31 Days Before Your CCNP and CCIE Enterprise Core Exam

A Day-By-Day Review Guide for the ENCOR 350-401 Certification Exam

ciscopress.com

Patrick Gargano

31 Days Before Your
CCNP and CCIE Enterprise Core Exam

A Day-by-Day Review Guide for the CCNP and CCIE Enterprise Core ENCOR 350-401 Certification Exam

Patrick Gargano

Cisco Press • 221 River Street • Hoboken, NJ 07030 USA

31 Days Before Your CCNP and CCIE Enterprise Core Exam

Patrick Gargano

Published by:
Cisco Press
221 River Street
Hoboken, NJ 07030 USA

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ScoutAutomatedPrintCode

Library of Congress Control Number: 2020913346

ISBN-13: 978-0-13-696522-0

ISBN-10: 0-13-696522-9

Warning and Disclaimer

This book is designed to provide information about exam topics for the Cisco Certified Networking Professional (CCNP) Enterprise and Cisco Certified Internetwork Expert (CCIE) Enterprise Infrastructure and Enterprise Wireless certifications. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The author, Cisco Press, and Cisco Systems, Inc., shall have neither liability for nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

About the Author

Patrick Gargano has been an educator since 1996, a Cisco Networking Academy Instructor since 2000, and a Certified Cisco Systems Instructor (CCSI) since 2005. He is currently working for Cisco as a Content Engineer on the Enterprise Technical Education team within DevCX. Until recently, he was based in Australia, where he worked as a Content Development Engineer at Skyline ATS, responsible for CCNP Enterprise course development with Learning@Cisco. He previously led the Networking Academy program at Collège La Cité in Ottawa, Canada, where he taught CCNA/CCNP-level courses, and he has also worked for Cisco Learning Partners NterOne and Fast Lane UK. In 2018 Patrick was awarded the Networking Academy Above and Beyond Instructor award for leading CCNA CyberOps early adoption and instructor training in Quebec, Canada. Patrick has also twice led the Cisco Networking Academy Dream Team at Cisco Live US. His previous Cisco Press publications include *CCNP and CCIE Enterprise Core & CCNP Advanced Routing Portable Command Guide* (2020), *31 Days Before Your CCNA Security Exam* (2016), and *CCNP Routing and Switching Portable Command Guide* (2014). His certifications include CCNA, CyberOps Associate, and CCNP Enterprise, as well as the Enterprise Core and Enterprise Advanced Infrastructure Implementation specialists. He holds BEd and BA degrees from the University of Ottawa, and he is completing a master of professional studies (MPS) degree in computer networking at Fort Hays State University.

About the Technical Reviewer

Akhil Behl is a Pre-Sales Manager with a leading service provider. His technology portfolio encompasses IoT, collaboration, security, infrastructure, service management, cloud, and data center. He has over 12 years of experience working in leadership, advisory, business development, and consulting positions with various organizations and leading global accounts, driving toward business innovation and excellence. Previously, he was in a leadership role with Cisco Systems.

Akhil has a bachelor of technology degree in electronics and telecommunications from IP University, India, and a master's degree in business administration from Symbiosis Institute, India. Akhil holds dual CCIE certifications in Collaboration and Security, PMP, ITIL, VCP, TOGAF, CEH, ISO/IEC 27002, and many other industry certifications.

He has published several research papers in national and international journals, including IEEE publications, and has been a speaker at prominent industry forums such as Interop, Enterprise Connect, Cloud Connect, Cloud Summit, Cisco Sec-Con, IT Expo, Computer Society of India, Singapore Computer Society, and Cisco Networkers.

Akhil is the author of several Cisco Press books. He also is a technical editor for Cisco Press and other publications. Akhil can be reached at akbehl@technologist.com.

Day 7

Virtualization

ENCOR 350-401 Exam Topics

Virtualization

- Describe device virtualization technologies
  - Hypervisor type 1 and 2
  - Virtual machine
  - Virtual switching
- Configure and verify data path virtualization technologies
  - VRF

Key Topics

Today we review topics related to *virtualization*—the process of running one or more virtual instances of an operating system in a layer abstracted from the underlying hardware. Virtualization commonly refers to simultaneously running multiple operating systems on general- or specific-purpose hardware. To the applications running on top of the virtualized machine, it may appear as if the applications are on their own dedicated machines, where the operating system, libraries, and other applications are unique to the (guest) virtualized system. Virtualization in networks allows for a single physical network topology to be utilized by different networks that have no interaction with each other. Today we explore server virtualization and network functions virtualization, as well as virtual routing and forwarding (VRF).

Server Virtualization

The heart of the virtualized server environment is the virtualization software. In this section, we explore virtualization and the VMware ESXi (Elastic Sky X integrated) software.

Standard deployment of applications on computer systems involves installing a single operating system on each computer. To provide the greatest stability in business environments, good practice dictates that a single type of application be present on an operating system; otherwise, compatibility issues could cause unforeseen problems. This approach provides good stability but is not very cost-effective. A growing number of applications greatly expands the number of physical servers in a data center, which increases the total cost of ownership (TCO) and can adversely impact the IT budget of a company.

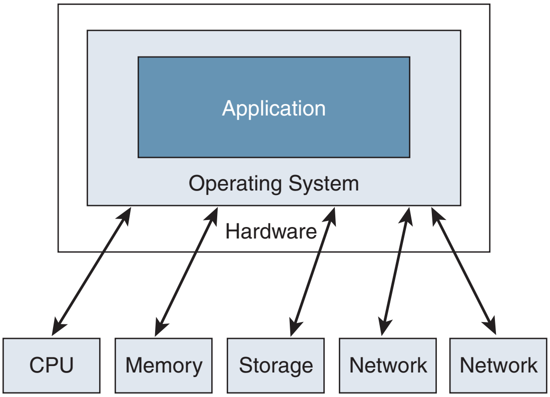
This standard deployment method of server usage was common in unified communications applications such as Communications Manager 7.x. The installation media installed the Voice Technology Group (VTG) software (a hardened version of Linux) on the bare-metal Media Convergence Server (MCS) 7800 series server and deployed a unified communications application such as Cisco Unified Communications Manager or Cisco Unity Connection on the VTG operating system. Each physical server represented one node in the application cluster.

Physical Server

Virtualization, in the context of IT, helps abstract physical resources from the services they provide. In server virtualization, a layer of software is installed between the server hardware and the operating system.

In a non-virtualized server, the operating system (Windows or Linux) has direct access to the physical resources on the server, as shown in Figure 7-1. If the physical server has four CPU cores, all four cores are available for use by the operating system.

Figure 7-1 Physical Server Resources

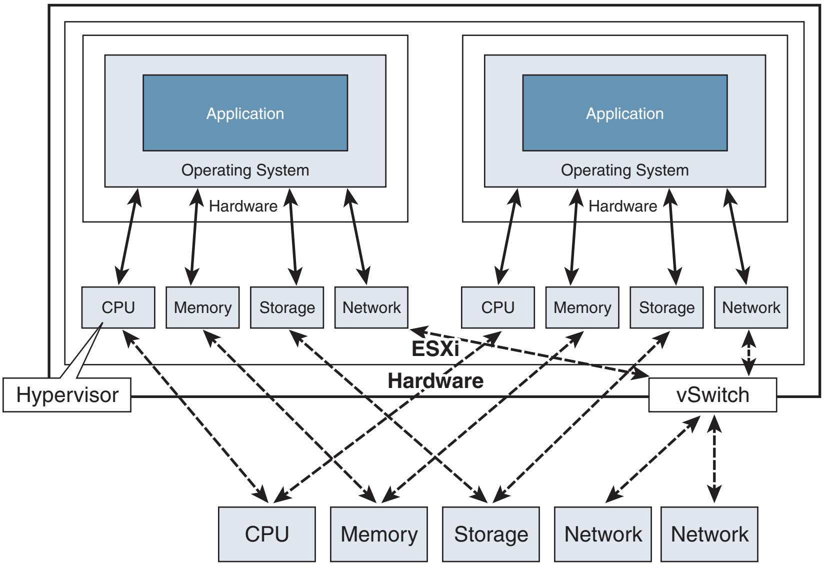


Virtualized Server

With a virtualized server, instead of loading an operating system, such as Linux, on the server, a software package called a hypervisor is installed on the server. The hypervisor is a combination of a lightweight operating system and additional software to create and manage the virtual environment, which abstracts the underlying hardware compute, storage, and network from the virtual servers running atop the hypervisor. An example of hypervisor software is VMware ESXi (vSphere).

Virtual machines (VMs) are created with software representations of CPU cores, memory, hard disk space, and network interface cards (vNICs). The application operating system is not aware that the resources are virtualized, and the resources that the VM has access to are not the physical CPUs or memory, as shown in Figure 7-2.

Figure 7-2 Virtualized Server Resources



The operating system has access to the assigned resources in the normal way. When the hypervisor receives a request, it forwards the request to the physical resources of the host server.

In each case, the virtual resource is a portion of the physical resources of the host server. One exception is the vNIC of the VM. Rather than connect directly to the physical Ethernet ports of the host server, the vNIC connects to an internal virtual Ethernet switch called a vSwitch via vPorts. The vSwitch can have many connections from VMs. The vSwitch maintains connections, called uplinks, to the physical Ethernet interfaces.

This approach has several benefits, including the following:

- The virtualization layer provides a uniform virtualized hardware interface to the operating system that is installed on top. Even if three different physical servers in the data center have been provided by three different vendors, they appear the same to the operating system that is installed on top of the virtualization layer. There may be differences in performance, but there are no differences in the type of resources, such as the network cards or graphics adapters, that appear to the operating system.
- The virtualization layer is capable of segmenting the physical hardware into multiple separate resource units that all draw from the same physical pool. This means you can have multiple instances of a single operating system or even different types of operating systems running simultaneously on a single server. These separate instances are all completely independent of each other. Unless a non-redundant hardware component fails, these instances are independent of each other, as if they were running on separate physical servers.

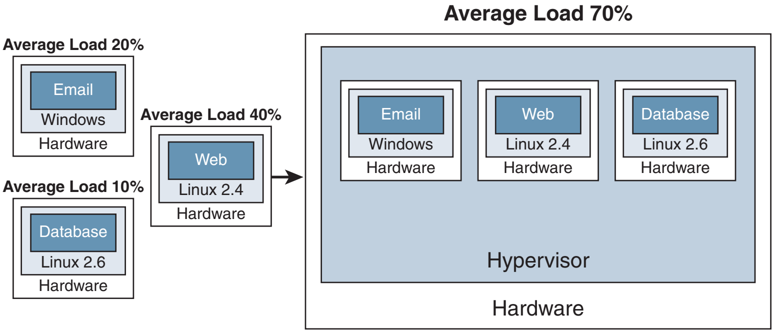
Basic Virtualized Server Environment

Server virtualization gives you the ability to run multiple operating systems on a single physical server. This ability can be useful in testing or production environments.

For example, a production virtual operating system can be copied and used to test upgrade procedures in a non-production or test environment. You can therefore try an upgrade procedure in a virtual environment that is the same as the production environment used by employees, but mistakes or compatibility problems will not affect users. After the upgrade has been thoroughly tested and the consequences evaluated, a real upgrade can take place, based on a full understanding of the impact on the production environment.

Physical servers are often underutilized in non-virtualized environments because typically a separate server is assigned for each application, as illustrated in Figure 7-3. An application such as Cisco Unified Communications Manager (CUCM), with low numbers of registered devices or applications, might use less than 20% CPU when operating normally. That leaves 80% of the system resources unused. Underutilization has a direct impact on both operations and operating costs as it increases the number of servers required. Every extra server requires additional physical space, power, and cooling requirements. As the number of servers grows, it becomes more difficult to manage them all.

Figure 7-3 Example of Underutilized Server Resources



Depending on the average load of an existing deployment, with virtualization, it is not unusual to be able to put 12 or more servers onto a single piece of hardware. If the data center environment is designed with virtualization in mind, the ratio of virtual servers to physical servers can be much higher. The limiting factor with collaboration applications, as well as other applications, is that you cannot exceed the physical resources of the host server.

For example, if a host server has 24 CPU cores installed (that is, 2 12-core processors), you could add 24 VMs that need 1 vCPU each. However, if a video conferencing application, such as Cisco Meeting Server (CMS), needs 20 vCPUs and other applications require 2 vCPUs each, you would be able to have only 3 VMs on that host.

Virtualization also provides more flexibility as the deployment of a new virtual server is much easier and faster than the deployment of a physical server.

Hypervisor: Abstraction Layer

A hypervisor acts as an abstraction layer that sits between the guest operating system and the hardware. A hypervisor, such as VMware ESXi, is a dedicated operating system that leverages VMkernel for separation of the virtual and physical layers. The abstraction layer supports the installation of multiple operating systems, which can be of different types, but a hypervisor supports only virtual machines and is not intended for the installation of applications. The component of the hypervisor that conveys commands from the software on the VM to the actual hardware on the physical server is known as the virtual machine monitor.

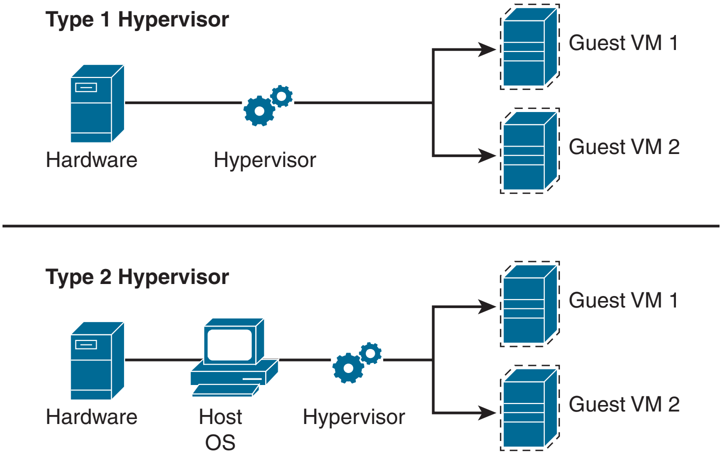
Cisco unified communications applications started supporting virtualization with VMware ESXi 4.0 and Cisco UCM Version 8.0. Most current versions of Cisco collaboration applications support ESXi Version 5.0 and later.

A hypervisor performs several tasks:

- Provides provisioned resources to individual guest operating systems or VMs by partitioning the resources of the physical server, or host, on which it is installed
- Provides connectivity between VMs and other VMs, as well as external network resources or physical environment
- Ensures separation between individual VMs from a security perspective

Two types of hypervisors are used in networks today: type 1 and type 2 (see Figure 7-4).

Figure 7-4 Type 1 and Type 2 Hypervisors



Type 1 Hypervisors

A type 1 hypervisor runs directly on the host machine's physical hardware and does not have to load an underlying OS first. This type of hypervisor is often referred to as a *bare-metal hypervisor*. With direct access to the underlying hardware and no other software, such as operating systems and device drivers, to contend with, type 1 hypervisors are regarded as the most efficient and best-performing hypervisors available for enterprise computing. Hypervisors such as VMware ESXi, Microsoft Hyper-V Server, and open-source KVM are examples of type 1 hypervisors.

Type 2 Hypervisors

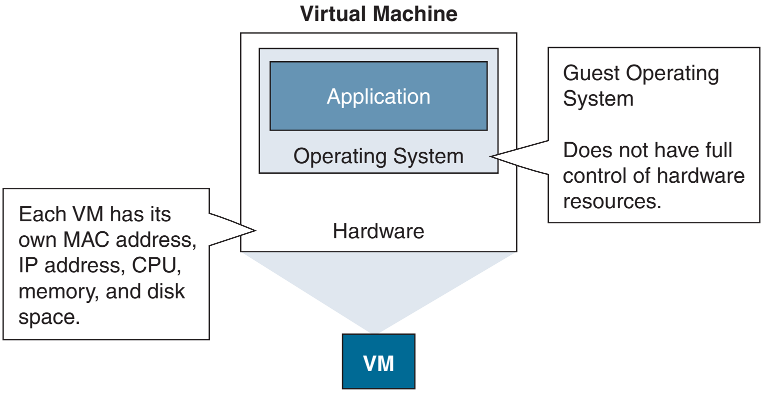
A type 2 hypervisor is typically installed on top of an existing OS and is called a *hosted hypervisor* because it relies on the host machine's preexisting OS to manage calls to CPU, memory, storage, and network resources. Type 2 hypervisors include VMware Fusion, Oracle VM VirtualBox, Oracle VM Server for x86, Oracle Solaris Zones, Parallels, and VMware Workstation. This type of nested deployment is usually avoided in production environments.

VM Definition

A *virtual machine* (*VM*) is a logical container that holds all the resources that an operating system requires for normal operation, such as a graphics adapter, memory, a processor, and networking resources. As far as the operating system running in a VM is concerned, there is no difference between these components and the components that would be available on a physical server. However, these components are virtualized representations of host resources. In the case of processor capacity or memory, the assigned resources are a percentage of the actual resources. In the case of a hard disk, the virtual component is a specially formatted file that is visible as a disk to the virtualized (guest) operating system. Regarding the vNIC, this is a simulated virtual component that is managed by the hypervisor, so it acts like a physical component.

A VM is the virtual equivalent of a physical PC or server, and it requires the same set of software and networking identifiers as any physical device, as shown in Figure 7-5. One of the benefits of a VM is that an administrator can set these identifiers to desired values by simply manipulating the VM configuration file.

Figure 7-5 Simple Virtual Machine Architecture Example


Virtual Machine
Application
Operating System
Hardware
VM
Guest Operating System
Does not have full control of hardware resources.
Each VM has its own MAC address, IP address, CPU, memory, and disk space.

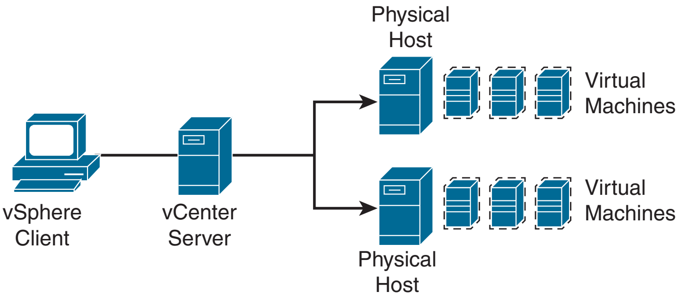
Managing Virtual Machines

There are two ways of managing virtual machines when using a solution like VMware vSphere. An ESXi host can be controlled and configured either individually or as part of a group of hosts that are centrally controlled via the vSphere GUI.

Individual access is possible by using the vSphere Client for Windows or the newer VMware Host Client. The more mature access method is the Windows client. The client is downloaded from the VMware website and installed on the client PC. The same client can connect to an individual VMware host or to VMware vCenter Server.

Typically, a larger VMware environment is managed using VMware vCenter Server (see Figure 7-6). vCenter is a unified control point for all the participating hosts. You can configure hosts on an individual basis in simpler environments, but some advanced functionalities of VMware, such as vMotion (which enables you to easily move a VM from one host to another), require vCenter.

Figure 7-6 Centralized vSphere Management



In addition to being able to configure multiple hosts from one interface, you can activate additional features, such as vSphere Distributed Switch (vDS), resource and power load balancing, and CPU compatibility. vDS provides a centralized interface for configuration, administration, and monitoring of virtual machine access and switching in a data center.

vCenter can be deployed as a separate physical server, or it can be installed as a VM. Starting in vSphere 5.0, vCenter is available as a Linux-based virtual appliance that can be downloaded and installed from a virtual appliance template.

Network Function Virtualization

Enterprise branch architectures today often host multiple devices for accomplishing required network functionality. Network appliances that perform WAN optimization, firewall protection, and Cisco intrusion prevention system (IPS) or voice services operate alongside routers and switches. In high-end branch environments, these devices are even deployed with redundancy, further adding to the hardware landscape that has to be installed and operated in the branch.

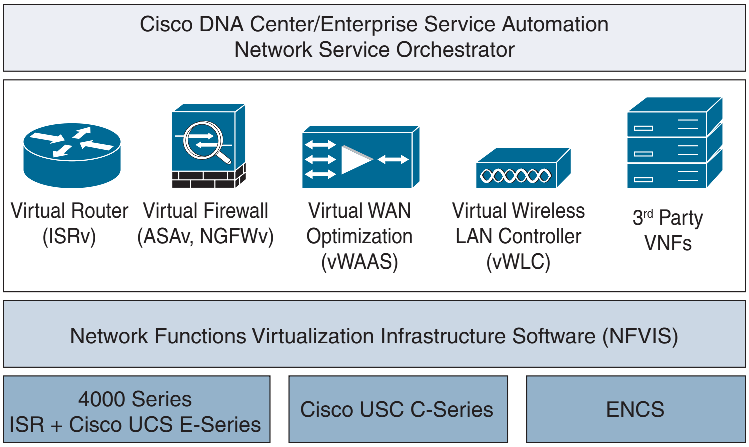
Cisco Enterprise Network Functions Virtualization (Enterprise NFV) reduces the operational complexity of such branch environments by running the required networking functions as software on standard x86-based hosts. In particular, the Cisco Enterprise NFV solution reduces the number of hardware elements to be managed at the branch, reducing the need to perform costly site visits for hardware installations or upgrades. Also, the Cisco Enterprise NFV solution automates the deployment, management, and operations of branch functions, which diminishes operating expenses.

Cisco Enterprise NFV Solution Architecture

Cisco Enterprise NFV is an end-to-end solution that addresses all requirements for deploying virtualized network and application services, such as orchestration and management, virtualization software packages, and options for different hardware platforms. Cisco Enterprise NFV also addresses the use of Cisco and third-party service functions.

Figure 7-7 illustrates the Cisco Enterprise NFV solution architecture.

Figure 7-7 NFV Solution Architecture


Cisco DNA Center/Enterprise Service Automation
Network Service Orchestrator
Virtual Router (ISRv)
Virtual Firewall (ASAv, NGFWv)
Virtual WAN Optimization (vWAAS)
Virtual Wireless LAN Controller (vWLC)
3rd Party VNFs
Network Functions Virtualization Infrastructure Software (NFVIS)
4000 Series ISR + Cisco UCS E-Series
Cisco USC C-Series
ENCS

The top layer of the Cisco Enterprise NFV architecture is focused on orchestration, automation, and management (OAM). This layer is composed of either Cisco Enterprise Service Automation (ESA) or Cisco DNA Center for network service automation and orchestration. It is also possible to use Cisco Network Services Orchestrator (NSO) for OAM in NFV environments. This layer provides common OAM across both virtual and physical devices.

The second layer is composed in the virtual network functions (VNFs). VNFs are the service functions that execute the desired tasks, such as Layer 3 routing, firewall and intrusion detection and prevention systems, WAN optimization and caching, SD-WAN vEdge and cEdge routers, and virtual cloud wireless LAN controllers. Also, third-party or custom VNFs can be deployed at this layer (for example, Palo Alto, Fortinet, ThousandEyes).

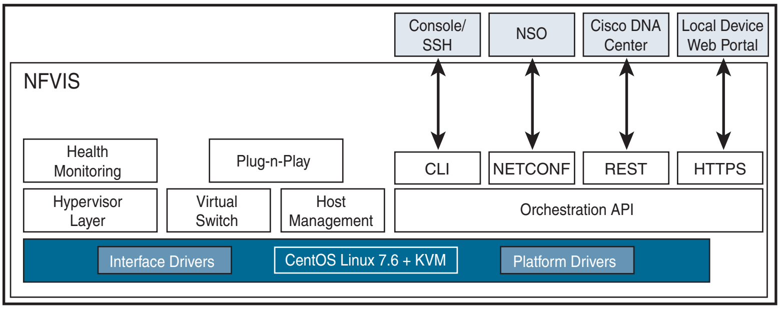
The third layer is Cisco Enterprise NFV Infrastructure Software (NFVIS). NFVIS is a software host that manages virtualization and hardware. An integrated hypervisor allows you to create and run network functions as virtual appliances using a GUI. Also, it manages the underlying hardware platform and exposes an API for orchestration and management.

The bottom layer is the platform hardware layer. It is composed of various host options for different branch sizes. The supported platforms are the Cisco Unified Computing System (UCS) C-Series and the Cisco Integrate Services Router (ISR) 4000 routers, which are equipped with Cisco UCS E-Series servers and Cisco Enterprise Network Compute System (ENCS). Cisco ENCS is a compute appliance that is designed for a virtualized software-defined network architecture. The system offers service flexibility and performance, and it lowers total cost of ownership for the next-generation branch office.

NFVIS Building Blocks

The Cisco Enterprise NFV solution includes the NFVIS virtualized software platform. NFVIS extends Linux by packaging additional functions for VNF lifecycle management, monitoring, device programmability, and hardware acceleration. Figure 7-8 shows the building blocks of NFVIS.

Figure 7-8 NFVIS Building Blocks



NFVIS delivers the following components and functionality:

- **Linux (CentOS Version 7.1) and OS kernel:** NFVIS is built on CentOS Version 7.1. The OS kernel component drives the underlying hardware platforms (for example, Cisco UCS C-Series servers, Cisco UCS E-Series servers, Cisco ENCS) and hosts the virtualization layer for VNFs, virtual switching APIs, and management.

- **Virtualization support and virtual switching:** The hypervisor for virtualization is built on top of a KVM, and it includes QEMU, libvirt, and other associated processes. Virtual switching enables multiple VNFs to share the physical interface resources and allows the traffic to be passed within the x86 host, between VNFs.

- **VM lifecycle management:** VM lifecycle management includes support to bring up VNFs dynamically and to control their liveliness using ESC-lite. Cisco Elastic Services Controller (ESC) is a VNF manager that performs lifecycle management. Cisco ESC provides agentless and multivendor VNF management by provisioning virtual services and monitors the health and load of these services. Cisco ESC provides the flexibility to define rules for monitoring and associates the actions to be triggered based on the outcomes of these rules.

- **Orchestration API:** Cisco Enterprise NFVIS provides orchestration APIs for OAM servers to automate and orchestrate the VNFs on NFVIS. REST, CLI, or NETCONF, and YANG are supported.

- **PnP client:** The PnP agent that is running on the NFVIS host first tries to discover a PnP server that is running on the Cisco DNA Center with which it can communicate. Cisco DNA Center is loaded with the right host configuration. When the PnP server is found and the connection is established, the agent performs deployment-related activities such as host configuration with the server.

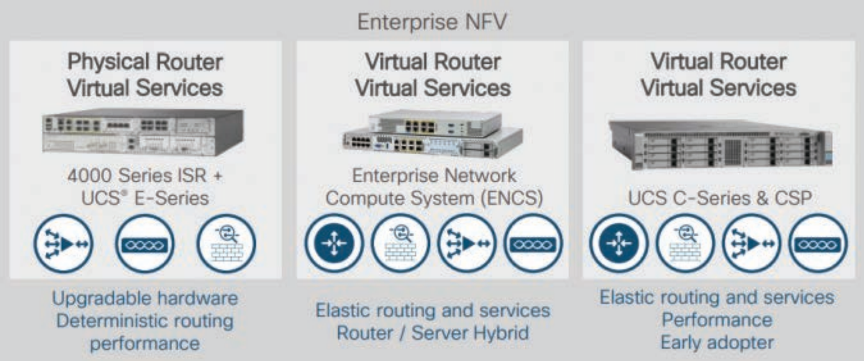
- **Web server:** The web server can enable connectivity to NFVIS through HTTPS, which is particularly useful for supporting local management tools and orchestration APIs.

- **Health monitor:** Tools like syslog, SNMP, and collectd help with statistics collection and reporting.

Cisco NFV Hardware Options

The compute resources to operate VNFs are offered in the Cisco Enterprise NFV solution in various form factors, as shown in Figure 7-9:

Figure 7-9 NFV Hardware Options


Enterprise NFV
Physical Router Virtual Services
4000 Series ISR + UCS® E-Series
Upgradable hardware Deterministic routing performance
Virtual Router Virtual Services
Enterprise Network Compute System (ENCS)
Elastic routing and services Router / Server Hybrid
Virtual Router Virtual Services
UCS C-Series & CSP
Elastic routing and services Performance Early adopter

- **Cisco UCS C-Series servers:** These servers can integrate into Cisco UCS through Cisco UCS fabric interconnects or can be used as standalone servers with Cisco or third-party switches. With ultrafast performance for two- and four-socket servers, Cisco rack and storage servers can integrate into Cisco UCS through a single set of cables. These servers provide a wide range of I/O, memory, internal disk, solid-state disk (SSD) drive, and NVMe storage device capacity, enabling you to easily match servers to workloads. The Cisco UCS C480 ML M5 server is designed for artificial intelligence and machine learning workloads. Its purpose-specific design incorporates eight NVIDIA V100 SMX2 32 GB graphics processing units (GPUs) to power compute-intensive deep learning applications.

- **Cisco ISR 4000 routers and Cisco E-Series servers:** Cisco 4000 Series ISR devices are branch office edge routers that are designed to meet Gigabit forwarding performance needs while running a broad set of network and application services on a single platform. The modularity of this series, with available interface modules, transforms the network edge at the branch into a premier NFV host. When equipped with Cisco UCS E-Series compute servers running NFVIS, the result is the most optimally deployed NFV services available in the market. This flexible platform is further extended using network interface modules to add a wide range of options for network interfaces, voice for unified communications, storage, and other x86 compute cards.

- **Cisco ENCS:** The Cisco 5000 Enterprise Network Compute System (ENCS) is a line of compute appliances designed for the Cisco SD-Branch and Enterprise NFV solution. It delivers a new standard of software-defined flexibility and performance and offers a low total cost of ownership (TCO). The 5000 ENCS is a hybrid platform that combines the best attributes of a traditional router and a traditional server and offers the same functionality with a smaller infrastructure footprint. Offered with the Cisco Integrated Services Virtual Router (ISRv) and NFVIS as the hosting layer, the platform offers a complete solution for a simplified deployment. It also accelerates some functions in hardware, such as inter-VM traffic flows, IP Security (IPsec) crypto, and RAID for storage. Cisco SD-WAN capability can be enabled on both ENCS 5100 and 5400 appliances. Cisco SD-WAN offers an entirely new way to manage and operate a WAN infrastructure, offering better user experience, greater agility, and advanced threat protection for branch office. Cisco ENCS is supported by Cisco DNA Center for management, orchestration, and automation.

Network Path Isolation

Path isolation refers to the creation of independent logical traffic paths over a shared physical network infrastructure. This involves the creation of VPNs with various mechanisms and mapping between various VPN technologies, Layer 2 segments, and transport circuits to provide end-to-end isolated connectivity between various groups of users.

The main goal when segmenting a network is to preserve and, often, improve the scalability, resiliency, and security services available in a non-segmented network. Any technology used to achieve virtualization must also provide the mechanisms necessary to preserve resiliency and scalability and to improve security.

A traditional hierarchical IP network is a combination of Layer 3 (routed) and Layer 2 (switched) domains. Both types of domains must be virtualized, and the virtual domains must be mapped to each other to keep traffic segmented. This can be achieved by combining the virtualization of the network devices (also referred to as *device virtualization*) with the virtualization of their interconnections (known as *data path virtualization*).

Layer 2 and Layer 3 Virtualization

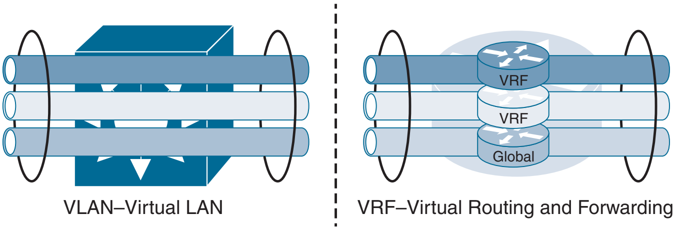
Virtualization in the Layer 2 domain is not a new concept: VLANs have been used for years. Within the campus, network virtualization began with the concept of VLANs. VLANs provide the most basic means of isolating network traffic at Layer 2 in a broadcast domain and require a Layer 3 device to route between those domains.

The goal of a robust network design is to minimize the extent of the broadcast domain and exposure to spanning tree loops. It is required to translate the Layer 2 VLAN to a Layer 3 virtual network or VPN. This Layer 3 VPN must be capable of supporting its own unique control plane completely, with its own addressing structure and routing tables for data forwarding that are completely isolated from any other Layer 3 VPN on that device and in the network. The technology that enables this type of functionality is known as virtual routing and forwarding (VRF).

A VRF instance is defined on a networking device that serves as the boundary between the Layer 2 (client-side) VLANs and the Layer 3 network. Each VRF instance consists of an IP routing table, a forwarding table, and the interface(s) assigned to it. Common routing protocols such as OSPF, EIGRP, BGP, and RIPv2 can be used to advertise and learn routes to populate the routing table that is unique to each virtual network. VRF instances can be compared to virtual routers co-resident on a single Layer 3 switch or router. Figure 7-10 illustrates Layer 2 virtualization (VLAN) and Layer 3 virtualization (VRF).

Data path virtualization is the virtualization of the interconnection between devices, which can be a single-hop or multi-hop interconnection. The type of data path virtualization varies depending on how far the VRF instances are from each other. If the virtualized devices are directly connected to each other (single hop), link or circuit virtualization is necessary. If the virtualized devices are connected through multiple hops over an IP network, a tunneling mechanism is necessary.

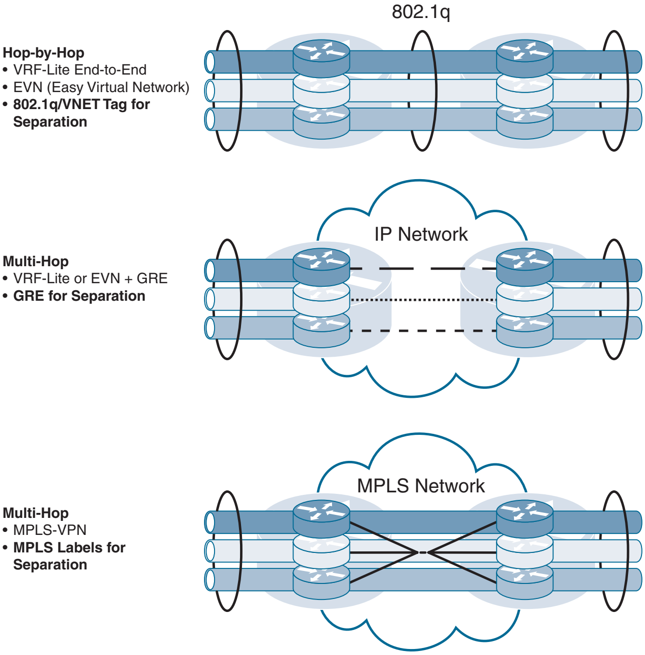
Figure 7-10 Layer 2 Virtualization and Layer 3 Virtualization



VRF-Lite combines the use of VRF instances with either 802.1Q trunking for hop-by-hop path isolation or Generic Routing Encapsulation (GRE)/Multipoint GRE (mGRE) for multi-hop path isolation. VRF-Lite with 802.1Q trunking would be found in a campus network where the IP routing is completely under the control of the organization deploying it and would typically make use of OSPF or EIGRP as the IGP for routing.

VRF-Lite can be used along with GRE or Multipoint GRE (mGRE) when it becomes necessary to extend a virtual network across a Layer 3 infrastructure or domain where virtualization is either not required or, as in the case of a service provider WAN, is beyond the control of the organization. In addition to GRE and mGRE, MPLS can also be used to extend VRFs across an IP infrastructure as well. These different options are illustrated in Figure 7-11.

Figure 7-11 Data Path Virtualization



In addition, within each networking device there are two planes to virtualize:

- **Control plane:** The control plane consists of all the protocols, databases, and tables necessary to make forwarding decisions and maintain a functional network topology free of loops or unintended black holes. This plane can be said to draw a clear picture of the topology for the network device. A virtualized device must have a unique picture of each virtual network it handles; thus, there is a requirement to virtualize the control plane components.

- **Forwarding plane:** The forwarding plane consists of all the processes and tables used to forward traffic. The forwarding plane builds forwarding tables based on the information provided by the control plane. Each virtual network has a unique forwarding table that needs to be virtualized.

The control and forwarding planes can be virtualized at different levels, which map directly to different layers of the OSI model. For instance, a device can be VLAN-aware and therefore can be virtualized at Layer 2, yet have a single routing table, which means it is not virtualized at Layer 3. The various levels of virtualization may or may not be useful, depending on the technical requirements of a deployment. In some cases, such as with a wiring closet, Layer 2 virtualization is enough. In other cases, virtualization of other layers may be necessary; for example, providing virtual firewall services requires Layer 2, 3, and 4 virtualization, plus the ability to define independent services on each virtual firewall, which perhaps is Layer 7 virtualization.

Virtual Routing and Forwarding

A virtual routing and forwarding (VRF) instance consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. The use of VRF technology allows a customer to virtualize a network device from a Layer 3 standpoint, creating different "virtual routers" in the same physical device.

The use of Cisco VRF-Lite technology has the following advantages:

- **Allows for true routing and forwarding separation:** Dedicated data and control planes are defined to handle traffic belonging to groups with various requirements or policies. This represents an extra level of segregation and security because no communication between devices belonging to different VRF instances is allowed unless explicitly configured.

- **Simplifies the management and troubleshooting of the traffic belonging to the specific VRF because separate forwarding tables are used to switch that traffic:** These data structures are different from the one associated to the global routing table. This also guarantees that configuring the overlay network does not cause issues (such as routing loops) in the global table.

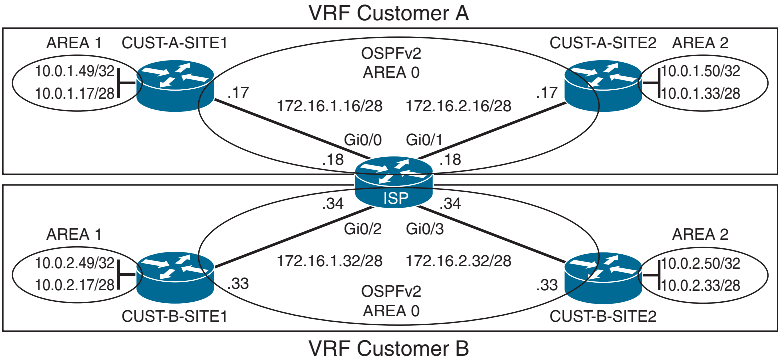
- **Enables support for alternate default routes:** The advantage of using a separate control and data plane is that it allows for the definition of a separate default route for each virtual network (VRF). This can be useful, for example, when providing guest access in a deployment when there is a requirement to use the default route in the global routing table just to create a black hole for unknown addresses to aid in detecting certain types of worm and network scanning attacks.

A VRF instance achieves the virtualization of the networking devices at Layer 3. VRF instances must be mapped to the appropriate VLANs at the edge of the network to provide continuous virtualization across the Layer 2 and Layer 3 portions of the network. The mapping of VLANs to VRF instances is as simple as placing the corresponding VLAN interface at the distribution switch into the appropriate VRF instance. The same type of mapping mechanism applies to Layer 2 virtual circuits (ATM, Frame Relay) and IP tunnels that a router handles as logical interfaces.

Configuring and Verifying VRF-Lite

The ISP router in Figure 7-12 is configured with VRF instances for Customer A and Customer B to achieve path isolation. The two customers use OSPFv2 for routing between their respective sites.

Figure 7-12 VRF-Lite Configuration Example



The use of separate routing tables solves part of the problems related to preventing one customer's packets from leaking into another customer's network because of overlapping prefixes, while allowing all sites for the same customer to communicate. A VRF instance exists inside a single router. Typically, a router needs at least one VRF instance for each customer attached to that router. In the example shown in Figure 7-12, the ISP router is connected to the two customers and must be configured with both VRF instances.

Each VRF instance has three main components:

- An IP routing table (RIB)
- A CEF FIB that is populated based on that VRF instance's RIB
- A separate instance or process of the routing protocol used to exchange routes with the CE devices that need to be supported by the VRF instance

Example 7-1 shows the commands required to deploy VRF-Lite for the scenario shown in Figure 7-12. This example assumes that the customer routers are already configured for OSPFv2.

Example 7-1 Configuring VRF-Lite

```
ISP(config)# ip vrf CUST-A
ISP(config-vrf)# exit
ISP(config)# ip vrf CUST-B
ISP(config-vrf)# exit
ISP(config)# interface GigabitEthernet 0/0
ISP(config-if)# ip vrf forwarding CUST-A
% Interface Ethernet0/0 IPv4 disabled and address(es) removed due to enabling VRF
  CUST-A
ISP(config-if)# ip address 172.16.1.18 255.255.255.240
ISP(config-if)# interface GigabitEthernet 0/1
ISP(config-if)# ip vrf forwarding CUST-A
% Interface Ethernet0/1 IPv4 disabled and address(es) removed due to enabling VRF
  CUST-A
ISP(config-if)# ip address 172.16.2.18 255.255.255.240
ISP(config-if)# interface GigabitEthernet 0/2
ISP(config-if)# ip vrf forwarding CUST-B
% Interface Ethernet0/2 IPv4 disabled and address(es) removed due to enabling VRF
  CUST-B
ISP(config-if)# ip address 172.16.1.34 255.255.255.240
ISP(config-if)# interface GigabitEthernet 0/3
ISP(config-if)# ip vrf forwarding CUST-B
% Interface Ethernet0/3 IPv4 disabled and address(es) removed due to enabling VRF
  CUST-B
ISP(config-if)# ip address 172.16.2.34 255.255.255.240
ISP(config-if)# exit

ISP(config)# router ospf 100 vrf CUST-A
ISP(config-router)# router-id 0.0.0.100
ISP(config-router)# network 172.16.1.16 0.0.0.15 area 0
ISP(config-router)# network 172.16.2.16 0.0.0.15 area 0
ISP(config-router)# router ospf 200 vrf CUST-B
ISP(config-router)# router-id 0.0.0.200
ISP(config-router)# network 172.16.1.32 0.0.0.15 area 0
ISP(config-router)# network 172.16.2.32 0.0.0.15 area 0
```

The first step is to configure the ISP router with a VRF instance for Customer A and a VRF instance for Customer B by using the **ip vrf** *vrf-name* command. Once the VRF instances are created, interfaces need to be assigned to each VRF instance by using the **ip vrf forwarding** *vrf-name* command. Notice that when an interface is assigned to a VRF instance, the router produces a warning message indicating that any IP address previously configured on the interface has been removed. This is because the interface has been moved from the global routing table to a specific VRF instance. The last step is to create two instances of OSPF, one for Customer A and one for Customer B, by using the **router ospf** *process-id* **vrf** *vrf-name* command. Each instance uses a unique OSPF process ID and a unique OSPF router ID.

To verify VRF-Lite, use the **show ip vrf** and **show ip vrf interfaces** commands, as shown in Example 7-2.

Example 7-2   Verifying VRF-Lite

```
ISP# show ip vrf
  Name                                Default RD             Interfaces
  CUST-A                              <not set>              Gi0/0
                                                             Gi0/1
  CUST-B                              <not set>              Gi0/2
                                                             Gi0/3

ISP# show ip vrf interfaces
Interface              IP-Address      VRF                              Protocol
Gi0/0                  172.16.1.18     CUST-A                           up
Gi0/1                  172.16.2.18     CUST-A                           up
Gi0/2                  172.16.1.34     CUST-B                           up
Gi0/3                  172.16.2.34     CUST-B                           up
```

Because the interfaces are now part of a VRF instance, you need to use the **show ip route vrf** *vrf-name* command to view specific VRF instance routing tables. The **show ip route** command will show you only the global routing table. The same applies to the **show ip protocols** command. You need to use the **show ip protocols vrf** *vrf-name* command to view VRF-specific routing information.

Study Resources

For today's exam topics, refer to the following resources for more study.

Resource	Module, Chapter, or Link
CCNP and CCIE Enterprise Core ENCOR 350-401 Official Cert Guide	27
CCNP and CCIE Enterprise Core & CCNP Advanced Routing Portable Command Guide	13