 2-23. VxLAN for Simple Network



Notice that the VMs now have contiguous IP addresses from 1 to 12, and that they are color-coded to indicate that they are on VxLAN 1 with IP address 10.10.1.0/24. So, VM1 on Host 1 has IP 10.10.1.1 and VM3 on Host 4 has IP 10.10.1.12, etc. Also note that the VTEPs are color-coded to the local sub-nets. So, VTEP 1 on the left has IP 192.168.1.1 and VTEP 2 on the right has IP 192.168.2.2, etc. The VMs (in this figure) now all belong to the same layer-2 network with the same IP subnet. There are many scenarios for communication across the virtual wire. Finally, notice that each VTEP is in the data path between a host's vSwitch and the pSwitch.

To provide a high-level understanding of how VxLANs work, this course describes the transmission of a TCP packet from VM1 to VM12. (The course Getting Started Developing a NSX Solution includes a deep-dive into VxLANs in vSphere environments).

Suppose VM1 is an SMTP client and VM12 is an SMTP server. At a high level, having the SMTP client on VM1 connect to the SMTP server on VM12 and send a single packet of data involves the following steps (some of which are simplified):

Note Assume that VM1 has MAC E1E100000000 and VM2 has MAC C15AID000000. Further, assume that as each VM starts, its IP address becomes known to the VTEP on its host, and that association is shared between VTEPs via the VxLAN management software. For example, at any given time VTEP 192.168.1.1 knows that VM 7 is reachable via VTEP 192.168.2.1, etc. Finally, assume that each VTEP listens for UDP packets on port 4796, the IANA default for VxLAN, and also has a UDP source port of 20480.

1 The SMTP client makes a socket() or analogous system call to create a TCP socket. Assume that results in the SMTP client getting allocated TCP port 10240.

2 The SMTP client makes a TCP `connect()` (or analogous) system call to connect to VM12 at port 25 :

   a The kernel creates a TCP packet with source port (for this example, assume 10240), destination port (25) and a connection (SYN) request, then hands the packet to the layer-3 (IP) module

   b The IP module does an IP lookup for VM12 (DNS, hosts file, etc.) and receives 10.10.1.12

   c The IP layer creates an IP packet with the TCP packet as payload, places the source (10.10.1.1) and destination module (10.10.1.12) and any other requisite information into the IP headers, and then hands the packet to the Layer-2 module

   d The layer-2 module (which could be Ethernet or 802.1q) looks up IP 10.10.1.12 in its ARP cache. If there is no hit, it must issue an ARP request. Assume that the ARP request succeeds.

   e The layer-2 module creates an Ethernet packet with the IP packet as payload, fills in the source address of E1E100000000, destination address of C15AID000000, and other requisite header fields, and then transmits the packet

   Note The Ethernet packet resulting from the steps so far is called the *inner packet*. The headers are referred to as the *inner source MAC*, *inner destination MAC*, *inner source IP* and *inner destination IP*, respectively. It is called the *inner packet* because it gets encapsulated inside another set of layer-2 / layer-3 packets, called the *outer packet* as described in the following steps.

3 The vSwitch sees the packet and realizes it isn't addressed to any of the VMs attached to it, so pushes the packet toward the pNIC.

4 The VTEP processes the packet on its way to the pNIC:

   a The VTEP knows (or discovers) via the VxLAN management data that VM1 belongs to VxLAN Network Identifier 1 (VNI 1).

   b The VTEP examines the packet to see the destination IP, then looks in VxLAN management data for the ID of the VTEP in VNI 1 that has a VM with IP address 10.10.1.12 and discovers it is VTEP 192.168.2.2.

   c The VTEP builds a VxLAN packet with the Layer-2 packet as payload (stripping the VLAN tag, if any), setting the VNI in the VxLAN header to 1.

   d The VTEP uses a `sendto()` (or analogous) system call to send the VxLAN packet via UDP to the VTEP at 192.168.2.2, with source port 20480 and destination port 4796. This typically pushes the packet to the UDP layer in the hypervisor's kernel,

5 The hypervisor's UDP module creates a UDP packet with the VxLAN packet as the payload, fills in the source and destination ports and any other applicable headers, then pushes the packet to the hypervisor's IP module.

6 The hypervisor's IP module creates an IP packet to transmit to the VTEP at 192.168.2.2:

   a It uses the UDP packet as payload, setting the source IP to 192.168.1.1 and destination IP to 192.168.2.2, and filling in any additional header fields.

b It determines that the source and destination addresses are not on the same subnet, so it needs a route.

c It finds a route to the 192.168.2.0/24 network through the router (assume the last octet of the IP address of each router port is 253) via IP 192.168.1.253.

d It sends the IP packet to the Layer 2 module for transmission to the router.

7 The layer-2 module performs an ARP lookup on IP 192.168.1.253 to receive the MAC address of the router's port. Assume the port's MAC is B2C200000001, then creates an Ethernet packet with the IP packet as its payload, filling in the source MAC (that of the pNIC on Host 1) and destination MAC (B2C200000001) and any other requisite headers, and then transmits the packet on out the pNIC.

Note The Ethernet packet that results form this step, and the IP packet resulting from the previous step are referred to as the *outer packet* with *outer source IP*, *outer destination IP*, *outer source MAC* and *outer destination MAC* addresses.

At this point, the packet transits the router to the destination VTEP (10.10.2.2) as any IP traffic would.

8 On arrival into Host 4's pNIC, the hypervisor kernel processes the packet and delivers it to the VTEP as a UDP datagram.

9 The VTEP on Host 4 reads the VxLAN header and validates that VM 10.10.1.12 is still present. If so, it strips the VxLAN header and sends the packet through the Host 4 vSwitch to VM 12.

10 VM 12's guest OS receives the packet and delivers it to the SMTP daemon listening on port 25 (assuming no firewall issues, etc.) to process the connection.

Assuming the SMTP server accepts the connection, the TCP protocol requires the guest OS to send a SYN-ACK back to the sender. The SYN-ACK packet goes through the steps just describes, except swapping the MAC and IP addresses for the packet to go in the reverse direction.

Its important to highlight that VxLAN's overlay network is a virtual layer-2 network. This is evidenced that the Ethernet packet that results from *step 2.e* has the destination MAC of VM 12. The fact that the packet is proxied through the VTEP is invisible to both VM 1 and VM 12.

Handling ARP

The procedure in the preceding section assumed that VM 1 knew the MAC address of VM 12, that it was already in VM 1's ARP cache. The first time VM 1 attempts to communicate with VM 12, this will not be true. So, how does VM 1 get the MAC address of VM 12 in a VxLAN environment?

The IETF specification for VxLAN suggests the use of multicast. At a high level:

- When the first VM belonging to a given VNI starts / resumes / moves in, the VTEP for that host uses an multicast group associated with the VNI (the mapping for each VNI is specified by the VxLAN management component and sent to each VTEP as the VNIs are created, so the VTEPs know which multicast to join)
- When the VM performs an ARP:

  a The VTEP code sees the packet is a broadcast ARP request.

  b The VTEP creates a multicast packet analogous to an ARP request and sends it to all VTEPs in the multicast group associated with the VNI of the vNIC that sent the ARP request

  c The multicast packet must get transmitted to all the other VTEPs in the VNI's multicast group

  d At most one of the VTEPs will know the MAC address of the requested IP. That VTEP will respond with a unicast packet to the requesting VTEP

  e The requesting VTEP generates an ARP response to send to the requesting VM

The IETF draft document for VxLAN acknowledges that using multicast is potentially undesirable, leaving the door open for other solutions, provided they are compatible with the VTEP sending out multicast packets for the request and receiving unicast responses. Version 5.5 of NSX on ESXi has an alternative solution that provides an alternative option that does not rely on routers being configured for multicast, and forwarding multicast traffic beyond the local subnet. Further discussion of the NSX solution is beyond the scope of this class.

Benefits of VxLAN

VxLAN implementations provide several benefits, including:

- The ability to have a (virtual) layer-2 network that spans (overlays) layer-3 subnets. This itself has many benefits. In a virtualized environment, one of the biggest is that a VM can be moved from one subnet / datacenter to another without having to change its IP address and without having to resort to using mobile IP protocols such as LISP (see `http://en.wikipedia.org/wiki/Locator/Identifier_Separation_Protocol`)
- The ability to have over 16 million virtual wires (VNIs) within a single virtualized environment. This is especially beneficial to multi-tenant data centers, who can now provide one VNI / tenant to keep their data separated.

VxLAN is a key component in enabling software-defined networking and software defined data centers. It is also key for cloud-based service providers.

Acknowledging Virtual Private Networks (VPNs) and Other Virtual Networking Concepts

The preceding topics in this section of the course have discussed virtual networking concepts that developers are likely to encounter in today's virtualized environments. There is one other technology that developers likely encounter, is typically well understood (like Ethernet basics), but is not part of the narrative related to VxLANs: Virtual Private Networks (VPNs).

Briefly, Virtual Private Network technology allows the connection of separate private endpoints or networks, via a public network. The key to keeping the data on the network private is to encrypt the data before it leaves the private endpoint or private network so that entities on the public network cannot see the actual data exchanged between the networks / endpoints. There are several different VPN products and schemes including (but not limited to):

- Endpoint VPN — RSA, Cisco and others have VPN products that allow people to attach their workstation to an organization's network. Microsoft and Apple also have VPN components that people can use. These have historically been technology OEMed from one of the main technology providers.

  The organization must provide a VPN server to which the endpoint VPN clients connect. The server must issue the client with an IP address that is on the organization's network. Most VPN servers thus with a DHCP and dynamic DNS server, and may include NAT capability.

  Historically, this has been a hardware device behind the customer edge router (for example a Cisco PIX® firewall). More recently, vendors have developed virtual appliances to provide this functionality. Most server operating systems also offer a software feature to provide this functionality.
- Layer 2 VPN (L2VPN) — This technology is used to connect two networks at layer-2. Most networking companies provide L2VPN capabilities in their customer edge products.

Note Some VPN protocols do not include encryption. One example is Multiprotocol Label Switching (MPLS) VPN. Such protocols assume that encryption is the responsibility of some a component higher in the networking stack.

There are other virtual networking concepts that exist but are rarely, if ever, relevant to software development in a virtualized environment. However, in the interest of completeness, it is appropriate to mention them here:

- Virtual Routing and Forwarding (VRF) — This is a technology that allows a single router to have multiple routing tables, in essence acting as two virtual routers. For more on this topic, see `http://en.wikipedia.org/wiki/VRF`.
- Secure Domain Router (SDR) — This is a technology created by Cisco to physically partition a single CRS router into multiple logical routers. These were originally called *logical routers* (*LRs*).
- Virtual Device Context (VDC) — This is a technology created by Cisco to have their Nexus 7000 series switches appear as multiple logical switches.

Understanding "Cloud"

The term *cloud* seems omnipresent in IT discussions today, for example cloud-based storage for backups, cloud-based computing, cloud-based applications, etc. While some people understand *cloud* well, many people misunderstand it plenty. This section, and its sub-sections, define the term *cloud* and *cloud-based service* types.

While some people claim that anything that uses the Internet, including traditional web applications, are *cloud-based services*, and while the industry has not created a generally accepted scientific definition of the term, this course and all courses in the VMware Certified Developer use the following definition for things that qualify as *cloud-based services*:

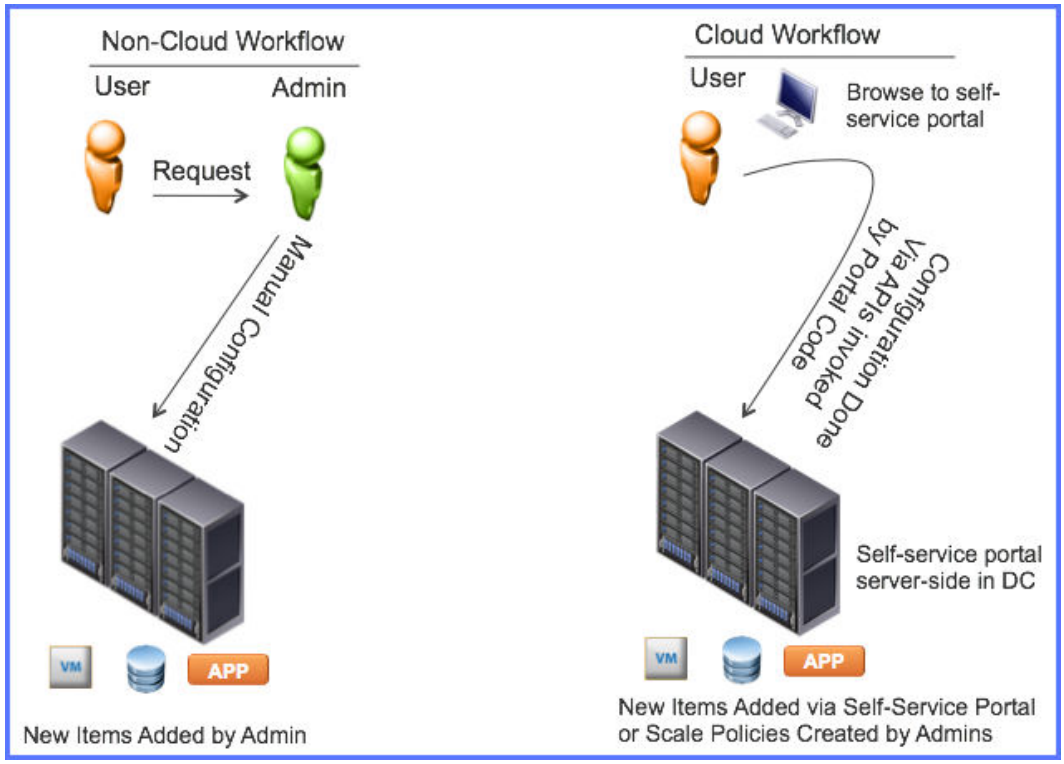
Cloud services

These are computer-related services that are:

- Provisioned *on demand*, via a web browser or API, without the need of human intervention (beyond the requestor) to instantiate. Another way to say this is that the service must be provisioned via *self-service portals*.
- Instantiated on virtualized technology (for example involving VMs, VxLANs or VLANs, virtual storage, etc.)

The following figure illustrates the differences between cloud and non-cloud based services:

Figure 2-24. Cloud and Non-Cloud Workflows



The business model of the service provider is irrelevant to the definition. It may be a for-profit organization offering the services to the world, an IT department within an organization that is considered a cost-center for accounting, or any combination thereof. The types of services offered is also irrelevant to the definition. They could be raw computing services (CPU, RAM, storage), desktop operating environments, development platforms, or complete applications. The latter can be hard to distinguish from traditional Web Applications, but there is a distinction.

Here are some well know examples of cloud-based services:

- Zoho Office Suite / Google Apps — These products (and their competitors) are office productivity tools that you use your browser to access and that run in web sites maintained by their respective vendors. You create an account at the vendors's site (some of which are free), then start creating documents. No humans are involved in provisioning your account, etc.
- Salesforce.com — Manages many aspects of customer relationships, including sales and support, all in a browser with web apps in the business's site, servers at the Salesforce site, and multi-tenancy for data protection.
- Surveymonkey.com — Allows you to create surveys and send invitations for people to take them, either from a sample provided by Surveymonkey, or from a list of email addresses you supply.
- Cloud Foundry by Pivotal Labs — This product (and its competitors) allow you to develop cloud-based software, in the cloud.
- A website on an organization's intranet that allows you to allocate a virtual desktop instance, specifying the desktop OS and select any (standardized) applications you may need. Again, this is only a *cloud* service if the virtual desktop is created and the applications are installed and all is licensed automatically, without the need to involve corporate IT, network management, etc.

Here are some examples of Inter/Intranet-based services that do not qualify as *cloud*, given the preceding definition:

- XYZ Accounting Solution running on an organization's IT equipment on their Intranet. Regardless of whether you access them from a web browser, there is no automated provisioning via self-service portals, etc. in a typical deployment of accounting software. Rather, this is "just" a Web App.
- Email running on an organization's Intranet. This is still just a web-app. One can argue that free Email service (for example Yahoo! / Google / Hotmail are cloud-based Email, since the account provisioning is automated.

Benefits of Cloud-Based Services

Wikipedia's discussion of Cloud Computing (see `http://en.wikipedia.org/wiki/Cloud_computing`) provides a list of other characteristics typically exhibited by cloud-based services, which you can also view as a list of benefits, including:

- Scalability — This term has several meanings. From a users perspective, the performance of the service should not degrade as other users' provision themselves into the system. Further, the provisioning portal may allow users to add resources to the service to improve performance as the user's demand increases, and to subtract them as their demands decrease. This ability to add and remove resources, ideally in an automated manner, is also called *elasticity*. From a provider perspective, they should ideally be able to add performance capacity and coverage by throwing more and faster servers, storage, and network capacity at the datacenter. The underlying virtualization software should be able to take advantage of the new resources with little or no other administrative action beyond adding the resources to the environment.
- Multitenancy — This means that users should not see, or be able to see, each other. Each user's data is private. For public offerings, where users are customers, this is critical. For example, if two competitors happen to choose the same cloud service for their accounting software, visibility into each other's data could be a disaster for either customer, and would definitely be a disaster for the service provider.
- Maintenance — Using cloud-based services offloads much of the IT maintenance from the users. For example, if a disk dies in a data center used by a service provider, the data center administrator should have the data protected by redundant technology, such as those discussed in "Understanding Storage Virtualization," on page 35, such that the service provider just has someone replace the disk and the user never notices.

There are other purported benefits of cloud-based services that are still debated. Two of the most debated issues are cost and security:

- Cost — Cloud-based services have little or no capital expenses (CAPEX) for users. One only needs a browser-enabled device to access the services. Further, the operational expenses (OPEX) of maintaining the infrastructure is not born (directly) by the customer. However, as the saying goes, "There ain't no such thing as a free lunch" (TANSTAAFL - Robert Heinlein). The cloud operator must recover their costs to remain in business. Their recovery mechanism is on-going customer fees to use their services. Over a long period of time, customers may end up paying more for a service than if they did it themselves. Ideally, it is a win-win: the CAPEX and maintenance OPEX of the cloud service provider is aggregated across enough customers that they can provide the services at a price lower than the customer can do it for themselves.
- Security — There is an old saying, "The only secure computer is the one that is still it its box, never powered on. It's also useless." By their nature, cloud-based services must be accessible via network connections, exposing them to attack from within the organization. If the network connections include the Internet, the services are exposed to attack from around the world, not just from within the organization.

Protecting any Internet-connected data center is difficult. An advantage had by multi-tenant cloud providers is the ability to aggregate the cost of that effort across all the tenants / customers. Arguably then, they should have better security than smaller data centers. However, a zero-day vulnerability exposes all data centers equally. Further, multi-tenant data centers are arguably a higher-value target than smaller, single-tenant centers.

There are also legal issues that may effect an organization's decision about locating certain personal data in cloud-based services, including the US HIPPA law, Canada's PIPEDA law, and the European Union's Directive 95/46/EC (Data Protection Directive).

To view debate the illuminates the two sides of the security argument, see `http://www.theregister.co.uk/2013/08/27/marc_andreessen_pat_gelsinger_in_verbal_vmworld_brawl/`. The debate continues.

The Big Picture

This introduction to Cloud provided some simple examples of cloud-based services and their benefits. To get a well-rounded understanding of *cloud*, you must understand the various types of cloud-based services, including: *infrastructure as a service* (*IaaS*), *platform as a service* (*PaaS*), *desktop as a service* (*DaaS*), and *software as a service* (*SaaS*), as well as the concepts of *public cloud*, *private cloud*, and *hybrid cloud* alluded to in the preceding definition. The sub-sections of this section provide that information.

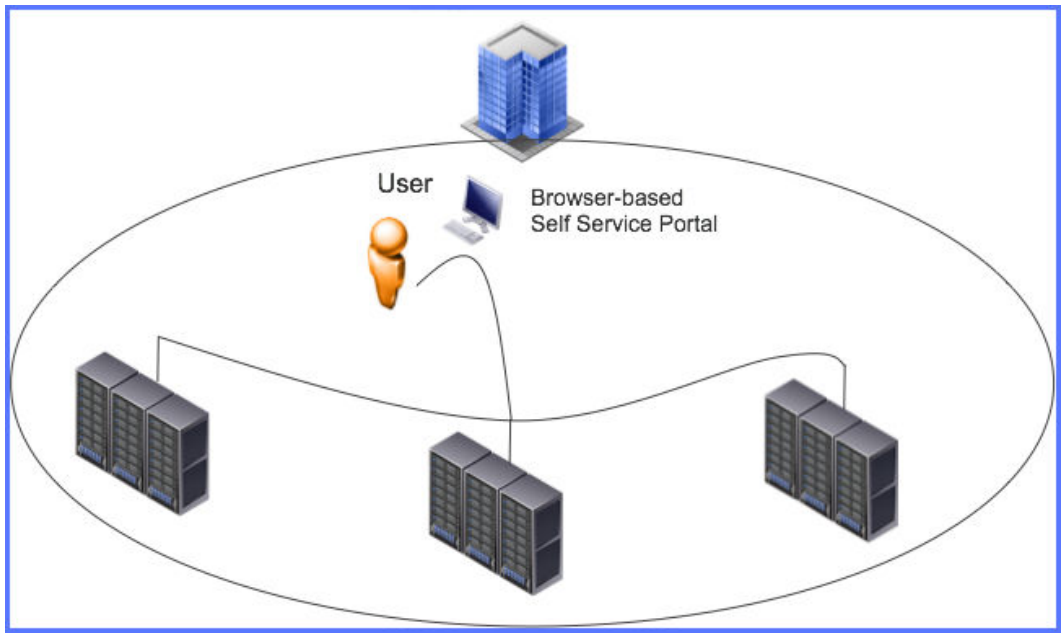
Understanding Private Cloud

The term *private cloud* refers to cloud-based services that are instantiated on an organization's own infrastructure, either within their own premises or at an off-premise datacenter. The key is, the infrastructure belongs to the organization. The distinction between just having an on-premise data-center and having a *private cloud* emanates from the definition of *cloud services* available within the datacenter. For example:

- A datacenter consisting of servers running Linux as a bare metal OS running web applications connected to databases running on other bare-metal servers does not constitute a private cloud. It is just a datacenter.
- A datacenter consisting of servers running hypervisors, with VMs created and deleted automatically in response to users provisioning services on demand does constitute a private cloud.

The following figure illustrates that private clouds remain within an organization's datacenter:

Figure 2-25. Private Cloud: Services all on an organization's infrastructure



There are as many business models for operating private clouds as you can imagine. Popular business models include:

- Cost center — The capital and operating expenses are a cost of doing business
- Charge back — The capital and operating expenses are charged back to the business units, departments, etc. based on their usage
- Federation — Each business unit / department / etc. within an organization brings their own hardware to the datacenter for installation, instantiates their own cloud services on their equipment, and are charged by the operators for space, power, and related staff time

There are hybrids of these models and many other models not listed.

Except in the Federated model, the cloud infrastructure is typically managed by an IT department consisting of system administrators, network administrators, etc. that are qualified to work in a virtualized environment.

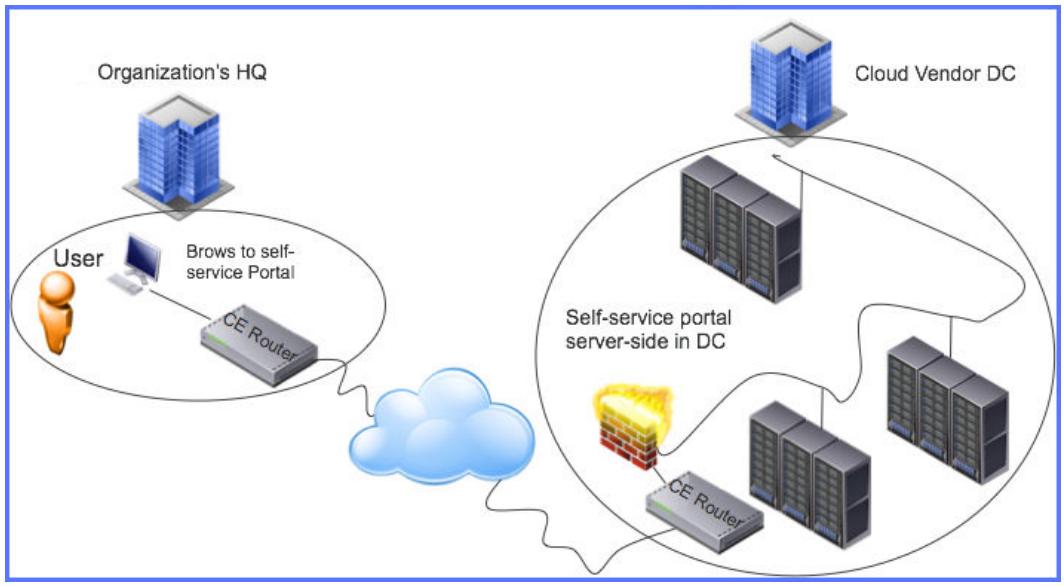
Organizations can use private clouds for any cloud-based service: Infrastructure as a Service (see "Understanding Infrastructure as a Service (IaaS)," on page 60, Software as a Service (see), etc.

The advantage most touted of private cloud over other types, specifically public cloud (see "Understanding Public Cloud," on page 58), is security, or more specifically regulatory compliance related to security. An organization that runs its own cloud is in charge of, and theoretically controls, its own security. The expertise required to truly secure any private data center, applications for managing cloud-based services, and the cloud-based services themselves, can be expensive and may not be wholly as effective as security provided by public-cloud vendors.

Understanding Public Cloud

The term *public cloud* refers to cloud-based services that are instantiated on infrastructure made available to the public via Internet connections, as shown in the following figure.

Figure 2-26. Understanding Public Cloud



Public cloud providers offer various cloud-based services, including Infrastructure and Software as a Service. The providers typically have a mix of free and pay-for services. For example, with Google Docs, you get free applications and a certain amount of free storage. For additional storage, you must pay. Apple's iCloud uses a similar business model.

The biggest advantage touted for public cloud services is that they allow the vendor to provide the services with lower capital and operational expenses than could be had by a customer hosting the same services in a private data center. Because the services must be reached via an Internet connection, it is vital that organizations have fast and reliable Internet connections and related equipment, such as customer edge routers.

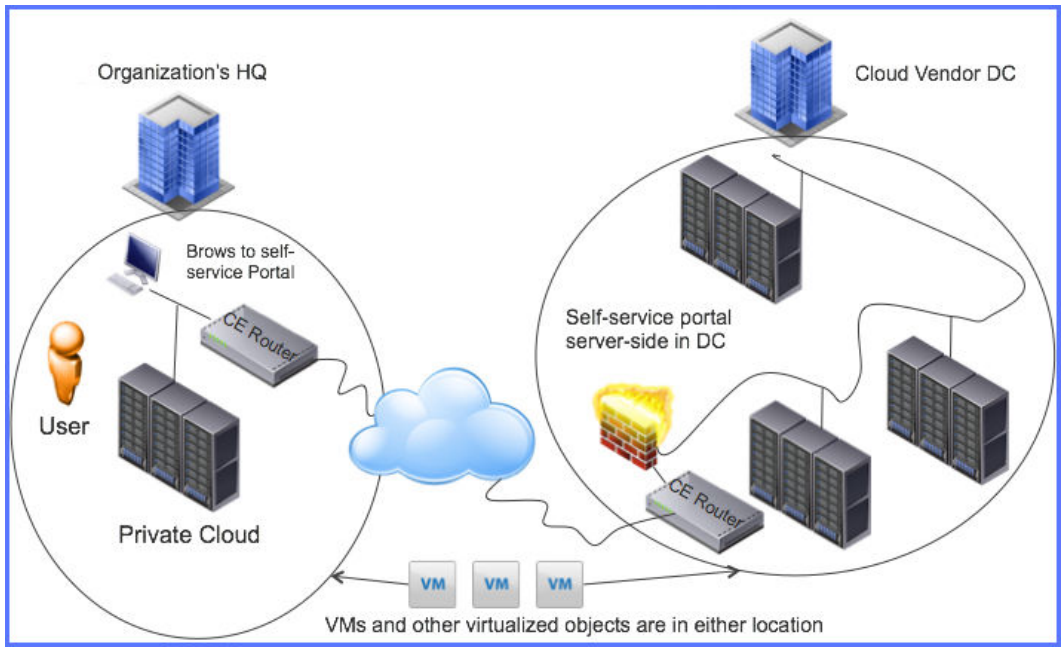
Understanding Hybrid Cloud

The term *hybrid cloud* refers to cloud-based services that reside on both private and public cloud instances. Examples of hybrid cloud uses include:

- An organization can create a private cloud for running their virtual desktops. They can then connect to a public cloud offering Infrastructure as a Service (see "Understanding Infrastructure as a Service (IaaS)," on page 60), allowing VMs to migrate to the public cloud instance when demand exceeds capacity in the private cloud instance. The organization's cloud administrators must be careful to observe privacy laws when configuring which VMs are eligible to migrate to the public instance.
- An organization can employ an email service that runs on the local cloud, but uses a public cloud front-end to filter incoming email for unwanted content, including malware. Several companies offer email filtering software that includes such public-cloud filtering capabilities, with a self-serve quarantine filter that aggregates emails quarantined in the public and private-cloud servers.

The following figure illustrates the concept of hybrid cloud as described above:

Figure 2-27. Hybrid Cloud Example



The advantage of hybrid cloud is that it allows organizations to have elastic capability beyond their own private cloud, but also allows them to observe privacy laws, keeping certain data and applications within their private cloud.

As with public and private cloud, hybrid cloud can be used to offer any of the cloud-based services: IaaS, SaaS, etc.

Understanding Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) is a service that offers computing infrastructure, including:

- OS instances, in the form of VMs, with vCPUs, vRAM, and vDisk. The storage and networking may come un-bundled and sold separately, as *Storage as a Service* and *Networking as a Service*. However, since OSes typically require a minimal amount of network connectivity and disk, these instances typically include these minimal resources.
- Networking infrastructure to connect the servers to one another, in a multi-tenant data center implementation (no customer can see another customer's servers via datacenter infrastructure). Typically, the service provider instantiates a new VLAN or VxLAN for each new customer, placing each of their VMs on said VLAN / VxLAN.
- (Typically) Internet connectivity with public IPs but no default firewalling

New servers (VMs), additional disk space, vRAM, vCPUs, etc. are all provisioned through a web interface, and possibly via an API, typically REST, Ruby, or Ruby on Rails-based. During VM provisioning, the customer selects the guest OS from a (typically) wide but finite list of supported options. The provisioning process may use templates or similar mechanisms to quickly instantiate the new VM with the selected guest OS, and provision it with an appropriate IP address. Customers then access the VM via the IP address.

Public IaaS providers charge for VMs with a base amount of resources (including CPU time), and then charge more for increased resource utilization, such as additional storage, vRAM, as well as Internet traffic. Private IaaS providers may have a similar business model, or may provide the resources for free (as a cost center).

There are several advantages of IaaS:

- Lower CAPEX — The customer does not have to purchase the servers, storage, networking infrastructure, etc. provided by IaaS. That expense is born by the service provider. There may still be some CAPEX for on-premise networking to access the IaaS, for example client systems and local networking.
- Lower OPEX for maintenance — The service provider bears the cost of hardware maintenance. They can aggregate the cost of system and network administrators across their client base, mark up that cost, pass it along to the customer base as part of their monthly access fees or usage fees. The maintenance portion of the fees are still often less than if a customer hired their own system and network administrators, in part because said administrators in smaller organizations are rarely fully utilized.
- Elastic resources — It is typically trivial to add or remove CPU, vRAM, and Storage resources to a customer's existing resources. This ability to expand and contract resources is often referred to as *elasticity*.

Organizations must be judicious in pricing IaaS services to ensure that they won't spend more in OPEX for IaaS than they would for both CAPEX and OPEX had they left the resources in-house.

For a list of the top 20 IaaS providers, see `http://www.clouds360.com/iaas.php`, and add VMware vCloud Hybrid Services® (vCHS) at `http://vcloud.vmware.com`, which was released after this list was created. Examples beyond VMware vCloud Hybrid Services include Google Compute Engine, and Rackspace Cloud Servers.

Understanding Platform as a Service (PaaS)

Platform as a Service (PaaS) is where service providers create an abstract platform, defined by APIs, that developers use to create and then run cloud-based applications. The platform APIs include methods for performing database operations, receiving and responding to web transactions, etc. The platforms typically support multiple languages by having multiple language bindings for their APIs as well as multiple language engines (script interpreters, Java Runtime Environments, Tomcat servers, etc.). While PaaS providers may provide *micro-platforms* for testing their code in a small virtual environment provided by the developer, they provide a virtualized, and ideally highly elastic, environment for running the applications.

PaaS enables software developers to focus their effort on the business logic and fresh UI experience, rather than on the details of how to write code for a specific OS, database vendor, web server, etc. All of those details are hidden in the generic *platform* provided by the PaaS vendor. PaaS enables the explosion of new web and mobile application development. To see this, consider the following:

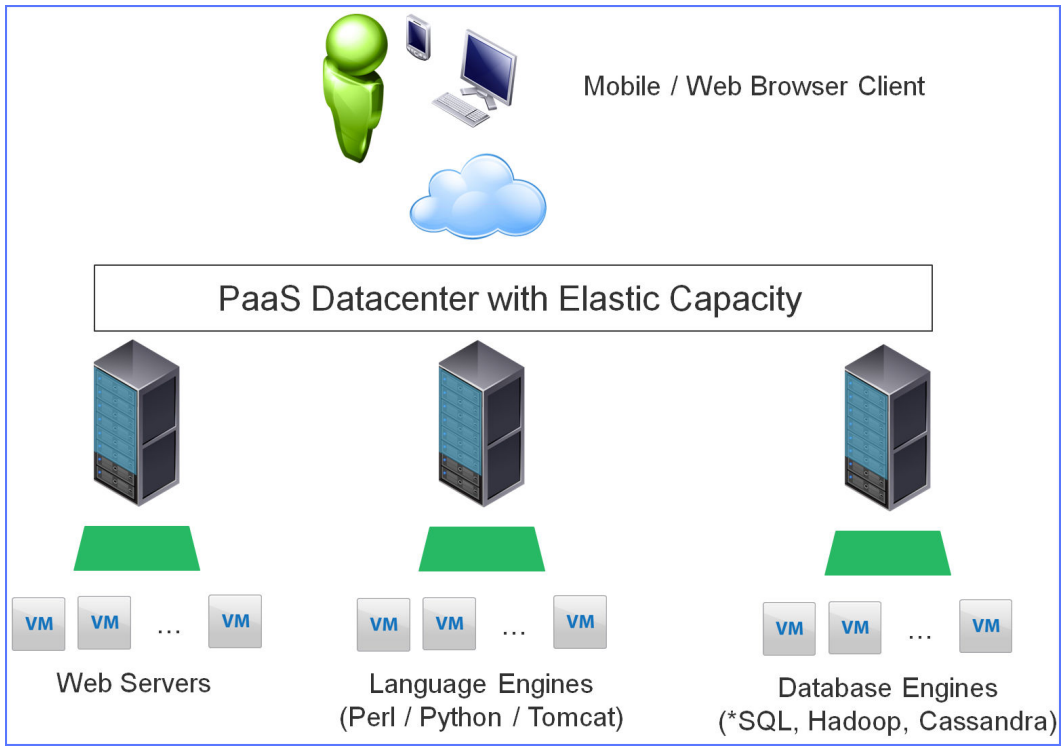
- We used to look in a phone book to find the phone number or address of a person or store — Now there are web and mobile applications that can do that, typically faster.
- We used to look at a map to determine how to get to the person or store we looked up in the phone book — Now there are web and mobile applications that will provide us with a map to get there. Those applications typically provides directions that don't take you down a one-way street the wrong way, provide you an estimated ETA, and may update the ETA based on traffic conditions.
- We used to hunt through isles for the things we wanted to buy, maybe looking at the signs over the ends of the rows for hints — Now there are applications that can locate you within the store, and guide you to the items you wish to purchase.
- We used to have no visibility into the current temperature inside our house, the state of the door locks, or data about our car, unless we were in our house, at the door, or in our car, respectively — Now there are applications for remotely: Querying the thermostat(s) in your house, and adjusting them; Querying the status of door locks, and locking or unlocking them; Querying the car to determine the fuel or battery level, lock or unlock the doors, set and enable the A/C or heating system, etc.

- We used to phone the pharmacy, or visit it, to renew our prescriptions — Now we use an application on our phone to scan the bar code on the medicine bottle, which sends the prescription number to the pharmacy as part of a renewal transaction.

These latter applications are being written by the retail vendors: the pharmacy chain or clothing or hardware store chain, not traditional software companies. PaaS provides these companies with a APIs for quickly developing, testing their code (at scale), and a platform for deploying that code in a way that scales. Scalability is key to these customers. For example, consider a hardware store that runs an ad on radio, TV, and social media just before Father's Day, and realizes a 1000-fold increase in demand from consumers. If the application does not scale, the consumers will turn away from the application and the retailer loses the sale. There are analogous examples in the not-for-profit space where a charity may experience spikes in giving during fundraising drives, telethons, etc.

Recall form the definition in the first paragraph that PaaS is targeted at web and mobile applications. Such applications typically have client / server architecture with the server-side having several components as shown in the following figure:

Figure 2-28. Multi-Tier, Cloud Enabled, Web / Mobile App Architecture



This figure shows that web and mobile applications typically require:

- A web browser or mobile device used for UI
- A web server, for example Apache, that is used to authenticate users (if necessary) and perform other transactions (such as find an address or a product)
- A language execution environment to run the application. The server-side code in a mobile / web app is typically Java or a scripting language such as PHP, Python, or Perl. Most server-side Java code is run in a container engine such as Tomcat or Virgo servers.
- A database engine, historically SQL-oriented, such as MySQL or Postgresql, but more recently Big Data solutions such as Hadoop, Cassandra, Mongodb, etc.

Such applications typically scale by having the PaaS virtualized environment add and remove web servers, instances of the language engines to run instances of the application, and instances of database servers (not databases) as needed. The PaaS virtualized environment may also add and remove VMs to run the web and database servers and language engines as needed.

While some PaaS solutions are free, some are commercial products based on commercial-grade IaaS-type data centers and that typically offer more features than free solutions. For example, Pivotal Lab's Cloud Foundry commercial product includes features that allow seamless upgrades (from the perspective of the user) of applications running on the PaaS as well as the PaaS infrastructure including hardware, libraries, database engines, etc.

The combination of automatic scaling and seamless updates allow application developers to concentrate on the business logic of their code and the refresh of their UI, enabling a culture of nearly-continuous deployment, especially compared to the previous waterfall methods of development and deployment in other environments.

PaaS offerings typically also include APIs and tooling that helps software engineers develop, test, and deploy cloud-based applications. Some PaaS offerings' offer plugins to existing IDEs, for example Eclipse or JBuilder.

For a list of popular PaaS vendors, see `http://www.clouds360.com/paas.php` and `http://www.cloudfoundry.com`. Immediate examples include Go Pivotal's Cloud Foundry, Google App Engine, and Engine Yard (which is both a product by the company of the same name).

Understanding Software as a Service (SaaS)

Software as a Service (SaaS) provides general purpose applications, via a web browser, with the application running on cloud infrastructure. Examples include:

- Fusion 360® (`360.autodesk.com`) — Provides 3D CAD (computer aided design) in the cloud, similar to running AutoCAD® from an IaaS server.
- Google Docs and Microsoft Office 365® (`docs.google.com` and `office.microsoft.com`, respectively) — Provide office productivity tools, including a word processor, spreadsheet, presentation software, and more. Google provides access to its applications for free, while Microsoft charges a fee. Both provide a base amount of storage for free, then charge for additional storage.

While the end product, applications delivered via the cloud, may seem the same as PaaS, there are some key differences, for example:

- Applications offered via SaaS are typically general purpose, where applications hosted on PaaS are typically specific to a vendor. For example, a word processor vs a specific coffee shop's application to advertise the special drink of the day and find their store closest to you.
- SaaS software has a revenue stream tied specifically to its use, where as PaaS applications are typically free, driving business to the reseller's stores, or selling pop-up space to advertisers.
- Perhaps most importantly, SaaS just provides the software. It does not provide a platform on which software engineers develop and then run their applications.

As with other cloud-based services, SaaS is multi-tenant, requiring users to authenticate to the service, for billing as well as security. That is, users cannot see each other's data (documents, databases, accounting legers, etc.) unless the owners of the days explicitly share it.

The benefits of SaaS include:

- The ability to share data with others without copying — This significantly increases collaboration over sending files between users via copy or email attachments. Some SaaS applications allow greater collaboration than others. For example, google docs allows real-time collaboration (one user can see another user's changes as they happen).

- The on-demand availability of the application — Users / Administrators are relieved of the need to install the software on a supported OS, update the software and OS, etc.
- Pay as you go — SaaS vendors may have time-based, usage-based, or mixed pricing models. This allows you to pay as you go, instead of having to purchase the software (and the equipment and OS to run it) up front. This is the same CAPEX / OPEX discussed in IaaS (see "Understanding Infrastructure as a Service (IaaS)," on page 60). The savings can be significant.

  Consider the case of CAD. A high-end CAD workstation needs the fastest graphics processor available to render drawings as quickly as possible, and the fastest CPU available and lots of RAM to perform analysis such as stress tests on buildings. However, most of the time, the user of a CAD system is making minor updates to the drawing that do not use the powerful CPU and huge amounts of RAM. That CAPEX is wasted while it is sitting idle. However, with CAD provided via SaaS, the user only pays for the extra vCPU and vRAM when it is needed.

For a list of the top 20 SaaS vendors, see `http://www.clouds360.com/saas.php`.

Understanding Storage as a Service

Storage as a Service provides storage space in the cloud as a service. Storage as a Service is packaged and priced in many different ways. Consider the following:

- Google Docs and Google Drive — Google provides a certain amount of cloud-based space for storing documents for free. Then it charges incremental fees for incremental space.

  Originally, the space was only available for use via google docs or upload / download links in the Google Docs web UI. In 2012, Google introduced GoogleDrive, which made the space available as a new drive in Windows and a mountable filesystem for Mac OS X.
- Amazon Simple Storage Service (S3) — Amazon provides storage accessible through a web interface including a browser UI and REST and SOAP APIs over HTTP. Third party vendors offer filesystem drivers that allow S3 customers to mount the S3 storage units (called *buckets*) as filesystems / drives on Linux / Windows systems, respectively. The Amazon S3 business model charges for both space and I/O to/from the Internet.
- Backup — Several vendors provide solutions for backing up data to storage in the cloud. Business models range from time-based subscriptions with relatively unlimited space to charging only for space, etc.

The vast majority of Storage as a Service vendors provide highly redundant protection of the data stored there, so the failure of a disk at the vendor's data center doesn't impact the user. They typically also provide both encrypted connections to the storage and encrypted within the storage itself. Encrypting the stored data mitigates against data theft should attacks against the vendor's site succeed.

The benefit of Storage as a Service is similar to other cloud services: Low or no CAPEX, and little or no OPEX for maintenance; It can provide off-site backup for disaster recovery.

There are several issues with Storage as a Service:

- OPEX can be high, especially for services that charge for I/O and where I/O is high
- Transfer times can be significant. This is especially true for organizations with asymmetric Internet connections with upload speeds that are 1/10th that of download speeds.
- Security is always an issue. Depending on the tool used, encrypting the data may increase its size, expanding the storage usage, increasing the data transmitted, and slowing effective I/O. Since most modern encryption algorithms also include compression, the net size may not be negligible, negating this as an issue.

For a list of popular Software as a Storage vendors, see `http://www.clouds360.com/storage.php`.

Note While Storage as a Service and Software as a Service both share the same acronym, SaaS is used for Software as a service, not Storage as a Service.

Acknowledging Other Cloud-based Services

The preceding topics in this section of the course have discussed the types of cloud-based services that developers are likely to encounter in today's virtualized environments. There are other types of cloud-based services emerging that you should be aware of, and can investigate in detail if desired, but don't (currently) need the level of detail given to the types of cloud-based services discussed thus far. Briefly, the other types of cloud-based service categories include (alphabetically):

- Desktop as a Service (DaaS) — You can think of this as IaaS + Storage as a Service + SaaS, bundled with a virtual desktop solution. The key value to customers is the virtual desktops as a cloud-based service, where the desktops are typically provisioned with standard productivity applications. This is a relatively new category (as of when the course was written). There are several vendors in this space, including AT&T and Desktone (recently acquired by VMware).
- Security as a Service — You can think of this as a special case of SaaS, where the software is focused on securing various aspects of a customer's IT, protecting on-premise, data in the could, data coming from or going to the Internet. A customer may employ several could-based security solutions to cover as many areas of exposure as possible. For example, one solution may protect an organization against malware for email and web services while another handles intrusion detection. Most of the products in this space have both appliance / hosted as well as cloud-based components. For example, a web protection system may pass URLs to a cloud service to determine whether it points to an infected web site.

  Note While Security as a Service and Software as a Service both share the same acronym, SaaS is used for Software as a service, not Security as a Service.

Understanding Trends in Virtualization: Software Defined *X*

Over time, vendors have been able to virtualize increasingly complex physical objects, progressing from just virtualizing CPUs and RAM to networking interfaces, to whole servers, etc. The progress continues. This progress has cause the introduction of additional cloud-based services (IaaS / PaaS / SaaS, etc, explained in a preceding section). The increased virtualization complexity and new services has created a need for more powerful management features, for both automated and manual operations.

Remember that, in general, *virtualization* of some *thing* decouples software implementation of that *thing* from its hardware implementation. A key aspect of *software defined X* technologies is that they continue this decoupling, treating the available hardware as a generic (white box) physical layer on which to define entities via software, for example Software Defined Networks (SDN), Software Defines Storage (SDS), etc.)

Another key aspect of *software defined X* technologies is that deployment of the software-defined object is driven by software that is driven by templates and policies. That is, once the underlying physical infrastructure is present (the servers with disks / switches / routers are racked and cabled together properly, etc.), the configuration of the virtual objects (hosts, datastores, networks, etc.) all happens through software, not through human configuration of each component.

The next three topics describe Software Defined Storage / Networking / Data Centers, respectively.

Understanding Software-Defined Storage (SDS)

Server virtualization has had a significant impact on storage, including:

- Increased need for Shared Storage, in large part driven by the increased frequency of virtualized clusters and vMotion type operations. Before virtualization, clusters were rarer.
- Increased performance demand due to location of `.vmdk` files on shared storage and consolidation of multiple applications onto the shared storage. Where a shared storage might have been dedicated to a database or similar application, now it is used by databases and VMs, each of which may be running multiple applications.
- Increased demand for object mobility between storage units (for example vMotion between two local datastores or between two volumes on separate storage devices)
- Increased expectation for reliability (if the storage is down, the VMs are down)
- Increased expectation of fast provisioning

Historically, storage provisioning has been labor intensive for administrators, including (but not limited to) the following tasks:

- Building disk pools
- Assigning RAID levels to pools
- Carving LUNs / logical volumes from pools and formatting them
- Configuring zoning and mapping

Tuning performance of storage has also historically been labor-intensive and require highly skilled administrators. Consider adding caching to a storage unit. How much cache is needed? What storage units should be allocated cache and on which is it wasted? Should caching be moved if a storage object is moved to different units or plexes?

Defining storage through software, using templates and policies, without the need for human intervention, automates the deployment of storage in a virtualized environment. This yields the following definition:

Software-Defined Storage (SDS) - Course definition	Software that creates virtual pools of disk, whether local to commodity servers or dedicated to storage hardware, automatically allocated to a datastore based on policies (including SLAs) rather than physical constraints, usable by virtual infrastructure, particularly hypervisors, but also third-party data services such as database engines. The software must also automatically manage available caching storage (such as SSDs) in a manner that does not impede VM mobility. The software must maintain SLAs even during storage mutations, such as increasing storage size, rebuilds due to physical disk loss, etc. The use of policies obviates the need for, but does not eliminate the possibility of, human management of the storage.

Several virtualization vendors are working to implement SDS as part of their SDDC offerings. VMware has several SDS-related initiatives, for example virtual SAN (vSAN), see GUID-49702645-2EAD-47C7-8820-55F05ECA12B3#GUID-49702645-2EAD-47C7-8820-55F05ECA12B3.

Understanding Software-Defined Networking (SDN)

Virtualization has increased the use of networking in many ways, including:

- Server and desktop consolidation has meant that each host system now homes multiple IPs, typically at least one per VM on the host.

- The need to create separate networks for management has always been around, but is more critical than ever in a virtualized environment.
- The need to create separate networks for vMotion and other infrastructure traffic
- The need for vSwitches to obviate *hair-pin traffic*, and careful placement of vRouters and VxLAN to vLAN gateways to eliminate *traffic tromboning*.

Historically, the provisioning and management of networking has been a labor-intensive activity with data center administrators racking new switches and routers and cabling devices and administrators configuring switch ports with vLAN tags, setting QoS policies and static routes, configuring firewall policies, etc.

The advent of server virtualization with vSwitches, distributed vSwitches, virtual firewalls and virtual routers (whether in virtual appliances or hypervisor kernel modules), VxLANs and other inventions have made it possible for all aspects of virtualization to be abstracted away from the physical network. Provided there is a physical fabric connecting virtual infrastructure, these advances have made it possible to define networks in software. Put another way, these advances have enabled *Software-Defined Networking* (*SDN*).

Wikipedia provides a the following definition for Software-Defined Networking (SDN):

Software-Defined Networking (from Wikipedia)

Software-defined networking (SDN) is an approach to computer networking which evolved from work done at UC Berkeley and Stanford University around 2008.[1] SDN allows network administrators to manage network services through abstraction of lower level functionality. This is done by decoupling the system that makes decisions about where traffic is sent (the control plane) from the underlying systems that forward traffic to the selected destination (the data plane). The inventors and vendors of these systems claim that this simplifies networking.[2]

[1] "Prof. Scott Shenker - Gentle Introduction to Software-Defined Networking - Technion lecture". YouTube. 2012-06-26. Retrieved 2014-01-23. [2] "Software-Defined Networking: The New Norm for Networks". White paper. Open Networking Foundation. April 13, 2012. Retrieved August 22, 2013.

Given the above definition, SDN can be said to have been achieved more than a decade ago by several networking hardware vendors that separate their control-plane and data-plane processing, for example Cisco's RPs/SUPs (route processor / supervisor) and LC (line card) separation and Juniper's SRM (switch route processor) and SFM (switch fabric module) separation.

However, SDN must be more than a separation of the control and data planes. It must include the ability to dynamically and automatically create networking entities (switches, routers, firewalls, NICs, etc.) in software, without the need for human intervention (though that must still be allowed). This requirement leads this course's definition of SDN:

Software-Defined Networking (course definition)

Software that is able to virtualize all aspects of networking: wires, switches, routers, firewalls, load balancers, vLANs, VXLANs, etc. (decoupling physical objects from their virtual equivalents). It treats the physical data-plane, and even physical routers and switches, as an extension of virtual wires and other equivalent virtualized objects, connecting those parts of the software defined network that do not reside on the same host. The software must be able to create the virtual networks and their components automatically, but also allow human interaction.

This definition includes the abstraction concept present in the Wikipedia definition, that the physical network (everything beyond the pNICs attached to the host systems) is considered a single physical fabric. It also allows for the idea that an entire network can exist within a single host, connected with virtual wires and other virtual network devices, all of which can be handled in software. That is, the course definition allows for an entire network to be defined in software.

Understanding Software-Defined Data Center (SDDC)

Creating data centers has been a labor-intensive process, with people racking and cabling all of the required individual components: servers, networking devices, storage devices, IP and storage fabrics, etc. and then installing, patching, and finally configuring each individual device's OS and (where appropriate) applications. This method of creating data centers is sometimes called *do it yourself (DIY) data centers*. Because devices of like type, such as servers / storage / datacenter switches, are typically racked together, DIY data centers are often said to have *siloed architectures*.

The *Software Defined Data Center* (*SDDC*) concept defines whole virtual data centers in software, relying on hypervisors to define compute resources (CPU, RAM, etc), and to provide networking via software-defined networking (SDN) and storage via software-defined storage (SDS) for the datacenter.

Wikipedia provides a formal definition of (SDDC) as follows (see `http://en.wikipedia.org/wiki/Software-defined_data_center`) :

Software-Defined Data Center (from wikipedia)

Software-defined data center (SDDC) is an architectural approach to IT infrastructure that extends virtualization concepts such as abstraction, pooling, and automation to all of the data center's resources and services to achieve IT as a service.[1] In a software-defined data center, "compute, storage, networking, security, and availability services are pooled, aggregated, and delivered as software, and managed by intelligent, policy-driven software."[2] Software-defined data centers are often regarded as the necessary foundational infrastructure for scalable, efficient cloud computing.[3]

Note The citations are, 1: Son, Emily A. "The Software-Defined-Data-Center (SDDC): Concept Or Reality? [VMware]". Softchoice Advisor Article. Softchoice Advisor. Retrieved 28 June 2013; 2: Manca, Pete (29 May 2013). "Software-Defined Data Centers: What's the Buzz All About?". Wired. Retrieved 28 June 2013; and [3] "The Storage Hypervisor, the missing link for The Software-Defined Data Center". Virsto blog post. Virsto. Retrieved 28 June 2013.

This course's simplified definition is:

Software-Defined Data Center (course definition)

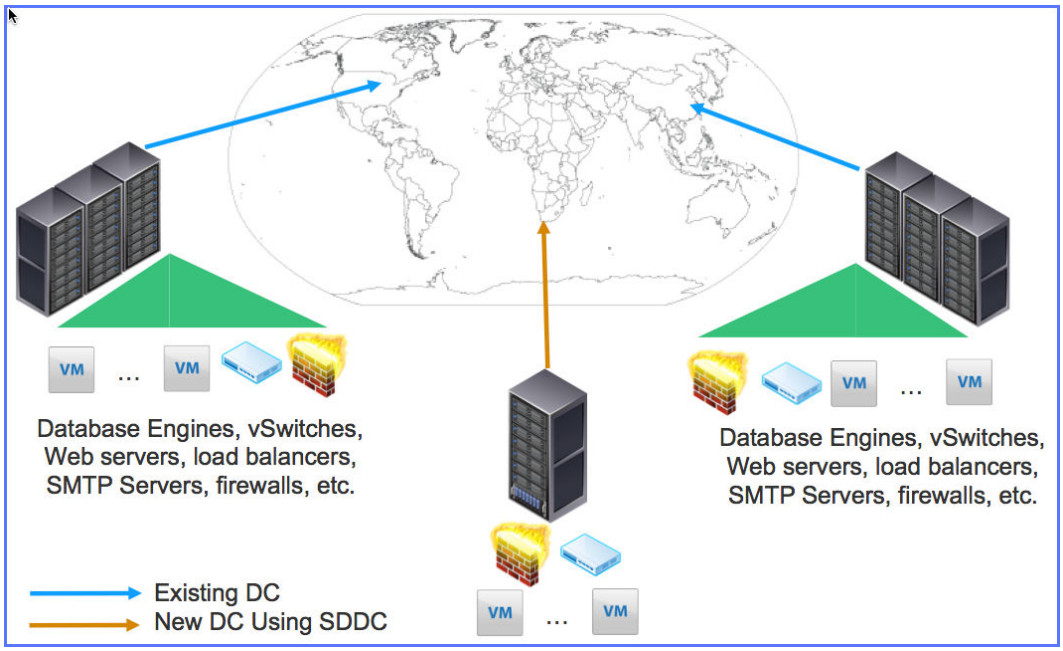
The ability to turn a set of physical servers, connected with basic networking (for example TOR switches) and / or sophisticated backplanes (in the case of blade servers) that have basic storage (local disks) into a functioning datacenter, with all resources not available in hardware virtualized in software, created through automation driven either by policies or, optionally, sparse human intervention. By extension, SDDC includes the ability to extend an existing SDDC, after appropriate physical resources are added, through automation (again) driven either by policies or, optionally, sparse human intervention.

In a fully SDDC, VMs must also be able to migrate anywhere within an organization's virtualized infrastructure. The SDDC must make all appropriate resources available to the VM. For example, its IP address within the organization cannot change, it must continue to have access to its datastores, etc. This requires that all of networking and storage infrastructure support SDDC as well.

The value of SDDC is the ability to deliver all aspects of Information Technology (IT) in an automated and flexible way. For example:

- Consider an organization that has a complex datacenter and needs to deploy a new one with comparable configuration, perhaps for geographical diversity, as illustrated in the following figure:

Figure 2-29. Datacenter Replication via SDDC



The organization can put together the physical resources, then click a menu item and have an entire existing datacenter's infrastructure replicated in the new datacenter, with the organization's VxLAN extending to the new datacenter.

- Consider an organization needs more compute or storage, they simply add servers into a rack and wire them to the TOR switch, or add a blade into a server, and the SDDC automatically recognizes the resource and deploys it where it is most needed.
- Consider the need to migrate web services away from infrastructure in area just hit by a natural disaster to infrastructure in other areas that are not affected, all without interrupting services to users / customers.

Consider what it takes to fully automate deployment or augmentation of any aspect of a datacenter using virtualization. Here are just some of the requirements (and VMware products that are discussed in another chapter of the course, as a teaser):

- Automated installation of hypervisors on bare metal servers, for example using ESXi and its Auto Deploy feature
- Automated creation of datastores using local or shared storage, for example using vSAN technology
- Automated creation of VMs running vApps such as databases, web servers, intrusion detection, firewalls, etc. for example using vCloud Director
- Automated creation of vSwitches and vDSes, DHCP servers, load balancers, and fixed IP addressing of appropriate nodes in the virtual network, for example using NSX for vSphere
- Automated creation of VLANs and VxLANs with VxLANs automatically connected to existing LAN segments under the virtual wire
- Automated creation of resource pools for storage, compute, RAM, etc. and assignment / migration of VMs and datastores to resources in applicable pools

After discussing the many forms of virtualization in the preceding sections, and looking at the first two sentences of the Wikipedia definition, it seems that the realization of the SDDC is near, if not already here. We can virtualize compute (CPUs, vRAM, etc.), storage (vDisks), networking (vNICs, vSwitches, virtual routers, VxLANs, etc.) and even security (virtual firewalls, including deep packet inspection, etc.). Further, given the discussion in "Understanding Management Solutions for Virtualized Environments," on page 29, it may seem that the realization is truly at hand.

The virtualization vendors are all working to complete their SDDC offerings, or make them better. VMware is generally recognized as the leader by many reputable industry organizations, for example Gartner, because it has all of the essential pieces, including automated management. That said, VMware is working to increase functionality, flexibility, and the speed at which SDDC functions occur with their infrastructure.

Understanding Virtualizing SAP Workloads

Understanding ESXi's Memory Management and its Benefits for SAP Workloads

The ESXi hypervisor creates a contiguous addressable memory space for a virtual machine when it runs. This allows the hypervisor to run multiple virtual machines simultaneously while protecting the memory of each virtual machine from being accessed by others. For details on how memory is managed, see Security of the VMware vSphere Hypervisor (http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/techpaper/vmw-white-paper-secrty-vsphr-hyprvsr-uslet-101.pdf).

The *VMkernel* manages all machine memory. The *VMkernel* dedicates part of this managed machine memory for its own use. The rest is available for use by virtual machines. Virtual machines use machine memory for two purposes: each virtual machine requires its own memory and the *virtual machine monitor (VMM)* requires some memory and a dynamic overhead memory for its code and data. The virtual and physical memory space is divided into blocks called pages. When physical memory is full, the data for virtual pages that are not present in physical memory are stored on disk. Depending on processor architecture, pages are typically 4 KB or 2 MB.

The configured size is a construct maintained by the virtualization layer for the virtual machine. It is the amount of memory that is presented to the guest operating system, but it is independent of the amount of physical RAM that is allocated to the virtual machine, which depends on the resource settings (shares, reservation, limit) explained later in this section.

For example, consider a virtual machine with a configured memory size of 1GB. When the guest operating system boots, it detects that it is running on a machine with 1GB of physical memory. The actual amount of physical host memory allocated to the virtual machine depends on its memory resource settings and memory contention on the ESXi host. In some cases, the virtual machine might be allocated the full 1GB. In other cases, it might receive a smaller allocation. Regardless of the actual allocation, the guest operating system continues to behave as though it is running on a machine with 1GB of physical memory. The following per-vm settings determine the amount of physical memory allocated to a vm:

- **Shares** — This setting specifies the relative priority for a virtual machine if more than the reservation is available.
- **Reservation** — This setting specifies the guaranteed lower bound on the amount of physical memory that the ESXi host reserves for the virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the virtual machine has sufficient memory to run efficiently, without excessive paging. After a virtual machine has accessed its full reservation, it is allowed to retain that amount of memory and this memory is not reclaimed, even if the virtual machine becomes idle. For example, some guest operating systems (for example, Linux) might not access all of the configured memory immediately after booting. Until the virtual machines accesses its full reservation, VMkernel can allocate any unused portion of its reservation to other virtual machines. However, after the guest's workload increases and it consumes its full reservation, it is allowed to keep this memory.

- **Limit** — This setting creates an upper bound on the amount of physical memory that the ESXi host can allocate to the virtual machine, up to the amount of the VM's configured memory size. Overhead memory includes space reserved for the virtual machine frame buffer and various virtualization data structures.

When a virtual machine requires memory, the VMkernel zeros out each memory page before providing them to the virtual machine. Memory isolation is maintained because any attempt by the OS, or any application running inside a virtual machine, to address memory outside of what has been allocated by the hypervisor causes a fault to be delivered to the guest OS. Such a fault typically results in an immediate system crash, panic, or halt in the virtual machine, depending on the OS.

Memory Over Commitment

For each running virtual machine, the system reserves physical memory for the virtual machine's reservation (if any) and for its virtualization overhead. Because of the memory management techniques the VMware ESXi host uses, your virtual machines can use more memory than the physical machine (the host) has available. For example, you can have a host with 2GB memory and run four virtual machines with 1GB memory each. In that case, the memory is overcommitted. Over commitment makes sense because, typically, some virtual machines are lightly loaded while others are more heavily loaded, and relative activity levels vary over time.

To improve memory utilization, the VMware ESXi host transfers memory from idle virtual machines to virtual machines that need more memory. Use the Reservation or Shares parameter to preferentially allocate memory to important virtual machines. This memory remains available to other virtual machines if it is not in use. In addition, memory compression is enabled by default on VMware ESXi hosts to improve virtual machine performance when memory is overcommitted.

Memory Sharing

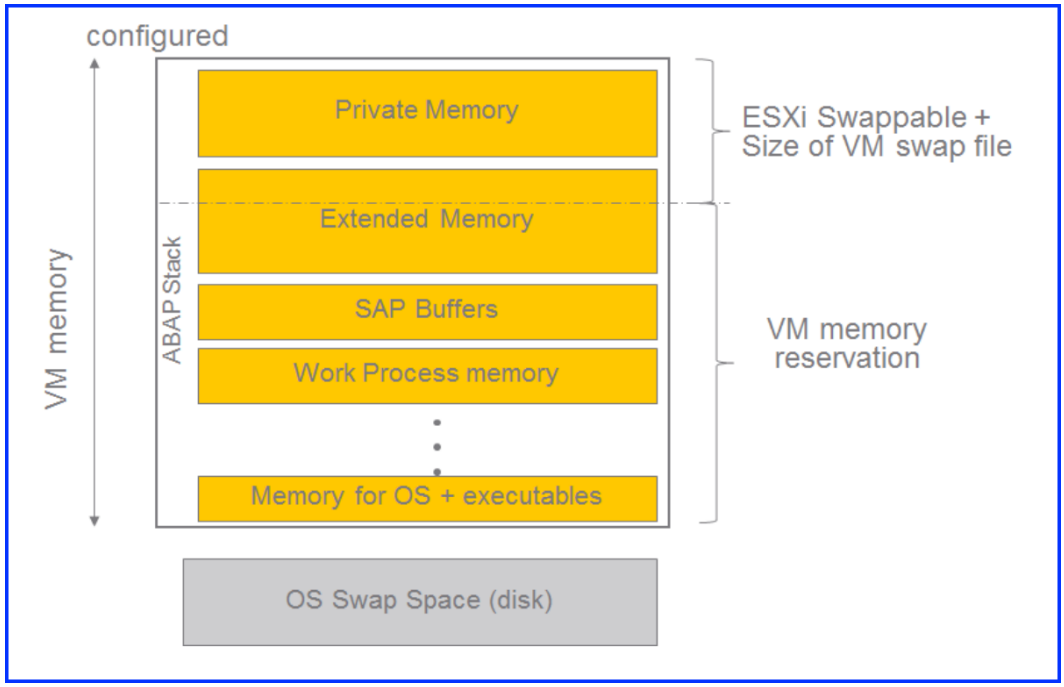
Many workloads present opportunities for sharing memory across virtual machines. For example, several virtual machines might be running instances of the same guest operating system, have the same applications or components loaded, or contain common data. VMware ESXi systems use a proprietary page-sharing technique to securely eliminate redundant copies of memory pages. With memory sharing, a workload consisting of multiple virtual machines often consumes less memory than it would when running on physical machines. As a result, the system can efficiently support higher levels of over commitment. The amount of memory saved by memory sharing depends on workload characteristics. A workload of many nearly identical virtual machines might free up more than thirty percent of memory, while a more diverse workload might result in savings of less than five percent of memory.

Under normal circumstances, the virtual machine has exclusive use of the memory page, and no other virtual machine can touch it or even detect it. The exception is when *Transparent Page Sharing* is in effect. This is a proprietary VMware ESXi technique to transparently share memory pages between virtual machines, thus eliminating redundant copies of memory pages. This can help with memory over commit to increase virtual machine consolidation.

Beginning with vSphere 6 (and included in patches or updates to some earlier releases), *Transparent Page Sharing* is enabled between virtual machines only when those virtual machines have the same salt value. For more information, see VMware KB Additional Transparent Page Sharing management capabilities and new default settings (2097593) (https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2097593). By default, in VMware ESXi salting is enabled (Mem.ShareForceSalting=2), and each virtual machine has a different salt value, meaning that *Transparent Page Sharing* will not happen across the virtual machines by default. This is not an issue for production SAP environments because this provides the highest security between virtual machines. Maximizing virtual machine consolidation through memory over commit is not a priority, and the best practice for databases is to use large pages that are not shared.

Virtual Machine Memory Settings

Figure 2-30. VM Memory Settings



Each virtual machine consumes memory based on its configured size, plus additional overhead memory for virtualization. The configured size is the amount of memory that is presented to the guest OS. The above figure shows the memory settings for a virtual machine running an *Advanced Business Application Programming (ABAP)* based application server.

The different memory areas inside the virtual machine correspond to the *Advanced Business Application Programming (ABAP)* stack. Consult SAP documentation for an exact breakdown of the different memory areas of an *Advanced Business Application Programming (ABAP)* based application server (such as http://help.sap.com/saphelp_nw74/helpdata/en/49/32f2b1e92e3504e10000000a421937/content.htm)

The following settings are shown in the above figure:

- Configured memory – Memory size of the virtual machine assigned at creation.
- VM memory reservation – Guaranteed lower bound on the amount of memory that the host reserves for the virtual machine, which cannot be reclaimed by VMware ESXi for other virtual machines.
- VMware ESXi swappable – Virtual machine memory that can be reclaimed by the balloon driver (*hypervisor memory reclamation technique*) or, in the worst case, by VMware ESXi swapping. This is the automatic size of the per-virtual-machine swap file that is created on the VMware file system (.vswp file). VMware ESXi swapping can cause severe performance degradation and should be avoided.
- OS swap space – The guest OS requires its own swap file configured according to SAP guidelines, and it resides in the virtual disk. This will function in the same manner as in physical environments.
- For production SAP, it is recommended to set the reservation to the configured size. The size of the VMware ESXi swap file is zero. The virtual machine will only start on or be live migrated to another VMware ESXi host if there is enough free memory equal to the reservation plus its overhead.

  You can allocate virtual machines on a single VMware ESXi host based on the following formula:

  ```
  Memory Available for SAP Virtual Machines = [Total ESXi Server Physical Memory] – [Memory Required by ESXi]
  ```

Memory required by a VMware ESXi host is comprised of memory required by VMkernel plus memory required for each virtual machine (which depends on the size of the virtual machine). The vSphere Resource Management guide (https://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-resource-management-guide.pdf) provides more details about this overhead memory. The memory guidelines are purposely conservative to avoid VMware ESXi kernel swapping. This is important due to the mission-critical nature of SAP business processes.

VMware Memory Management Techniques — Summary

VMware has perfected some powerful techniques by which RAM can be managed and optimized on a vSphere host in order to provide additional scalability on a per-host basis and to keep a host operating a peak levels.

- VMware Oversubscription/Overcommit - Allows administrators to assign more aggregate RAM to virtual machines than is actually physically available in the server
- Transparent Page Sharing - This is basically a deduplication method applied to RAM rather than storage
- Guest Ballooning - When VMware Tools is installed inside a virtual machine, along with everything else is a memory balloon process. The guest operating system can swap processes out to help free up memory that is then assigned to the balloon
- Memory compression attempts to fit multiple pages of RAM into a smaller number of pages in order to postpone for as long as possible the need for the hypervisor to swap to disk. Disk swapping is expensive in terms of performance

When it comes to memory, assigning too much is not a good thing and there are several reasons for that. The first reason is that the OS and applications tend to use all available memory for things like caching that consume extra available memory. This makes the hypervisors job of managing memory conservation, via features like TPS and ballooning, more difficult. Another thing that happens with memory is when you assign memory to a VM and power it on; you are also consuming additional disk space. The hypervisor creates a virtual swap (vswp) file in the home directory of the VM equal in size to the amount of memory assigned to a VM (minus any memory reservations). The reason this happens is to support vSphere's ability to over-commit memory to VMs and assign them more than a host is physically capable of supporting. Once a host 's physical memory is exhausted, it starts uses the vswp files to make up for this resource shortage, which slows down performance and puts more stress on the storage array

Virtualize CPU

VMware uses the terms *virtual CPU (vCPU)* and physical CPU (pCPU) to distinguish between the processors within virtual machines and the underlying physical x86-based processors. Virtual machines with more than one vCPU are also called *SMP (symmetric multiprocessing)* virtual machines.

VMware Virtual Symmetric Multi-Processing (Virtual SMP) enhances virtual machine performance by enabling a single virtual machine to use multiple physical processors simultaneously. The biggest advantage of an SMP system is the ability to use multiple processors to execute multiple tasks concurrently, thereby increasing throughput (for example, the number of transactions per second). Only workloads that support parallelization (including multiple processes or multiple threads that can run in parallel) can really benefit from SMP. The SAP architecture (application and database tiers) includes multiple processes that can take advantage of multiple threads, making it a good candidate to take advantage of Virtual SMP.

Hyper-Threading

Hyper-threading (also called simultaneous multithreading, or SMT) allows a single physical processor core to behave like two logical processors, essentially allowing two independent threads to run simultaneously. Unlike having twice as many processor cores that can roughly double performance, hyper-threading can provide anywhere from a slight to a significant increase in system performance by keeping the processor pipeline busier.

Virtual machines can be assigned up to 128 virtual CPUs in vSphere 6.0. The amount you can assign to a VM, of course, depends on the total amount a host has available. This number is determined by the number of physical CPUs (sockets), the number of cores that a physical CPU has, and whether hyper-threading technology is supported on the CPU.

For example, consider a server that has a single physical CPU that is quad-core, but it also has hyper-threading technology. So, the total CPUs seen by vSphere is 8 (1 x 4 x 2). Unlike memory, where you can assign more virtual RAM to a VM then you have physically in a host, you can't do this type of over-provisioning with CPU resources.

The performance impact of hyper-threading for a *SAP OLTP* workload was derived from SAP OLTP testing, documented in the VMware blog post SAP Three-Tier Shows Excellent Scaling on vSphere (http://blogs.vmware.com/performance/2010/03/sap-threetier-shows-excellent-scaling-on-vsphere.html).These results show that hyper-threading increased OLTP throughput by approximately 24 percent. Note that a different workload might yield different results.

NUMA (Non-Uniform Memory Access)

NUMA (non-uniform memory access) is a method of configuring a cluster of microprocessors in a multiprocessing system so that they can share memory locally, improving performance and the ability of the system to be expanded. NUMA is used in a symmetric multiprocessing (SMP) system. An SMP system is a "tightly-coupled," "share everything" system in which multiple processors working under a single operating system access each other's memory over a common bus or "interconnect" path. Ordinarily, a limitation of SMP is that as microprocessors are added, the shared bus or data path get overloaded and becomes a performance bottleneck. NUMA adds an intermediate level of memory shared among a few microprocessors so that all data accesses don't have to travel on the main bus. SMP and NUMA systems are typically used for applications such as data mining and decision support system in which processing can be parceled out to a number of processors that collectively work on a common database.

In a NUMA system, multiple NUMA nodes consist of a set of processors and the memory. The access to memory in the same node is local and the access to other nodes is remote. The remote access requires more cycles because it involves a multihop operation. Due to this asymmetric access latency, keeping the memory access local or maximizing the memory locality improves performance. The NUMA load balancer in ESXi assigns a home node to a virtual machine so that all vCPUs of the virtual machine are scheduled within the home node. For the virtual machine, the memory is allocated from the home node. Because the virtual machine rarely migrates away from the home node, the memory access from the virtual machine is mostly local. This is feasible with SAP application servers where there is flexibility in the virtual machine size. That is, multiple smaller application server virtual machines can fit within a NUMA node because SAP can scale out horizontally in the application tier. However, depending on sizing requirements, the SAP database virtual machine might need to scale vertically beyond the size of a NUMA node. Virtual machines with more vCPUs than the number of cores in each physical NUMA node are called wide virtual machines. These virtual machines are assigned to two (or more) NUMA nodes and are preferentially allocated memory local to those NUMA nodes. Because vCPUs in these wide virtual machines might sometimes need to access memory outside their own NUMA node, they might experience higher average memory access latencies than virtual machines that fit entirely within a NUMA node. This potential increase in average memory access latencies can be mitigated by appropriately configuring *virtual NUMA (vNUMA)*

vCPU Assignment

Most VMs will be fine with one vCPU. You may want to start with one, in most cases, and add more, if needed. The applications and workloads running inside the VM will dictate whether you need additional vCPUs or not. The exception here is if you have an application (i.e. Exchange, transactional database, etc.) that you know will have a heavy workload and need more than one vCPU.

So, why not just give VMs lots of CPUs, and let them use what they need? CPU usage is not like memory usage, which often utilizes all the memory assigned to it for things like pre-fetching. The real problem with assigning too many vCPUs to a VM is scheduling. Unlike memory, which is directly allocated to VMs and is not shared (except for TPS), CPU resources are shared and must wait in a line to be scheduled and

processed by the hypervisor which finds a free physical CPU/core to handle each request. Handling VMs with a single vCPU is pretty easy: just find a single open CPU/core and hand it over to the VM. With multiple vCPU's it becomes more difficult, as you have to find several available CPUs/cores to handle requests. This is called *co-scheduling*.

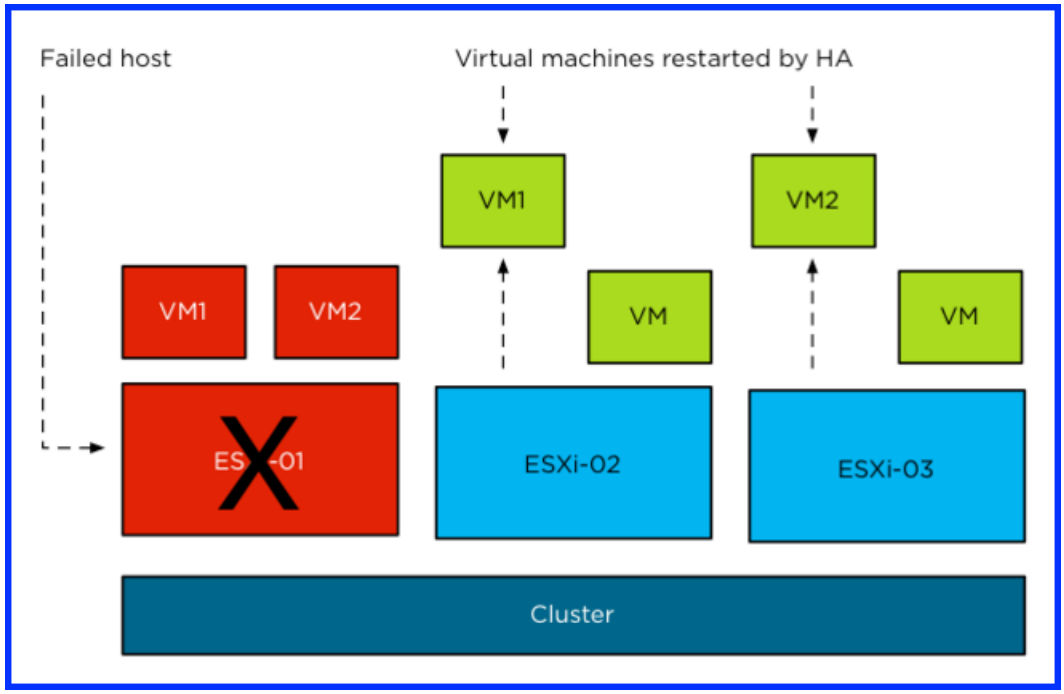
Performance Implications

CPU virtualization adds varying amounts of overhead depending on the workload and the type of virtualization used. An application is CPU-bound if it spends most of its time executing instructions rather than waiting for external events such as user interaction, device input, or data retrieval. For such applications, the CPU virtualization overhead includes the additional instructions that must be executed. This overhead takes CPU processing time that the application itself can use. CPU virtualization overhead usually translates into a reduction in overall performance. For applications that are not CPU-bound, CPU virtualization likely translates into an increase in CPU use. If spare CPU capacity is available to absorb the overhead, it can still deliver comparable performance in terms of overall throughput.

vSphere High Availability

Availability has traditionally been one of the most important aspects when providing services. When providing services on a shared platform like VMware vSphere, the impact of downtime significantly increases as many services run on a single physical machine. VMware vSphere High Availability (HA), provides a simple and cost effective solution to increase availability for any application running in a virtual machine regardless of its operating system. It is configured using a couple of simple steps throughvCenter Server (vCenter) and as such provides a uniform and simple interface. HA enables you to create a cluster out of multiple ESXi hosts. This will allow you to protect virtual machines and their workloads. In the event of a failure of one of the hosts in the cluster, impacted virtual machines are automatically restarted on other ESXi hosts within that same VMware vSphere Cluster (cluster).

Figure 2-31. VMware vSphere High Availability (HA)



In addition, in case of a Guest OS level failure, HA can restart the failed Guest OS. This feature is called *VM Monitoring*, but is sometimes also referred to as *VM-HA*. HA was designed to protect any virtual machine regardless of the type of workload within, but also can be extended to the application layer through the use of VM and Application Monitoring.

Note that in HA, a fail-over incurs downtime as the virtual machine is literally restarted on one of the remaining hosts in the cluster. If that is the case, why do you want to use HA when a virtual machine is not only getting restarted and the service is also temporarily lost ? The reason is that, not all virtual machines (or services) need 99.999 % uptime. For many services the type of availability HA provides is more than sufficient. Also HA does not require any changes to the guest as it is provided at the hypervisor level. Also *VM monitoring* does not require any additional software or OS modifications except for VMware Tools which should be installed anyway as a best practice. In case even higher availability is required, VMware also provides a level of application awareness through Application Monitoring. HA thus reduces complexity, cost associated with downtime, resource overhead and unplanned downtime with minimal additional costs.

There is often confusion between VMware HA and fault tolerance. Note that VMware HA is NOT fault tolerant in that if a host fails, the VMs on it also fail. HA thus deals only with restarting those VMs on other ESXi hosts with sufficient resources. On the other hand Fault tolerance, provides an uninterruptible access to resource in the event of a host failure. It hence eliminates the small amount of downtime that HA requires.

VMware HA maintains a communication channel with all the other ESXi hosts that are members of the same cluster using heartbeat that it sends at regular intervals. When an ESXi server misses a heartbeat, the other hosts will wait for a specific duration to get a response. Once the duration elapses, the cluster initiates the restart of the VMs on the failing ESXi host on the remaining hosts in the cluster. VMware HA also constantly monitors the ESXi hosts that are members of the cluster and ensures that resources are always available to satisfy requirements in the event of a host failure.

Virtual Machine Fault Monitoring is a technology that is disabled by default. Its function is to monitor virtual machines which it queries at regular intervals via a heartbeat. It does this by using the VMware Tools that are installed inside the VM. When a VM misses a heartbeat, VMware HA deems this VM as failed and attempts to reset it. *Virtual Machine Failure Monitoring* can detect whether a virtual machine was manually powered off, suspended, or migrated, and thereby does not attempt to restart it.

Figure 2-32. Overview of High Availability Solutions

	Reduce Unplanned Downtime	Reduce Planned Downtime
INFRASTRUCTURE	**vSphere HA** – in the event of ESXi host failure VMs are restarted on another host. *Central Services failover time: VM restart + Guest OS boot + Central Services start* *Database failover time: VM restart + Guest OS boot + database crash recovery + database start*	**vSphere vMotion** – live migrate VMs off ESXi host so hypervisor can be patched and BIOS updated. There is no downtime impact to the application. Need "N+1" type ESXi cluster setups for this to work.
	vSphere FT – applicable for Central Services in standalone VM and protects against ESXi host failure. vSphere 6 FT is limited to 4 vCPUs .Majority of virtual SAP databases are sized above 4 vCPU. *Central Services failover time: ~ zero, no loss of SAP locks.*	**vSphere vMotion** – live migration of primary and secondary VMs possible off ESXi host so hypervisor can be patched and BIOS updated.
	VM Monitoring – protects against Guest OS crash. Will restart VM if VMware Tools heartbeats are not received and the VM isn't generating any storage or network IO.	N/A
INFRASTRUCTURE + APPLICATION	**Partner solutions** which integrate with VMware HA and provide agents to monitor Central Services and the database *(not available out-of-the-box in vSphere)* Software failures are detected and automatically restarted. After pre-configured number of failed restarts VMware HA event is invoked restarting the virtual machine.	N/A
APPLICATION	**Database vendor agent** – database vendor specific functionality that monitors health of the database services and in case of failure restarts the database automatically	N/A
	3rd party Cluster software - active-passive configuration of two VMs on separate ESXi hosts. VMs connected to shared storage. Cluster resource groups configured for database and Central Services to automate service failover or restart in case of ESXi host or software failure. Standalone + Replicated enqueue possible (http://help.sap.com/saphelp_nwpi711/helpdata/en/47/e023f3bf423c83e10000000a42189c/frameset.htm) *Central Services failover time: message server restart + restart standalone enqueue and transfer of SAP locks – no loss of locks* *Database failover time: database crash recovery + database start*	Rolling patch upgrade of the Guest OS is possible; patch the passive VM; failover resources patch the other VM. Depending on the vendor, rolling patch upgrade of the database software is possible. Consult the database vendor documentation for details Rolling kernel upgrade of Central Services is covered in http://service.sap.com/sap/support/notes/953653
	Database Replication* – database vendors have solutions that enable transactions to be shipped/replayed from a primary to a standby database (in two VMs on separate ESXi hosts). Database storage is duplicated/non-shared. Solution from database vendor or third party required to automate failover. Relatively expensive solution for local HA, typically used in DR scenarios. *Database failover time: DB IP address failover; make standby DB primary; start DB. (no crash recovery is required as standby database is in consistent state)*	Depending on the database vendor solution it is possible to perform rolling patch upgrades e.g. upgrade standby database first, perform switchover, patch primary and then switch back. Consult the database vendor documentation for details – e.g. for HANA see http://service.sap.com/sap/support/notes/1917506

The high availability solutions in the figure are grouped into infrastructure and application. This corresponds to different IT groups who would be responsible for the two areas. VMware infrastructure teams are responsible for the setup of vSphere HA and vSphere FT, and the guest OS, SAP and database administrators would be responsible for the application-level solutions that require cluster or database availability solutions installed inside the virtual machine. Consider the following points —

- Overall availability is a function of unplanned and planned downtime of both layers, infrastructure and application.
- The vSphere HA features in the infrastructure layer are available out-of-the-box with VMware vSphere installations.
- ESXi cluster design involves sizing for spare capacity such that spare resources are available so virtual machines can restart in the event of host failure. This is also required for successful vSphere vMotion operations.
- With cluster software installed in the guest OS, there are two levels of clustering that provide high availability: vSphere HA at the ESXi host level, and cluster resources inside the virtual machine enabling failover of application services, such as database and Central Services.

- Application-level solutions can increase availability —
  - Reduce planned downtime with rolling patch upgrades
  - Faster fail over and reduced RTO
  - Software monitoring of the database and Central Services
  - Whether these solutions are needed depends on the business requirements. There is a trade-off because deployment introduces an extra level of configuration and complexity

vSphere Fault Tolerance

VMware vSphere Fault Tolerance (FT) is another form of VM clustering developed by VMware for systems that require extreme uptime.

When protecting VMs with FT, a secondary VM is created in lockstep of the protected VM, the first VM. FT works by simultaneously writing to the first VM (Protected VM or Primary VM) and the second VM (Duplicate VM or Secondary VM) at the same time. Every task is written twice. If you Click on the Start menu on the first VM, the Start menu on the second VM will also be Clicked. The power of FT is its capability to keep both VMs in sync.

The secondary VM on another ESXi host shares the same virtual disk file as the primary VM and then the CPU and virtual device inputs are transferred from the primary VM (record) to the secondary VM (replay) via a FT logging NIC so it is in sync with the primary and ready to take over in case of a failure. While both the primary and secondary VMs receive the same inputs, only the primary VM produces output such as disk writes and network transmits. The secondary VM's output is suppressed by the hypervisor and is not on the network until it becomes a primary VM, so essentially both VMs function as a single VM.

If the protected VM should go down for any reason, the secondary VM immediately takes its place, seizing its identity and its IP address, continuing to service users without an interruption. The newly promoted protected VM then creates a secondary for itself on another host and the cycle restarts.

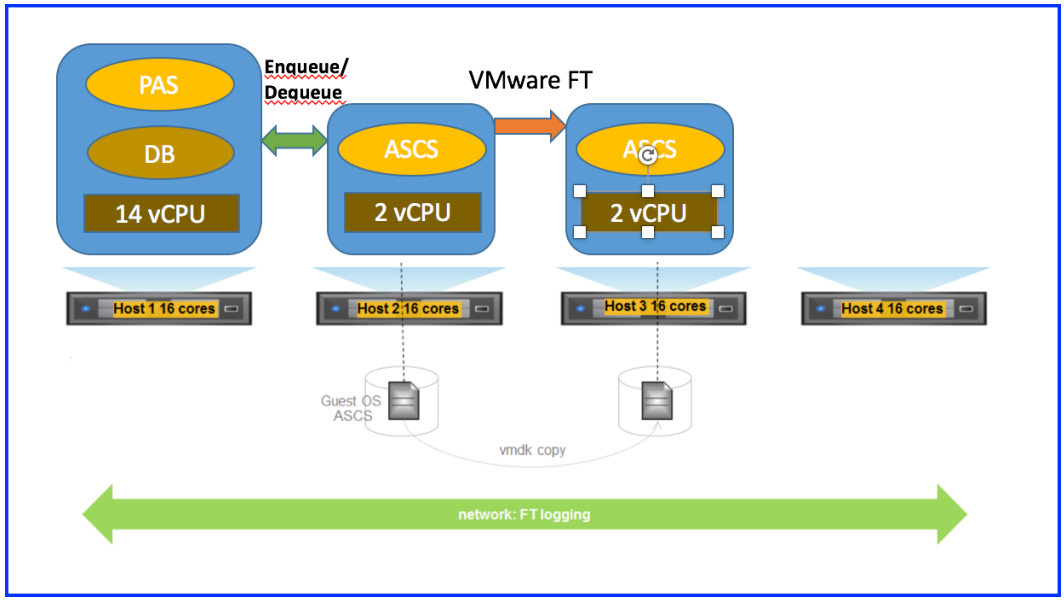
Salient Features of vSphere FT functionality

- Host Requirements
  - CPUs: Only recent HV-compatible processors (AMD Barcelona+, Intel Harpertown+), processors must be the same family.
  - All hosts must be running the same build of VMware ESXi.
  - Storage: shared storage (FC, iSCSI, or NAS)
  - Hosts must be in an HA-enabled cluster
  - Network and storage redundancy to improve reliability: NIC teaming, storage multipathing
  - Separate VMotion NIC and FT logging NIC, each Gigabit Ethernet (10GB recommended). Hence, minimum of 4 NICs(VMotion, FT Logging, two for VM traffic/Service Console)
- **VM Requirements**
  - VMs must be single-processor (no vSMP)
  - All VM disks must be "thick" (fully-allocated)
  - No non-replayable devices (USB, sound, physical CD-ROM, physical floppy, physical RDMs)
  - Make sure para virtualization is not enabled by default (Ubuntu Linux 7/8 and SUSE Linux 10)
  - All applications and guest OSes are supported-both 32-bit and 64-bit
- If the host running the primary VM fails, the secondary VM is immediately activated to replace it with no interruption of service to users. A new secondary VM is started and vSphere FT redundancy is reestablished automatically.

- If the host running the secondary VM fails, it is also immediately replaced
- A fault-tolerant virtual machine and its secondary copy are not allowed to run on the same host
- The primary VM can be up to 4 vCPUs and 64 GB
- The secondary VM has its own copy of the primary VM's virtual disks
- Dedicated 10 GbE NIC is recommended for vSphere FT logging traffic. vSphere FT logging traffic between primary and secondary VMs contains guest network and storage I/O data, as well as the memory contents of the guest operating system
- For more details on vSphere FT and best practices and limitations see the vSphere Availability guide — http://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-601-availability-guide.pdf
- For details on Performance Best Practices for VMware vSphere 6.0 refer to — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-perfbest-practices-vsphere6-0-white-paper.pdf

Example of SAP Central Services protected by vSphere FT

Figure 2-33. *SAP Central Services* protected by vSphere FT Example



SAP Central Services is a good candidate for vSphere FT. The above figure describes an example scenario of vSphere FT with Central Services managing an OLTP workload of numerous concurrent users executing SAP Sales and Distribution transactions.

In this scenerio, the *Primary Application Server* (PAS) running the ABAP stack and database is installed in the same virtual machine, and the *ABAP SAP Central Services* (ASCS) component is installed in a standalone 2 vCPU, 8 GB virtual machine. No other virtual machines are running, so each virtual machine had full access to host cores based on their vCPU count. SAP lock requests and deletes (enqueue and dequeue) generate network traffic between the PAS and ASCS.

ASCS is protected via VMware vSphere Fault Tolerance

Recovery with SRM

VMware Site Recovery Manager (SRM) provides disaster recovery protection for virtual environments.

Disaster recovery testing comprises a logistical plan for how an organization will recover and restore partially or completely interrupted critical functions within a predetermined time after a disaster or extended disruption. A disaster recovery plan is only as good as its last successful test. Disaster recovery testing is often difficult because it is usually very disruptive, extremely complex, and expensive in terms of resources. By leveraging virtualization, Site Recovery Manager addresses this problem while making planning and testing simpler to execute.

Two sites are involved when using Site Recovery Manager: a protected site and a recovery site. Site Recovery Manager leverages array-based replication between a protected site and a recovery site to copy virtual machines.

Recovery point objective (RPO) and recovery time objective (RTO) are the two most important performance metrics to keep in mind while designing and executing a disaster recovery plan. RPO is traditionally addressed at the storage layer, where certified storage replication adapters integrate with Site Recovery Manager to enable a fully automated test or real recovery. There are two additional options available for data replication: vSphere Replication (which can also be used to supplement storage replication adapters from the storage vendors) and database vendor-specific replication solutions.

RTO is addressed by the Site Recovery Manager recovery plans that automate the startup sequence of multiple virtual machines that comprise a virtualized SAP landscape and automate network connectivity at the remote site.

For further details on VMware Site Recovery Manager refer to —
https://www.vmware.com/support/pubs/srm_pubs.html

Additional References - Virtualization Overview

Following are some additional useful references —

1 Site Recovery Manager — http://www.vmware.com/products/site-recovery-manager.html

2 VMware vSphere 6 Documentation - https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-6-pubs.html

Overview of VMware vCloud Suite 3

VMware's product lines allows customers to implement virtualization from a small datacenter to a private cloud to hybrid cloud to public cloud as a service provider. To facilitate cloud deployments, VMware has bundled relevant products into a single product, *vCloud Suite*. The features of the **vCloud Suite** are an aggregation of the features of each of the products in the bundle. The suite itself does not add any features. The benefit of the suite is the bundled pricing and licensing. After introducing the term this chapter describes the components of vCloud Suite deployed in a SAP landscape environment and then provides an example architecture of a virtualized SAP environment using these components. The last section provides links to various other additional sources of information.

This chapter includes the following topics:

What is vCloud Suite ?

VMware vCloud Suite is an integrated set of products that provide infrastructure virtualization, disaster recovery and automation, and cloud management for on-premises vSphere environments.
VMware vCloud Suite enables IT to build and manage a vSphere based private cloud resulting in strategic IT outcomes. This is done by assembling an integrated set of products that are engineered to work better together.

Component Description

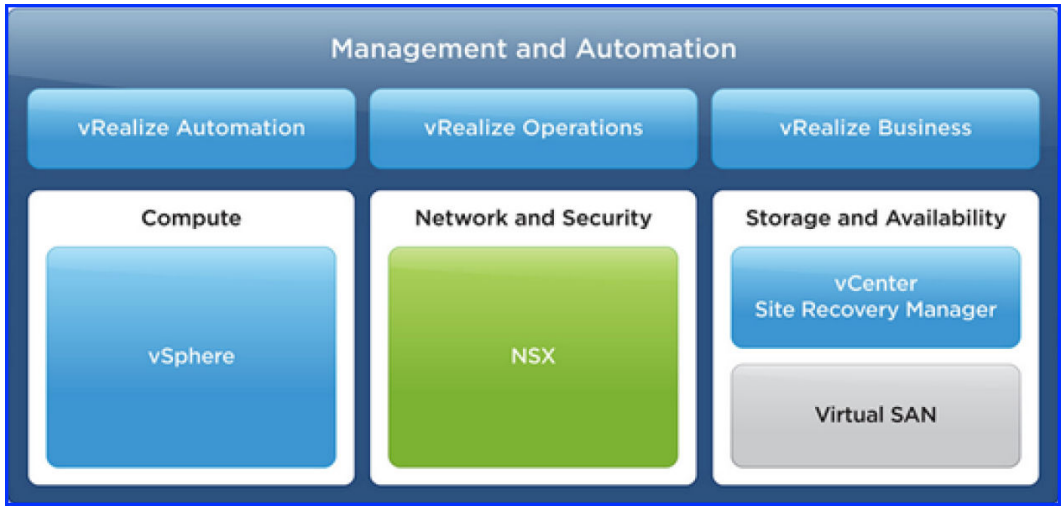
VMware vCloud Suite contains the following integrated products:

- VMware vSphere 6.0 – Industry leading server virtualization platform.
- VMware Site Recovery Manager (SRM) – Policy based disaster recovery and testing for all virtualized applications
- Cloud Management Platform —
  - VMware vRealize Operations – Intelligent performance, capacity and configuration management for vSphere environments
  - VMware vRealize Automation – Self-service and policy based infrastructure and application provisioning for vSphere environments

- VMware vRealize Business – Automated costing, usage metering, and service pricing of virtualized infrastructure for vSphere environments.

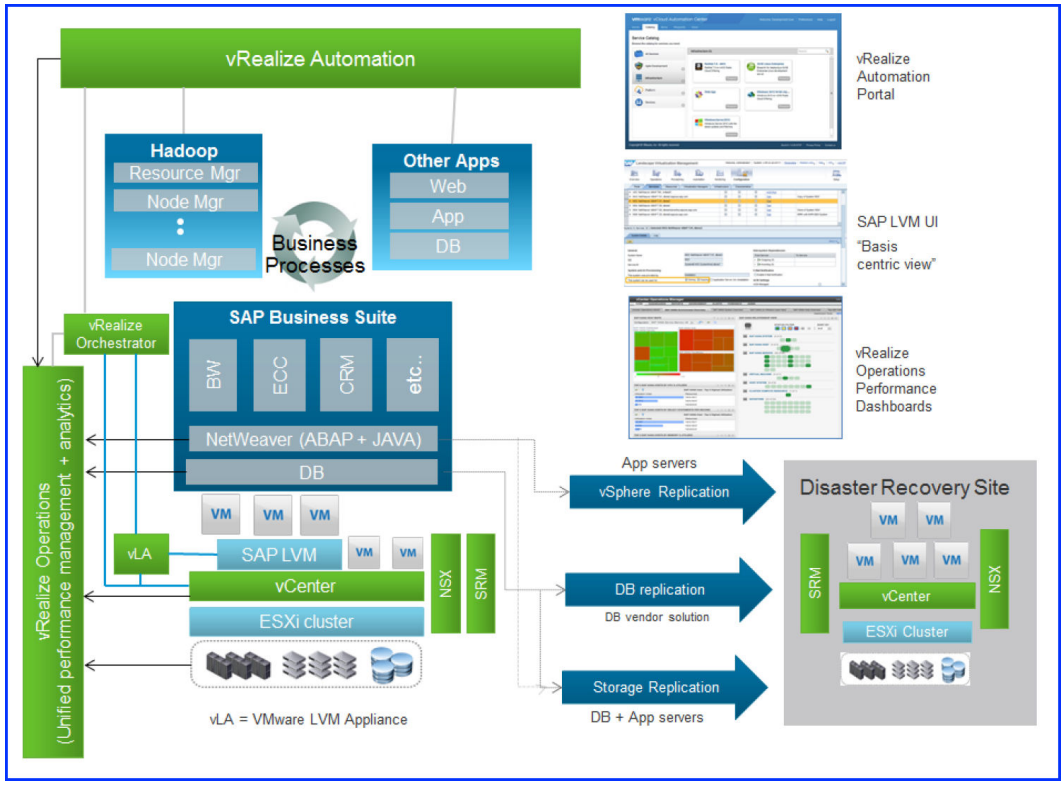
- VMware Virtual SAN – Software defined storage platform that extends VMware vCloud Suite by abstracting and pooling storage to deliver data center virtualization and standardization.
- VMware NSX – Security and network virtualization that is fully decoupled from hardware. VMware NSX extends VMware vCloud Suite by virtualizing networking to deliver data center virtualization and standardization and security controls native to the infrastructure.

Figure 3-1. VMware vCloud Suite



Example Architecture of Virtualized SAP Landscape with vCloud Suite

Figure 3-2. Example Architecture SAP Landscape



The above figure shows an example architecture of a *virtualized SAP landscape* along with components of VMware vCloud Suite.

- **Note**

  The *SAP Business Suite* of applications along with other applications that SAP might need to integrate with to deliver complete business processes (example, integration between SAP HANA and Hadoop to combine structured and unstructured data for data analytics requirements)

- *LaMa* with the VMware Adapter for SAP Landscape Management handles SAP provisioning. The adapter consists of three components —
  - The VMware LaMa Appliance (VLA) is a virtual machine that contains all the application code
  - An adapter that the VLA deploys on to the *LaMa*. The adapter tells LaMa how to communicate with the VLA
  - vCenter Orchestrator workflows that execute the commands LaMa sends to the VLA. vCenter Orchestrator executes these instructions on the vCenter Server
- The automated provisioning of other applications that integrate with SAP is managed by VMware vRealize Automation. For example, with VMware vRealize Automation you can run Hadoop as a service to support any integration requirements with SAP landscapes.
- VMware vRealize Operations management packs are available to extract performance metrics from SAP CCMS, databases and storage arrays. This enables performance data from multiple layers to be coordinated with vSphere metrics into unified dashboards.
- For database and application virtual machines are deployed on the same ESXi cluster, depending on security requirements. VMware NSX micro-segmentation can provide security firewalls between the database and application tier.
- VMware Site Recovery Manager (SRM) manages disaster recovery testing and execution. VMware Site Recovery Manager workflows manage the startup order of virtual machines at the disaster recovery site, which helps reduce the recovery time objective (RTO). Three methods are possible to replicate data and virtual machines to the disaster recovery site that address the recovery point objective (RPO) —
  - vSphere Replication
  - Database vendor replication solutions
  - Storage array replication

For more details on SAP support on VMware vSphererefer to — https://wiki.scn.sap.com/wiki/display/VIRTUALIZATION/SAP+on+VMware+vSphere

SAP Architecture on VMware Virtualized Infrastructure

This section briefly discusses SAP architecture concepts and terminology.

SAP uses the term *System Landscape*, which contains all the SAP systems that have been installed. It can consist of several system groups where SAP systems are linked by transport routes. Transport routes refer to the path of code migrations between SAP systems, for example from Development (DEV) to Quality Assurance (QAS) to Production (PRD) — https://help.sap.com/saphelp_nw74/helpdata/en/63/a30a4ac00811d2851c0000e8a57770/content.htm

The architecture of a single SAP system is multi-tier and consists of the following components —

- Application Servers (*SAP Web Application Servers*) – These are *Advanced Business Application Programming (ABAP)* and/or Java (J2EE) based, depending on the specific SAP product or module. Two types exist —
  - *Primary Application Server (PAS)* – An application server instance that is installed with *SAP Central Services* in newer *NetWeaver* releases and is part of the base installation.

- *Additional Application Servers (AAS)* – Applications servers installed as required for horizontal scalability.

- *SAP Message Service* – The SAP Message Service is used to exchange and regulate messages between SAP instances in a SAP system. It manages functions such as determining which instance a user logs onto during client connect, and scheduling batch jobs on instances configured for batch.
- *SAP Enqueue Service* – The SAP Enqueue Service manages the locking of business objects at the SAP transaction level. Locks are set in a lock table stored in the shared memory of the host on which the SAP Enqueue Service runs.
- *Database Server* – SAP supports several databases. The most common databases include Sybase, HANA, Microsoft SQL Server, Oracle, and IBM DB2.

The following SAP services are defined based on the Message and Enqueue Services —

- *Central Instance (CI)* – Comprised of the Message and Enqueue Services and other SAP work processes that allow the execution of online and batch workloads. In newer NetWeaver releases, the CI is replaced with *SAP Central Services* and the *Primary Application Server*.
- *SAP Central Services* – In newer versions of SAP, the Message and Enqueue Services have been grouped into a standalone service. Separate Central Services exist for ABAP and Java based application servers. For ABAP variants, it is called *ABAP SAP Central Services (ASCS)*, and for J2EE variants is called *SAP Central Services (SCS)*
- *Replicated Enqueue Server* – This component consists of the standalone Enqueue Server and an Enqueue Replication Server. The Replicated Enqueue Server runs on another host and contains a replica of the lock table (replication table). If the standalone Enqueue Server fails, it must be restarted on the host on which the Enqueue Replication Server is running, because this host contains the replication table in a shared memory segment. The restarted Enqueue Server uses this shared memory segment to generate the new lock table, after which the shared memory segment is deleted

SAP Standard Application Benchmarks are available to assist customers and partners find the appropriate hardware configuration for their IT solutions. You can find more details at — http://sap.com/benchmark

SAP sizing is conducted using the SAPS (SAP Application Performance Standard) metric. SAPS is a hardware-independent unit of measurement that describes the performance of a system configuration in the SAP environment. It is derived from the Sales and Distribution (SD) benchmark. The SAPS metric is used to size in the virtual environment to make x86 server purchasing and design decisions similar to physical.

Additional References - Overview of VMware vCloud Suite

1 VMware vCloud Suite — https://www.virtualizationworks.com/datasheets/VMware-vCloud-Suite-Datasheet.pdf

2 http://www.vmware.com/in/products/vcloud-suite.html

Virtualize SAP HANA 4

When it comes to real-time analytics and the next generation of enterprise IT applications, you don't need to look much further than SAP. Between the numerous modules available for a wide variety of use cases and SAP HAN - one of the most powerful and versatile in-memory data platforms - any organization can build a tech stack that meets its corporate needs.

Complex SAP systems modules and SAP HANA must be hosted on the right infrastructure. Beyond the sheer power and performance of the foundations hardware of these environments, it is also important to look at the other features and capabilities of the SAP infrastructure. One of the most important aspects of SAP environment is virtualization. As enterprise IT progresses, it's becoming increasingly obvious that decoupling compute resources from underlying hardware will be a key to success, speed and performance. And when it comes to SAP HANA, it's hard to overstate the importance of virtualization. In brief, it's a must-have capability.

This chapter includes the following topics:

What is SAP HANA ?

SAP HANA is an in-memory database that massively improves performance of existing SAP applications, and enables business transformation via real-time analytics and transaction execution. A few salient points to note about SAP HANA are:

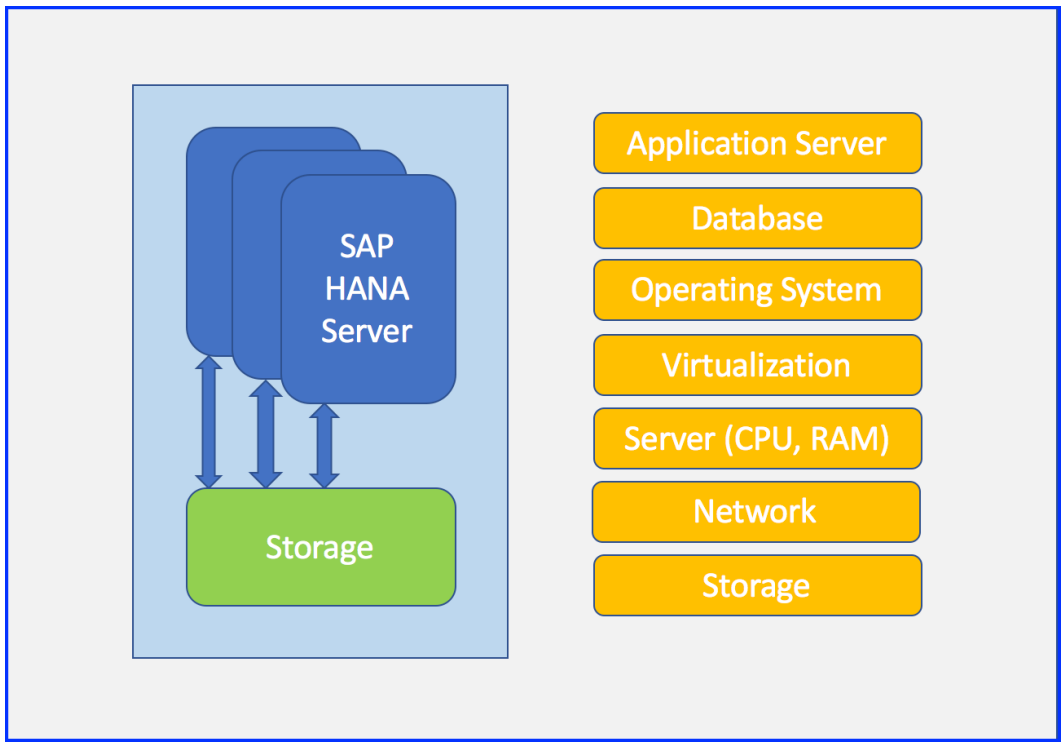
- SAP HANA is a combination of hardware and software made to process massive real time data using in-memory computing.
- SAP HANA combines row-based, column based database technology.
- Data now resides in main memory (RAM) and no longer on a hard disk. Since everything is in RAM all the time, it gives CPUs quick access to data for processing. The speed advantage is further accelerated by the use of multi-core CPUs, and multiple CPUs per board, and multiple boards per server appliance. Complex calculations on data are not carried out in the application layer, but are moved to the database.

SAP HANA Deployment Types

SAP HANA is now deployable in the cloud or as an on-premises appliance that is pre-installed and configured by certified partners, including HPE, IBM, Fujitsu, Hitachi, Cisco, Dell, Huawei, NEC, and VCE. Organizations can run SAP HANA on existing certified enterprise-class storage using the *SAP HANA Tailored Data Center Integration model*.

Appliance Delivery Model — SAP delivers SAP HANA in the form of standardized and highly optimized appliances. It is possible for companies to choose between several SAP HANA hardware partners. SAP has partnered with leading hardware vendors (HP, Fujitsu, IBM, Dell etc) to sell SAP certified hardware for HANA. SAP is selling licenses and related services for the SAP HANA product which includes the HANA database, easy to use data modeling tool called HANA studio and other software to load data in the database. If you prefer the delivery of a pre-configured hardware setup with pre-installed software packages that can be quickly implemented by your SAP HANA hardware partner of choice, then an appliance is the right delivery model for you. It is fully supported by both the hardware partner and SAP. While the appliance delivery is easy and comfortable, it might introduce some limitations regarding hardware flexibility and it may require changes to your established IT operation processes.

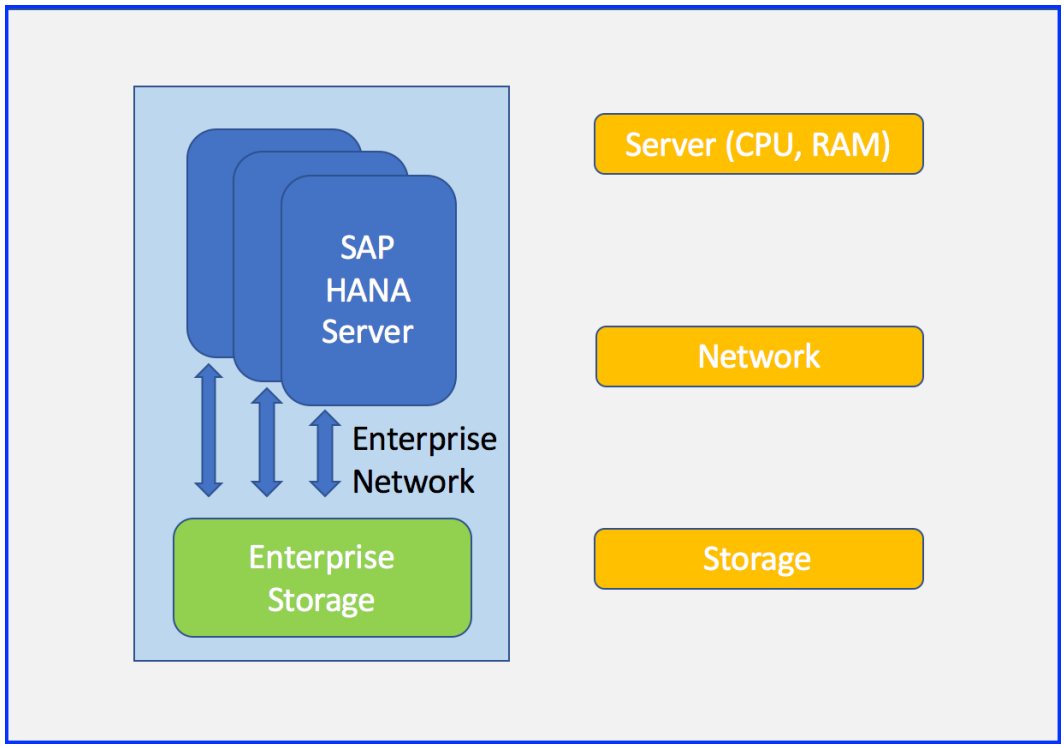
Figure 4-1. SAP HANA Appliance Delivery Model



SAP HANA Tailored Data Center Integration Model — This option allows customers to use certain parts of their existing hardware and infrastructure components instead of using the corresponding components that are delivered with a SAP HANA appliance. Since many VMware features require shared storage, leveraging *SAP HANA TDI* to deploy SAP HANA on shared storage in customer environments is the preferred deployment model to leverage features like VMware vSphere High Availability, VMware vSphere vMotion, VMware vSphere Distributed Resource Scheduler, and VMware vSphere Fault Tolerance. This model thus provides you with more flexibility regarding hardware components (compute servers, storage and network) required to run SAP HANA

In summary, SAP supports the scale-up deployment of SAP HANA for production use on VMware vSphere 6.0, part of the VMware vCloud Suite. Customers can now achieve the benefits of virtualization for SAP HANA environments, while leveraging all the components of VMware vCloud Suite 6.0 to build and run a vSphere based private cloud.

Figure 4-2. SAP HANA TDI Model



Comparison - Appliance Delivery Model vs SAP HANA TDI Approach —

Criteria	Appliance Delivery Model	SAP HANA TDI Approach
Hardware Selection	Little Flexibility ■ Customer can choose between different appliance vendors ■ No possibility to replace certain components by hardware already used in customer data center	Save IT budget and existing investments ■ Use preferred storage ■ Use preferred network components ■ A choice of computer server processor (example E5, E7)
Implementation Effort	Low for customer ■ Pre-configured hardware plus pre-installed software	Only hardware is delivered ■ Installation to be done by customer ■ Extensive documentation available (guides, SAP notes etc)
Safeguarding / Solution Validation	Done together by SAP and Hardware Partners	■ SAP HANA Going-Live check offered by SAP AGS ■ Self managed infrastructure tests possible ■ SAP HANA hardware configuration check tool
Support	Fully provided by SAP	Individual support agreements (with hardware partners required)
OS Services Contract	Appliance vendor is reseller of OS provider's service contract	Customer to care for getting the OS provider's service contract

SAP HANA - Benefits

SAP supports the scale-up deployment of SAP HANA for production use on VMware vSphere 6.0, the foundation of VMware vCloud Suite. Combining the power of SAP HANA in memory platform with VMware vSphere 6.0 help achieve the following benefits:

- Faster Time-to-Value:
  - Accelerate and automate provisioning
  - Reduce deployment time to hours from days (Source : Based on EMC IT internal analysis)
  - Use template provisioning to ensure consistency and scalability across environments along with cloning capabilities
  - Easier lifecycle management by leveraging VMware Adapter for SAP Landscape Management for SAP HANA on vSphere
- Higher Service Levels:
  - With VMware vSphere vMotion, live migrate SAP HANA across hosts in minutes with zero downtime and zero data loss
  - In the case of server outage, VMware vSphere High Availability ensures 99.9% (Source : EMC IT, 2/14 EMC Perspective, H12853) high availability by automatically restarting virtual machines on other hosts in the cluster, without manual intervention. Maximum uptime by automatically restarting SAP HANA virtual machines with VMware vSphere High Availability
  - Continually changing the business and workload demands of mission-critical apps make it challenging to meet service levels and end-user expectations alike.
    VMware vSphere Distributed Resource Scheduler allows IT to automatically manage peak analytic workloads easily, by adjusting resource allocation levels to meet fluctuating demands
  - Ensure configuration consistency and compliance checks by leveraging *VMware Host Profiles*
  - Unified management tools allow management of SAP HANA environments using the same tools used to manage VMware-virtualized data centers. The unified management portal provides access to VMware-enabled abstraction, pooling and automation of the entire compute layer, enabling rapid application provisioning on any hardware stack and on vSphere private or public clouds
  - The on-demand self-service portal and catalog allows designated users to easily self provision infrastructure, platform and desktop services in minutes. At the same time, best-practice application architectures allow the building of reusable templates that can be shared across teams, organizations, and clouds. Both of these features greatly reduce time-to-market for business initiatives and new products
- Lower *Total Cost of Ownership* (TCO):
  - Reduce *CapEx* by 70% and *OpEx* (Source – Taneja Group Research 2014) by 56% through greater utilization of existing resources and infrastructure
  - Unify and manage SAP HANA with rest of your virtualized data center
  - Improve resource utilization through simplified operations management
  - Enabling scenario-based data center capacity planning further allows IT to grow the environment in a sustainable fashion, keeping long-term costs down.
- SAP HANA databases can be virtualized up to the maximum size of a virtual machine on the vSphere 6.0 release, which is 128 vCPUs and 4 TB of memory
- Unified Disaster Recovery for SAP HANA environments with Automated DR leveraging Site Recovery Manager and storage block level based replication

- Increase adoption of SAP HANA in the enterprise providing self service provisioning of instances to the private/public cloud with vCloud Automation Center
- Manage health, risk, and efficiency of SAP HANA virtual machines with the rest of the VMware virtualized private cloud environment with *VMware vCenter Operations Manager*

SAP HANA is typically deployed on a complete, pre-configured appliance, which is provided by a certified SAP HANA hardware partner. However, the *SAP HANA Tailored Data Center Integration (TDI)* option allows customers to leverage existing assets for some elements of the SAP HANA environment. The SAP TDI program, and the support for all SAP HANA and S/4HANA deployment options, coupled with VMware production support, provides greater flexibility for deploying SAP HANA beyond the physical appliance model. TDI reduces both *CapEx* and *OpEx* by allowing the reuse of existing hardware, as well as leveraging current operational management processes

To summarize, running SAP HANA on vSphere 6.0 is the next step in the evolution toward a software defined data center. It enables customers to transform and virtualize their entire SAP landscape – from transactional to analytic workloads.

Benefits of Virtualizing SAP HANA - References to Customer Opinions (Videos)

You can get a good perspective of customer opinions with respective to how they perceive the benefits of virtualizing SAP HANA, by watching the following vieos:

- Increase Agility with SAP HANA on the VMware Platform — http://bcove.me/lzsyf7sg
- More deployment options with SAP HANA on the VMware Platform — http://bcove.me/4l83u8h0
- Transform Business – Achieve Better Control and Faster Analytics — http://bcove.me/fro4rqig

SAP HANA Architecture

SAP HANA is an in-memory database and platform that is deployable on-premise or in the cloud. The SAP HANA platform is a flexible data source agnostic in-memory data platform that allows customers to analyze large volumes of data in real-time. It is also a development platform, providing an infrastructure and tools for building a high-performance applications based on SAP HANA Extended Application Services. It is the foundation of various SAP HANA editions, like the SAP HANA Platform Edition, providing core database technology, and the SAP HANA Enterprise Edition, bundling additional components for data provisioning.

The SAP HANA platform edition comprises of the following software components:

- SAP HANA Database
- SAP HANA Client
- SAP HANA Studio
- SAP HANA XS advanced runtime
- SAP HANA XS Engine
- SAP HAN Advanced Data Processing
- SAP HANA Spatial

SAP HANA features, SAP HANA capabilities, SAP HANA Options — They provide additional functions and you need a dedicated license for the options or capabilities that you want to use. Following are some of the SAP HANA options and the SAP HANA capabilities:

- SAP HANA Accelerator for SAP ASE
- SAP HANA Dynamic Tiering

- SAP HANA Remote Data Sync
- SAP Landscape Transformation Replication Server
- SAP HANA Smart Data Streaming

SAP HANA Multitenant Database Containers:

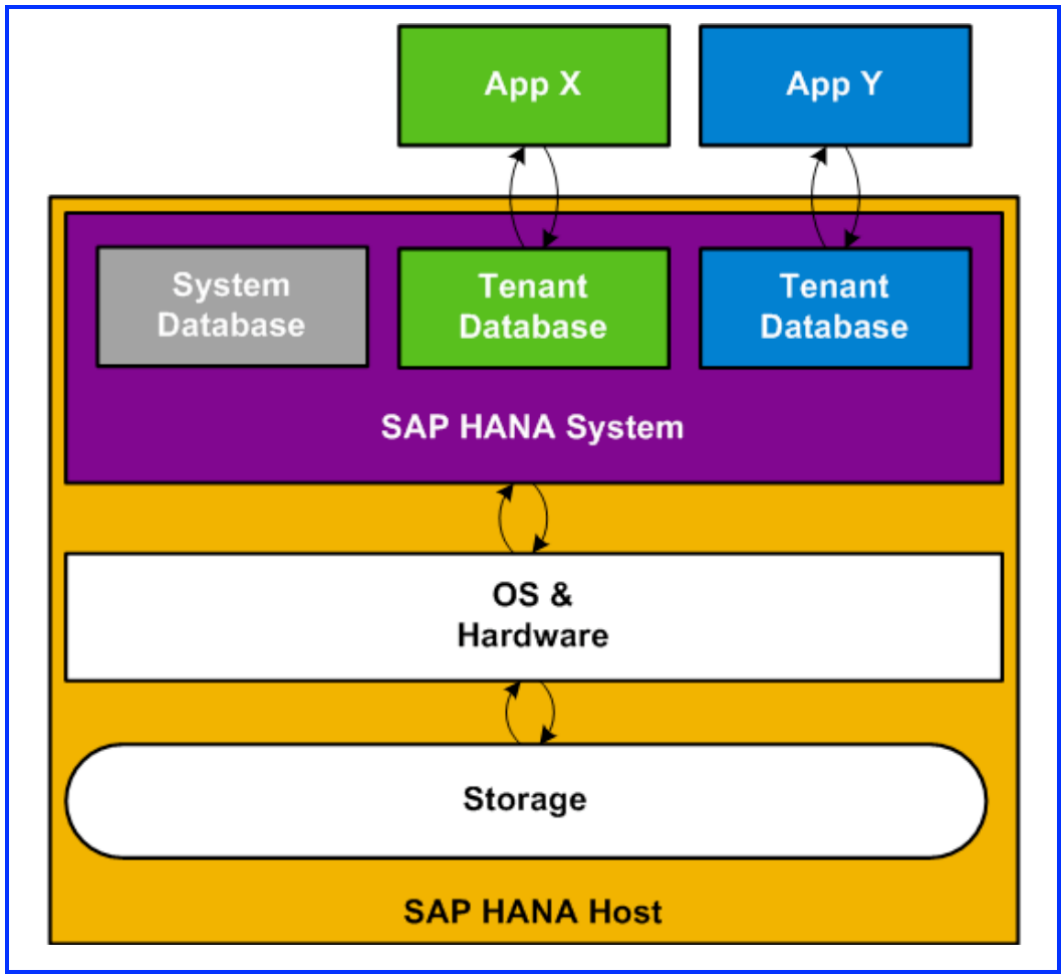
It si possible to install SAP HANA to support multitenant database containers. A SAP HANA system installed in this mode is capable of containing more than one multitenant database containers. Otherwise it is a single container system. Single container system can be converted into a multiple container system. A multiple container system always has one system database that is used for central system administration. There can be any number of multitenant database containers (including zero) also called tenant databases. A SAP HANA system installed in multiple container mode is identified by a single system ID (SID). Databases are identified by a SID and a database name. From the administration perspective, there is a distinction between tasks performed at system level and those performed at database level. Database clients such as the SAP HANA studio connect to specific databases.

All databases in a multiple container system share the same installation of database system software, the same computing resources and the same system administration. However each database is self contained and fully isolated with its own:

- Set of database users
- Database catalog
- Repository
- Persistence
- Backups
- Traces and logs

Although database objects such as schemas, tables, views, procedures and so on are local to the database, cross database SELECT queries are possible. This supports cross application reporting too. If you use a multiple container system you have one system database and any number of tenant databases. Multiple applications run in different tenant databases as depicted in the following figure:

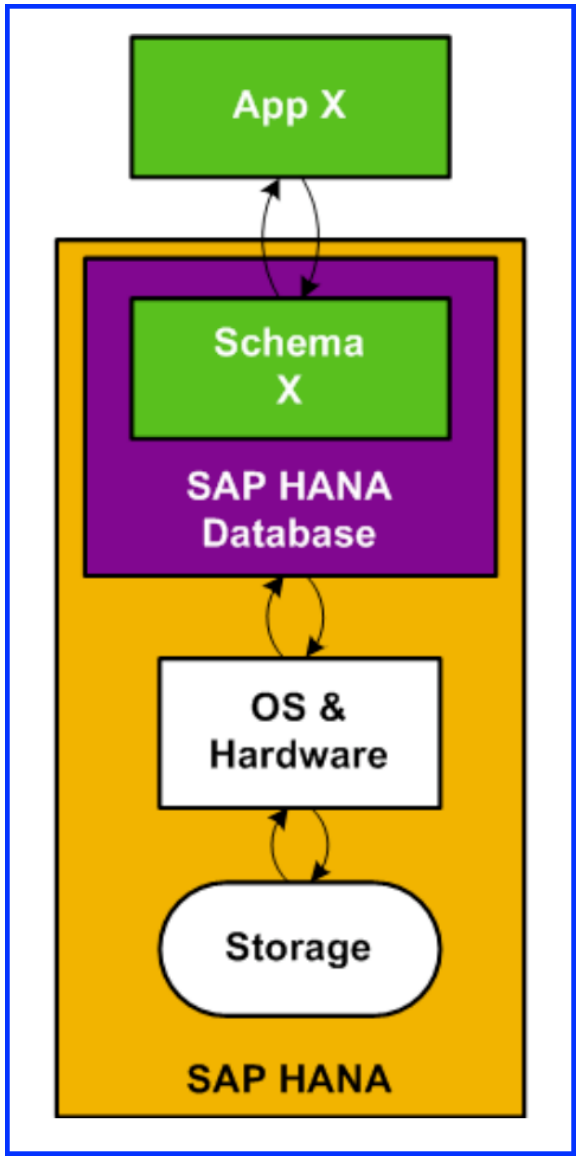
Figure 4-3. Multitenant Database Container



Single Application on One SAP HANA System (SCOS)

In this simple and straight forward scenario a single application runs in a single schema, in a single SAP HANA database as part of a SAP HANA system. It is pictorially depicted in the following figure. It is also referred to as Single Component on One System (SCOS):

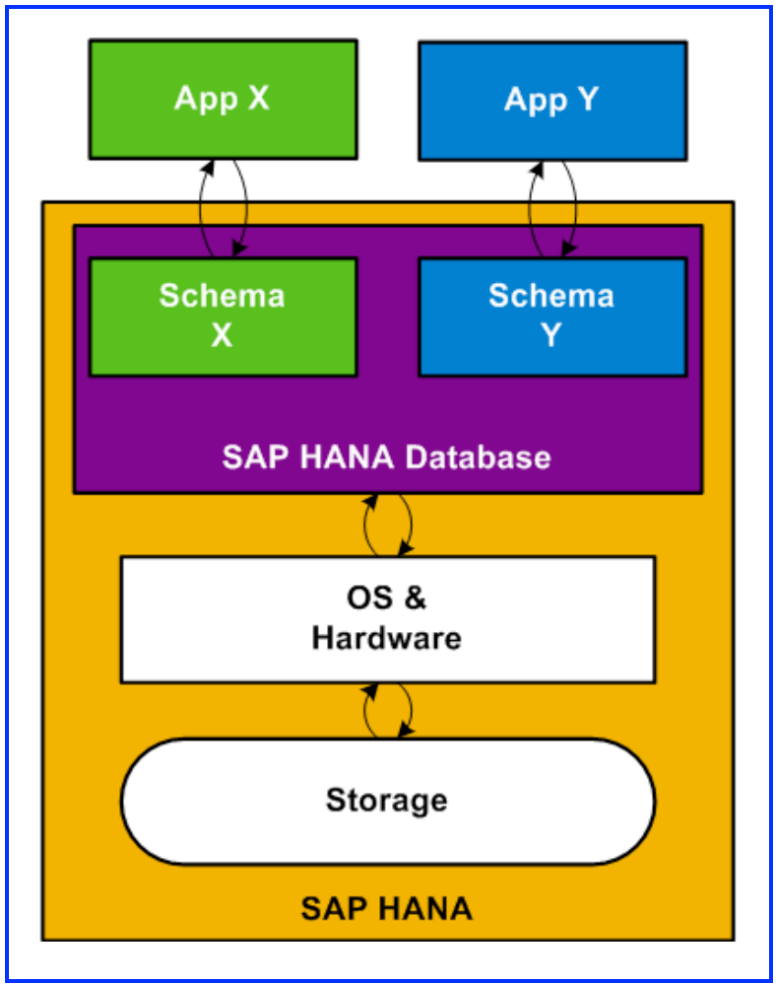
Figure 4-4. Single Component on One System (SCOS)



Multiple Applications on One SAP HANA System (MCOD)

Multiple applications on one SAP HANA system is also known as Multiple Components on One Database (MCOD). In this scenario more than one application run on a single SAP HANA system. This deployment type is available with restrictions for production SAP HANA systems.

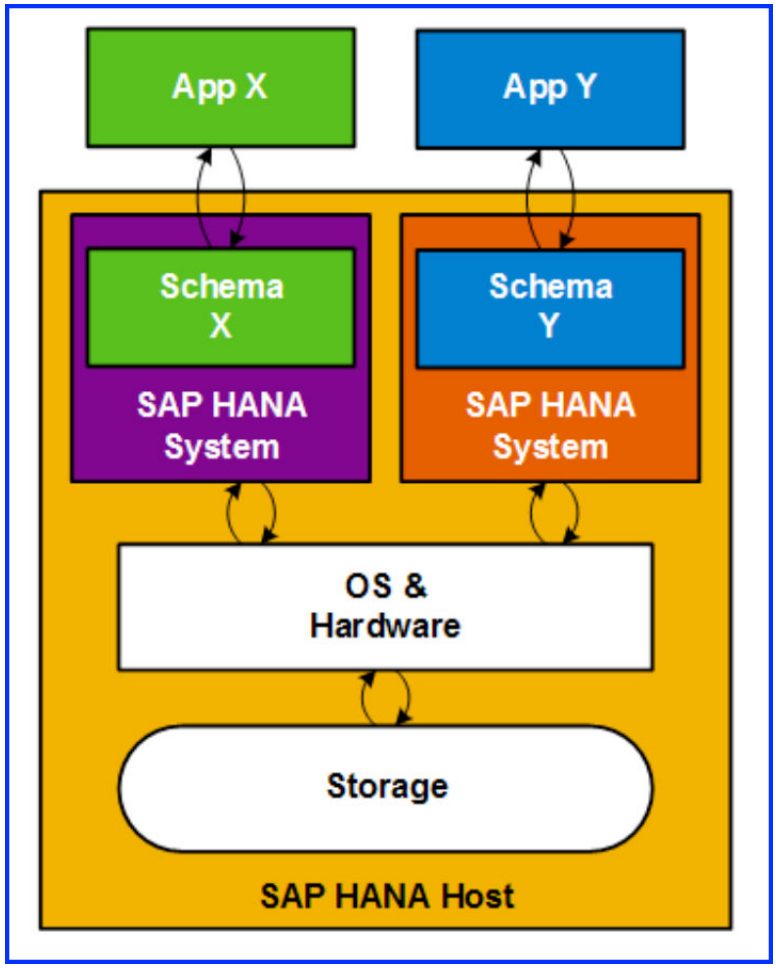
Figure 4-5. Multiple Components on One Database (MCOD)



Multiple SAP HANA Systems on One Host (MCOS)

Multiple SAP HANA systems on one host are also known as Multiple Components on One System (MCOS). SAP does support running multiple SAP HANA systems (SIDs) on a single production SAP HANA host. Note that multi DID requires significant attention to various detailed tasks related to system administration and performance management. Running multi-SID on one SAP HANA host may impact performance of various types of operations, due to resource contention issues.

Figure 4-6. Multiple Components on One System (MCOS)

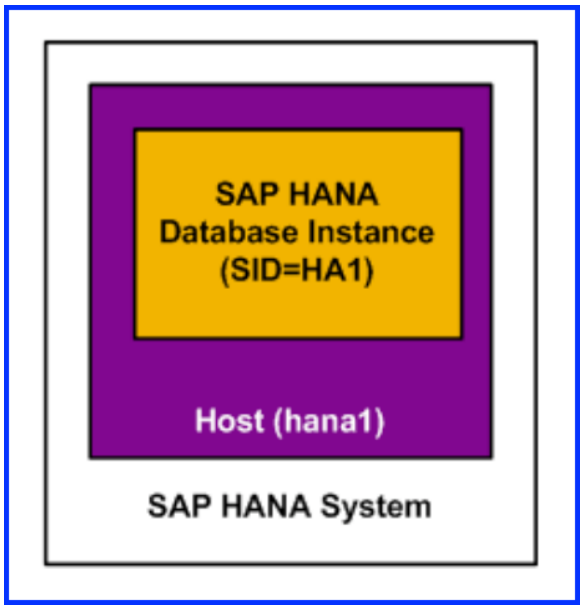


SAP HANA System Types

The number of hosts in a SAP HANA system landscape determines the SAP HANA system type. The host is the operating environment in which the database runs. The host provides all the resources and services (CPU, memory, network and OS) that the SAP HANA database requires. The host provides links to the installation directory and log directory or to the storage itself. The storage needed for an installation does not have to be on the host. In particular, shared data storage is required for distributed systems. A SAP HANA system can be configured as one of the following types:

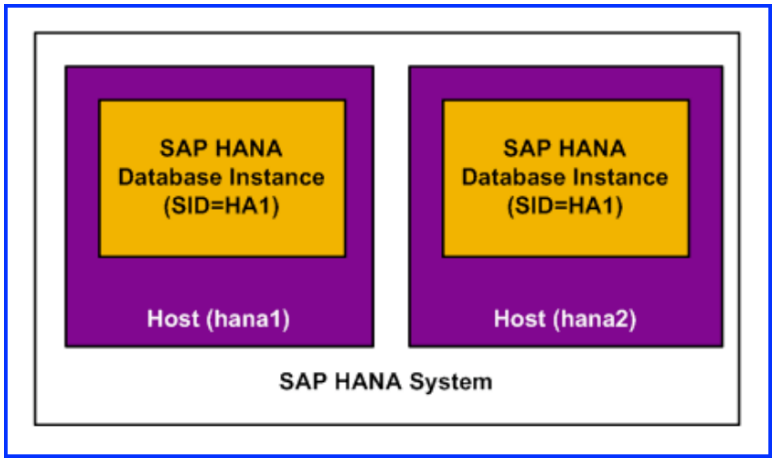
- Single host system — One SAP HANA instance on one host

Figure 4-7. Single Host System



- Distributed system (Multiple host system) — Multiple SAP HANA instances distributed over multiple hosts, with one instance per host. A distributed system might be necessary in the following cases —
  - You can scale SAP HANA either by increasing RAM for a single server, or by adding hosts to the system to deal with larger workloads. This allows you to go beyond the limits of a single physical server.
  - Distributed systems can be used for fail over scenarios and to implement high availability. Individual hosts in a distributed system have different roles (master, slave and standby), depending on the task.

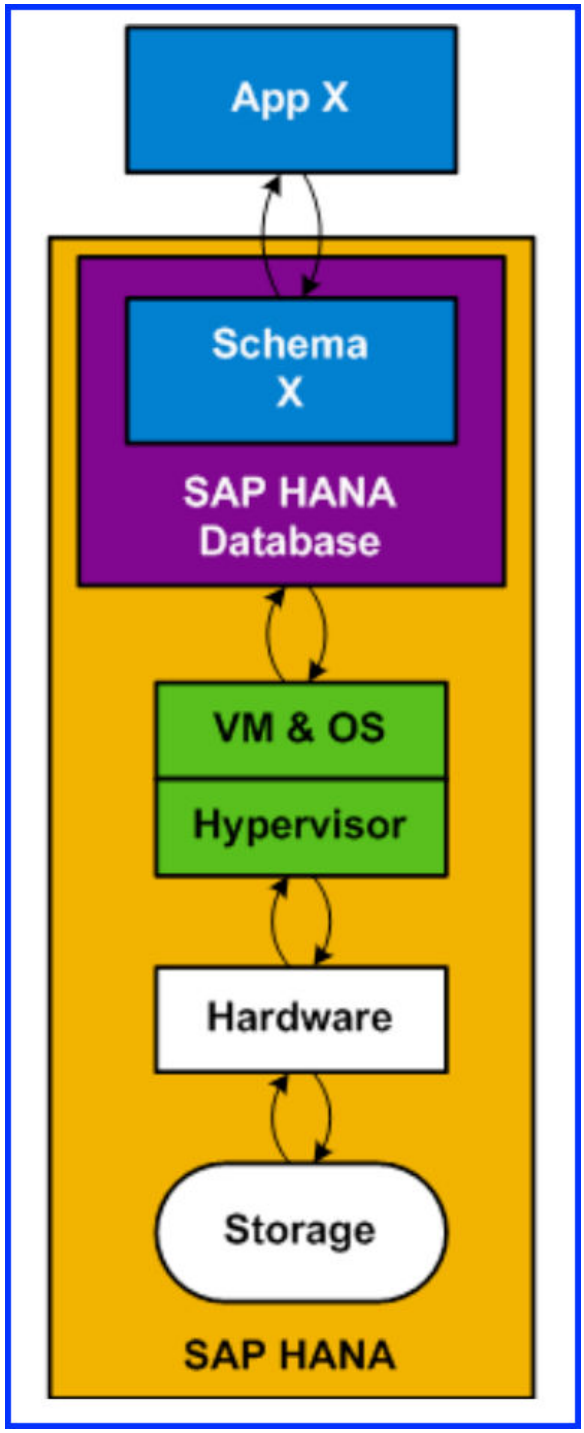
Figure 4-8. Multiple Host System



SAP HANA with Virtualization

Here one or more SAP HANA database SIDs are deployed on one or more virtual machines running on SAP HANA hardware.

Figure 4-9.



A SAP HANA data center deployment can range from a database running on a single host to a complex distributed system with multiple hosts located on one or more secondary sites, and supporting a distributed multi-terabyte database with full high availability and disaster recovery. In terms of network connectivity, SAP HANA supports traditional database client connections and with SAP HANA Extended Application Services, Web-based clients, SAP HANA can be integrated with transaction oriented databases using

replication services, as well as with high speed event sources, SAP HANA based applications can be integrated with external services such as email, web and R-code execution. The setup of a SAP HANA system and the corresponding data center and network configurations, depends on your company's environment and implementation considerations, some of which are:

- Support for traditional database clients, Web-based clients and administrative connections
- The number of hosts used for the SAP HANA system, ranging from a single host system to a complex distributed system with multiple hosts
- Support for high availability through the use of standby hosts and support for disaster recovery through the use of multiple datacenters
- Security and performance

SAP HANA had different types of network communication channels to support the different SAP HANA scenarios and setups:

- Channels used for external access to SAP HANA functionality by end-user clients, administration clients, application servers, and the data provisioning via SQL or HTTP
- Channels used for SAP HANA internal communication within the database or in a distributed scenario for communication between hosts.

SAP HANA supports the isolation of internal communication from outside access. To separate external and internal communication, SAP HANA hosts use a separate network adapter with a separate IP address for each of the different networks. SAP HANA can also be configured to use SSL for secure communication.

Separate network zones, each with its own configuration, allow you to control and limit network access to SAP HANA to only those channels required for your scenarios, while ensuring the required communication between all components in the SAP HANA network. The network zones can be described as follows:

- Client zone — the network in this zone is used by SAP application servers, by clients such as the SAP HANA studio or Web applications running against the SAP HANA XS server, and by other data sources such as SAP Business Warehouse.
- Internal zone — This zone covers the interhost network between hosts in a distributed system as well as the SAP HANA system replication network
- Storage zone — This zone refers to the network connections for backup storage and enterprise storage.

SAP HANA on VMware vSphere - Features

By running the SAP HANA platform virtualized on VMware vSphere, SAP customers can leverage an industry standard data center platform, optimized for agility, high availability, cost savings and easy provisioning. SAP customers will not only gain the ability to provision instances of SAP HANA in virtual machines much faster but also benefit from unique capabilities like:

- Increased security and SLAs (example through NSX or DRS)
- Live migration of running SAP HANA instances with VMware vSphere vMotion
- Standardized High Availability, based on VMware vSphere High Availability (HA)
- Built-in multi-tenancy support, through system encapsulation in a virtual machine (VM)
- Abstraction of the hardware layer
- Higher hardware utilization rates
- All the above features help lower the total cost of ownership and ensure the best operational performance and availability

SAP now supports SAP HANA on VMware vSphere 6 deployments for production workloads (see SAP note 2315348 for further details). The support for VMware vSphere 6 allows customers to increase the RAM up to 4 TB of existing virtual SAP HANA systems when migrated to VMware vSphere 6 and allows to react on increased memory needs due to data growth or newly deployed SAP HANA scenarios and solutions. In addition, upto 128 vCPUs can now be configured and used by a single SAP HANA VM. Supporting more physical compute resources inside the VM provides more power to the virtualized SAP HANA system.

The following table lists the capabilities, supported deployment options and best practices for SAP HANA VMs on VMware vSphere:

Table 4-1.

Capability / Option	vSphere 5.5		vSphere 6.0	
	Supported in Production	No SAP Support, VMware best effort support	Supported in Production	No SAP Support, VMware best effort support
SAP HANA Scale-Up VM<= 1024 GB	Yes		Yes	
SAP HANA Scale-UP VM <= 4080 GB	No		Yes	
SAP HANA Scale-Out VM <= 1024 GB	Yes		No	Yes
SAP HANA Scale-Out <= 4080 GB	No		No	Yes
SAP HANA Multi-VM	Yes		Limited	Yes
SAP HANA Version	SPS07 and above		SPS11 and above	SPS09 and above
	VMware vSphere and SAP HANA HA and Operation Features		**VMware vSphere and SAP HANA HA and Operation Features**	
vSphere HA	Yes		Yes	
SAP HANA Host Auto-Failover	Yes		Yes	
SAP HANA System Replication	Yes		Yes	
VMware FT	No		No	
VMware SRM	Yes		Yes	
vSphere vMotion	Yes		Yes	
VMware DRS	Yes		Yes	
	Supported HW Configurations		**Supported HW Configurations**	

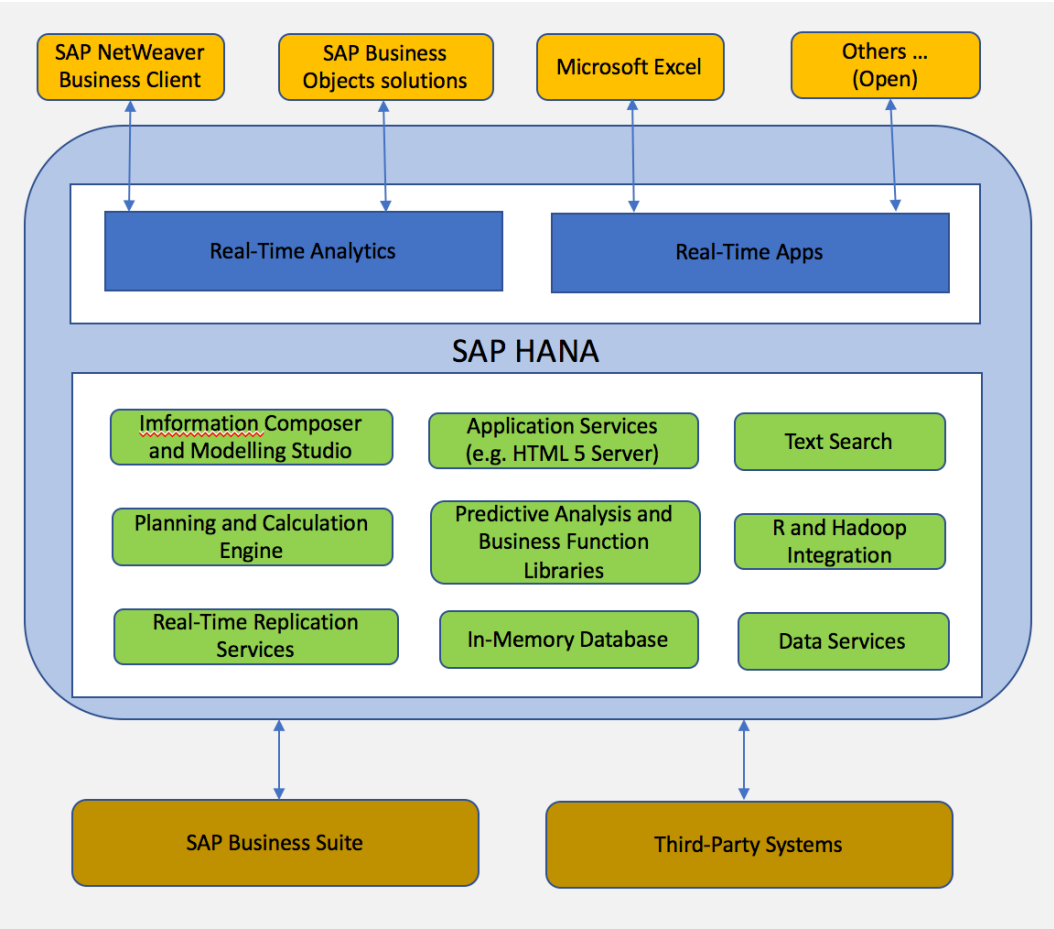
Table 4-1. (Continued)

Capability / Option	vSphere 5.5		vSphere 6.0	
Supported SAP HANA Systems for VMware Virtualization	Only SAP HANA and VMware certified two-, four- and eight socket Intel E7 v2 (Ivy Bridge) and later Intel processor-based server systems and Intel Xeon E5 v3 and v4 based two-socket single node SAP HANA entry level systems, with a minimum of eight cores per CPU are supported.		Only SAP HANA and VMware certified two-, four- and eight socket Intel E7 v2 (Ivy Bridge) and later Intel processor-based server systems and Intel Xeon E5 v3 and v4 based two-socket single node SAP HANA entry level systems, with a minimum of eight cores per CPU are supported.	
Intel Xeon E7 CPU Support	Ivy Bridge, Haswell		Ivy Bridge, Haswell, Broadwell (SPS12 or later)	
NUMA Nodes per server	4	up to 8 (4 with Broadwell)	up to 8 (4 with Broadwell)	
Intel Xeon E5 CPU Support	Ivy Bridge, Haswell, Broadwell (SPS12 or later)		Ivy Bridge, Haswell, Broadwell (SPS12 or later)	
NUMA Nodes per server	2		2	
NUMA Node Sharing	No	Yes	No	Yes
Enable Hyperthreading	Yes		Yes	
Maximal RAM installed in server (Host RAM)	up to 6 TiB		up to 12 TiB	
	Supported Storage Configuration		**Supported Storage Configuration**	
Supported SAP HANA Storage Systems for Virtualization	All SAP HANA TDI and VMware certified / supported storage solutions can get used.		All SAP HANA TDI and VMware certified / supported storage solutions can get used.	
TDI Storage KPI Requested	Yes, per HANA VM	No	Yes, per HANA VM	No
	SAP HANA Virtual Machine Configuration		**SAP HANA Virtual Machine Configuration**	
Max VM size #vCPU	64		128	
Max VM size #vRAM	1024 GB		4080 GB	
Minimal VM size #vCPU	All threads of a single CPU socket	10	All threads of a single CPU socket	10
Minimal VM size #vRAM	RAM of CPU socket	As Sized	RAM of CPU socket	As Sized
	Supported OS for virtualized SAP HANA		**Supported OS for virtualized SAP HANA**	
SLES 11 and 12	Yes		Yes	
RHEL 6 and 7	Yes		Yes	

SAP HANA Deployment on VMware vSphere 6.0 for Production Environments

SAP provides production support for SAP HANA scale-up deployments on VMware vSphere 6.0 part of the VMware vCloud Suite. You know that SAP HANA is an in-memory database that massively improves performance of existing SAP applications, and enables business transformation via real-time analytics and transaction execution. It is deployable both in the cloud and as an on-premise appliance that is pre-installed and configured by certified partners. It is a revolutionary platform that is best suited for performing real-time analytics and developing and deploying real-time applications. SAP HANA platform converges database and application platform capabilities in-memory to transform transactions, analytics, text analysis, predictive and spatial processing to enable businesses to operate in real-time. At the core of this real-time data platform is the SAP HANA database. SAP HANA platform architecture enables converged, online transaction processing (OLTP) and online analytical processing (OLAP) within a single in-memory, column-based data store using the ACID (Atomicity, Consistency, Isolation, Durability) compliance model and it eliminates data redundancy and latency.

Figure 4-10. SAP HANA Platform Architecture



You can run SAP HANA on existing certified enterprise class storage using the *SAP HANA Tailored Datacenter Integration Model*. You can purchase virtual SAP HANA (SAP HANA on vCloud Suite) from your existing SAP HANA OEM appliance vendor or via installation of virtual SAP HANA on existing IT infrastructure via the *SAP HANA Tailored Datacenter Integration Model*.

SAP and VMware support SAP HANA systems in scale-up deployment, up to the maximum size of a virtual machine (VM) on vSphere 6.0 which is 128 vCPUs and 4 TB of memory. For example, a 1 TB SAP HANA database is comprised of approximately 512 GB of compressed data, the remainder of the RAM is utilized for Linux OS, temporary tables, intermediate calculations and other SAP HANA database structures. A single production level SAP HANA virtual machine on a dedicated SAP HANA certified server is supported. Co-deployment of non-production level SAP HANA or non SAP HANA systems is allowed, as long as the production level SAP HANA VM gets configured with resource commitments. Two, four or eight socket SAP HANA certified Intel E7 v2 Ivy Bridge EX or Intel E7 v3 Haswell processor based single node configurations are supported. The correspondingly supported entry level systems are Intel Xeon E5 v2 and later based two socket single node systems with a minimum of eight cores per CPU. You can use VMware vSphere vMotion, VMware vSphere Storage DRS(DRS) and VMware vSphere High Availability(HA) to achieve operational performance and availability. Note that the SAP HANA databases are sized in an identical manner for physical ad virtual environments.

Multiple virtual machines can be deployed on a single server. With vSphere 5.5 co-deployment of multiple production SAP HANA VMs is supported. But with vSphere 6.0 co-deployment of one vSphere 6.0 production SAP HANA VM and several other non-SAP HANA VMs or non-production level SAP HANA VMs is supported. SAP and Vmware will jointly support virtual SAP HANA in production, adhering to the service-level agreements (SLAs) defined in teh customer support contract. If a reported problem is a known SAP HANA issue with a validated fix, SAP support will recommend the appropriate fix directly to the customer. For all other performance related issues, the customer will be referred within SAP's OSS system to VMware staff for support. VMware will take ownership and work with the SAP HANA hardware/operating system partner, SAP and the customer to identify the root cause and resolve the issue.

You can choose to deploy vCloud Suite on any SAP HANA appliance which comes either pre-installed from the appliance vendor for SAP HANA or can also be installed and verified as documented in the SAP HANA TDI approach. You should be aware that SAP HANA is only supported on SAP HANA and VMware certified hardware in your production environment.

Additional References - Virtualize SAP HANA

Following are some of the additional useful sources of Information —

- Single SAP HANA VM on VMware vSphere 6 in production - http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/vmware-sap-support-note.pdf
- SAP HANA Tailored Data Center Integration - FAQ — http://sapassets.edgesuite.net/sapcom/docs/2016/05/e8705aae-717c-0010-82c7-eda71af511fa.pdf

SAP HANA - Deployment in Virtualized Environments, benefits and experiences (Videos):

- VMware vRealize Operations Management Pack for SAP HANA — http://bcove.me/0t3hnh5s
- Manage Business for Real-time - SAP HANA session at VMworld 2014 — http://bcove.me/794ogypm
- EMC Data Center Transformation with SAP HANA and VMware vSphere EMC — http://bcove.me/fwkwi7s6
- Transform Business - Achieve Better Control and Faster Analytics HP — http://bcove.me/fro4rqig
- Deliver Fast, Agile and Resilient Performance in a Virtualized Environment IBM — http://bcove.me/xcw1wrxw
- Reduce Complexity with SAP HANA on the VMware Platform Capgemini — http://bcove.me/a5fmmdaz
- Increase Agility with SAP HANA on the VMware Platform Deloitte — http://bcove.me/lzsyf7sg
- More Deployment Options with SAP HANA on the VMware Platform Fujitsu — http://bcove.me/4l83u8h0
- Running SAP HANA on VMware vCloud Suite - vMotion Demo — https://www.youtube.com/watch?v=vx_TQA5Gfx8

- SAP HANA Disaster Recovery over longer distances — https://www.youtube.com/watch?v=eksfJzrfFmE

Useful Technical Resources:

- SAP HANA on vSphere and NetApp FAS Systems - http://www.netapp.com/us/media/tr-4338.pdf
- SAP HANA on vSphere 5.5 Best Practices Guide — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/sap_hana_on_vmware_vsphere_best_practices_guide-white-paper.pdf
- SAP HANA on VMware vSphere Technical Resource Guide — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/sap-hana-on-vmware-vsphere-5-5-best-practices-resource-guide.pdf
- SAP Sybase IQ on VMware vSphere — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-sap-sybase-iq-deployment-guide.pdf
- Distributed Query Processing in SAP IQ on VMware vSphere and Virtual SAN — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-sap-iq-vsan-perf.pdf
- SAP Sybase Adapter Server Enterprise on VMware vSphere — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/sap-sybase-adaptive-server-enterprise-on-vmware-vsphere.pdf
- Rubicon Phase 1 Whitepaper — http://www.emc.com/collateral/white-papers/h13185-project-rubicon-sap-hana-team-wp.pdf
- Virtual SAP HANA Disaster Recovery on VMware vSphere Using EMC Recoverpoint — https://community.emc.com/docs/DOC-36121
- Scale-Out Deployments on SAP HANA on VMware vSphere — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/vmware-sap-hana-scale-out-deployments-on-vsphere.pdf

Solutions Overviews and White Papers:

- HP ConvertedSystem 500 for SAP HANA and VMware vSphere Brochure — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/business-critical-apps/vmware-hp-convergedsystem-500-for-sap-hana-and-vsphere-brochure.pdf
- SAP HANA on VMware vSphere for Production Environments Solution Brief — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/business-critical-apps/vmware-saphana-brief-hires.pdf
- SAP HANA on VMware vSphere FAQ — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/business-critical-apps/vmware-sap-hana-on-vsphere-5.5-for-production-environments-faq.pdf
- Virtualize your SAP Environment - A Joint Solution from SAP and VMware to Increase IT Agility and Minimize Virtualization Risk — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/business-critical-apps/vmware-virtualize-your-sap-environment-joint-solutions-brief.pdf
- SAP Sybase Adaptive Server Enterprise on VMware vSphere - Essential Deployment Tips — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/business-critical-apps/vmware-sap-sybase-adaptive-server-enterprise-on-vSphere.pdf
- TCO and ROI Analysis of SAP Landscapes using VMware Technology — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/partners/sap/sap-tcoroi-customers-final-white-paper.pdf

- Get Professional Services for Virtualizing SAP — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/business-critical-apps/vmware-assess-create-and-adopt-a-virtualization-and-cloud-strategy.pdf
- Disaster Recovery of Tier 1 Applications on VMware Site Recovery Manager — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/disaster-recovery-tier1-app-on-vmware-vcenter-site-recovery-manager.pdf

VMware Adapter for SAP Landscape Management 5

The *SAP Landscape Manager (LaMa)* is used to manage the entire SAP landscape. But how does it integrate into the *VMware Software Defined Data Center (SDDC)* environment ? The first topic in this chapter addresses this question. You get to know what the VMware Adapter for SAP Landscape Management is and what it does. In the reference architecture section you get the big picture of the *VMware LaMa Appliance (VLA)* execution environment, its components and the relationship between them. The following section points you to the *VMware Adapter for SAP Landscape Management Installation, Configuration, and Administration Guide for VI Administrators* guide. You need to go through this document to understand how to install and configure the VMware Adapter for SAP Landscape Management. Finally the last section points you to various additional sources of information.

This chapter includes the following topics:

Understanding VMware Adapter for SAP Landscape Management

Introducing the VMware LaMa Adapter

SAP has a solution to manage *SAP landscapes* called *Landscape Manager (LaMa)*. *LaMa* is a central management point for SAP Basis administrators that allows mass operations, automation of day-to-day administrative and lifecycle management tasks, and automation of the copy, clone, and refresh of SAP systems. VMware has an adapter that integrates *SAP Landscape Management* with *VMware Software Defined Data Center (SDDC)* technologies for automated provisioning and management of a virtualized SAP system. For more information on the VMware Adapter for SAP Landscape Management, http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/sap-lvm-datasheet.pdf

What the VMware Adapter for SAP Landscape Management Does ?

The VMware Adapter for SAP Landscape Management enables following functionalities —

1 Mass SAP System Start and Stop – Start and stop of SAP systems on VMware Virtual Infrastructure

2 Copy a VMware based virtual SAP system – Enables creation of a system copy of a virtual SAP system while the system is live and online. The system copy operation can be either a linked copy or a full copy

3 Migrate a SAP system from a Source Host to a Target Host – Enables migration of a SAP system on a virtual host from a source system to a target system, leveraging the VMware vSphere vMotion Technology

4 Clone a VMware based virtual SAP system – Enables creation of a system clone of a virtual SAP system while the system is live and online. The system clone can be either a linked clone of a full clone

5 Migrate Storage in a SAP system from a Source Datastore to a Target Datastore – Enables migration of storage of a SAP system on a Source Datastore to a Target Datastore, leveraging the VMware vSphere Storage vMotion Technology

6 Refresh a VMware based virtual SAP system – Create a system copy of a virtual SAP system for a QA refresh of the production system while the system is live and online. The system copy can be either a linked copy or a full copy

7 Migrate a SAP system from a Source Network to a Target Network – Migrate a SAP system on a source virtual network to a target virtual network, leveraging the VMware vCenter Orchestrator, workflows and vCenter Server network selection technology.

Useful References — VMware Adapter for SAP Landscape Management

1 Release Notes — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/vmw-adapter-sap-lvm-release-notes.pdf

2 Simplify DevOps for VMware Virtualized SAP Landscapes — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/infographic/vmw-lvm-infographic.pdf

3 Product Open Source Download — http://www.vmware.com/go/dl-sap-lvm-oss

4 FAQ — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vmware-sap-lvm-faq.pdf

Reference Architecture

The following diagram illustrates the components of a VLA execution environment and their relationship to one another:

Figure 5-1. VLA Execution Environment

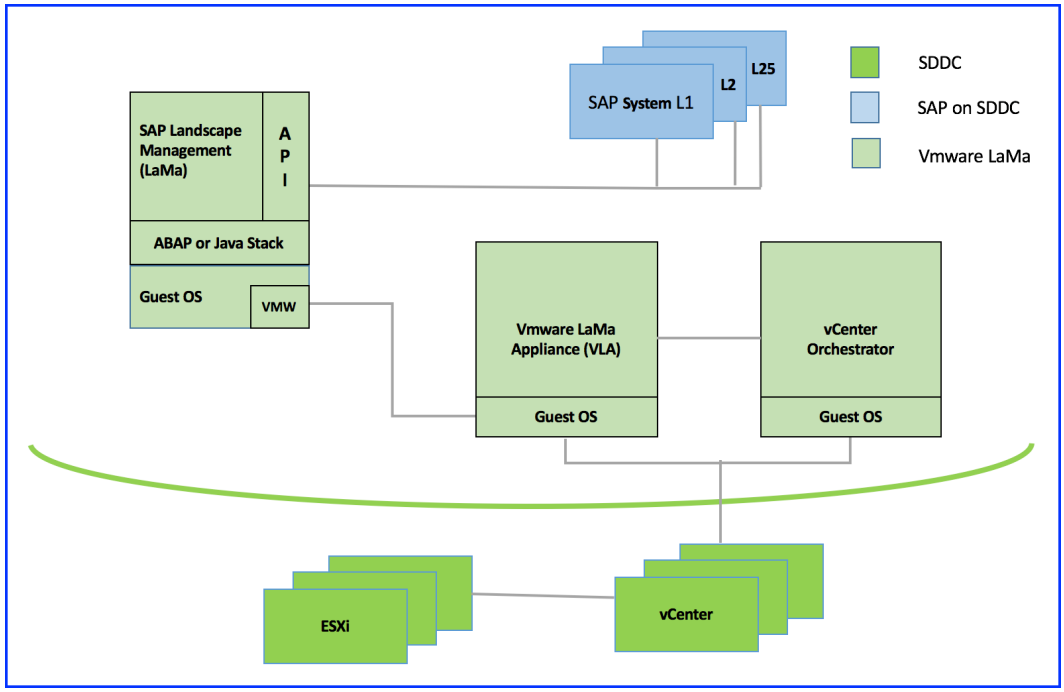
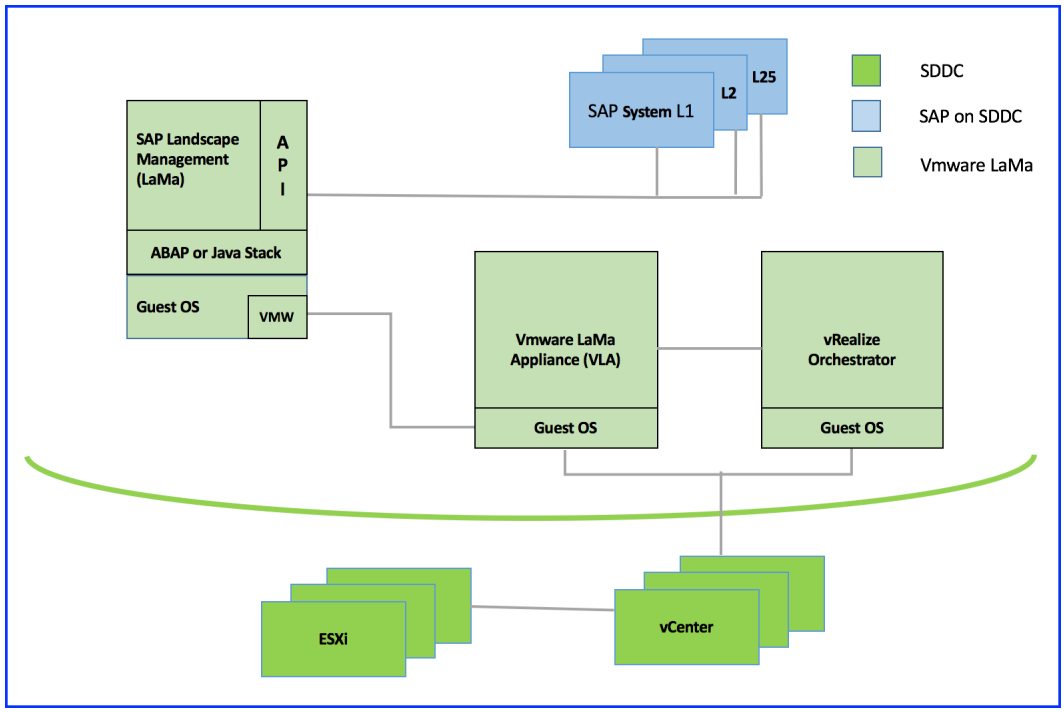


Figure 5-2. VLA Execution Environment



The key components in this diagram are :

- SAP Systems – Each of these systems consist of software running on one or more machines (bare metal, or in the case of VLA environments, virtual machines [VMs] hosted on VMware vSphere™ products [ESXi systems managed by vCenter Server™]) that perform some business function, such as order processing, accounts payable, general ledger, inventory management, etc. Each SAP System consists of one or more components. When all of the components are up and running, the SAP System is running. When all of the components are stopped, the SAP system is stopped. If some systems are running and some are not, the SAP system is in an intermediate state.
- The *SAP Landscape Management (LaMa)* VM – The SAP Landscape Management (LaMa) application runs on ABAP or Java stack in a Linux based guest OS. It provides a web-based user interface for SAP BASIS administrators to create / destroy / configure / and otherwise operate on and provision SAP Systems and their underlying machinery (bare metal or virtualized).

  The *SAP Landscape Management (LaMa)* has an extensible architecture that allows SAP and third-party vendors, for example VMware, to create plugins to extend certain features.
- The VMware Adapter for SAP Landscape Management — This is a plugin to LaMa that extends how LaMa integrates with the underlying systems virtualized with VMware vSphere (see next bullet), optimizing and extending the functionality for certain operations, such as activating (powering on) and deactivating (powering off), copying and cloning systems, and automation of these copying and cloning operations.

  Note Automation of copying and cloning SAP Systems involves the concept of a *LaMa Provisioning Template (LPT)*, which is different from a VMware vSphere Template (see next bullet).
- ESXi and vCenter Server (collectively called vSphere) – ESXi is VMware's premier hypervisor product. VI administrators typically install it on bare-metal server-class computers, with VMs running guest operating systems (OSes) with SAP Systems as applications within the guests. vCenter Server is VMware's premier product for managing environments virtualized with ESXi. Collectively called

vSphere, these products provide an enterprise-class environment with features for creating clusters, load balancing VMs between host systems (ESXi instances), fault tolerance, virtual networking, virtual storage, and more. In VLA environments, the VLA appliance (next bullet) runs in a VM on this infrastructure.

Note It is possible to run ESXi in a nested environment. In this case, VI administrators install ESXi in a VM running in a vSphere environment. This can be especially advantageous for `SA-API` developers, as it can allow you to quickly deploy new development environments from vSphere virtual application (vApp) Template, which are different from *SLPT*. For more information on vSphere Templates, see http://pubs.vmware.com/vsphere-60/index.jsp?topic=%2Fcom.vmware.vsphere.hostclient.doc%2FGUID-F40130B0-0194-4A41-91FA-1A967721924B.html. For more information about vApps, see http://pubs.vmware.com/vsphere-60/index.jsp?topic=%2Fcom.vmware.powercli.ug.doc%2FGUID-CFCCBEAC-74DD-4259-9D9D-1FCCCB185218.html

---

- VMware vCenter Orchestrator™ – This VMware product helps VI administrators automate their environments by creating work flows (essentially scripts) that perform VI administrative actions, including complex actions that may take multiple steps, involve loops, conditions, etc. VMware vCenter Orchestrator workflows can handle exceptions automatically or can pause waiting for a VI administrator to mitigate an issue. See the next bullet for how VLA uses VMware vCenter Orchestrator.
- VMware Landscape Management Appliance (VLA) – This part of the VLA product is a virtual appliance. Collectively, it consists of one or more web services that accepts commands from (previously discussed) *LaMa* VLA Adapter and/or SA-API clients and take appropriate actions to implement the commands, typically with the help of the (previously discussed) VMware vCenter Orchestrator. For example::
  - When a SAP BASIS administrator activates (powers on) a SAP System via LaMa, the VLA Adapter sends commands to the `vla-service` (discussed later in this topic) to power on the underlying VMs. The `vla-service` in turn invokes a VLA-specific workflow on the VMware vCenter Orchestrator to turn on the VMs in the underlying vSphere infrastructure. An analogous action occurs when a SAP BASIS administrator deactivates (powers off) a SAP System.
  - When a SAP BASIS administrator copies a SAP System, the VLA Adapter sends commands to the `vla-service` which in turn invokes a VLA-specific VMware vCenter Orchestrator workflow to create vSphere copies of the VMs on which the SAP Systems reside, configuring the VMs according to the parameters provided by the SAP BASIS administrator in the LaMa web user interface.
  - When a vCloud Director (vCD) script invokes the SA-API to power-on a business-critical application, the sa-api-service (discussed later in this topic) sends commands to VLA-service similar to those discussed in the first example in this list.

  The VLA Appliance contains several components, including:

  - A purpose-configured and hardened operating system (OS)
  - A minimalist set of OS utiltiies and VLA-specific programs and configuration files required to provide the functionality described here. These include:
    - The `vla-service` — A web service running in `tomcat` that receives and processes commands from the VLA Adapter. It also serves out the VLA dashboard web UI. By default, this server listens on port 8443.
    - Tomcat user database — Database with usernames / passwords used to authenticate access to that instance's services. VI Administrators create an entry in the database for the VLA instance during deployment of the VLA environment using the `vla_user` command as detailed later in this document.

- A `credentials store` (separate from the username / password database for tomcat access) that contains information needed for the various components of the VLA environment to communicate with one another. Each entry in the credentials store includes a component type (vCenter Orchestrator, *LaMa*, vCenter Server etc), the hostname and port (if configurable) for the component's API, and a username / password used to authenticate to the component's API. You create entries in this database using the `vla_credentials` command as detailed later in this document.
- `sa_api-server` — A web service running in `tomcat` that serves out the SA-API. This is also known to as the *SA-API engine*. The purpose of this API is the furtherance of VMware's Software Defined Data Center (SDDC) strategy by allowing automating the management of business critical applications, such as SAP.

Install and Configure - VMware Adapter for SAP Landscape Management

VMware Adapter for SAP Landscape Management Installation Configuration and Administration Guide for VI Administrators — You need to go through this guide to better understand the Installation, Configuration and Administration tasks for the VMware Adapter for SAP Landscape Management.

Additional References - VMware Adapter for SAP Landscape Management

Following are some useful technical references:

- Installation and Configuration Guide — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vmw-adapter-sap-lvm-installation-guide.pdf
- Users Guide — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vmw-adapter-sap-lvm-user-guide.pdf
- SAP on VMware vSphere — https://wiki.scn.sap.com/wiki/display/VIRTUALIZATION/SAP+on+VMware+vSphere
- VMware vRealize Orchestrator Documentation — https://www.vmware.com/support/pubs/orchestrator_pubs.html
- vSphere Resource Management — https://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-resource-management-guide.pdf
- vRealize Operations Manager Customization and Administration Guide — https://pubs.vmware.com/vrealizeoperationsmanager-6/topic/com.vmware.ICbase/PDF/vrealize-operations-manager-601-cust-admin-guide.pdf
- http://www.vmware.com/in/products/adapter-sap-lvm.html

SAP Solution with VMware Products 6

High availability is the name given to a set of techniques, engineering practices and design principles that support the goal of business continuity. High availability is achieved by eliminating single points of failure (fault tolerance), and providing the ability to rapidly resume operations after a system outage with minimal business loss. Fault recovery is the process of recovering and resuming operations after an outage due to a fault. Disaster recovery is the process of recovering operations after an outage due to a prolonged data center or site failure. Preparing for disasters may require backing up data across longer distances, and may thus be more complex and costly.

This chapter includes the following topics:

SAP Solution with VMware High Availability

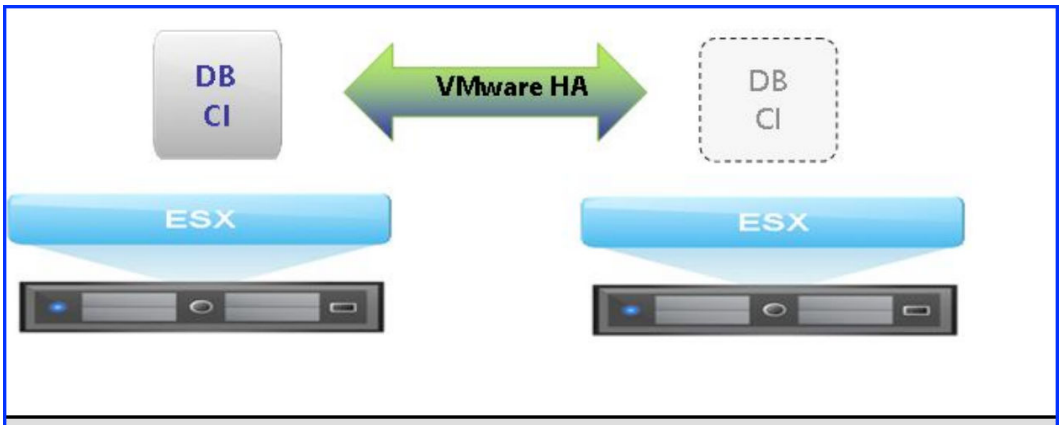
Business continuance (sometimes referred to as business continuity), describes the processes and procedures an organization puts in place to ensure that essential functions can continue in case of unplanned downtime. Unplanned downtime refers to an outage in system availability due to infrastructure failure (server, storage, network), or site disaster. SAP products and solutions provide mission-critical business processes that need to be highly available even in a site disaster.

SAP provides a range of enterprise software applications and business solutions to manage and run the complete business of a company. These mission critical systems require continuous availability. SAP has a scalable, fault-tolerant, multi-tier architecture, components of which can be protected either by horizontal scalability (e.g., NetWeaver application servers) or by cluster and switch over solutions that protect the single points of failure in the SAP architecture – they include the database, message and locking services. The latter two are included in constructs referred to as the Central Instance (CI) or ABAP SAP Central Services (ASCS).

VMware vSphere High Availability (HA) continuously monitors all VMware ESXi™ hosts in a cluster and detects hardware failures. The VMware HA agent placed on each host maintains a heartbeat with the other hosts in the cluster using the service console network. Each server sends heartbeats to the other servers in the cluster at regular intervals. If any servers lose heartbeat, VMware HA initiates a failover action of restarting all affected virtual machines on other hosts.

The following figure depicts a scenario with the SAP database and Central Instance running in a single virtual machine with VMware HA applied. You can also see the summary of the available features in this configuration.

Figure 6-1. SAP with High Availability



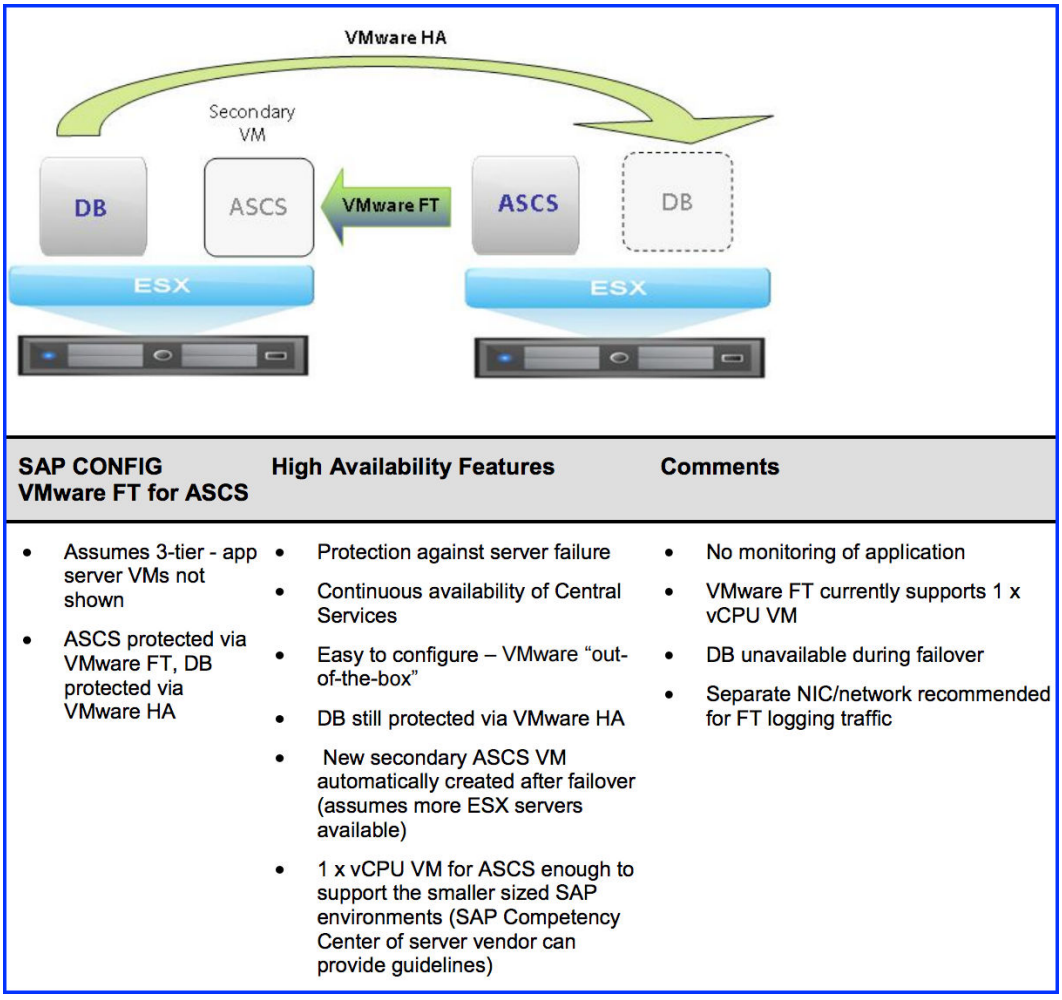
SAP Config VMware HA	High Availability Features	Comments
• 2-tier or 3-tier – app server VMs not shown • protected via VMware HA	• Protection against server failure • Auto restart of VMs • Startup scripts/service required to auto-start SAP/DB instances in guest OS • VMware HA easy to configure (VMware “out-of-the-box”)	• No monitoring of application • DB unavailable during failover • No enqueue and message services during CI failover

SAP Solution With VMware Fault Tolerance

Fault Tolerance (FT) relies on *VMware vLockstep Technology* to establish and maintain an active secondary virtual machine that runs in virtual lockstep with the primary virtual machine. The secondary virtual machine resides on a different host and executes exactly the same sequence of virtual (guest) instructions as the primary virtual machine. The secondary observes the same inputs as the primary and is ready to take over at any time without any data loss or interruption of service should the primary fail. Both virtual machines are managed as a single unit, but run on different physical hosts. By allowing instantaneous fail over between the two virtual machines, FT enables zero downtime for the application deployed within the virtual machine.

VMware vSphere Fault Tolerance only one virtual CPU and is a good candidate for the *Central services component*. The following figure depicts a high availability configuration of the SAP database and ASCS. Look at the table that summarizes the features of this setup. The database virtual machine is protected by VMware vSphere High Availability and the ASCS virtual machine by VMware vSphere Fault Tolerance.

Figure 6-2. Example - SAP Solution with VMware FT



SAP CONFIG VMware FT for ASCS	High Availability Features	Comments
• Assumes 3-tier - app server VMs not shown • ASCS protected via VMware FT, DB protected via VMware HA	• Protection against server failure • Continuous availability of Central Services • Easy to configure – VMware "out-of-the-box" • DB still protected via VMware HA • New secondary ASCS VM automatically created after failover (assumes more ESX servers available) • 1 x vCPU VM for ASCS enough to support the smaller sized SAP environments (SAP Competency Center of server vendor can provide guidelines)	• No monitoring of application • VMware FT currently supports 1 x vCPU VM • DB unavailable during failover • Separate NIC/network recommended for FT logging traffic

SAP Solution with VMware SRM

VMware vCenter Site Recovery Manager (SRM) provides business continuity and disaster recovery protection for virtual environments.

Disaster recovery testing comprises a logistical plan for how an organization will recover and restore partially or completely interrupted critical function(s) within a predetermined time after a disaster or extended disruption. Common wisdom states that any disaster recovery plan is only as good as the last (successful) test. Disaster recovery efforts can fail because the IT team neglects to test often, the result being an insurance policy that does not pay off when disaster hits.

Disaster recovery testing is often difficult because it is usually very disruptive, expensive in terms of resources and extremely complex. By leveraging virtualization, SRM addresses this problem while making planning and testing simpler to execute.

Using SRM, two sites are involved—a protected site and a recovery site. SRM leverages array-based replication between a protected site and a recovery site to copy virtual machines.

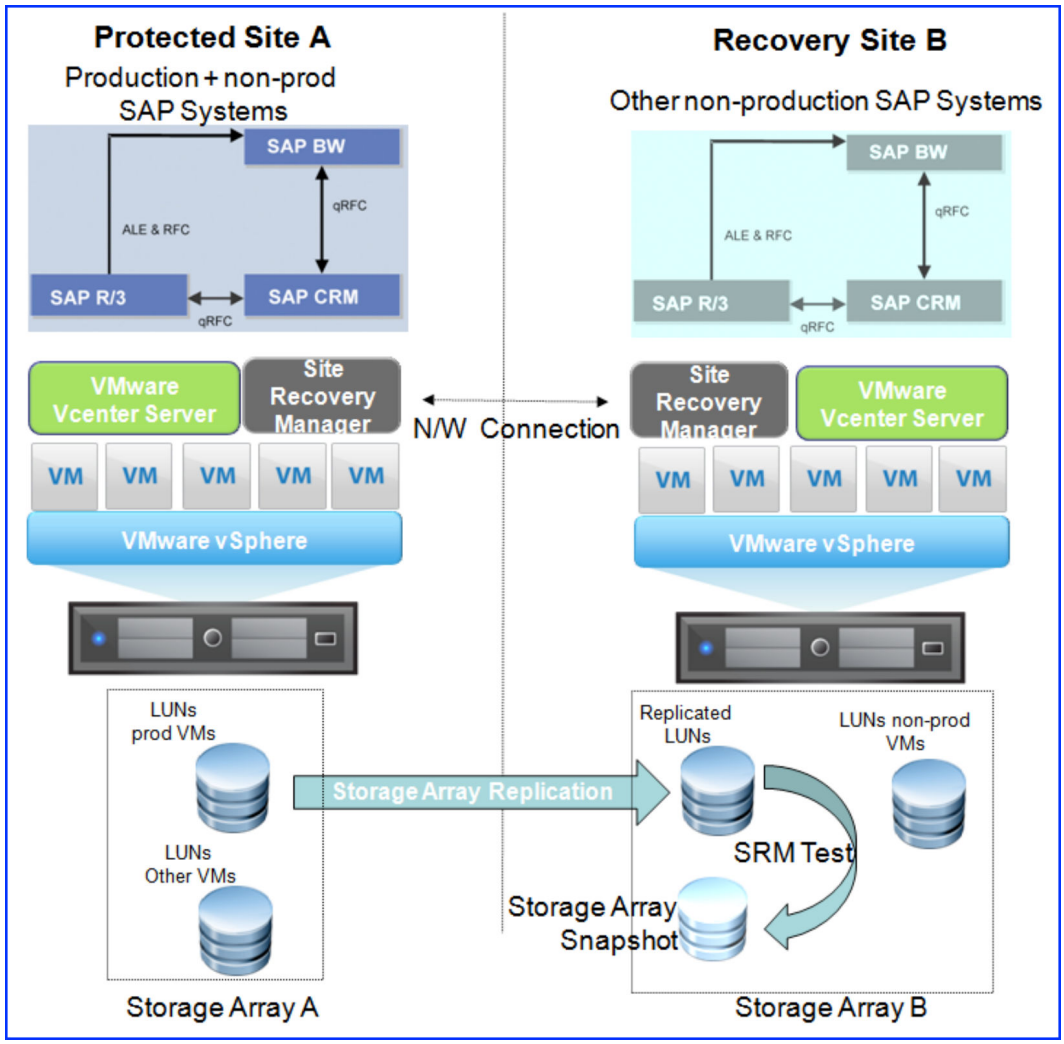
Recovery Point Objective (RPO) and *Recovery Time Objective (RTO)* are the two most important performance metrics IT administrators need to keep in mind while designing and executing a disaster recovery plan. RPO is addressed by the storage provider who provides certified storage replication adapters that integrate with SRM to enable a fully automated test or real recovery. RTO is addressed by the SRM recovery plans that automate the startup sequence of multiple virtual machines that comprise a virtualized SAP landscape and automate network connectivity at the remote site.

Site Recovery Manager Architecture

An SAP landscape can consist of a considerable number of separate systems to host the multiple SAP products, each with separate production and non-production systems. In the production environment, multiple SAP systems typically interface to a myriad of third-party bolt-on applications. In addition, the multi-tier architecture of Netweaver may result in separate tiers of application and database servers. Hence, a fully virtualized SAP environment results in numerous virtual machines with data interfaces/flows between these virtual machines. Such a volume of virtual machines can be managed with the workflow features of SRM that process the correct sequence and order of recovery of virtual machines after a site failure.

The following figure shows the architecture of a deployment of a virtualized SAP landscape with SRM. In this example, production SAP systems are replicated from the protected to a recovery site. Each site hosts a separate storage array. Customer-specific business requirements determine if non-production systems also need to be replicated and protected against site failure. Other non-production SAP systems are hosted at the protected site. The SAP landscape is logically depicted here by three SAP systems for simplicity, each of which is connected via interfaces to demonstrate that business processes can traverse separate systems.

Figure 6-3. SAP with SRM



Architecture Overview

- VMware ESXi hosts at the recovery site run some non-production systems to maximize resource usage; these servers do not need to be idle. Another scenario (not shown here) based on two-way storage array replication is feasible with SRM whereby production systems can be split between the two sites and each can be acting as a fail over to the other.
- A SRM server is installed both at the protected and recovery site. Both sites are managed by their own VMware vCenter Server. The SRM Server operates as an extension to the VMware vCenter Server and the SRM user interface installs as a vSphere client plug-in
- A certified storage array vendor is required that has an adapter which integrates with Site Recovery Manager. Storage array replication needs to be correctly installed and configured for Site Recovery Manager to operate. This should follow the same process as the physical environments. Site Recovery Manager automatically detects the replicated LUNs that contain virtual machines
- The protected and recovery sites should be connected by a reliable IP network. Storage arrays might have additional network requirements for replication. The SRM servers at both sites communicate with each other during normal operations
- On the protected site, production virtual machines are replicated via storage array replication. On the recovery site (storage array B), the replicated LUNs are not visible to the VMware ESXi hosts

Executing Recovery Plans

- Protection groups are created on the protected site. A protection group is a collection of virtual machines that all use the same set of replicated LUNs and fail over together.
- Recovery plans are created at the recovery site and are created from the protection groups. The recovery plan is essentially an automated runbook that consists of a set of steps that control what happens during a fail over:
  - The recovery plan determines the order of production virtual machine startup during a fail over and also can suspend non-production virtual machines already running at the recovery site. Enough server resources are required at the recovery site to run the production systems, as well as any non-production systems that are also needed to run per business requirements (otherwise, non-production systems can be suspended by the recovery plan).
  - Call outs to custom scripts can be included in the recovery plan for customer specific requirements.
  - The SAP application can be configured to auto start after a guest OS boot within the virtual machine.
  - The execution of the recovery plans enable customers to achieve faster RTO.
- Recovery plan can be executed in either of two modes:
  - Actual fail over – array replication is halted and the replicated LUNs on the recovery site are enabled for read and write capabilities, and SRM initiates the power up of the virtual machines in the recovery site according to the startup order in the recovery plan. SRM does not automatically detect a site disaster—recovery has to be manually started via the SRM user interface at the recovery site.
  - Test fail over – the replicated LUNs on the recovery site still remain unavailable to the VMware ESXi hosts. They are copied using storage array snapshot functionality and these copied snapshot LUNs are presented to the VMware ESXi hosts. The snapshot is reasonably quick as data is not duplicated (this is part of the storage array feature). The production virtual machines are started according to the recovery plan and can then be user tested. After testing is complete, a manual step to continue via the SRM user interface stops the production virtual machines and removes the storage array snapshot. Meanwhile, any suspended non-production virtual machines are started again on the protected site. During this test cycle the replicated LUNs are still being refreshed per the storage array replication schedule, and production systems continue to function normally on the protected site.

Network Customization

Typically there are separate networks at the protected and recovery sites. While each network should be connected via routers, the subnet and IP address will differ between the locations. Therefore, when performing site fail over, IT administrators can be faced with the following challenges on the recovery site:

- Network properties of the production virtual machines need to be customized according to the network specification of the recovery site.
- Domain Name Server (DNS) records pertaining to these virtual machines need to be updated.

After fail over to a disparate network, the network properties of virtual machines such as IP addresses, Gateway and DNS domain all need to change to return to a functional state. SRM addresses this at the recovery site via the following features:

- Customization Specification Manager – this allows administrators to create a custom network specification for each production virtual machine that is replicated from the protected site. Network properties (IP address, gateway, etc) can be assigned to the virtual machine so that when it starts up in a recovery plan it will function correctly on the recovery site network. The hostname of the guest OS in the virtual machine needs to remain the same so as not to impact the SAP application (SAP instance files once installed have the hostname of the OS in various configuration and startup files, but IP address is not hard coded in the files)

- After changing the IP addresses of virtual machines, DNS records of the virtual machines need updating.

Refer to the following document for further details — http://communities.vmware.com/docs/DOC-11516

Storage Array Replication

As previously mentioned, the SRM solution requires storage array tools to replicate the LUNs from the protected to the recovery site. Storage array replication needs to be installed and configured in the same manner as in physical environments, and administrators should follow guidelines from their storage vendor. Similarly, SAP database LUN layout on the storage array should follow the same recommendations as for physical environments. The major storage array vendors have SAP practices that have developed best practice guidelines for LUN layouts of SAP databases and how they should be replicated between separate sites in a disaster recovery scenario. The same guidelines should be followed with SRM. For example:

- Best practice for I/O performance requires production database virtual machines not to be shared with other virtual machines.
- Where applicable, some storage array vendors may prefer the use of RDMs as they are compatible with their disaster recovery tools. In these cases the virtual machine guest OS drive (—root‖ or —C:\‖) would be VMFS format and the database datafiles would be RDM-based

The RPO objective is managed by the storage array replication schedule. The frequency of replication and subsequent cost with respect to bandwidth requirements over a long distance is managed by the storage vendor specifications and is balanced against the business requirements. Two broad replication methods are available from storage vendors that impact RPO, and in both cases SRM does not manage the consistency of the SAP database during replication (quiescing of the database). This is addressed by the storage vendor technology or by separate procedures:

- Synchronous Replication – Guarantees zero data loss, where a write either completes on both sides or not at all. The storage vendor technology typically guarantees consistency of the database that is spread across multiple LUNs
- Asynchronous Replication – Write is considered complete as soon as local storage acknowledges it. The remote storage is not guaranteed to have the current copy of data. A potential scenario to guarantee database consistency in this situation involves putting the database into online backup mode before replicating. On a separate, more frequent schedule, replicate the database log files. Database recovery then involves starting the database and applying logs to roll forward the database. Such a process may be created manually or be part of tools/products from the storage vendor.

SAP on Virtual SAN

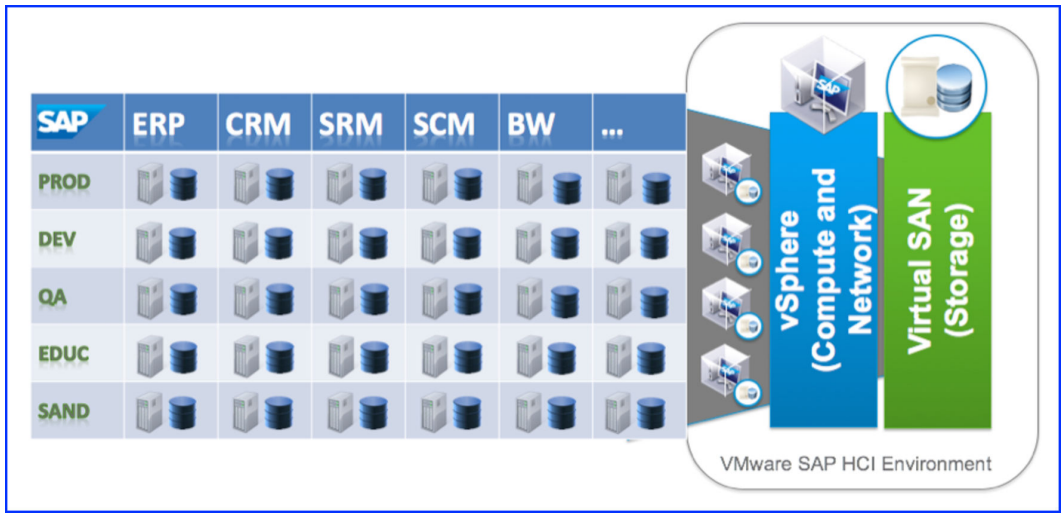
For SAP classic, non-SAP HANA applications, SAP does not require a specific storage certification. For details, refer to SAP note 2273806 (https://websmp230.sap-ag.de/sap/support/notes/2273806). As an integral part of the hypervisor, Virtual SAN is embedded in the vSphere kernel. Therefore, SAP treats it as a feature of the supported vSphere hypervisor and as such customers can readily use Virtual SAN for their SAP environments. In other words, Virtual SAN is fully supported for SAP applications, like SAP NetWeaver based products or SAP Business One, in production. For a complete list of currently supported SAP products on vSphere please review SAP notes 1492000 (https://websmp130.sap-ag.de/sap/support/notes/1492000) and 2161991 (https://websmp230.sap-ag.de/sap/support/notes/2161991).

In addition, VMware provides the one-stop support experience to customers running SAP applications on Virtual SAN. To escalate Virtual SAN related issues in these SAP environments, simply collect a VMware performance snapshot as described in note 1158363 (https://websmp130.sap-ag.de/sap/support/notes/1158363) and vSphere support bundles. Next open a ticket directly with VMware.

VMware takes it forward from there. If you are using other HCI solutions and land into a similar situation, you will have to first work with the respective solution vendors and prove that it is a vSphere related issue before you can escalate to VMware. Certainly the former is a simpler and faster solution especially in the context of Business Critical Applications (BCAs).

Customers wanting to modernize their existing aged and complex SAP environments, as shown in the figure below, can now consolidate their SAP landscapes on Virtual SAN based HCI solutions running on both VMware and SAP certified x86 servers, to eliminate traditional IT silos of compute, storage, and networking. All the intelligence and management moves into a single Hyper-Converged Software (HCS) stack, allowing a VM and application-centric policy-based control and automation. This brings better quality of service, security, higher performance, operational simplicity, and cost-effectiveness into SAP application environments.

Figure 6-4. SAP on vSAN



Traditional database systems that store majority of their data on disk will have VMs that have relatively small footprints. In contrast SAP HANA VM can easily be configured to reach the vSphere VM maximums of 128 vCPUs or 4 TB of RAM.

SAP HANA *Tailored Datacenter Integration* (TDI) is a program for allowing customers to leverage existing hardware and infrastructure components for their SAP HANA environment. Currently TDI targets enterprise network and enterprise storage solutions. All of the TDI certified enterprise storage solutions are SAN or NAS arrays today. Virtual SAN represents a totally different architecture, it is a paradigm shift of how storage is architected and consumed by HANA. Virtual SAN as a software-defined solution is able to satisfy HANA performance requirements and can therefore fulfill the needs of any other SAP applications.

Additional References - SAP Solution with VMware Products

- https://www.vmware.com/support/pubs/

SAP on VMware Best Practices 7

This chapter discusses some useful guidelines that provide general recommendations and deployment best practices for SAP HANA on VMware vSphere for the computing environment, networking and storage.

This chapter includes the following topics:

Deployment Best Practices - Computing Environment

When SAP HANA is deployed on the VMware vSphere platform, it enables an optimized, purpose-built virtual machine and computing environment. The first step to creating this type of environment is to carefully examine the BIOS settings, and disable any unnecessary processes and peripherals in order to direct the compute resources (CPU, memory, network and IO) to SAP HANA. The computing environment features that we cover in this section are:

- VMware vSphere host server BIOS settings
- Virtual Machine Guest Operating System
- Hardware-assisted memory virtualization
- Large (huge) memory pages
- Non-Uniform Memory Access (NUME)
- Wide Virtual Machines
- Virtual CPUs
- Hyperthreading
- Memory Considerations

VMware vSphere **Host Server BIOS Settings** — BIOS settings for and x86 server can be set to disable unnecessary processes and peripherals and to maximize performance. Some of the useful settings are described in the following table:

Table 7-1. Recommended BIOS Settings

BIOS Feature	Setting	Description
VT-x, AMD-V, EPT, RVI, VT-d, AMD-Vi	YES	Hardware-based virtualization support
Node Interleaving	NO	Enables NUMA
Turbo Mode	YES	Balanced workload over unused cores
Hyper-Threading	YES	Hyper-Threading is recommended
Power-Saving	NO	Disable if performance is more important than saving power
Power Management	OS Controlled Mode	Allow ESXi to control CPU power-saving features
C1E Halt State	NO	Disable for performance
Execute Disable	YES	Required for VMware vSphere vMotion and VMware vSphere Distributed Resource Scheduler features

Some of the settings that are not necessary and you may want to disable are — Video BIOS Cacheable, Video BIOS Shadowable, Video RAM Cacheable, On Board Audio, on Board modem, On Board Firewire, On Board Serial Ports, On Board Parallel Ports and On Board Game Port

Note Disabling hardware devices can free interrupt resources.

Virtual Machine Guest Operating System

SAP HANA is supported on SUSE Linux Enterprise Server (SLES) 11 Service Pack 2 or later. You should plan the operating system installation to ensure that it takes advantage of VMware virtualization and creates an optimized computing environment. It is important to ensure that the peripheral components are not re-enabled during operating system installation. But, simply disabling the peripheral components in the BIOS also does not guarantee that these components are fully disabled. After the installation of the OS, you should disable unnecessary foreground and background processes — for example, anacron, apmd, atd, autofs, cups, cupsconfig, gpm, isdn, iptables, kudzu, netfs and portmap. Turn off the SLES kernel dump function (kdump) if it is not needed for specific reasons. Configure the SLES kernel parameter as — net.ipv4.tcp_slow_start_after_idle = 0

Adhere to the shared memory settings (if they are not already set during installation) as per the following table:

Table 7-2. Recommended Shared Memory Settings

SIZE	SHMMNI VALUE	PHYSICAL MEMORY
Small	4096	>= 24GB and < 64 GB
Medium	65536	>64 GB and < 256 GB
Large	53488	> 256 GB

It is also recommended that you install the latest version of VMware Tools in the guest operating system. VMware Tools is a suite of utilities that enhances the performance of a virtual machine. Although a guest operating system can run without VMware Tools, many VMware features are not available until you install VMware Tools.

To minimize the time drift in virtual machines, follow the guidelines in SAP note 989963 — Linux: VMware timing problem and Timekeeping best practices for Linux guests (1006427) — http://kb.vmware.com/kb/1006427

You can set the Linux kernel IO scheduler to NOOP or Deadline. ESXi uses an asynchronous intelligent IO scheduler, so virtual guests should experience improved performance by allowing ESXi to handle IO scheduling — http://kb.vmware.com/kb/2011861

You can continue to use the Linux OS LVM (*Logical Volume Manager*) to manage disks for convenience and flexibility — http://kb.vmware.com/kb/1006371

Once the installation is complete, create a SAP HANA "gold" template or clone for consistency. For further details refer to the SAP HANA Server Installation Guide — https://help.sap.com/hana/SAP_HANA_Installation_Guide_en.pdf

Hardware Assisted Memory Virtualization

Some recent processors include a feature that addresses overhead due to memory management unit (MMU) virtualization by providing hardware support to virtualize the MMU. Without hardware-assisted MMU virtualization, VMware ESXi maintains shadow page tables that directly map guest virtual memory to host physical memory addresses. These shadow page tables are maintained for use by the processor and are kept consistent with the guest page tables. This allows ordinary memory references to execute without additional overhead, since the hardware translation lookaside buffer (TLB) caches direct guest virtual memory to host physical memory address translations that are read from shadow page tables. However, extra work is required to maintain the shadow page tables. Hardware assistance eliminates software memory virtualization overhead. In particular, it eliminates the overhead required to keep the shadow page tables in synchronization with the guest page tables, although the TLB miss latency is significantly higher. This means, hardware assistance provides workload benefits depending primarily on the memory virtualization overhead that is caused when using software memory virtualization. If a workload involves a small amount of page table activity, such as process creation, mapping the memory, or context switches, software virtualization does not cause significant overhead. However, workloads having a large amount of page table activity, such as workloads from a database, are likely to benefit from hardware assistance. Hence, enabling VMware vSphere to choose the best virtual machine monitor based on the CPU and guest operating system combination is recommended.

Large (Huge) Memory Pages

The operating system requirements for SAP HANA hardware appliances also apply to the SAP HANA virtual machines. It is recommended to use transparent HugePages to enable the use of large pages with SAP HANA. Large (huge) pages can potentially increase the TLB access efficiency, thereby improving the database performance. The use of large pages can significantly improve the performance of SAP HANA databases on VMware vSphere.

Non-Uniform Memory Access (NUMA)

The *Non-Uniform Memory Access (NUMA)* architecture is common on servers with multiple processor sockets. It is important to understand this memory access architecture when sizing memory-bound and latency-sensitive systems.

A NUMA node is equivalent to one CPU socket. For a server with two sockets, there are two NUMA nodes. Therefore, the available number of physical CPU cores and RAM can be divided equally among NUMA nodes. This is critical when sizing virtual machines to fit within one NUMA node. For example, a 4-socket, 40-core (10 cores on each socket) server with 512 GB RAM, has four NUMA nodes, each with 10 physical cores (CPUs) and 128 GB RAM (512 divided by 4). When sizing virtual machines, it is important to carefully consider NUMA node boundaries. For this example, if there are 10 vCPUs and 128 GB RAM. Exceeding the CPU and RAM size boundaries of a NUMA node causes the virtual machine to fetch memory from a remote location, and this can diminish performance.

It is recommended that you size the virtual machine to fit within the NUMA node. The ESXi scheduler has algorithms to keep the virtual machines within a NUMA node to optimize memory accesses.

Wide Virtual Machines

In cases where the CPU or RAM of a single NUMA node does not provide enough resources for a given workload, it is recommended to use a virtual machine that is larger than a single NUMA node. In this instance, performance becomes better, despite remote memory access, since the need for additional CPU or RAM has a larger effect on performance.

In general, size virtual machines to fit within as few NUMA nodes as possible, and do not oversize the virtual machine. This could cause unnecessary remote memory accesses. In addition, the vSphere ESXi CPU Scheduler is NUMA-aware and it has been designed to avoid remote memory accesses to the extent possible.

```
Wide virtual machine = number of vCPUs of virtual machine > number of cores in a socket/NUMA node
```

To summarize, configure vNUMA sockets for wide virtual machines that must cross NUMA nodes, for example, database virtual machines. ESXi exposes NUMA topology to the guest OS, allowing NUMA aware guest OSes and databases to make the most efficient use of the underlying hardware's NUMA architecture. When creating a virtual machine, you have the option to specify the number of virtual sockets and the number of cores per virtual socket. In general, leave this at the default value of one core per socket with the number of virtual sockets equal to the number of vCPUs. ESXi will automatically configure vNUMA to match the NUMA architecture of the host.

Virtual CPUs

When configuring SAP HANA virtual machines for production environments, ensure that the total vCPU resources for the virtual machines running on the system do not exceed the CPU capacity of the host. Do not over-commit CPU resources on the host. If the host CPU capacity is overloaded, the performance of the virtual database might degrade.

Configuring virtual SAP HANA with excess vCPUs can impose a small resource requirement on vSphere because unused vCPUs continue to consume timer interrupts. vSphere co-schedules virtual machine vCPUs and attempts to run the vCPUs in parallel to the best extent possible. Unused vCPUs impose scheduling constraints on the vCPU being used and can degrade its overall performance.

Hyperthreading

Hyperthreading for processors allows for multiple instruction threads to execute on a single physical core. While many of the core's resources are actually shared between cores, the additional logical thread allows for an increase in performance, usually in the range of 10 to 20 percent. It is recommended to enable hyperthreading on any system running vSphere and SAP HANA. This is very significant in the context of SAP OLTP workloads.

Virtual CPUs that are assigned to a vSphere virtual machine are mapped to a logical thread on the physical server on which they are running. When hyperthreading is enabled, each physical core has two logical threads. In order to assign all of the CPU resources to virtual machines, the number of virtual CPUs assigned needs to equal the number of logical threads on the server. For example, an Intel® Xeon® Processor E7 Series (Westmere-EX) based server that has 10 cores per socket and four sockets has a total of 40 physical cores and 80 logical threads.

See SAP Three-Tier Shows Excellent Scaling on VMware — http://blogs.vmware.com/performance/2010/03/sap-threetier-shows-excellent-scaling-onvsphere.html

Memory Considerations

For memory, the best practice is to configure memory reservations equal to the size of the SAP HANA configured RAM. Do not over commit memory. When consolidating multiple non-production SAP HANA instances on the same host, SAP HANA can share memory across all of the virtual machines that are running the same operating system. In this case, SAP HANA uses a proprietary, transparent page-sharing

technique to reclaim memory. This allows databases to run with less memory than when physical RAM is used. In this case, leave memory for overhead of the VMware ESXi kernel and for the virtual machines. As a conservative estimate, it would be safe to use up to 5 percent of system memory for this overhead. Do not assign memory overhead to virtual machines.

Note Beginning with SAP HANA 6.0, Transparent Page Sharing is enabled by default within virtual machines (intra-VM sharing), but is only enabled between virtual machines (inter-VM sharing) when those virtual machines have the same salt value, but by default each virtual machine has a different salt value. This provides the highest security between virtual machines, and will only have a small effect in most SAP deployments, because memory overcommit is not a priority. See http://kb.vmware.com/kb/2097593.

Set memory reservations equal to the size of the virtual machine for production systems under strict performance SLAs. This avoids memory ballooning, and therefore swapping in the virtual machine, and kernel swapping of the VMware vSphere hypervisor/ESXi host. Do not disable the balloon driver and memory compression. These are ESXi hypervisor techniques to reclaim virtual machine memory to increase consolidation.

Try to determine the right size for the configured memory of a virtual machine. Use SAP monitoring tools over a reasonable period to determine memory utilization. As memory reservations are used, an over-sized virtual machine wastes memory and reduces server consolidation.

Use large memory pages for databases. Large page support is enabled by default in ESXi and is supported in Linux and Windows. Using large pages can potentially increase TLB access efficiency and improve program performance. Follow SAP guidelines for using large pages.

Follow the same SAP Notes as per a physical deployment to configure the size of the OS swap space inside the virtual machine. This is independent of VMware. Any recommendations on OS swap sizing for SAP should be addressed by SAP or SAP Support.

If memory reservations are not set, VMware vSphere Distributed Resource Scheduler load-balancing recommendations could be suboptimal for SAP systems with large memory requirements. Follow the guidelines in DRS performs unwanted memory load balancing moves or DPM excessively consolidates memory — http://kb.vmware.com/kb/2059868

Deployment Best Practices - Networking

For SAP HANA the networking configuration includes Virtual Distributed Switch (vDS) and VMXNET3 optimization.

Virtual Distributed Switch (vDS) — SAP HANA uses Virtual Distributed Switch. vDS enables virtual machine networking that spans multiple vSphere hosts to be managed as a single virtual switch from a centralized VMware vCenter Server through VMware vSphere Web Client / VMware vSphere Client.

When a vSphere host is added, the networking for that host does not require configuration. Instead, the host is added to a defined port group that is dedicated to SAP HANA traffic or other application-specific traffic. In addition to supporting Private VLANs (PVLANs), vDS can also be used to shape both inbound and outbound network traffic. VMware Standard Switches can be easily migrated to vDS in a non-disruptive manner with the vCenter Server management user interface.

VMXNET 3 — The best practice is to use VMXNET 3 virtual NICs for SAP HANA virtual machines. VMXNET 3 is the latest generation of paravirtualized NICs that are designed from the ground-up for performance and latency-sensitive workloads. VMXNET3 offers several advanced features including multi-queue support, Receive Side Scaling, IPv4/IPv6 offloads, and MSI/MSI-X interrupt delivery. By default, VMXNET 3 also supports an adaptive interrupt coalescing algorithm for the same reason that physical NICs implement interrupt coalescing. Virtual interrupt coalescing helps drive high throughput to the virtual machines with multiple vCPUs with parallelized workloads. Red Hat Enterprise Linux 6 and SUSE Linux Enterprise Server 11 SP1 ship with built-in support for VMXNET 3 NICs.

Deployment Best Practices - Storage

For SAP HANA deployments, a typical storage configuration includes *Virtual Machine File System (VMFS)*, datastores, and Paravirtualized SCSI adapters.

The storage features are described in the following sections:

- Virtual Machine File System (VMFS)
- Datastores
- Paravirtual SCSI Adapters
- Multiple Virtual SCSI Controllers
- File System Considerations and Alignment
- SUSE Linux I/O Scheduler

Virtual Machine File System

VMware vSphere VMFS provides high performance, clustered storage virtualization that is optimized for virtual machines. With VMFS, each virtual machine is encapsulated into a small set of files. VMFS is the default storage management interface that is used to access those files on physical SCSI disks and partitions. VMFS allows multiple ESXi instances to access shared virtual machine storage concurrently. It also enables virtualization-distributed infrastructure services, such as VMware vMotion, DRS, and High Availability, to operate across a cluster of ESXi hosts.

In order to balance performance and manageability in a virtual environment, it is an accepted best practice to deploy databases using VMFS. Raw Device Mapping (RDM) is sometimes erroneously selected to provide increased performance. The two dominant workloads associated with databases, random read/write and sequential writes, have nearly identical performance throughput characteristics when deployed on VMFS or using RDM.

Datastores

vSphereere uses datastores to store virtual disks. Datastores provide an abstraction of the storage layer that hides the physical attributes of the storage devices from the virtual machines. VMware administrators can create datastores that can be used as a single consolidated pool of storage, or many datastores that can be used to isolate various application workloads.

In traditional Storage Area Network (SAN) deployments, it is an accepted best practice to create a dedicated datastore if the application has a demanding I/O profile. Databases fall into this category. The creation of dedicated datastores with vSphere allows database administrators to define individual service level agreements (SLAs) for different applications. This is analogous to provisioning dedicated logical units (LUNs) in the physical world.

In summary, with SAP HANA on vSphere, datastores can be used as follows:

- Create separate and isolated datastores for SAP HANA data and logs
- Multiple SAP HANA virtual machines can have their data and log virtual machine disk files provisioned on the same class of storage
- Provision virtual machine disk files as eager zeroed thick to avoid lazy zeroing

Paravirtual SCSI Adapters

It is a best practice to create a primary adapter for use with a disk that hosts the system software and SAP HANA binaries, and to separate paravirtual SCSI (PVSCSI) adapters for the SAP HANA data and log devices. PVSCSI adapters are high performance storage adapters that can result in greater throughput and lower CPU utilization. To configure PVSCSI adapters for use with the SAP HANA, see the VMware Knowledge Base article "Configuring disks to use VMware Paravirtual SCSI (PVSCSI) adapters (1010398)" at — https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1010398

Multiple Virtual SCSI Controllers

VMware recommends creating multiple virtual SCSI controllers to distribute the I/O associated with database workloads. When creating multiple SCSI controllers, map these controllers to the database or operating system workload profile. Ensure that the operating system and SAP HANA binaries reside on one SCSI controller, and the SAP HANA data files and log files reside on separate SCSI controllers. The primary purpose for using multiple virtual SCSI controllers is to parallelize the units of work in a database transaction or query. In this case, carefully consider the implications when using multiple SCSI controllers to parallelize a single unit of work within a transaction.

File System Considerations and Alignment

File system misalignment can severely impact performance. File system misalignment not only manifests itself in databases, but with any high I/O workload. VMware makes the following recommendations for VMware VMFS partitions:

- Similar to other disk-based file systems, VMFS suffers a penalty when the partition is not aligned. Use VMware vCenter to create VMFS partitions, since it automatically aligns the partitions along the 64 KB boundary
- In order to manually align your VMware VMFS partitions, check your storage vendor's recommendations for the partition starting block (for example, EMC VNX uses 128K offsets)

SUSE Linux I/O Scheduler

The default I/O Scheduler is Completely Fair Queuing (CFQ). The scheduler is an effective solution for nearly all workloads. The default scheduler affects all disk I/O for VMDK-based and RDM-based virtual storage solutions. In virtualized environments, it is often not beneficial to schedule I/O at both the host and guest layers. If multiple guests use storage on a file system or on a block device managed by the host operating system, the host might be able to schedule I/O more efficiently, since it recognizes requests from all guests and the physical layout of storage, which might not map linearly to the guests' virtual storage.

Testing shows that NOOP performs better with virtualized Linux guests. ESXi uses an asynchronous intelligent I/O scheduler. For this reason, virtual guests obtain improved performance by allowing ESXi to handle I/O scheduling. For additional information refer to — : http://kb.vmware.com/kb/2011861

Additional References - SAP on VMware Best Pratices

Following are some useful technical references:

- Virtualize Business Critical Applications — http://blogs.vmware.com/apps/sap
- VMware Documents regarding SAP Products — https://wiki.scn.sap.com/wiki/display/VIRTUALIZATION/VMware+documents+regarding+SAP+products
- Best Practices and Recommendations for Scale-up Deployments of SAP HANA on VMware vSphere — http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/sap_hana_on_vmware_vsphere_best_practices_guide-white-paper.pdf
- SAP on VMware Best Practices — http://www.vmware.com/files/pdf/business-critical-apps/sap-on-vmware-best-practices.pdf

Appendix

8

Useful Terminology:

- VMware vSphere – A virtualization platform that allows pooling resources to deliver IT as a service.
- VMware ESXi™ Host – An x86 server running the VMware bare-metal hypervisor ESXi, which allows virtual machines to be run.
- *VMkernel* – A component of the ESXi hypervisor, which is a POSIX-like operating system developed by VMware. *VMkernel* is designed specifically to support running multiple virtual machines, and provides core functionality such as resource scheduling, I/O stacks, and device drivers.
- *Virtual CPU (vCPU)* – VMware uses the terms *virtual CPU* (vCPU) and physical CPU to distinguish between the processors within the virtual machine and the underlying physical x86-based processors. The number of vCPUs assigned to a virtual machine can be regarded as the number of physical cores of processing power a virtual machine can consume.
- Virtual Machine (VM) – A software implementation of a computer that executes programs like a physical machine. It is configured with virtual CPU, memory, and disk. The disk encapsulates both the operating system and application software. VMware vSphere 6 supports up to 128 vCPUs and 4 TB of RAM per virtual machine.
- Guest Operating System (Guest OS) – The operating system that you install and run in a virtual machine.
- VMware vCenter Server™ – Provides centralized management of virtualized hosts and virtual machines from a single console.
- VMware vSphere Web Client™ – Provides GUI-based access to vCenter Server to perform management tasks.
- VMware vSphere Web Client – Provides access to a vCenter Server system to manage an ESXi host through a browser.
- VMware vSphere High Availability – Provides easy-to-use, cost-effective high availability solutions for applications running in virtual machines. In the event of server failure, affected virtual machines are automatically restarted on other servers with spare capacity.
- VMware vSphere Fault Tolerance – Provides continuous availability for applications in the event of server failures by creating a live shadow instance of a virtual machine that is always up to date with the primary virtual machine. In the event of a hardware outage, VMware vSphere Fault Tolerance automatically triggers failover, providing zero downtime and preventing data loss. As of VMware vSphere 6, VMware vSphere Fault Tolerance supports up to four virtual CPU virtual machines and is a viable solution for *SAP Central Services*.

- ESXi Cluster – A group of ESXi hosts defined in vCenter Server that enables the group to behave as a cluster, so that if one ESXi host fails, all the virtual machines running on the failed ESXi host are restarted on the remaining ESXi hosts in the cluster (using vSphere HA).
- VMware vSphere vMotion® – Enables the live migration of running virtual machines from one physical server to another with zero downtime and continuous service availability.
- VMware vSphere Distributed Resource Scheduler™ (DRS) – Dynamically balances computing capacity across ESXi hosts grouped in an ESXi cluster. This is achieved by live migrating (with vSphere vMotion) virtual machines between ESXi hosts to optimize the utilization of memory and CPU.
- VMware vSphere Replication™ – A hypervisor-based, asynchronous replication solution for vSphere virtual machines. It is fully integrated with VMware vCenter Server and the VMware vSphere Web Client.
- VMware vCenter Orchestrator™ – A development and process automation platform that provides a library of extensible workflows that allow you to create and run automated, configurable processes to manage the vSphere infrastructure and other VMware and third-party technologies. VMware vCenter Orchestrator is installed as a virtual appliance.
- Virtual Disk – Used by a virtual machine to store its OS and application data. A virtual disk is a large physical file, or a set of files, that can be copied, moved, archived, and backed up.
- VMware Tools™ – A suite of utilities installed in the guest operating system that enhances the performance of the virtual machine's guest operating system and improves management of the virtual machine.
- *SAP Landscape Virtualization Management* - SAP has a solution to manage SAP landscapes called *Landscape Manager (LaMa)*. *LaMa* is a central management point for SAP Basis administrators that allows mass operations, automation of day-to-day administrative and lifecycle management tasks, and automation of the copy, clone, and refresh of SAP systems. VMware has an adapter that integrates with LaMa to enable these tasks for a virtualized SAP system. For more information on the VMware Adapter for SAP Landscape Management, see http://www.vmware.com/products/adapter-sap-lvm.html.

Index