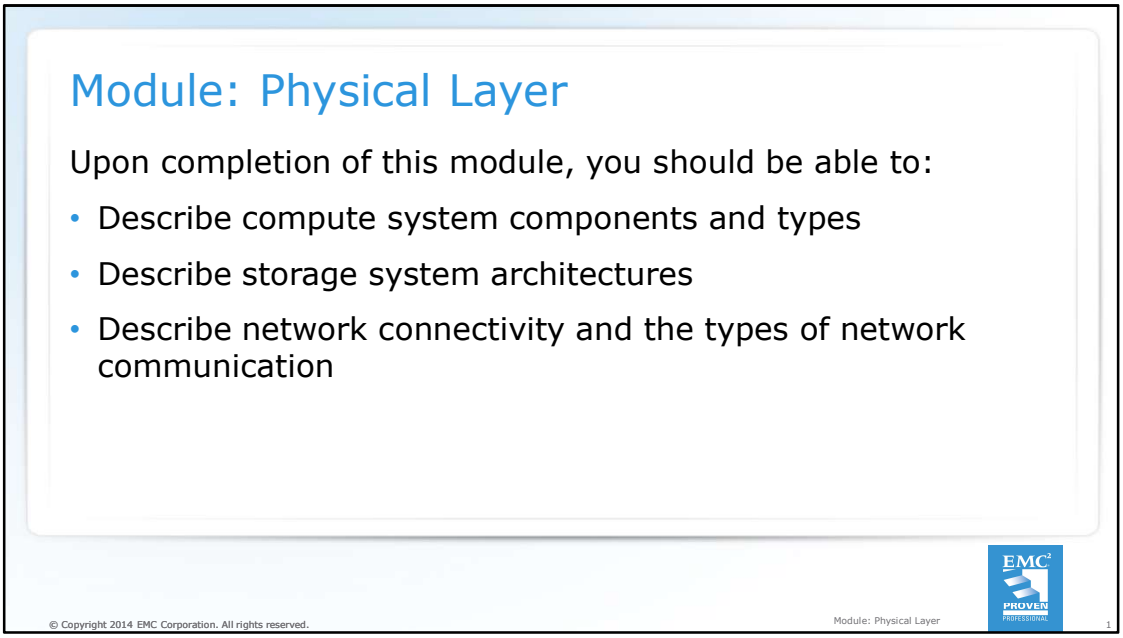
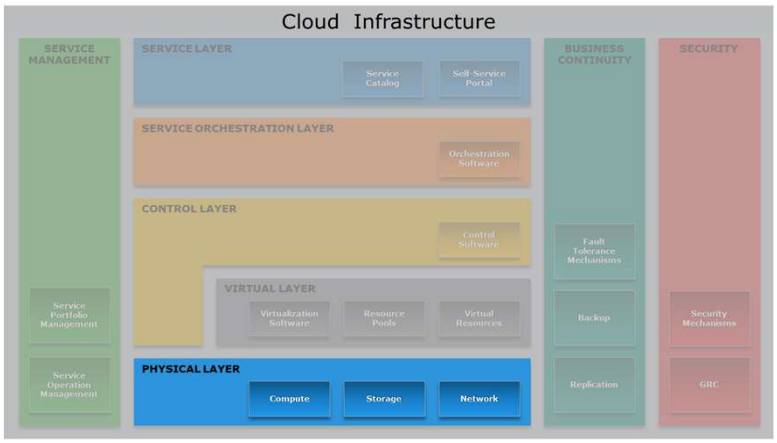


This module focuses on the physical layer of the cloud computing reference model. This module focuses on physical compute system, its components, and its types. This module also focuses on storage system architectures. Further, this module focuses on network connectivity and the types of network communication.

Cloud Computing Reference Model

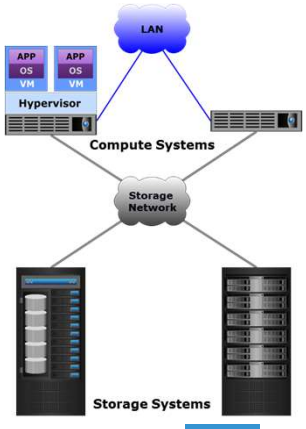
Physical Layer



The physical layer—highlighted in the figure on the slide—is the foundation layer of the cloud reference model. The process of building a cloud infrastructure is typically initiated with the cloud service provider setting up the physical hardware resources of the cloud infrastructure.

Physical Layer Overview

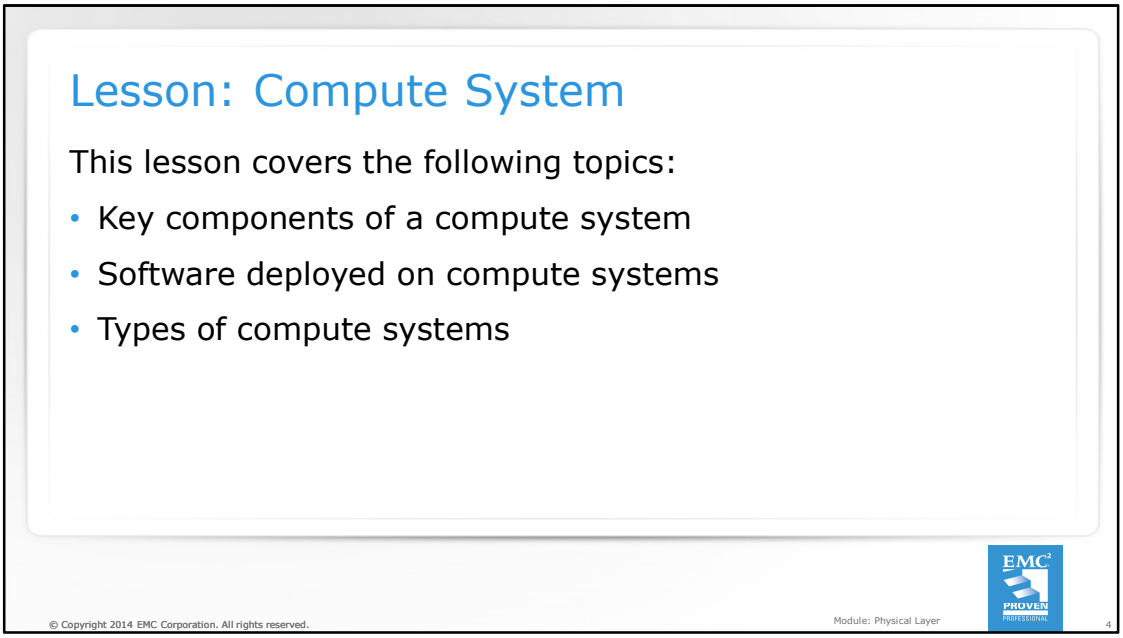
- The physical layer comprises physical compute, storage, and network resources
- Compute systems execute software of providers and consumers
- Storage systems store business and application data
- Networks connect compute systems with each other and with storage systems
  - Networks also connect multiple data centers or multiple clouds to one another



The physical layer comprises compute, storage, and network resources, which are the fundamental physical computing resources that make up a cloud infrastructure. As discussed in the 'Introduction to Cloud Computing' module, the physical resources are typically pooled to serve multiple consumers.

Physical compute systems host the applications that a provider offers as services to consumers and also execute the software used by the provider to manage the cloud infrastructure and deliver services. A cloud provider also offers compute systems to consumers for hosting their applications in the cloud. Storage systems store business data and the data generated or processed by the applications deployed on the compute systems. Storage capacity may be offered along with a compute system or separately (for example, in case of cloud-based backup). Networks connect compute systems with each other and with storage systems. A network, such as a local area network (LAN), connects physical compute systems to each other, which enables the applications running on the compute systems to exchange information. A storage network connects compute systems to storage systems, which enables the applications to access data from the storage systems. If a cloud provider uses physical computing resources from multiple cloud data centers to provide services, networks connect the distributed computing resources enabling the data centers to work as a single large data center. Networks also connect multiple clouds to one another—as in case of the hybrid cloud model—to enable them to share cloud resources and services.

Based on several requirements such as performance, scalability, cost, and so on, a cloud provider has to make a number of decisions while building the physical layer, including choosing suitable compute, storage, and network products and components, and the architecture and design of each system. The subsequent lessons describe various physical components and architectures that are available to cloud providers to build the physical layer.



This lesson covers an introduction to compute systems and describes the key components of a compute system. This lesson also covers the key software deployed on compute systems in a cloud environment, and the types of compute systems.

Introduction to Compute System

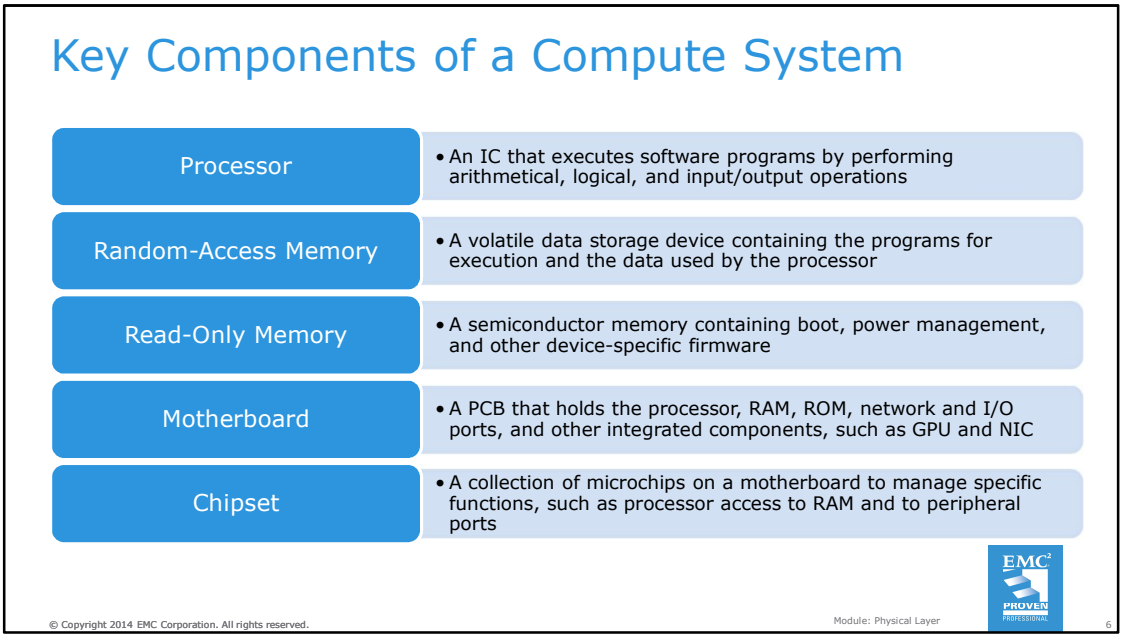
- A computing platform (hardware, firmware, and software) that runs platform and application software
  - Executes the provider's as well as the consumers' software
  - Typically x86-based servers or hosts
- Compute systems are provided to consumers in two ways:
  - Shared hosting: Multiple consumers share compute systems
  - Dedicated hosting: Individual consumers have dedicated compute systems
- Typically providers use compute virtualization and offer compute systems in the form of virtual machines

A compute system is a computing platform (hardware, firmware, and software) that runs platform and application software. Examples of physical compute systems include desktops, laptops, servers, mobile devices, and so on. A compute system consists of processor(s), memory, I/O devices, and a collection of software to perform computing operations. The software includes the operating system (OS), file system, logical volume manager, device drivers, and so on. The OS may include the other software or they can be installed individually. The OS manages the physical components and application execution, and provides a user interface (UI) for users to operate and use the compute system.

In a cloud environment, providers typically deploy x86-based servers or hosts to build the physical layer. These compute systems execute a provider's as well as the consumers' software. Consumers may deploy their applications entirely on cloud compute systems or may leverage the cloud for specific scenarios, such as application development and testing, or during peak workloads. Two or more compute systems are typically combined together into a cluster – a group of compute systems that function together, sharing certain network and storage resources, and are viewed as a single system. Compute clusters are typically implemented to provide high availability and for balancing computing workloads. Compute clustering is covered in detail in the 'Business Continuity' module.

A cloud provider typically offers compute systems to consumers in two ways: shared hosting and dedicated hosting. In shared hosting, the compute systems are shared among multiple consumers. For example, a provider hosts a consumer's website on the same compute system as the websites of other consumers. In dedicated hosting, a provider offers to a consumer dedicated compute systems that are not shared with any other consumer.

Providers typically install compute virtualization software (hypervisor) on a compute system and create multiple virtual compute systems, known as virtual machines (VMs), each capable of running its own OS. In this case, the hypervisor performs compute system management tasks and allocates the compute system's resources, such as processor and memory, dynamically to each VM. The provider allocates the VMs running on a hypervisor to consumers for deploying their applications. The provider may pre-install an OS on a VM or may enable the consumers to install an OS of their choice. Compute virtualization is covered in detail in the 'Virtual Layer' module.



A compute system typically comprises the following key physical hardware components assembled inside an enclosure:

- **Processor:** A processor, also known as a Central Processing Unit (CPU), is an integrated circuit (IC) that executes the instructions of a software program by performing fundamental arithmetical, logical, and input/output operations. A common processor/instruction set architecture is the x86 architecture with 32-bit and 64-bit processing capabilities. Modern processors have multiple cores (independent processing units), each capable of functioning as an individual processor.

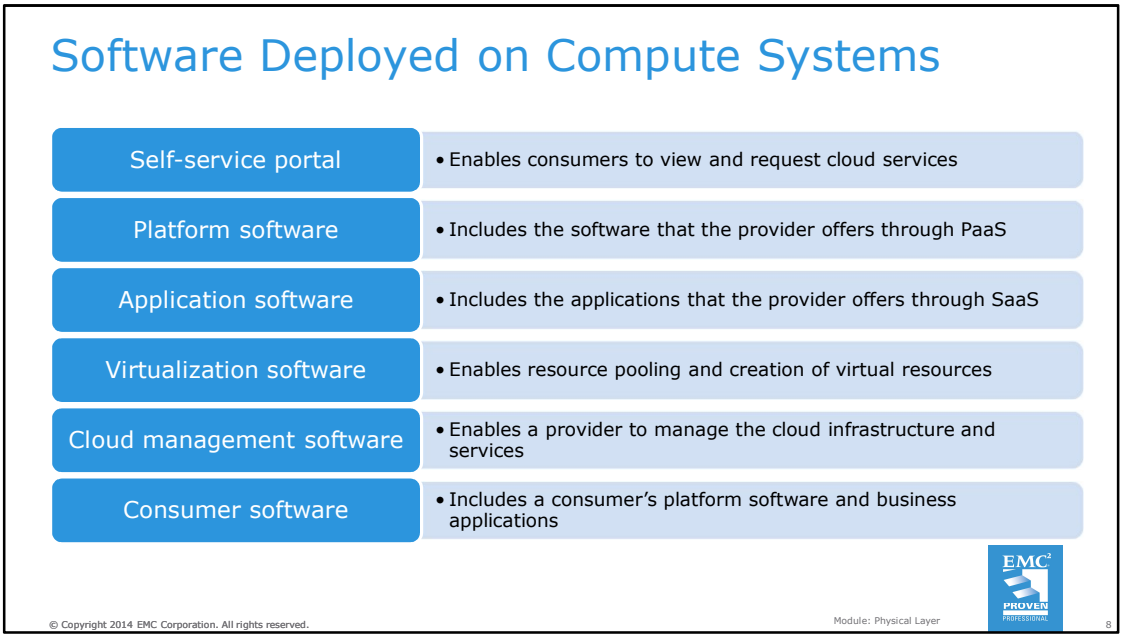
- **Random-Access Memory (RAM):** The RAM or main memory is a volatile data storage device internal to a compute system. The RAM holds the software programs for execution and the data used by the processor.

- **Read-Only Memory (ROM):** A ROM is a type of semiconductor memory that contains the boot firmware (that enables a compute system to start), power management firmware, and other device-specific firmware.

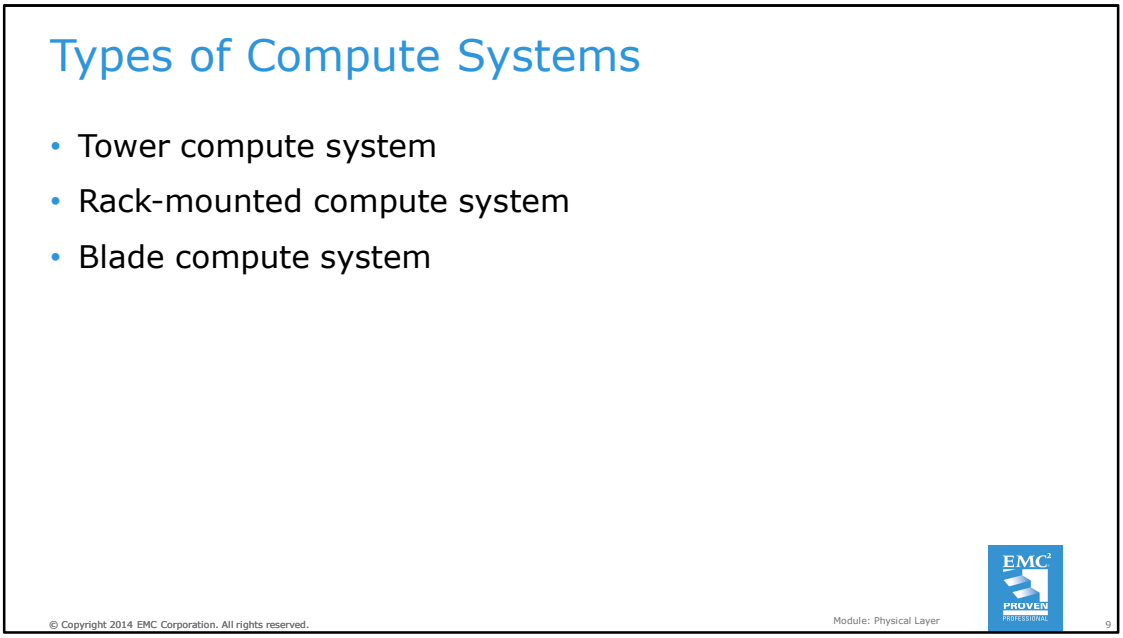
- **Motherboard:** A motherboard is a printed circuit board (PCB) to which all compute system components connect. It has sockets to hold components such as the microprocessor chip, RAM, and ROM. It also has network ports, I/O ports to connect devices such as keyboard, mouse, and printers, and essential circuitry to carry out computing operations. A motherboard may additionally have integrated components, such as a graphics processing unit (GPU), a network interface card (NIC), and adapters to connect to external storage devices.

- **Chipset:** A chipset is a collection of microchips on a motherboard and it is designed to perform specific functions. The two key chipset types are Northbridge and Southbridge. Northbridge manages processor access to the RAM and the GPU, while Southbridge connects the processor to different peripheral ports, such as USB ports.

(Cont'd)



On a compute system, a cloud provider deploys software such as the self-service portal, the application software and platform software that are offered as services (PaaS and SaaS) to consumers, virtualization software, cloud infrastructure management software, and so on. The provider also enables consumers to deploy their platform software and business applications on the compute systems. The slide provides a list and a brief description of the software that are deployed on compute systems in a cloud environment.



The compute systems used in building data centers and cloud infrastructure are typically classified into three categories:

- Tower compute system
- Rack-mounted compute system
- Blade compute system

Tower Compute System

- Built in an upright enclosure called a "tower"
- Has integrated power supply and cooling
- A group of towers occupies significant floor space, requires complex cabling, and generates noise from cooling units
- Deploying in large environments may involve substantial expenditure

A tower compute system, also known as a tower server, is a compute system built in an upright enclosure called a "tower", which is similar to a desktop cabinet. Tower servers have a robust build, and have integrated power supply and cooling. They typically have individual monitors, keyboards, and mice. Tower servers occupy significant floor space and require complex cabling when deployed in a data center. They are also bulky and a group of tower servers generates considerable noise from their cooling units. Tower servers are typically used in smaller environments. Deploying a large number of tower servers in large environments may involve substantial expenditure.

Rack-mounted Compute System

- Designed to be fixed on a frame called a "rack"
  - A rack is a standardized enclosure with mounting slots for vertically stacking compute systems
- Simplifies network cabling, consolidates network equipment, and reduces floor space use
- Administrators may use a console mounted on the rack to manage the compute systems

A rack-mounted compute system, also known as a rack server or an industrial server, is a compute system designed to be fixed on a frame called a "rack". A rack is a standardized enclosure containing multiple mounting slots called "bays", each of which holds a server in place with the help of screws. A single rack contains multiple servers stacked vertically in bays, thereby simplifying network cabling, consolidating network equipment, and reducing floor space use. Each rack server has its own power supply and cooling unit. A "rack unit" (denoted by U or RU) is a unit of measure of the height of a server designed to be mounted on a rack. One rack unit is 1.75 inches. A rack server is typically 19 inches (482.6 mm) in width and 1.75 inches (44.45 mm) in height. This is called a 1U rack server. Other common sizes of rack servers are 2U and 4U. Some common rack cabinet sizes are 27U, 37U, and 42U.

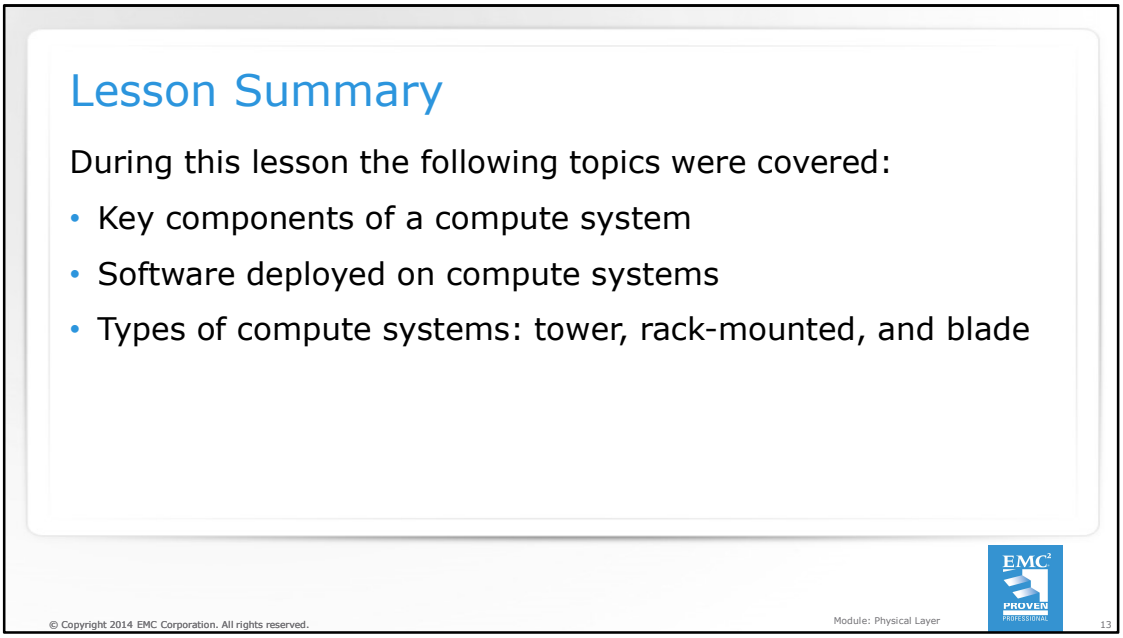
Typically, a console with a video screen, keyboard, and mouse is mounted on a rack to enable administrators to manage the servers in the rack. A keyboard, video, and mouse (KVM) switch connects the servers in the rack to the console and enables the servers to be controlled from the console. An administrator can switch between servers using keyboard commands, mouse commands, or touchscreen selection. Using a KVM switch eliminates the need for a dedicated keyboard, monitor, and mouse for each server and saves space and reduces cable clutter. Some concerns with rack servers are that they are cumbersome to work with, and they generate a lot of heat because of which more cooling is required, which in turn increases power costs.

Blade Compute System

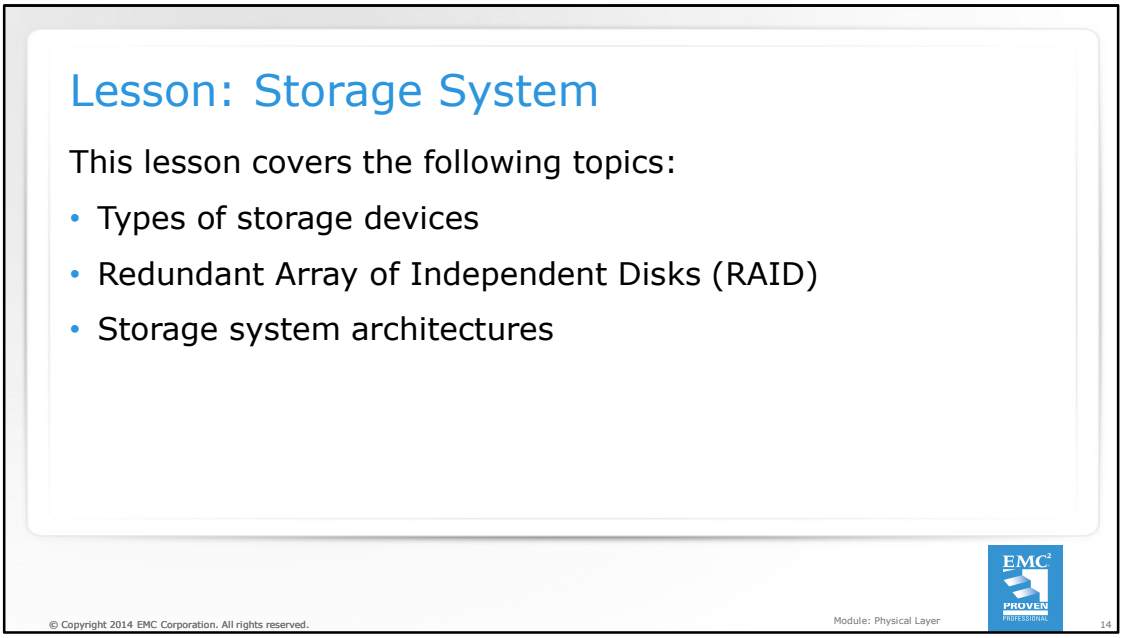
- Comprises an electronic circuit board with only the core processing components
- Multiple blades are housed in a blade chassis
  - The chassis provides integrated power supply, cooling, networking, and management
- Blades are interconnected via a high speed bus
- Modular design increases compute system density and scalability

A blade compute system, also known as a blade server, is an electronic circuit board containing only core processing components, such as processor(s), memory, integrated network controllers, storage drive, and essential I/O cards and ports. Each blade server is a self-contained compute system and is typically dedicated to a single application. A blade server is housed in a slot inside a blade enclosure (or chassis), which holds multiple blades and provides integrated power supply, cooling, networking, and management functions. The blade enclosure enables interconnection of the blades through a high speed bus and also provides connectivity to external storage systems.

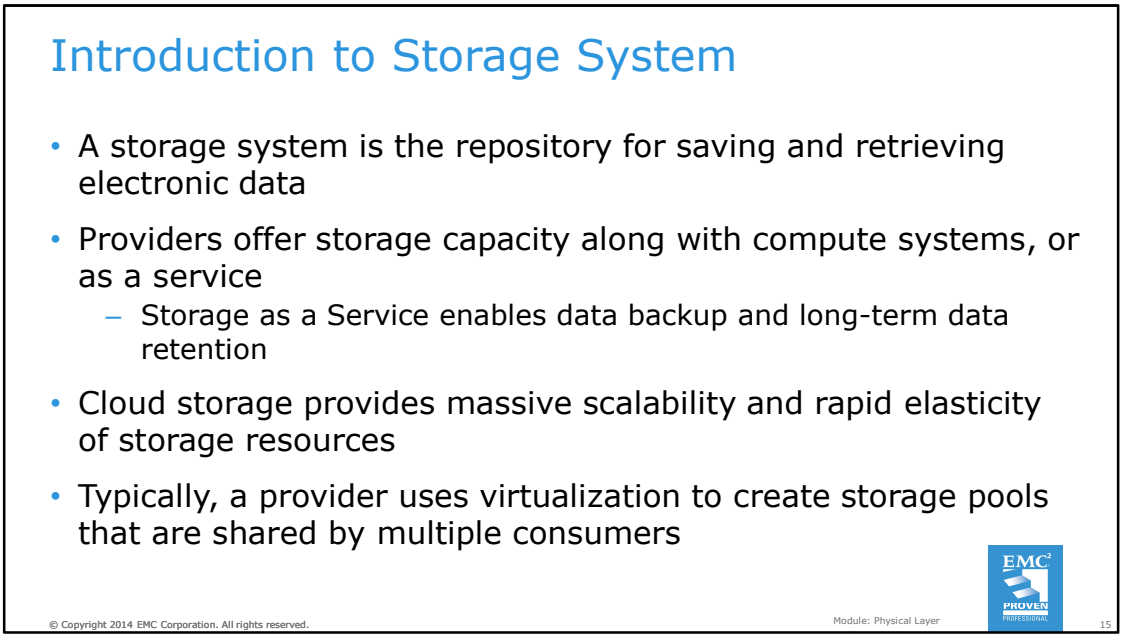
The modular design of blade servers makes them smaller, which minimizes floor space requirements, increases compute system density and scalability, and provides better energy efficiency as compared to tower and rack servers. It also reduces the complexity of the compute infrastructure and simplifies compute infrastructure management. It provides these benefits without compromising on any capability that a non-blade compute system provides. Some concerns with blade servers include the high cost of a blade system (blade servers and chassis), and the proprietary architecture of most blade systems due to which a blade server can typically be plugged only into a chassis from the same vendor.



This lesson covered the key software that are deployed on a compute system in a cloud environment. This lesson also covered the key components of a compute system, such as the processor, RAM, ROM, motherboard, and the chipset. Finally, this lesson covered the three common types of physical compute systems—tower, rack-mounted, and blade—that are used in building a cloud infrastructure.

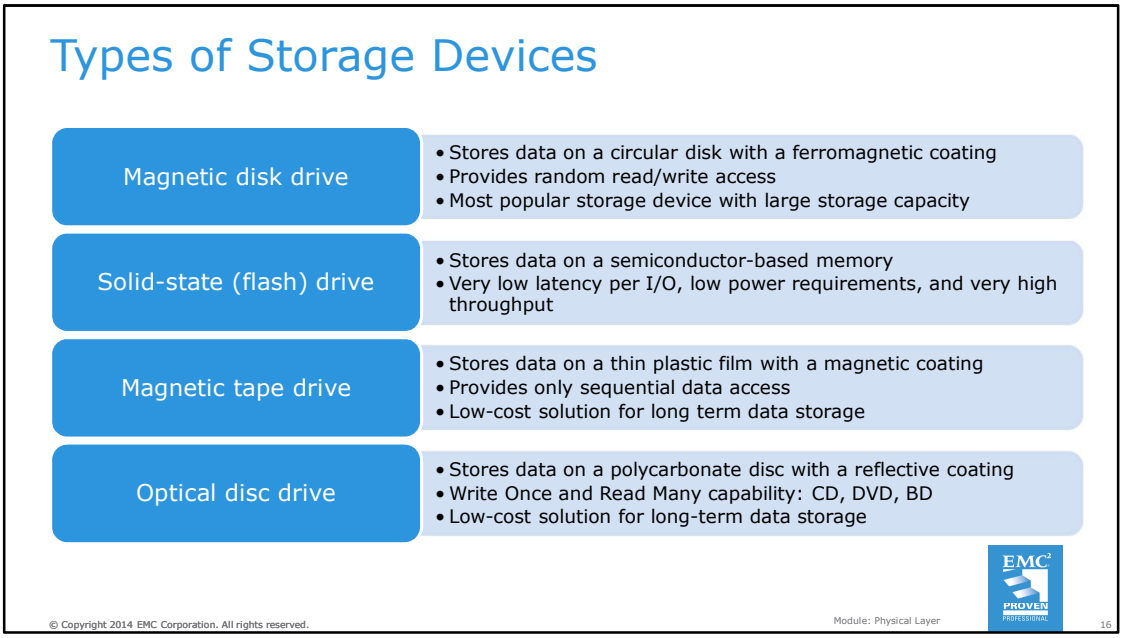


This lesson covers the common types of persistent storage devices. This lesson also covers Redundant Array of Independent Disks (RAID) and its use in data protection and storage performance improvement. Further, this lesson also covers the different types of storage system architectures, namely block-based, file-based, object-based, and unified storage systems.



Data created by individuals, businesses, and applications need to be persistently stored so that it can be retrieved when required for processing or analysis. A storage system is the repository for saving and retrieving electronic data and is integral to any cloud infrastructure. A storage system has devices, called storage devices (or storage) that enable the persistent storage and the retrieval of data. Storage capacity is typically offered to consumers along with compute systems. Apart from providing storage along with compute systems, a provider may also offer storage capacity as a service (Storage as a Service), which enables consumers to store their data on the provider's storage systems in the cloud. This enables the consumers to leverage cloud storage resources for purposes such as data backup and long-term data retention.

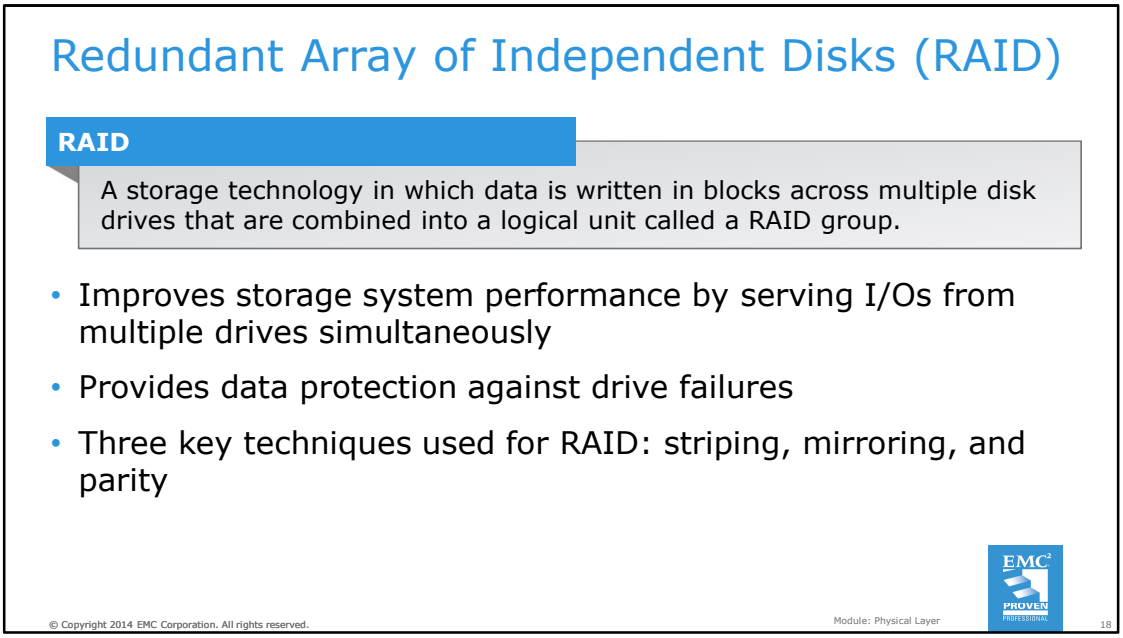
A cloud storage infrastructure is typically created by logically aggregating and pooling the storage resources from one or more data centers to provide virtual storage resources. Cloud storage provides massive scalability and rapid elasticity of storage resources. The cloud storage infrastructure is typically shared by multiple tenants or consumers which improves the utilization of storage resources.



A magnetic disk is a circular storage medium made of non-magnetic material (typically an alloy) and coated with a ferromagnetic material. Data is stored on both surfaces (top and bottom) of a magnetic disk by polarizing a portion of the disk surface. A disk drive is a device that comprises multiple rotating magnetic disks, called platters, stacked vertically inside a metal or plastic casing. Each platter has a rapidly moving arm to read from and write data to the disk. Disk drives are currently the most popular storage medium for storing and accessing data for performance-intensive applications. Disks support rapid access to random data locations and data can be written or retrieved quickly for a number of simultaneous users or applications. Disk drives use pre-defined protocols, such as Advanced Technology Attachment (ATA), Serial ATA (SATA), Small Computer System Interface (SCSI), Serial Attached SCSI (SAS), and Fibre Channel (FC). These protocols reside on the disk interface controllers that are typically integrated with the disk drives. Each protocol has its unique performance, cost, and capacity characteristics.

A solid-state drive (SSD) uses semiconductor-based memory, such as NAND and NOR chips, to store and retrieve data. SSDs, also known as "flash drives", deliver the ultra-high performance required by performance-sensitive applications. These devices, unlike conventional mechanical disk drives, contain no moving parts and therefore do not exhibit the latencies associated with read/write head movement and disk rotation. Compared to other available storage devices, SSDs deliver a relatively high number of input/output operations per second (IOPS) with very low response times. They also consume less power and typically have a longer lifetime as compared to mechanical drives. However, flash drives do have the highest cost per gigabyte ($/GB) ratio.

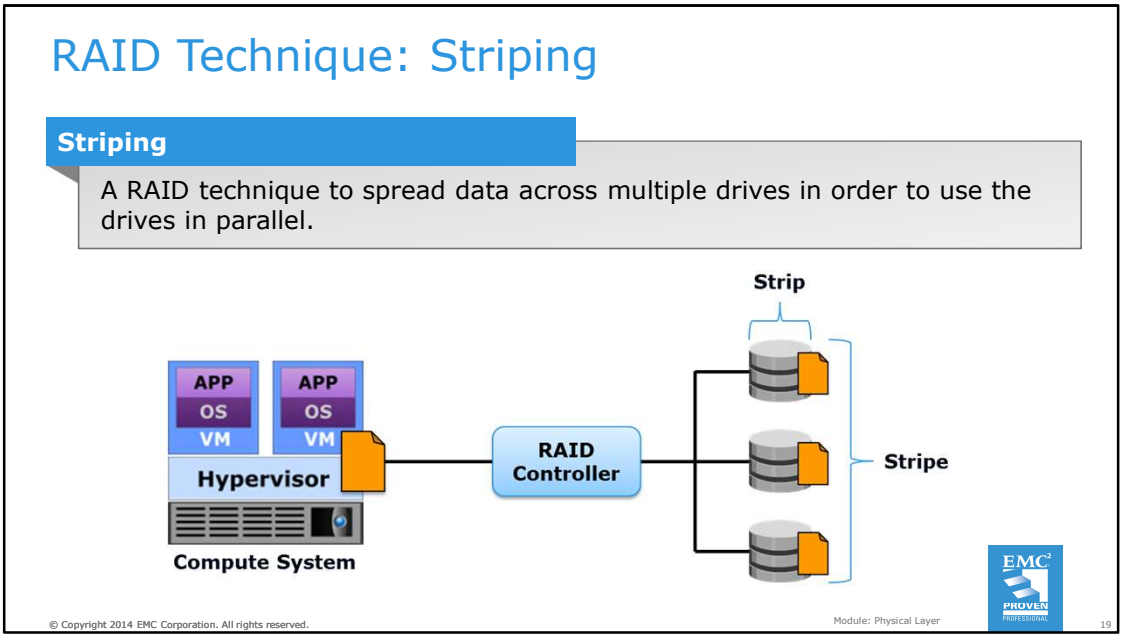
(Cont'd)



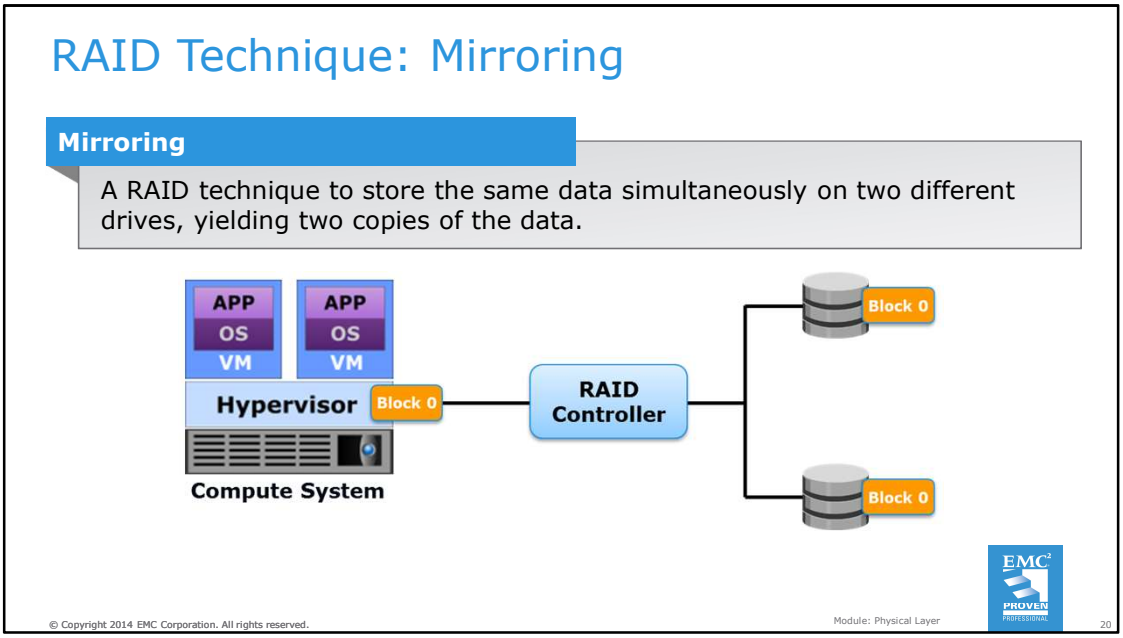
Redundant Array of Independent Disks (RAID) is a storage technology in which multiple disk drives are combined into a logical unit called a RAID group and data is written in blocks across the disks in the RAID group. RAID protects against data loss when a drive fails, through the use of redundant drives and parity. RAID also helps in improving the storage system performance as read and write operations are served simultaneously from multiple disk drives. For example, if the RAID group has four disk drives, data is written across all four of them simultaneously, which provides four times better write performance as compared to using a single drive. Similarly, during read operation, the data is retrieved simultaneously from each drive.

RAID is typically implemented by using a specialized hardware controller present either on the host or on the array. The key functions of a RAID controller are management and control of drive aggregations, translation of I/O requests between logical and physical drives, and data regeneration in the event of drive failures.

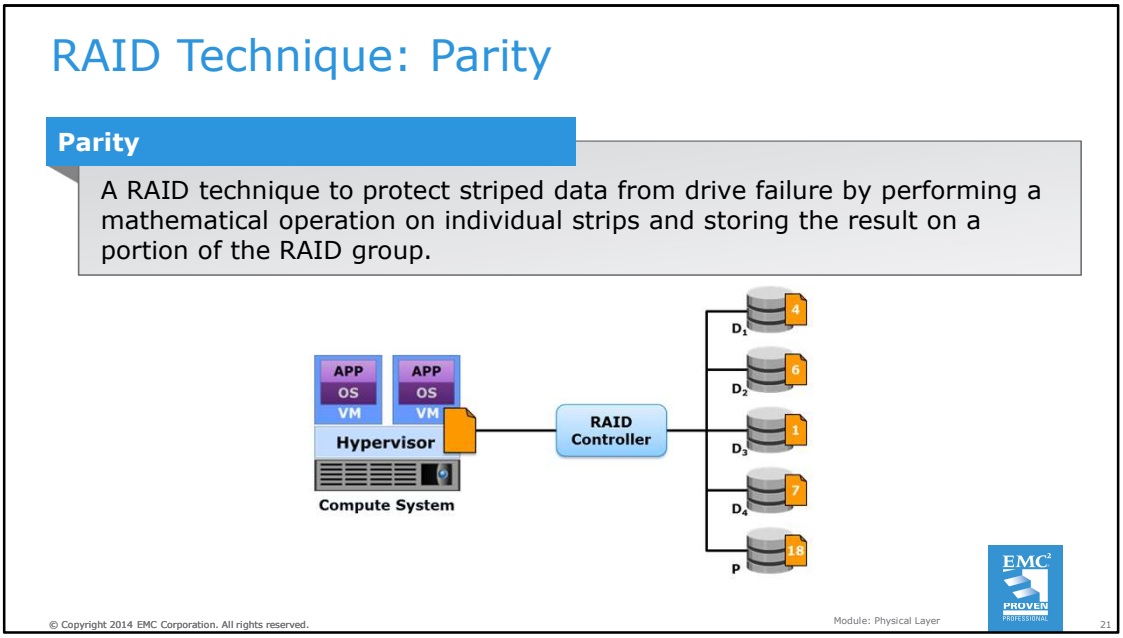
The three different RAID techniques that form the basis for defining various RAID levels are striping, mirroring, and parity. These techniques determine the data availability and performance of a RAID group as well as the relative cost of deploying the storage solution. A cloud provider must select the appropriate RAID levels to meet the requirements of cloud service delivery.



Striping is a technique to spread data across multiple drives in order to use the drives in parallel and increase performance as compared to the use of a single drive. Each drive in a RAID group has a predefined number of contiguously addressable blocks (the smallest individually addressable unit of storage) called a "strip". A set of aligned strips that span across all the drives within the RAID group is called a "stripe". All strips in a stripe have the same number of blocks. Although striped RAID provides improved read-write performance, it does not provide any data protection in case of disk failure.

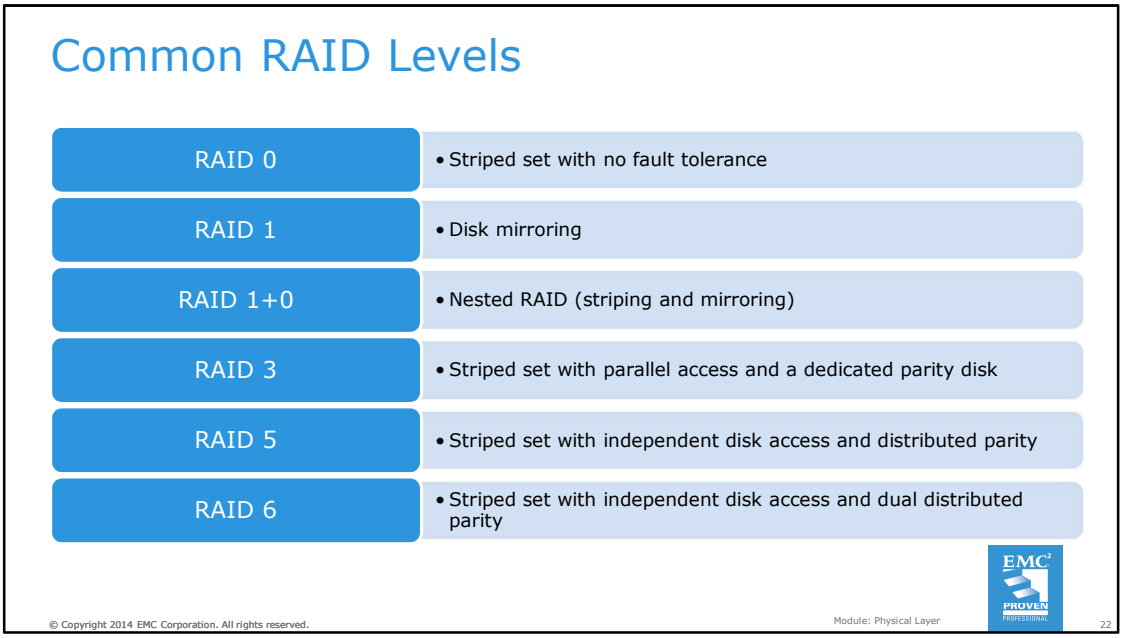


Mirroring is a technique in which the same data is stored simultaneously on two different drives, resulting in two copies of the data. This is called a "mirrored pair". Even if one drive fails, the data is still intact on the surviving drive and the RAID controller continues to service data requests using the surviving drive of the mirrored pair. When the failed disk is replaced with a new disk, the controller copies the data from the surviving disk of the mirrored pair to the new disk. This activity is transparent to the host. In addition to providing data redundancy, mirroring enables fast recovery from disk failure. Since mirroring involves duplication of data, the amount of storage capacity needed is twice the amount of data being stored. This increases costs because of which mirroring is typically preferred for mission-critical applications that cannot afford the risk of any data loss. Mirroring improves read performance because read requests can be serviced by both disks. However, compared to a single disk and striping, write performance is slightly lower in mirroring because each write request manifests as two writes on the disk drives.

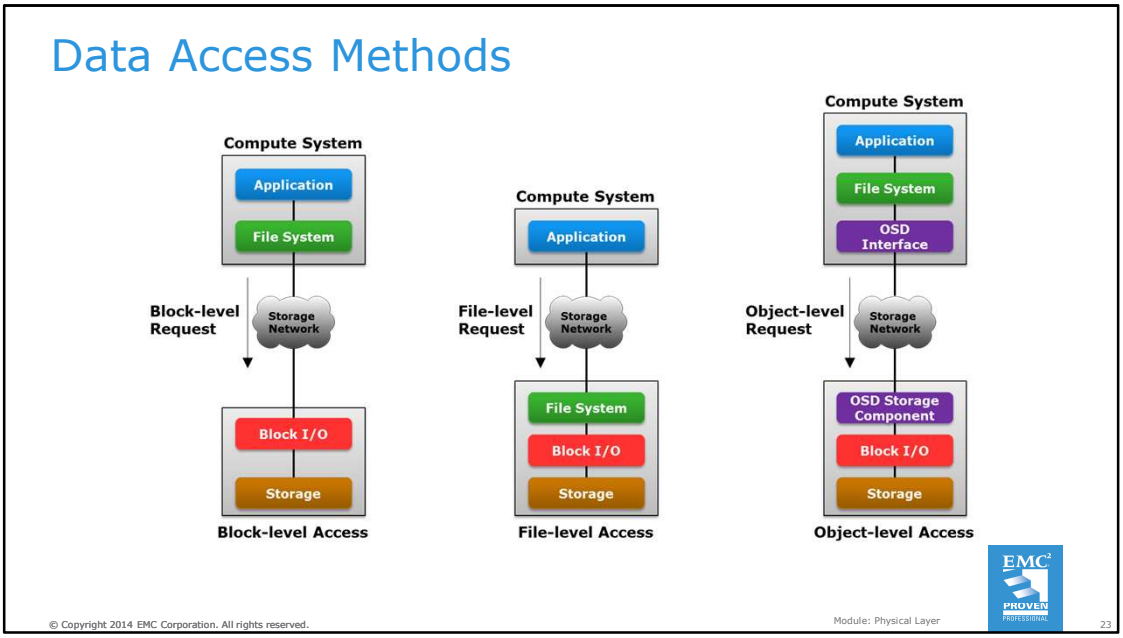


Parity is a value derived by performing a mathematical operation on individual strips of data and stored on a portion of a RAID group. It enables the recreation of missing data in case of a drive failure. Parity is a redundancy technique that ensures data protection without maintaining a full set of duplicate data. The RAID controller calculates the parity using techniques such as "bitwise exclusive or" (XOR). Parity information can be stored on separate, dedicated disk drives or distributed across the drives in a RAID group. Compared to mirroring, parity implementation considerably reduces the cost associated with data protection. However, a limitation of parity implementation is that parity is recalculated every time there is a change in data, which may affect the performance of the RAID array.

In the figure on the slide, the first four disks, labeled D1 to D4, contain data. The data elements are 4, 6, 1, and 7. The fifth disk, labeled P, stores the parity information i.e. 18, which is the sum of the data elements. If one of the drives fails, the missing value can be calculated by subtracting the sum of the remaining elements from the parity value.



RAID levels are implementations of the striping, mirroring, and parity techniques. Some RAID levels use a single technique, while others use a combination of the techniques. The commonly used RAID levels are RAID 0 – that uses striping, RAID 1 – that uses mirroring, RAID 1+0 – which is a combination of RAID 1 and RAID 0, and RAID 3, 5, and 6 – that use a combination of striping and parity.



Data can be accessed from a compute system (or a compute cluster) through block-level, file-level, or object-level schemes. External storage systems can be connected to the compute system directly or over a network. An application on the compute system stores and accesses data using the underlying infrastructure comprising an OS, a file system, network connectivity, and storage. In general, an application requests data by specifying the file name and the location. The file system maps the file attributes to the logical block address (LBA) of the data and sends it to the storage system. The LBA simplifies addressing by using a linear address to access the block of data. The storage system converts the LBA to a physical address called the cylinder-head-sector (CHS) address and fetches the data.

In *block-level access*, a storage volume (a logical unit of storage composed of multiple blocks, typically created from a RAID set) is created and assigned to the compute system to house created file systems. In this case, an application data request is sent to the file system and converted into a block-level (logical block address) request. This block level request is sent over the network to the storage system. The storage system then converts the logical block address to a CHS address and fetches the data in block-sized units.

In *file-level access*, the file system is created on a separate file server, which is connected to storage. A file-level request from the application is sent over the network to the file server hosting the file system. The file system then converts the file-level request into block-level addressing and sends the request to the storage to access the data.

In *object-level access*, data is accessed over the network in terms of self-contained objects, each having a unique object identifier. In this case, the application request is sent to the file system. The file system communicates to the object-based storage device (OSD) interface, which in turn sends the object-level request by using the unique object ID over the network to the storage system. The storage system has an OSD storage component that is responsible for managing the access to the object on the storage system. The OSD storage component converts the object-level request into block-level addressing and sends it to the storage to access the data.

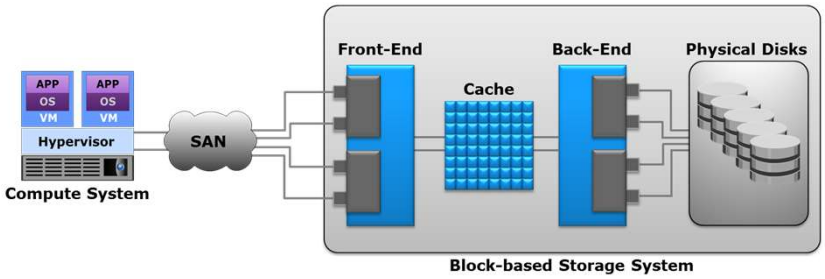
Storage System Architecture

- Storage system architectures are based on the data access methods
- Common storage system options are:
  - Block-based
  - File-based
  - Object-based
  - Unified

Storage system architecture is a critical design consideration for building cloud infrastructure. A cloud provider must choose the appropriate storage, and ensure adequate capacity to maintain the overall performance of the environment. Storage system architectures are based on the data access methods. The common variants are block-based, file-based, object-based, and unified storage systems. A unified storage system architecture uses all the three data access methods. A cloud provider may deploy one or more types of these storage systems to meet the requirements of different applications.

Block-based Storage System

- Enables creating and assigning storage volumes to compute systems
  - Compute system discovers the volumes as local drives
  - Required file system can be created on the volumes



A block-based storage system enables the creation and assignment of storage volumes to compute systems. The compute OS (or hypervisor) discovers these storage volumes as local drives. A file system can be created on these storage volumes, for example NTFS in a Windows environment, which can then be formatted and used by applications.

A block-based storage system typically comprises four key components:

- Front-end Controller(s)
- Cache Memory
- Back-end Controller(s)
- Physical disks

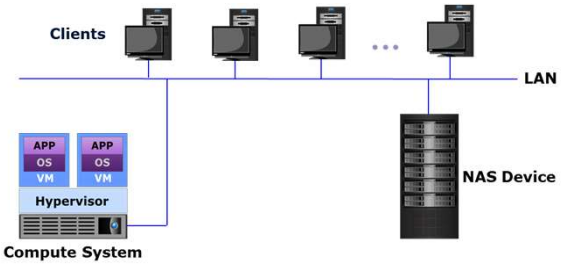
The *front-end controller* provides the interface between the storage system and the compute systems. Typically, there are redundant controllers in the front-end for high availability, and each controller contains multiple ports. Each front-end controller has processing logic that executes the appropriate transport protocol, such as Fibre Channel, iSCSI, or FCoE (discussed later in this module) for storage connections. Front-end controllers route data to and from a cache memory via an internal data bus.

The *cache* is a semiconductor memory where data is placed temporarily to reduce the time required to service I/O requests from the compute system. The cache improves storage system performance by isolating compute systems from the mechanical delays associated with disk drives. Accessing data from the cache typically takes less than a millisecond.

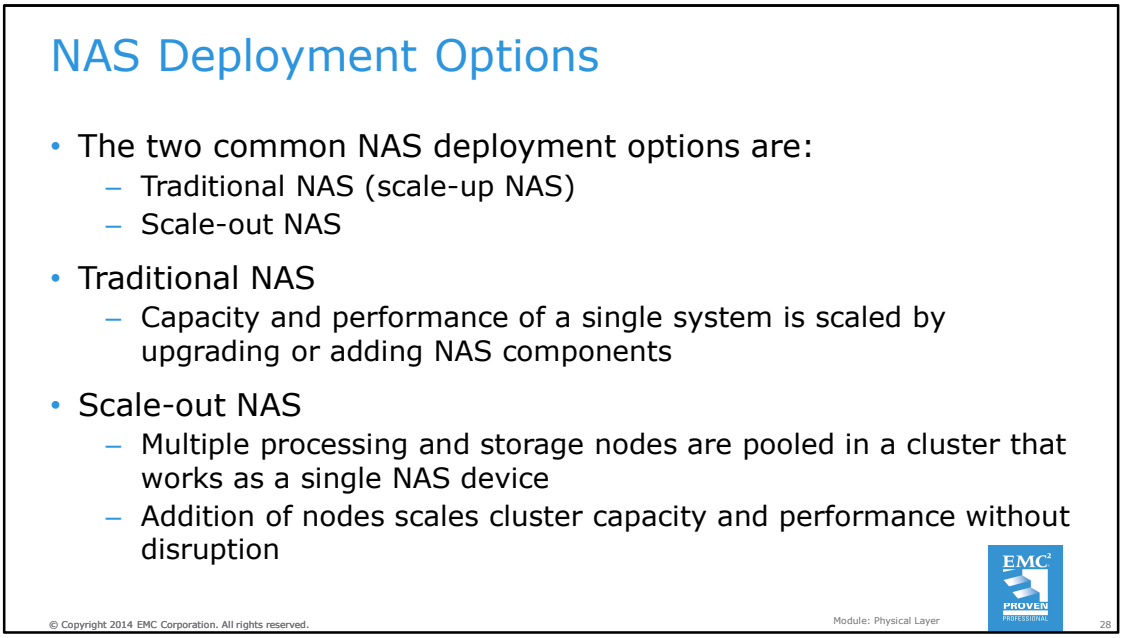
(Cont'd)

File-based Storage System

- A dedicated, high performance file server with storage (also known as Network-attached Storage)
- Enables clients to share files over an IP network
  - Supports data sharing for UNIX and Windows users
- Uses a specialized OS that is optimized for file I/O



A file-based storage system, also known as Network-Attached Storage (NAS), is a dedicated, high-performance file server having either integrated storage or connected to external storage. NAS enables clients to share files over an IP network. NAS supports NFS and CIFS protocols to give both UNIX and Windows clients the ability to share the same files using appropriate access and locking mechanisms. NAS systems have integrated hardware and software components, including a processor, memory, NICs, ports to connect and manage physical disk resources, an OS optimized for file serving, and file sharing protocols. A NAS system consolidates distributed data into a large, centralized data pool accessible to, and shared by, heterogeneous clients and application servers across the network. Consolidating data from numerous and dispersed general purpose servers onto NAS results in more efficient management and improved storage utilization. Consolidation also offers lower operating and maintenance costs.



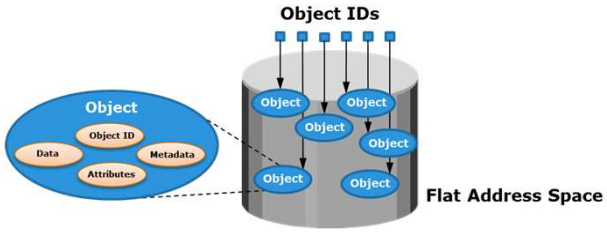
There are two common NAS deployment options: traditional NAS systems (scale-up NAS) and scale-out NAS systems.

A traditional NAS solution provides the capability to scale the capacity and performance of a single NAS system. Scaling up a NAS system involves upgrading or adding NAS components and storage to the NAS system. These NAS systems have a fixed capacity ceiling, and performance is impacted as the capacity limit is approached.

Scale-out NAS is designed to address the rapidly growing area of unstructured data (data that does not fit in tables and rows), especially Big Data (data sets whose size or scale break traditional tools). Scale-out NAS enables the creation of a clustered NAS system by pooling multiple processing and storage nodes together. The cluster works as a single NAS system and is managed centrally. The capacity of the cluster can be increased by simply adding nodes to the it. A node contains common server components and may or may not have disks. As each node is added to the cluster, it increases the aggregated disk, cache, processor, and network capacity of the cluster as a whole. Nodes can be non-disruptively added to the cluster when more performance and capacity is needed. Scale-out NAS creates a single file system that runs on all nodes in the cluster. As nodes are added, the file system grows dynamically and data is evenly distributed (or redistributed) to all nodes in the cluster.

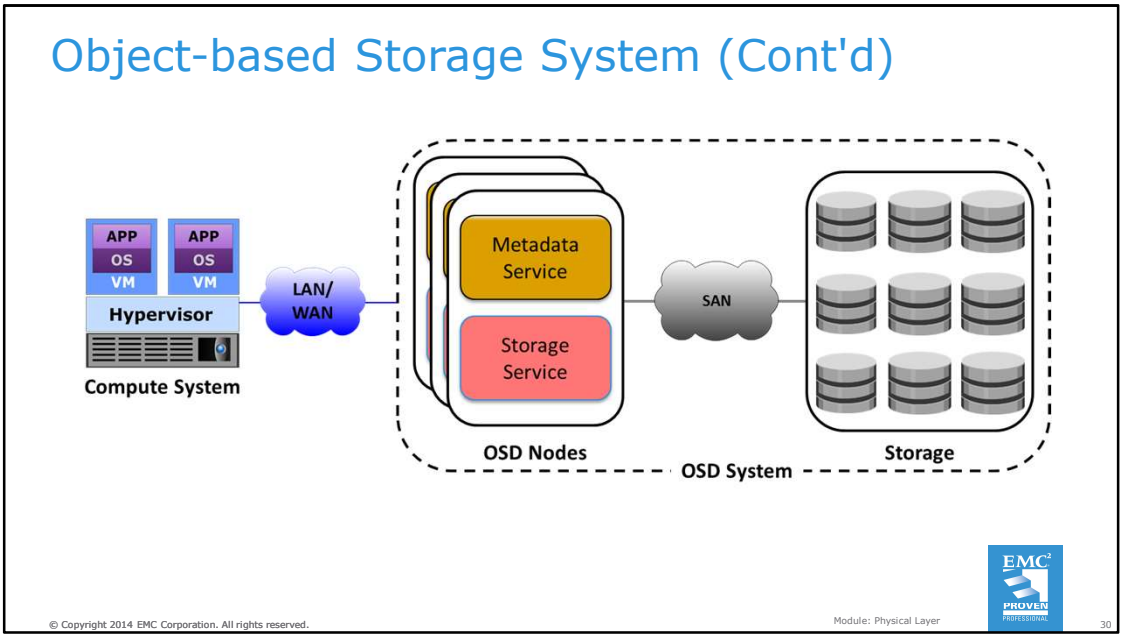
Object-based Storage System

- Stores file data in the form of objects based on data contents and attributes
  - Uses a flat, non-hierarchical address space
- Object contains user data, related metadata, and user-defined attributes
  - Objects are uniquely identified using object ID



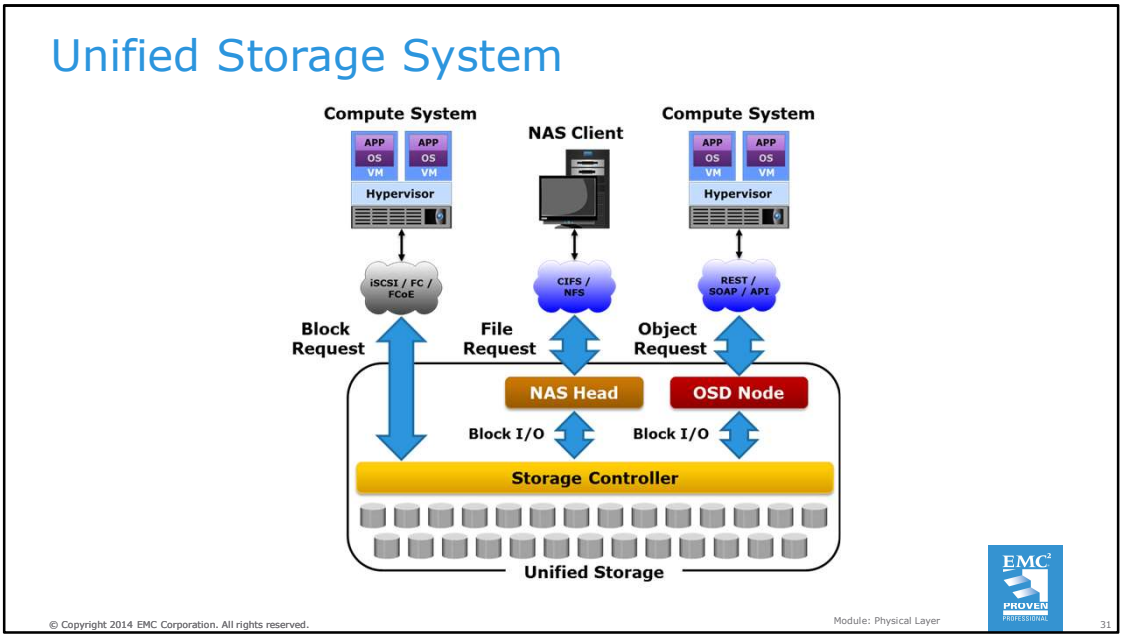
Object-based storage is a way to store file data in the form of objects based on the content and other attributes of the data rather than the name and location of the file. An object contains user data, related metadata (size, date, ownership, etc.), and user defined attributes of data (retention, access pattern, and other business-relevant attributes). The additional metadata or attributes enable optimized search, retention and deletion of objects. For example, when an MRI scan of a patient is stored as a file in a NAS system, the metadata is basic and may include information such as file name, date of creation, owner, and file type. When stored as an object, the metadata component of the object may include additional information such as patient name, ID, attending physician's name, and so on, apart from the basic metadata.

Each object stored in the object-based storage system is identified by a unique identifier called the *object ID*. The object ID allows easy access to objects without having to specify the storage location. The object ID is generated using specialized algorithms (such as a hash function) on the data and guarantees that every object is uniquely identified. Any changes in the object, like user-based edits to the file, results in a new object ID. This makes object-based storage a preferred option for long term data archiving to meet regulatory or compliance requirements. The object-based storage system uses a flat, non-hierarchical address space to store data, providing the flexibility to scale massively. Cloud service providers leverage object-based storage systems to offer Storage as a Service because of its inherent security, scalability, and automated data management capabilities. Object-based storage systems support web service access via REST and SOAP.



The object-based storage system has three key components: nodes, internal (private) network, and storage. The object-based storage system is composed of one or more nodes. In this context, a node is a server that runs the object-based storage operating environment and provides services to store, retrieve, and manage data in the system. The object-based storage system node has two key services: metadata service and storage service. The metadata service is responsible for generating the object ID from the contents of a file. It also maintains the mapping between the object IDs and the file system namespace. The storage service manages a set of drives on which the data is stored. The nodes connect to the storage via an internal network. The internal network provides both node-to-node connectivity and node-to-storage connectivity. The application server accesses the object-based storage node to store and retrieve data over an external network. In some implementations, the metadata service might reside on the application server or on a separate server.

Object-based storage provides the capability to automatically detect and repair corrupted objects, and to alert the administrator of any potential problem. It also provides on-demand reporting and event notification. Some object-based storage systems support storage optimization techniques such as single instance storage, where only one instance of an object is stored, thereby optimizing the usable capacity.

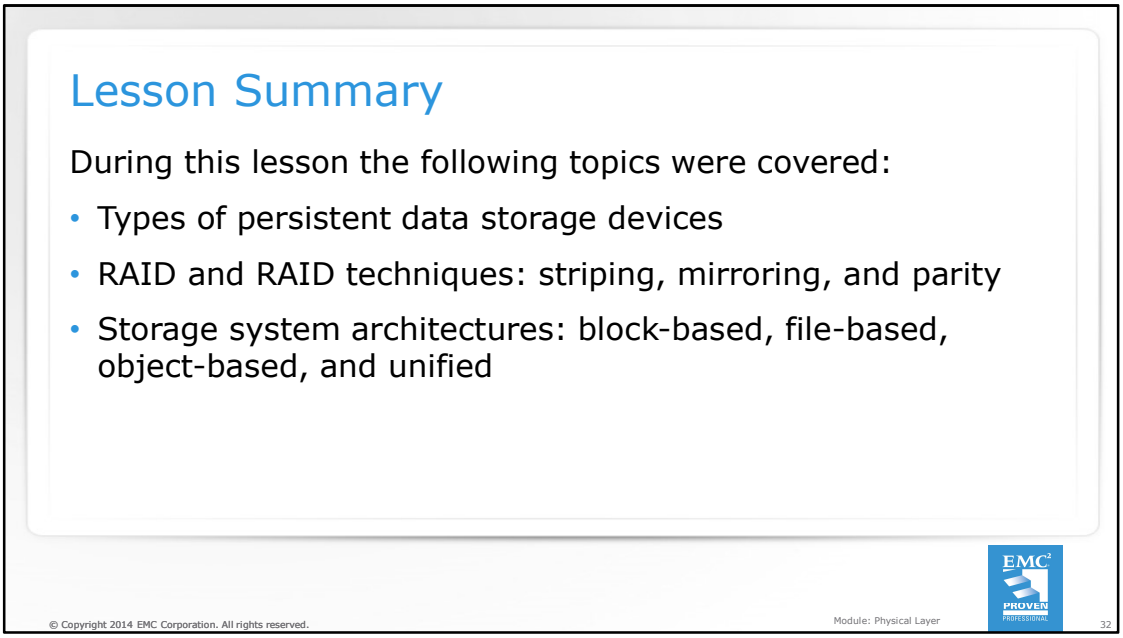


Unified storage or multiprotocol storage has emerged as a solution that consolidates block, file, and object-based access within one storage platform. It supports multiple protocols such as CIFS, NFS, iSCSI, FC, FCoE, REST, and SOAP for data access. Such a unified storage system is managed using a single interface. A unified storage system consists of the following key components: storage controller, NAS head, OSD node, and storage. The storage controller, NAS head, and OSD node may be present either separately or be part of a single unit.

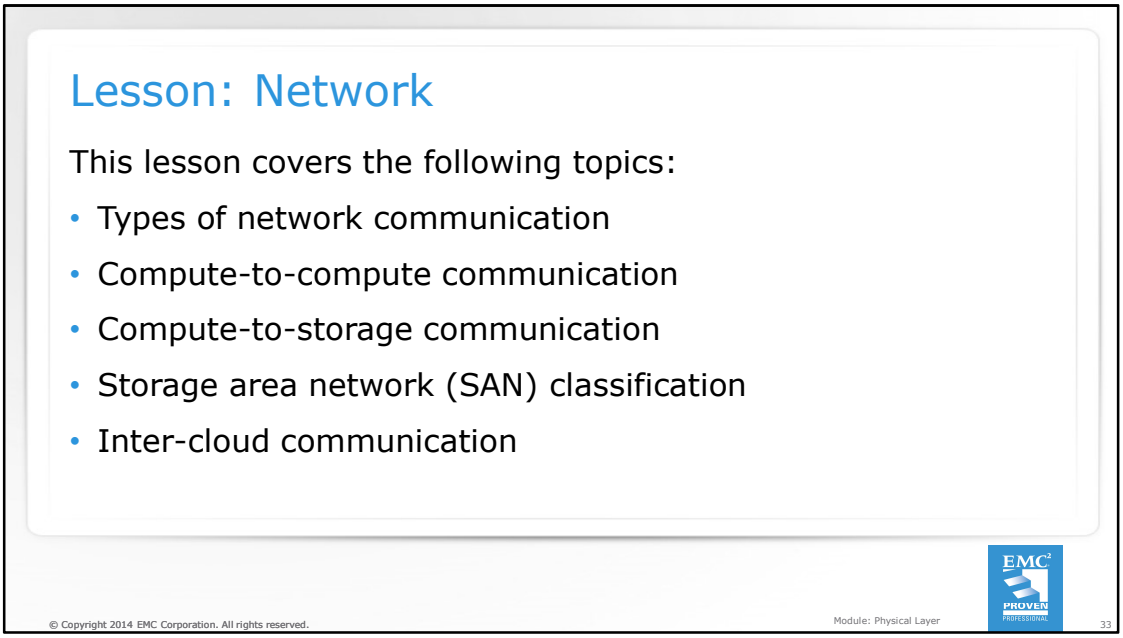
The storage controller provides block-level access to compute systems through various protocols. It contains front-end ports for direct block access. The storage controller is also responsible for managing the back-end storage pool in the storage system. The controller configures storage volumes and presents them to NAS heads and OSD nodes, as well as to the compute systems.

A NAS head is a dedicated file server that provides file access to NAS clients. The NAS head connects to the storage via the storage controller. The system usually has two or more NAS heads for redundancy. The NAS head configures the file systems on assigned volumes, creates NFS, CIFS, or mixed shares, and exports the shares to the NAS clients.

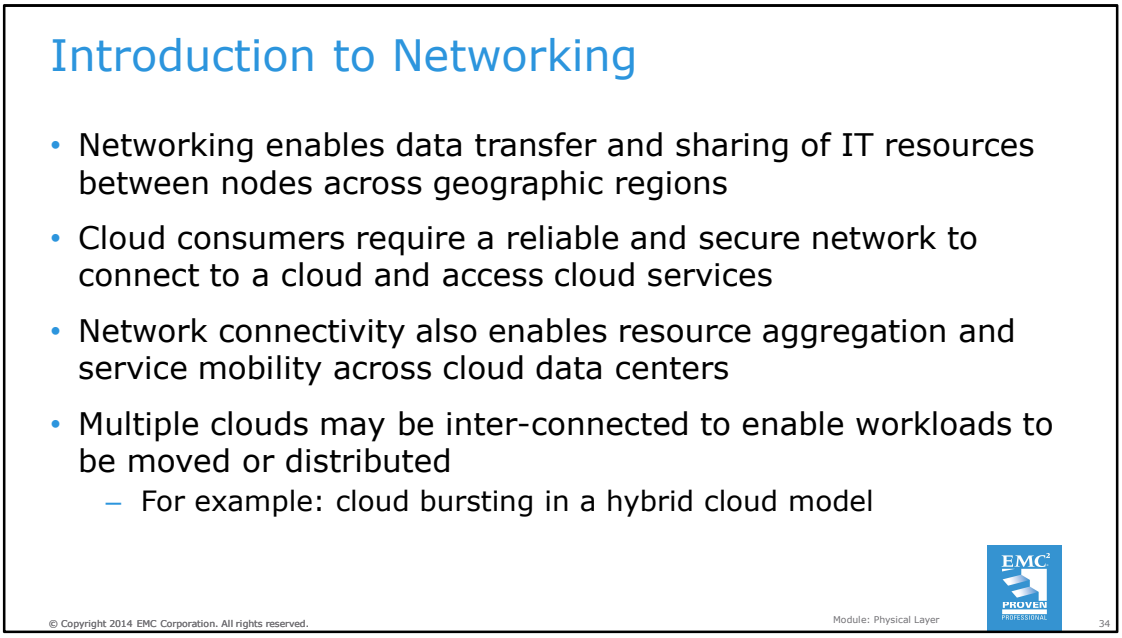
The OSD node also accesses the storage through the storage controller. The volumes assigned to the OSD node appear as physical disks. These disks are configured by the OSD nodes, enabling them to store object data.



This lesson covered the common types of persistent data storage devices. This lesson also covered the different RAID techniques (striping, mirroring, and parity) used for data protection and for improving storage performance. Finally, this lesson covered the different data access methods and the storage system architectures based on them, including block-based, file-based, object-based, and unified storage systems.



This lesson covers the types of network communication and describes compute-to-compute communication. This lesson also covers compute-to-storage communication via a storage area network (SAN) and the classification of SAN. Further, this lesson covers inter-cloud communication.



A network establishes communication paths between the devices in an IT infrastructure. Devices that are networked together are typically called "nodes". A network enables information exchange and resource sharing among a large numbers of nodes spread across geographic regions and over long distances. A network may also be connected to other networks to enable data transfer between nodes.

Cloud providers typically leverage different types of networks supporting different network protocols and transporting different classes of network traffic. As established in the discussion of fundamental cloud characteristics, cloud consumers require reliable and secure network connectivity to access cloud services. A provider connects the cloud infrastructure to a network enabling clients (consumers) to connect to the cloud over the network and use cloud services. For example, in an on-premise private cloud, the clients typically connect to the cloud infrastructure over an internal network, such as a LAN. In case of a public cloud, the cloud infrastructure connects to an external network, typically the Internet, over which consumers access cloud services.

Cloud service providers may also use IT resources at one or more data centers to provide cloud services. If multiple data centers are deployed, the IT resources from these data centers may be logically aggregated by connecting them over a wide area network (WAN). This enables both migration of cloud services across data centers and provisioning cloud services using resources from multiple data centers. Also, multiple clouds may be inter-connected over a WAN to enable workloads to be moved or distributed across clouds. This scenario was covered in the 'Introduction to Cloud Computing' module as part of the discussion on cloud bursting in a hybrid cloud environment.

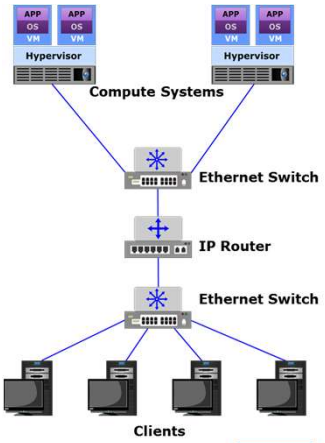
Types of Network Communication

- Based on the nodes connected by a network, the network communication is broadly categorized as:
  - Compute-to-compute communication
  - Compute-to-storage communication
  - Inter-cloud communication

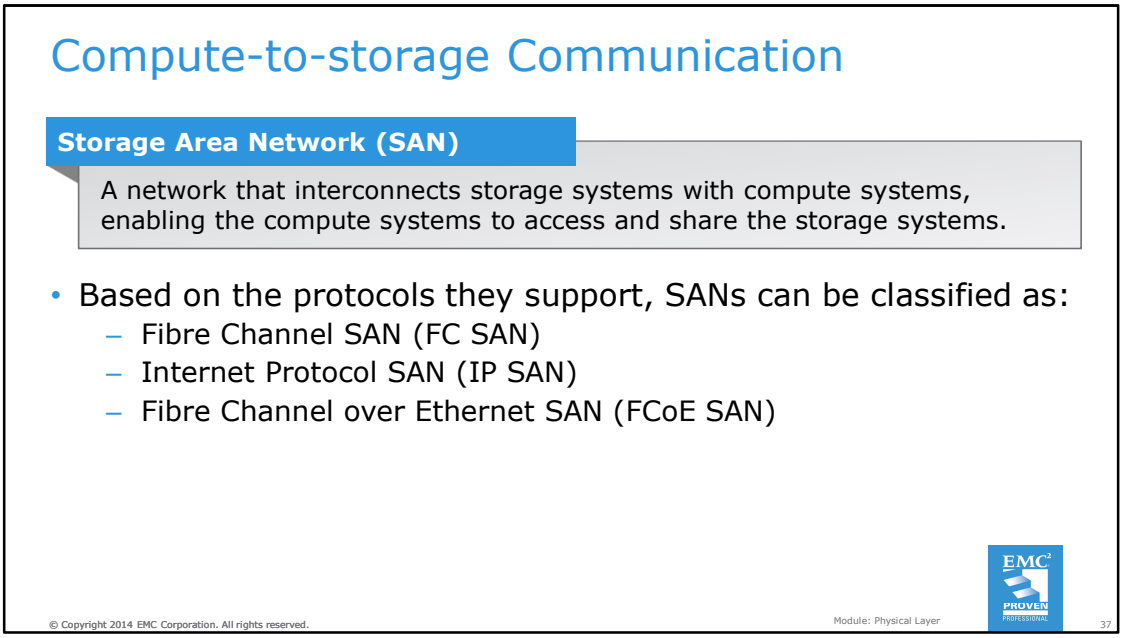
Networks in a cloud environment may be classified into various types based on attributes such as communication protocol, topology, transport medium, and so on. Generally network communication may be categorized into: compute-to-compute communication, compute-to-storage communication, and inter-cloud communication.

Compute-to-compute Communication

- Interconnecting physical compute systems enables compute-to-compute communication
- Compute-to-compute communication typically uses IP-based protocols
- Compute systems connect to a network through physical network card(s)
- Physical switches and routers are common interconnecting devices



Compute-to-compute communication typically uses protocols based on the Internet Protocol (IP). Each physical compute system (running an OS or a hypervisor) is connected to the network through one or more physical network cards, such as a network interface controller (NIC). Physical switches and routers are the commonly-used interconnecting devices. A switch enables different compute systems in the network to communicate with each other. A router enables different networks to communicate with each other. The commonly-used network cables are copper cables and optical fiber cables. The figure on the slide shows a network (Local Area Network – LAN or Wide Area Network – WAN) that provides interconnections among the physical compute systems. The cloud provider has to ensure that appropriate switches and routers, with adequate bandwidth and ports, are in place to ensure the required network performance.



A network of compute systems and storage systems is called a storage area network (SAN). A SAN enables the compute systems to access and share storage systems. Sharing improves the utilization of the storage systems. Using a SAN facilitates centralizing storage management, which in turn simplifies and potentially standardizes the management effort.

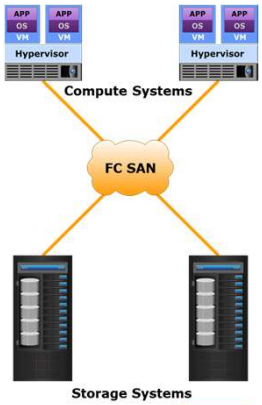
SANs are classified based on protocols they support. Common SAN deployments types are Fibre Channel SAN (FC SAN), Internet Protocol SAN (IP SAN), and Fibre Channel over Ethernet SAN (FCoE SAN).

FC SAN

FC SAN

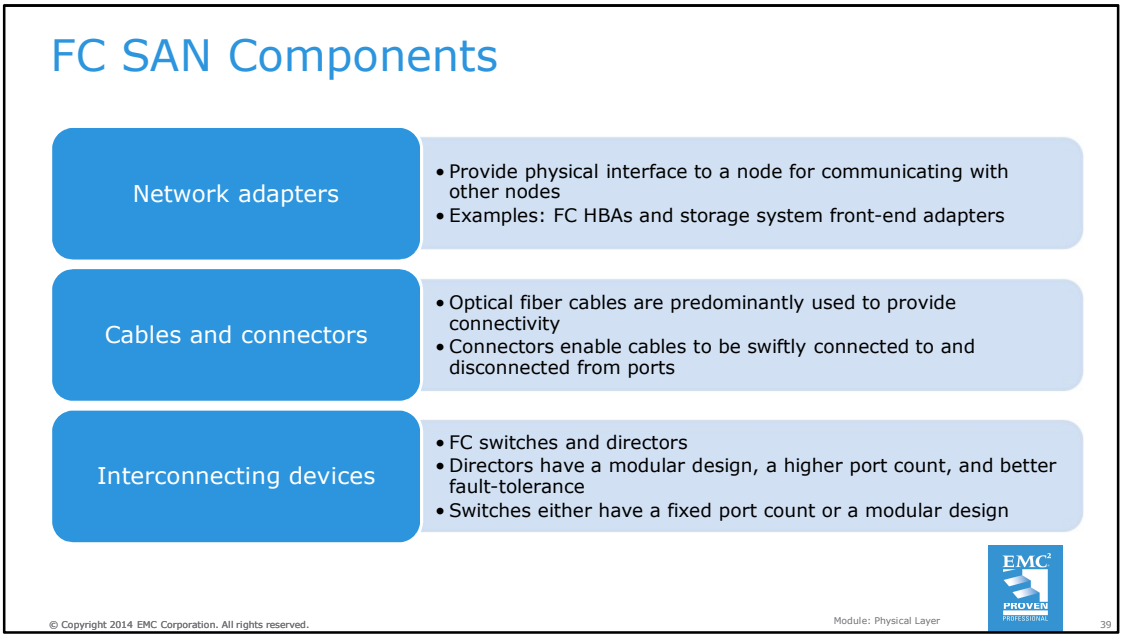
A SAN that uses Fibre Channel (FC) protocol to transport data, commands, and status information between compute and storage systems.

- FC provides block-level access to storage
- FC offers data transfer speeds up to 16 Gbps
- Theoretically, an FC SAN can connect approximately 15 million nodes



An FC SAN is a high speed, dedicated network of compute systems and shared storage systems that uses Fibre Channel (FC) protocol to transport data, commands, and status information between the compute and the storage systems. The FC protocol primarily implements the Small Computer System Interface (SCSI) command set over FC, although it also supports other protocols such as Asynchronous Transfer Mode (ATM), Fibre Connection (FICON), and IP. SCSI over FC overcomes the distance and accessibility limitations associated with traditional, direct-attached SCSI protocol systems. FC protocol provides block-level access to the storage systems. It also provides a serial data transfer interface that operates over both copper and optical fiber cables. Technical committee T11, a committee within International Committee for Information Technology Standards (INCITS), is responsible for FC interface standards. The latest FC implementations of 16 Gigabit Fibre Channel (GFC) offers data transfer speeds up to 16 Gbps. The FC architecture is highly scalable, and theoretically a single FC SAN can accommodate approximately 15 million nodes.

Note: The term "Fibre" refers to the protocol, whereas the term "fiber" refers to the medium.



The key FC SAN components include network adapters, cables and connectors, and interconnecting devices.

Each node requires one or more network adapters to provide a physical interface for communicating with other nodes. Examples of network adapters are FC host bus adapters (HBAs), and storage system front-end adapters. An FC HBA has SCSI-to-FC processing capability. It encapsulates OS (or hypervisor) storage I/Os (usually SCSI I/O) into FC frames before sending the frames to FC storage systems over an FC SAN.

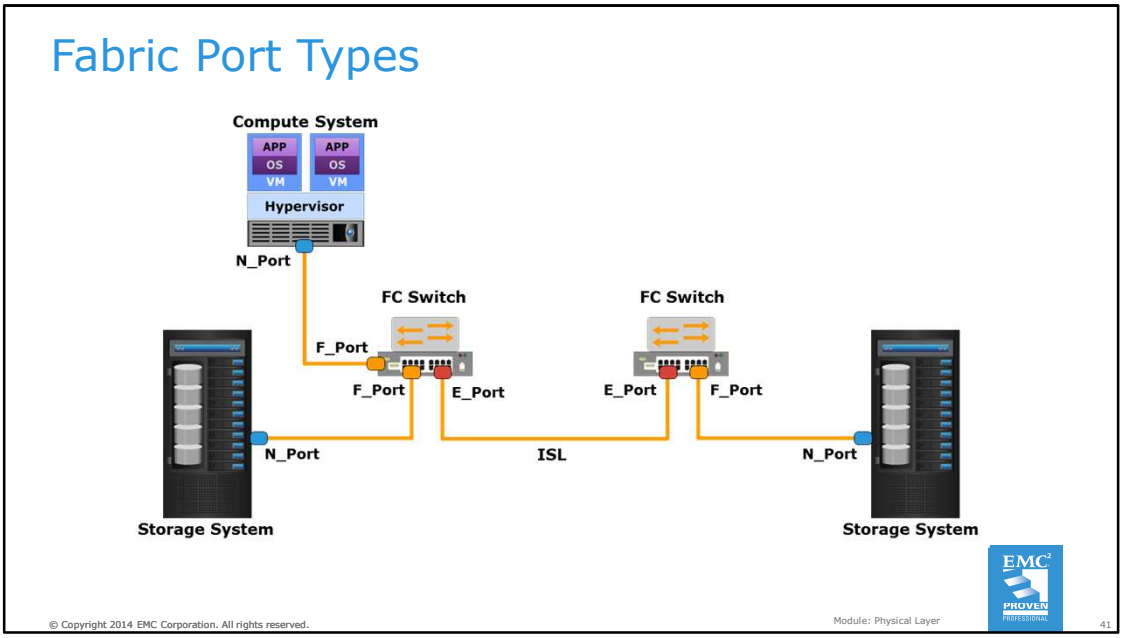
FC SAN predominantly uses optical fiber to provide physical connectivity between nodes. Copper cables might be used for shorter distances. A connector may attach at the end of a cable to enable swift connection and disconnection of the cable to and from a port.

FC switches and directors are the interconnecting devices commonly used in an FC SAN to forward data from one physical switch port to another. Directors are high-end switches with a higher port count and better fault-tolerance capabilities than smaller switches (also known as "departmental" switches). Switches are available with a fixed port count or with a modular design. In a modular switch, the port count is increased by installing additional port cards into empty slots. Modular switches enable online installation of port cards. The architecture of a director is usually modular, and its port count is increased by inserting line cards or blades to the director's chassis.

Fabric Connect and Addressing

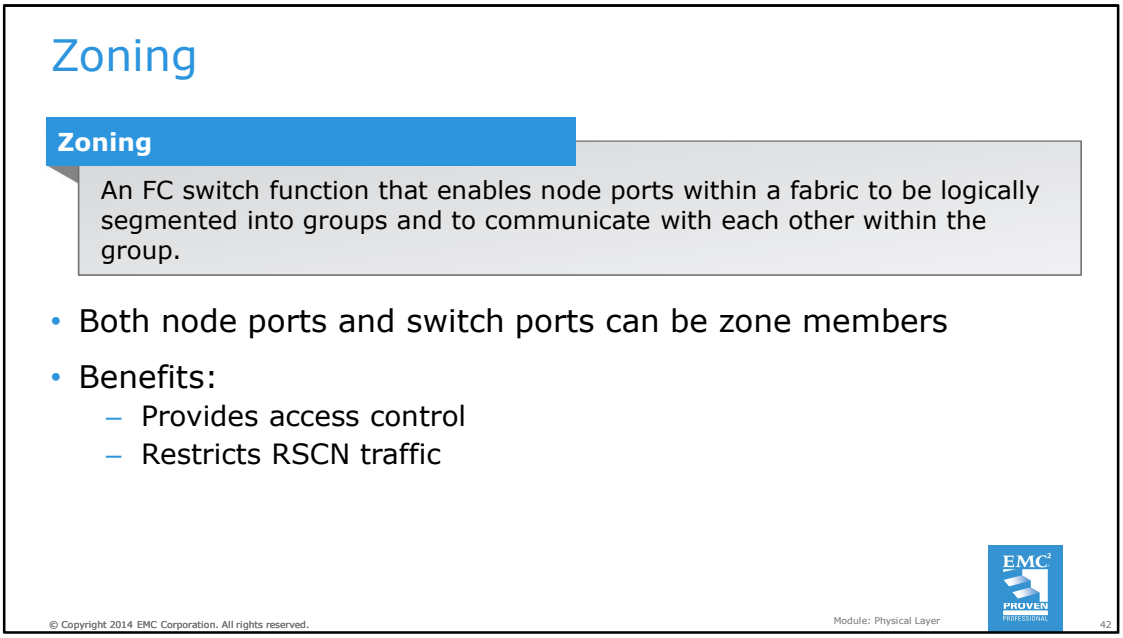
- A fabric created with FC switches connects all nodes and enables them to communicate
- Each switch in a fabric contains a unique domain identifier (ID)
- Each network adapter is physically identified by a 64-bit World Wide Node Name (WWNN)
- Each adapter port is physically identified by a 64-bit World Wide Port Name (WWPN)
- Each adapter port in a fabric has a unique 24-bit FC address
  - Fabric assigns FC addresses to adapter ports dynamically

A fabric is created with an FC switch (or an FC director) or a network of switches that enable all nodes to connect to each other and communicate. Each switch in a fabric contains a unique domain identifier (ID), which is part of the fabric's addressing scheme. Each network adapter and network adapter port in the FC environment has a globally unique 64-bit identifier called the World Wide Name (WWN). Unlike an FC address, which is assigned dynamically, a WWN is a static name. WWNs are burned into the hardware or assigned through software. An FC network adapter is physically identified by a World Wide Node Name (WWNN), and an FC adapter port is physically identified by a World Wide Port Name (WWPN). For example, a dual-port FC HBA has one WWNN and two WWPNs. Further, each FC adapter port in a fabric has a unique 24-bit FC address for communication.



A port in a switched fabric can be one of the following types:

- **N_Port** is an end-point in the fabric. This port is also known as the node port (or FC adapter port). Typically, it is a compute system port (on an FC HBA) or a storage system port connected to a switch in a fabric.
- **E_Port** is a switch port that forms a connection between two FC switches. This port is also known as an expansion port. The E_Port on an FC switch connects to the E_Port of another FC switch in the fabric through ISLs.
- **F_Port** is a port on a switch that connects an N_Port. It is also known as a fabric port.
- **G_Port** is a generic port on some vendors' switches. It can operate as an E_Port or an F_Port and determines its functionality automatically during initialization.



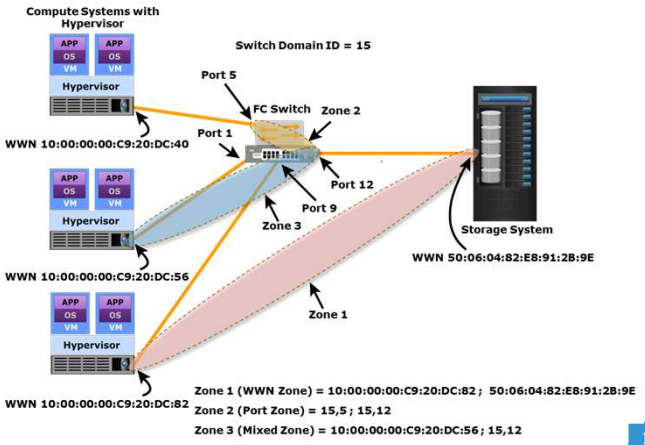
Zoning is an FC switch function that enables node ports within the fabric to be logically segmented into groups and to communicate with each other within the group. Whenever a change takes place in the fabric, the fabric sends a registered state change notification (RSCN) to the nodes in the fabric. If zoning is not configured, the RSCNs are received by all nodes in the fabric. This includes nodes that are not impacted by the change, resulting in increased fabric-management traffic. For a large fabric, the amount of FC traffic generated due to this process can be significant and might impact the compute-to-storage data traffic. Zoning helps to limit the number of RSCNs in a fabric. In the presence of zoning, a fabric sends the RSCNs to only those nodes in the zone where the change has occurred.

Both node ports and switch ports can be members of a zone. A port or node can be a member of multiple zones. Nodes distributed across multiple switches in a fabric may also be grouped into the same zone.

Single-initiator-single-target zoning is considered as an industry best practice to configure zones. In an FC SAN, the HBA ports and the storage system ports are called initiator ports and target ports respectively. A single-initiator-single-target zone consists of one initiator port and one target port. Single-initiator-single-target zoning eliminates unnecessary compute-to-compute interaction and minimizes RSCNs. Single-initiator-single-target zoning in a large fabric leads to configuring a large number of zones and more administrative actions. However, this practice improves the FC SAN performance and reduces the time to troubleshoot FC SAN-related problems.

Types of Zoning

- WWN zoning
- Port zoning
- Mixed zoning

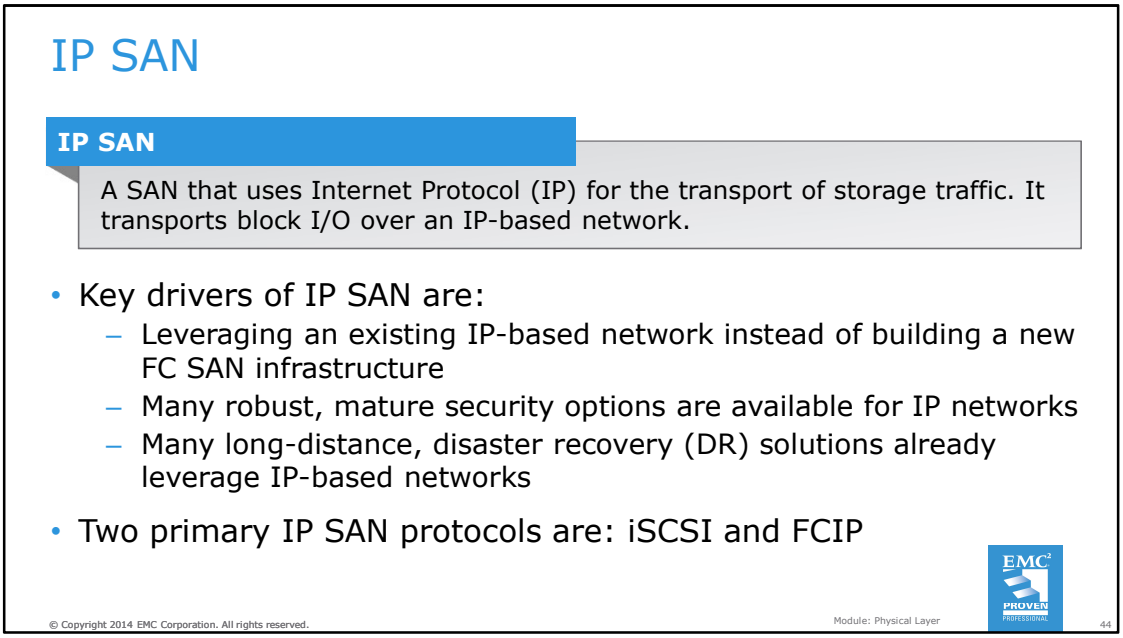


Zoning can be categorized into three types: WWN zoning, port zoning, and mixed zoning.

WWN zoning uses WWNs to define zones. The zone members are the unique WWPN addresses of the ports in HBA and its targets (storage systems). A major advantage of WWN zoning is its flexibility. WWN zoning allows nodes to be moved to another switch port in the fabric and to maintain connectivity to their zone partners without having to modify the zone configuration. This is possible because the WWN is static to the node port.

Port zoning uses the switch port identifier to define zones. In port zoning, access to data is determined by the physical switch port to which a node is connected. The zone members are the port identifier (switch domain ID and port number) to which an HBA and its targets are connected. If a node is moved to another switch port in the fabric, then zoning must be modified to allow the node, in its new port, to participate in its original zone. However, if an HBA or a storage system port fails, an administrator just has to replace the failed device without changing the zoning configuration.

Mixed zoning combines the qualities of both WWN zoning and port zoning. Using mixed zoning enables a specific node port to be tied to the WWN of a node.



IP SAN uses the Internet Protocol (IP) for the transport of storage traffic. It transports block I/O over an IP-based network. IP is a mature technology, and using IP SAN as a storage networking option provides several advantages. Cloud providers may have an existing IP-based network infrastructure, which could be used for storage networking. Leveraging an existing IP-based network therefore may be a more economical option than investing in building a new FC SAN infrastructure. In addition, many robust and mature security options are available for IP networks. Many long-distance, disaster recovery (DR) solutions already leverage IP-based networks. Therefore, with IP SAN, providers can extend the geographical reach of their storage infrastructure.

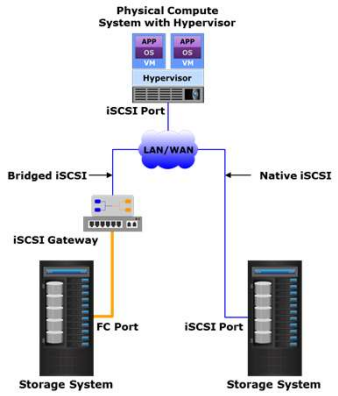
Two primary protocols that leverage IP as the transport mechanism for block-level data transmission are Internet SCSI (iSCSI) and Fibre Channel over IP (FCIP).

iSCSI Networking

iSCSI

iSCSI encapsulates SCSI commands and data into IP packets that are transported over an IP-based network.

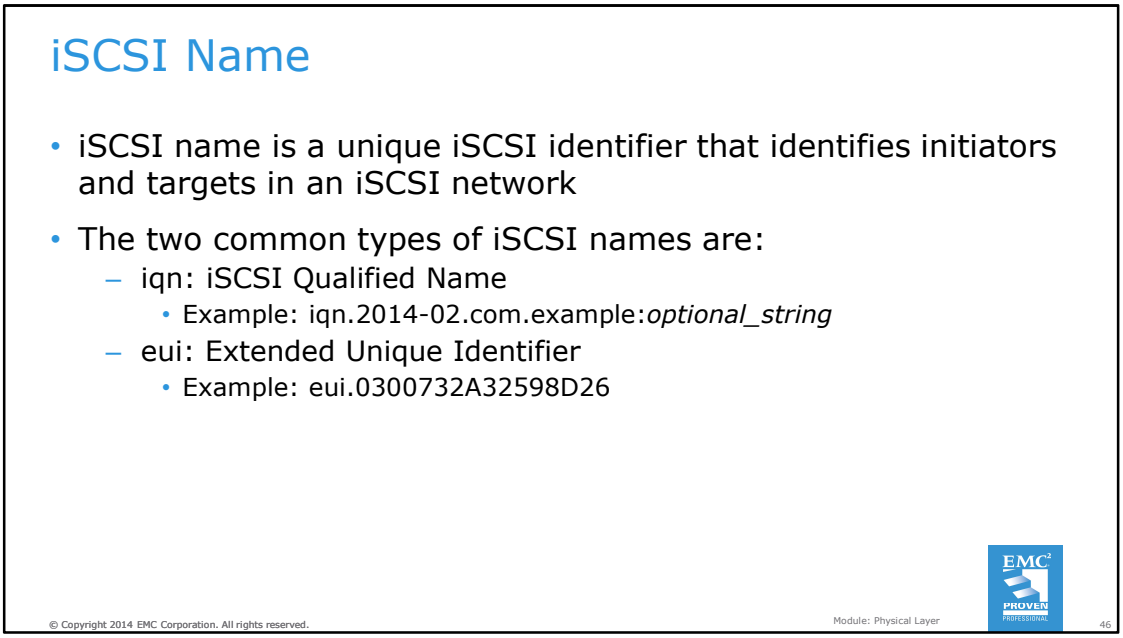
- iSCSI network components are:
  - iSCSI initiators
    - Example: iSCSI HBA
  - iSCSI targets
    - Example: storage system with iSCSI port (Native iSCSI)
    - Example: iSCSI gateway (Bridged iSCSI)
  - IP-based network



iSCSI encapsulates SCSI commands and data into IP packets. These IP packets are transported over an IP-based network. iSCSI network components include:

- iSCSI initiators such as a software iSCSI adapter and an iSCSI HBA
- iSCSI targets such as a storage system with iSCSI port or an iSCSI gateway
- IP-based network

An iSCSI initiator sends commands and associated data to a target and the target returns data and responses to the initiator. The software iSCSI adapter is an OS (or hypervisor) kernel-resident software that uses an existing NIC of the compute system to emulate an iSCSI initiator. An iSCSI HBA has a built-in iSCSI initiator and is capable of providing performance benefits over software iSCSI adapters by offloading the entire iSCSI and TCP/IP processing from the processor of the compute system. If an iSCSI-capable storage system is deployed, then an iSCSI initiator can directly communicate with the storage system over an IP-based network. This type of iSCSI implementation is called *native iSCSI*. Otherwise, in an iSCSI implementation that uses a storage system with only FC ports, an iSCSI gateway is used. This gateway device performs the translation of IP packets to FC frames and vice versa, thereby bridging the connectivity between the IP and the FC environments. This type of iSCSI implementation is called *bridged iSCSI*. The figure on the slide shows both native and bridged iSCSI implementations.



A unique worldwide iSCSI identifier, known as an iSCSI name, identifies the initiators and the targets within an iSCSI network to facilitate communication. The unique identifier can be a combination of the names of the department, application, or manufacturer, serial number, asset number, or any tag that can be used to recognize and manage the devices. The two types of iSCSI names that are commonly used are:

- **iSCSI Qualified Name (IQN):** An organization must own a registered domain name to generate iSCSI Qualified Names. This domain name does not need to be active or resolve to an address. It just needs to be reserved to prevent other organizations from using the same domain name to generate iSCSI names. A date is included in the name to avoid potential conflicts caused by the transfer of domain names. An example of an IQN is iqn.2014-02.com.example:*optional_string.* The *optional_string* provides a serial number, an asset number, or any other device identifiers. An iSCSI Qualified Name enables storage administrators to assign meaningful names to iSCSI devices, and therefore, manage those devices more easily.

- **Extended Unique Identifier (EUI):** An EUI is a globally unique identifier based on the IEEE EUI-64 naming standard. An EUI is composed of the "eui" prefix followed by a 16-character hexadecimal name, such as eui.0300732A32598D26.

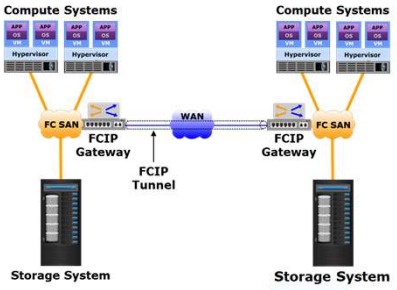
In either format, the allowed special characters are dots, dashes, and blank spaces.

FCIP Networking

FCIP

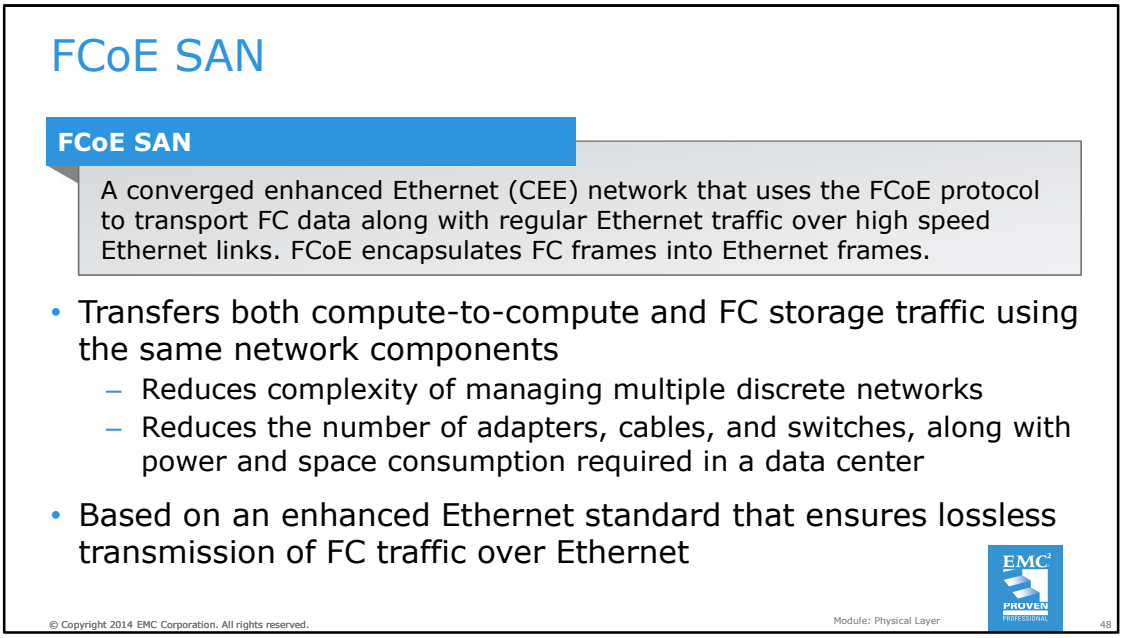
FCIP is an encapsulation of FC frames into IP packets that are transported between disparate FC SANs over an IP-based network through FCIP tunnel.

- An FCIP entity (e.g. FCIP gateway) exists at either end of an FCIP tunnel
  - Encapsulates FC into IP
  - Transfers IP packets to remote gateway
  - Decapsulates FC from IP
- Widely used in disaster recovery implementations



FCIP is an encapsulation of FC frames into IP packets. FCIP is a tunneling protocol that enables distributed FC SAN islands to be interconnected over the existing IP-based networks. This enables transporting FC data between disparate FC SANs that may be separated by a long distance. In an FCIP environment, an FCIP entity such as an FCIP gateway is deployed at either end of the tunnel between two FC SAN islands, as shown in the figure on the slide. An FCIP gateway encapsulates FC frames into IP packets and transfers them to the remote gateway through the tunnel. The remote FCIP gateway decapsulates the FC frames from the IP packets and sends the frames to the remote FC SAN. FCIP is extensively used in disaster recovery implementations in which data is replicated to storage located at a remote site.

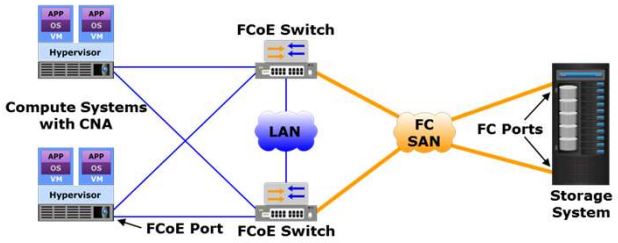
An FCIP implementation is capable of merging interconnected fabrics into a single fabric. In a merged fabric, the fabric service related traffic travels between interconnected FC SANs through the FCIP tunnel. However, only a small subset of nodes at either end of the FCIP tunnel requires connectivity across the tunnel. Thus, the majority of FCIP implementations today use some switch-specific features to prevent the fabrics from merging and also restrict the nodes that are allowed to communicate across the fabrics.



FCoE SAN is a converged enhanced Ethernet (CEE) network that is capable of transporting FC data along with regular Ethernet traffic over high speed (such as 4 Gbps, 8 Gbps, 10 Gbps, or higher) Ethernet links. It uses the FCoE protocol that encapsulates FC frames into Ethernet frames. FCoE is based on an enhanced Ethernet standard that supports Data Center Bridging (DCB) functionalities. DCB ensures lossless transmission of FC traffic over Ethernet.

FCoE SAN provides the flexibility to deploy the same network components for transferring both compute-to-compute traffic and FC storage traffic. This helps in reducing the complexity of managing multiple discrete network infrastructures. FCoE SAN uses multi-function network adapters and switches. Therefore, FCoE reduces the number of adapters, cables, and switches, along with power and space consumption required in a data center.

FCoE SAN Components: CNA and S/W FCoE Adapter



Component	Description
Converged network adapter (CNA)	• Provides functionality of both NIC and FC HBA in a single device • Encapsulates FC traffic onto Ethernet frames (FCoE traffic) • Consolidates both FC and regular Ethernet traffic over CEE links
Software FCoE adapter	• A software on the compute system performs FCoE processing • Supported NICs transfer both FCoE and regular Ethernet traffic

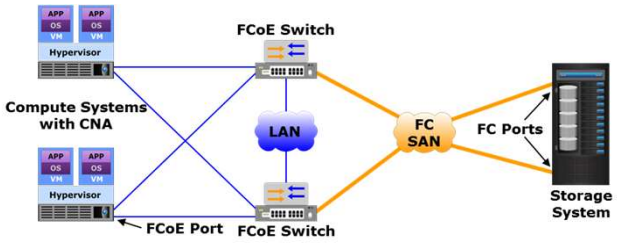
An FCoE SAN consists of converged network adapters (CNAs), FCoE switches, cables, and FCoE storage ports.

A CNA is a physical adapter that provides the functionality of both NIC and FC HBA in a single device. It consolidates both FC traffic and regular Ethernet traffic on a common Ethernet infrastructure. CNAs connect compute systems to FCoE switches. They are responsible for encapsulating FC traffic onto Ethernet frames and forwarding them to FCoE switches over CEE links.

Instead of CNA, a software FCoE adapter may also be used. A software FCoE adapter is software on the compute system that performs FCoE processing. FCoE processing consumes compute system processor cycles. With software FCoE adapters, the compute system implements FC protocol in software that handles SCSI to FC processing. The software FCoE adapter performs FC to Ethernet encapsulation. Both FCoE traffic (Ethernet traffic that carries FC data) and regular Ethernet traffic are transferred through supported NICs on the compute system.

The figure on the slide shows an FCoE implementation that consolidates both FC SAN traffic and LAN (Ethernet) traffic on a common Ethernet infrastructure.

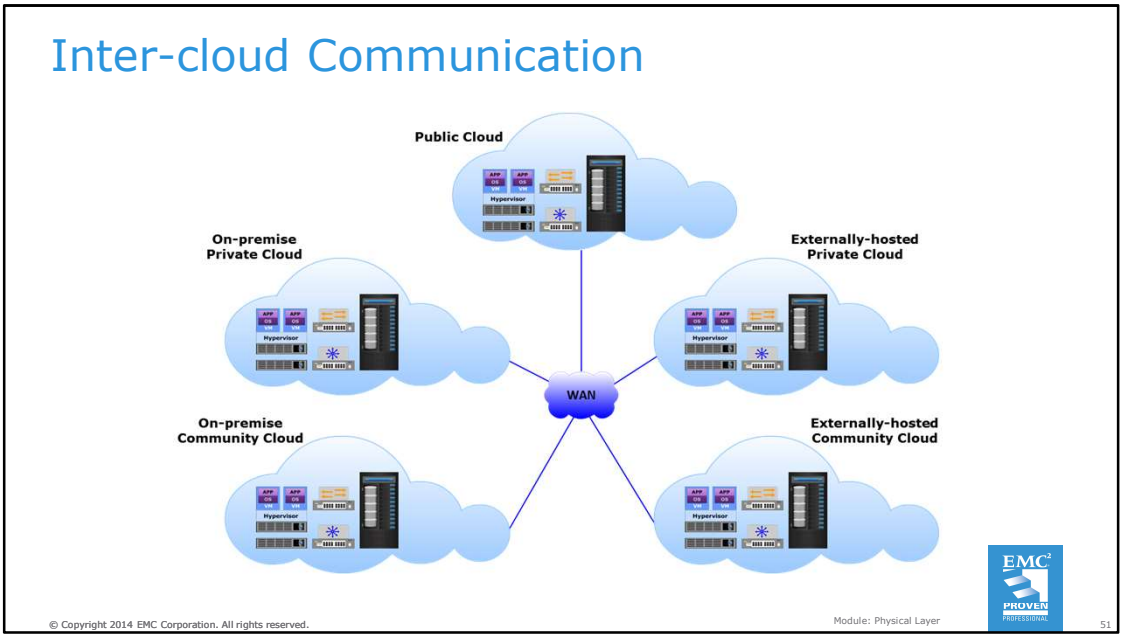
FCoE SAN Components: FCoE Switch and Storage Port



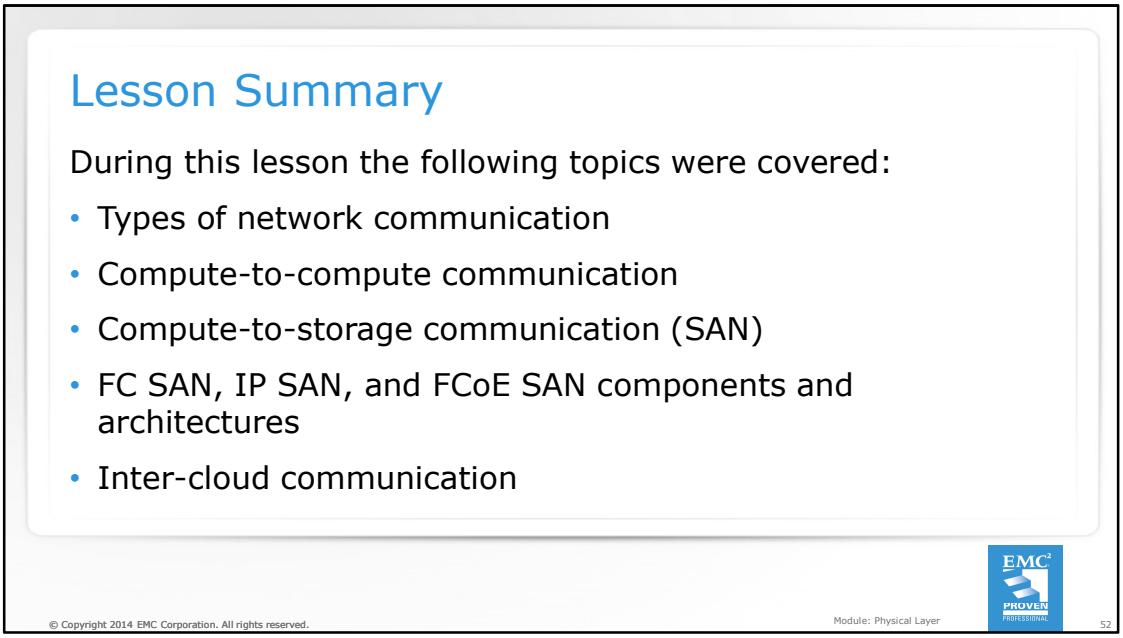
Component	Description
FCoE switch	• Contains Fibre Channel Forwarder (FCF), Ethernet Bridge, and a set of ports for FC, Ethernet, or FCoE connectivity • FCF encapsulates FC frames into Ethernet frames (FCoE frames) and decapsulates FCoE frames to FC frames
FCoE storage port	• Connects to FCoE switch, enabling end-to-end FCoE environment

An FCoE switch has the functionalities of both an Ethernet switch and an FC switch. It has a Fibre Channel Forwarder (FCF), an Ethernet Bridge, and a set of ports that can be used for FC, Ethernet or FCoE connectivity. The function of the FCF is to encapsulate the FC frames received from an existing FC SAN into the Ethernet frames, and also to decapsulate the Ethernet frames received from the Ethernet Bridge to the FC frames.

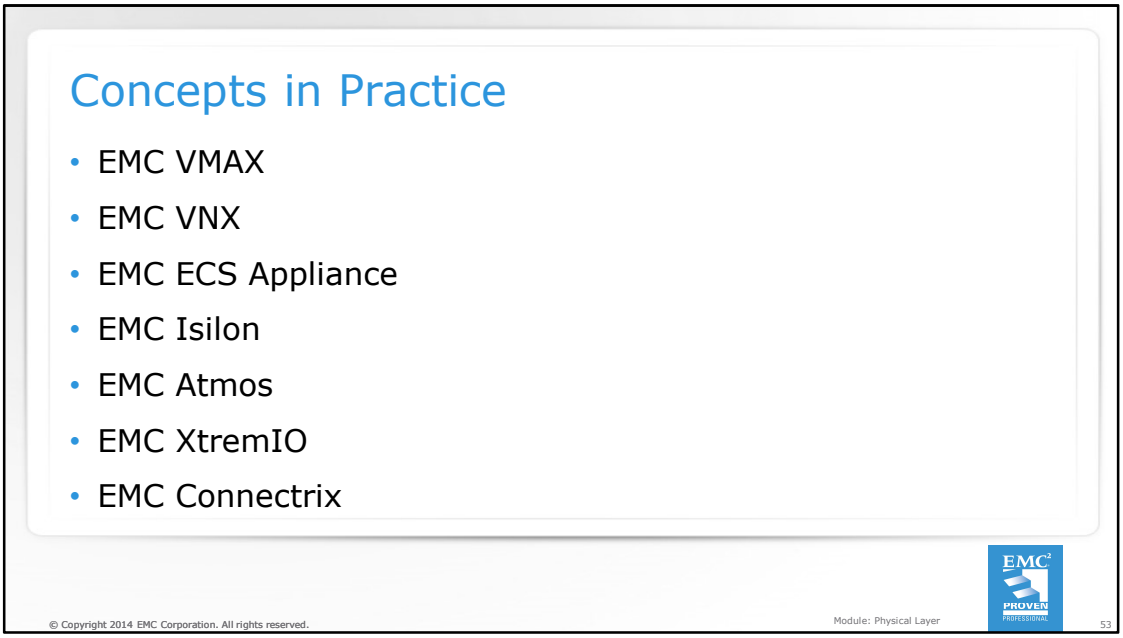
Some vendors offer FCoE ports in their storage systems. These storage systems connect directly to FCoE switches. The FCoE switches form FCoE fabrics between compute and storage systems and provide end-to-end FCoE support. The figure on the slide shows an FCoE implementation with an FCoE-capable storage system.



The cloud tenets of rapid elasticity, resource pooling, and broad network create a sense of availability of limitless resources in a cloud infrastructure that can be accessed from any location over a network. However a single cloud does not have an infinite number of resources. A cloud that does not have adequate resources to satisfy service requests from clients, may be able to fulfill the requests if it is able to access the resources from another cloud. For example, in a hybrid cloud scenario, a private cloud may access resources from a public cloud during peak workload periods. There may be several combinations of inter-cloud connectivity as depicted in the figure on the slide. Inter-cloud connectivity enables clouds to balance workloads by accessing and using computing resources, such as processing power and storage resources from other cloud infrastructures. The cloud provider has to ensure network connectivity of the cloud infrastructure over a WAN to the other clouds for resource access and workload distribution.



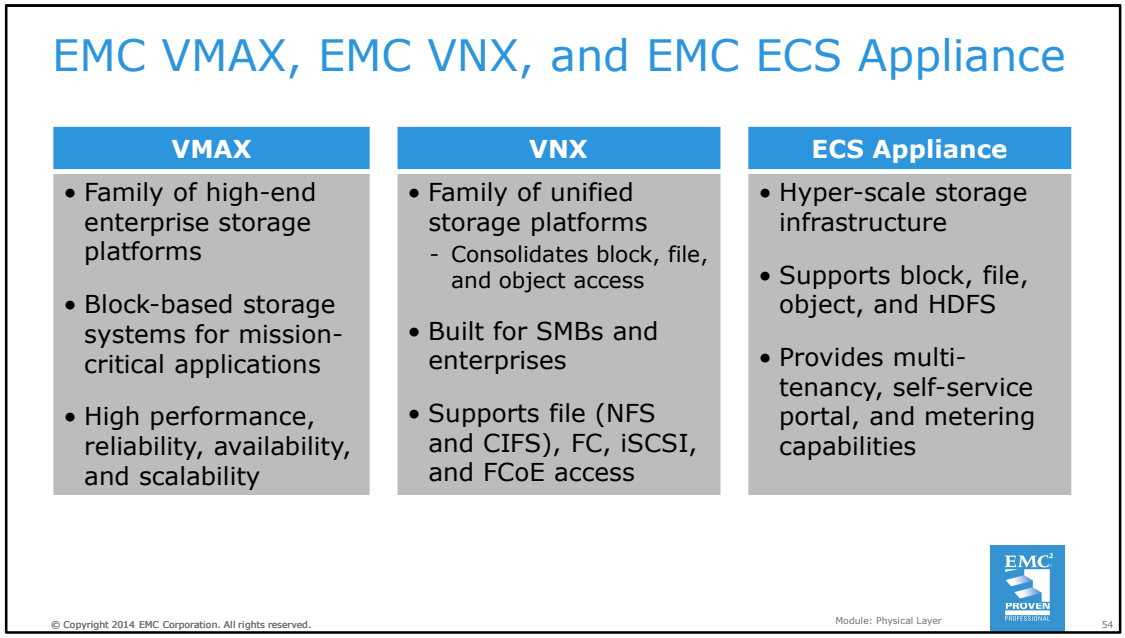
This lesson covered the types of network communication, compute-to-compute communication, and compute-to-storage communication over a storage area network (SAN). This lesson also covered the classification of SAN—FC SAN, IP SAN, and FCoE SAN—and described the components and architecture of each. Finally, this lesson covered inter-cloud communication.



The Concepts in Practice section covers EMC VMAX, EMC VNX, EMC ECS Appliance, EMC Isilon, EMC Atmos, EMC XtremIO, and EMC Connectrix.

Note:

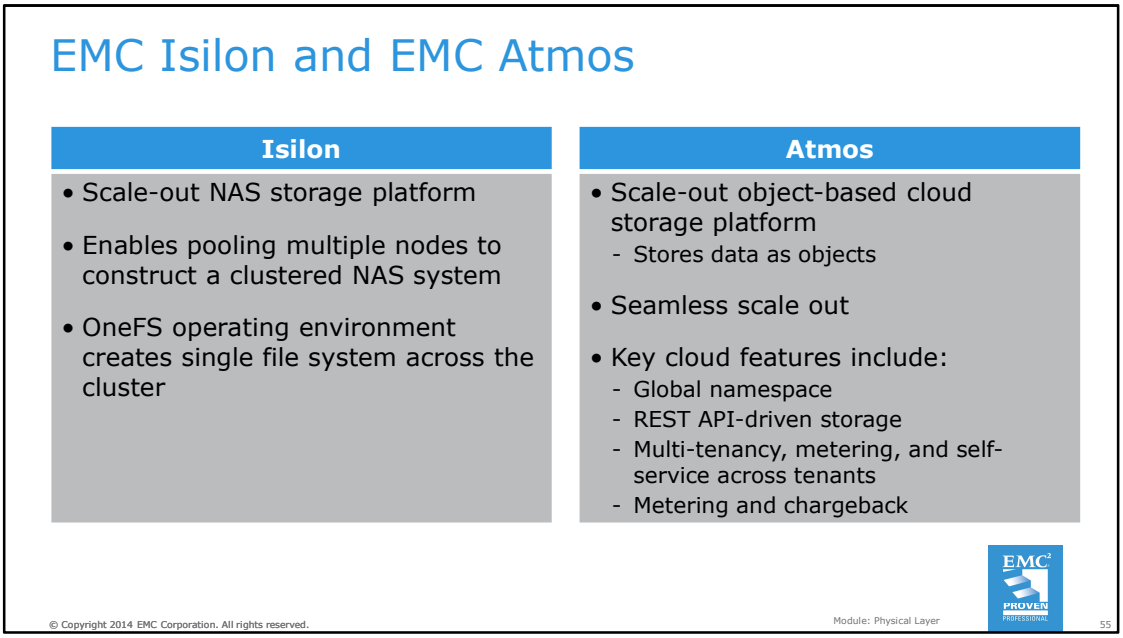
For the latest information on EMC products, visit www.emc.com.



The **EMC VMAX** family of storage arrays is a full line of high-end enterprise storage platforms from EMC. The VMAX series provides block-based storage that delivers enterprise storage with scalability, performance, and availability to meet the business requirements. The VMAX series is an innovative platform built around a scalable Virtual Matrix architecture to support the future storage growth demands of virtual data centers and cloud environments. It also supports multiple protocols for host connectivity. VMAX storage systems provide business continuity solution by supporting various local and remote replications.

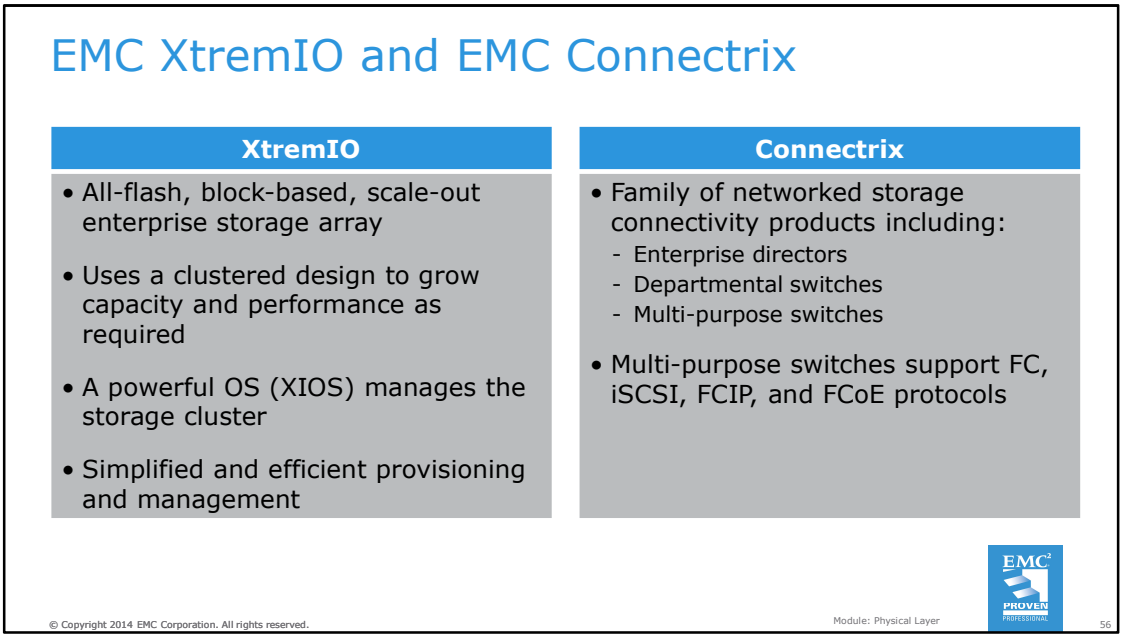
The **EMC VNX** family is a group of products that provide a unified storage platform that consolidates block, file, and object access into one solution. The VNX series is built for small to medium-sized businesses and enterprises. It enables organizations to dynamically grow, share, and manage multi-protocol file systems and multi-protocol block storage access. The VNX operating environment enables Windows and UNIX/Linux users to share files using NFS and CIFS. It also supports FC, iSCSI, and FCoE access.

EMC ECS Appliance is a hyper-scale storage infrastructure that provides universal protocol support in a single, highly-available platform for block, file, object, and Hadoop Distributed File System (HDFS) storage. ECS Appliance enables cloud providers to deliver competitive cloud storage services at scale. It provides geo-efficient protection, multi-tenancy, self-service portal, and detailed metering capabilities and enables scaling to Exabyte levels. ECS provides a single platform for all web, mobile, Big Data, and social media applications. There are two types of data services within ECS: Block Data Services and Unstructured Data Services that support unstructured, block, and mixed use cases.



EMC Isilon is a scale-out NAS storage product family powered by the OneFS operating environment. Isilon enables pooling multiple nodes together to construct a clustered NAS system. OneFS is the operating environment that creates a single file system that spans across all nodes in an Isilon cluster. EMC Isilon provides the capability to manage and store large (petabyte-scale), high-growth data in a single system with the flexibility to meet a broad range of performance requirements.

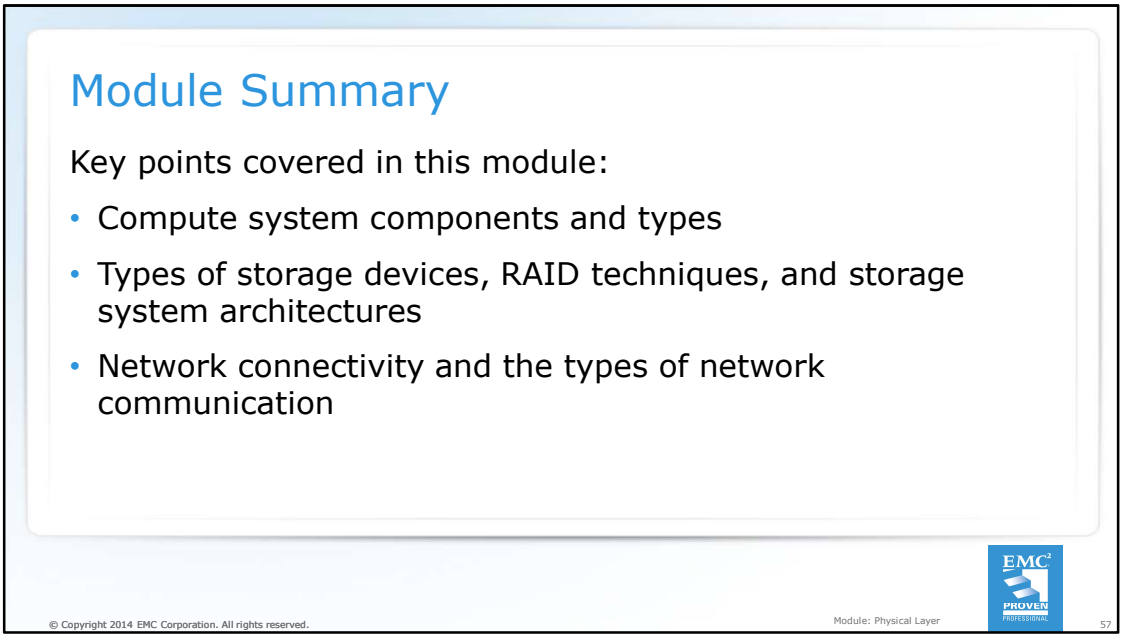
EMC Atmos is a cloud storage platform for enterprises and service providers to deploy public, private, or hybrid cloud storage. It enables to store, manage, and protect globally distributed, unstructured content at scale. Atmos is a scale-out object architecture that stores data as objects with the associated metadata. It enables storage to be scaled out without the need to rewrite applications. Some of the key cloud features of Atmos include a global namespace, REST API-driven storage, multi-tenancy, self-service, and metering and chargeback.



EMC XtremIO is an all-flash, block-based, scale-out enterprise storage array that provides substantial improvements to I/O performance. It is purpose-built to leverage flash media and delivers new levels of real-world performance, administrative ease, and advanced data services for applications. It uses a scale-out clustered design that grows capacity and performance linearly to meet any requirement. XtremIO arrays are created from building blocks called "X-Bricks" that are each a high-availability, high-performance, fully active/active storage system with no single point of failure. XtremIO's powerful operating system, XIOS, manages the XtremIO storage cluster. XIOS ensures that the system remains balanced and always delivers the highest levels of performance with no administrator intervention. XtremIO helps the administrators to become more efficient by enabling system configuration in a few clicks, provisioning storage in seconds, and monitoring the environment with real-time metrics.

The **EMC Connectrix** family is a group of networked storage connectivity products. EMC offers the following connectivity products under the Connectrix brand:

- Enterprise directors: Ideal for large enterprise connectivity. Offer high port density and high component redundancy. Deployed in high-availability or large-scale environments
- Departmental switches: Designed to meet workgroup-level, department-level, and enterprise-level requirements. Provide high availability through features such as non-disruptive software and port upgrade, and redundant and hot-swappable components
- Multi-purpose switches: Support various protocols such as FC, iSCSI, FCIP, FCoE, and FICON. Include FCoE switches, FCIP gateways, and iSCSI gateways. Multiprotocol capabilities offer many benefits, including long-distance SAN extension, greater resource sharing, and simplified management.



This module covered the key components of a compute system and the common types of physical compute systems: tower, rack-mounted, and blade. This module also covered the common types of persistent storage devices, the different RAID techniques (striping, mirroring, and parity), and the types of storage system architectures: block-based, file-based, object-based, and unified storage systems. Finally, this module covered compute-to-compute communication, compute-to-storage communication (SAN), and SAN classification: FC SAN, IP SAN, and FCoE SAN. Finally, this module covered inter-cloud communication.