Amirkabir University of Technology
(Tehran Polytechnic)

Cloud Computing

Virtualization
(OS-level virtualization, Docker, Etc.)

Seyyed Ahmad Javadi

sajavadi@aut.ac.ir

Spring 2020

Based on slide deck of Dr. Abrishami (Ferdowsi Uni. of Mashhad)

Main source: Mastering Cloud Computing By Rajkumar Buyya.

Part 1

Execution Virtualization

Operating System-level Virtualization

- It offers the opportunity to create different and separated execution environments for applications that are managed concurrently.

- Differently from hardware virtualization***, there is no virtual machine manager or hypervisor***, and the virtualization is done within a single operating system, where the OS kernel allows for multiple isolated user space instances.

Operating System-level Virtualization (Cont.)

- The kernel is also responsible ***for sharing the system resources among instances*** and for ***limiting the impact of instances on each other***.

- A user space instance in general contains a proper view of the file system, which is ***completely isolated***, and ***separate IP addresses***, software configurations, and access to devices.

Operating System-level Virtualization (Cont.)

- This virtualization technique can be considered an evolution of the ***chroot mechanism in Unix systems***.
- The chroot operation changes the file system root directory for a process and its children to a specific directory.
- As a result, the process and its children ***cannot have access to other portions of the file system than those accessible under the new root directory***.

Operating System-level Virtualization (Cont.)

- Because Unix systems also ***expose devices as parts of the file system***, by using this method it is possible to completely ***isolate a set of processes.***

- Following the same principle, operating system-level virtualization aims to ***provide separated and multiple execution containers*** for running applications.

Operating System-level Virtualization (Cont.)

- This technique is an efficient solution for server consolidation scenarios in which multiple application servers share the same technology:
  - Operating system
  - Application server framework
  - Other components.

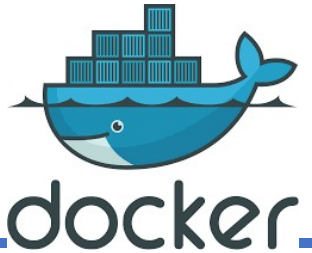
Operating System-level Virtualization (Cont.)

- When different servers are aggregated into one physical server, each server is run in a different user space, completely isolated from the others.
- Examples of operating system-level virtualizations are:
  - FreeBSD Jails
  - IBM Logical Partition (LPAR)
  - SolarisZones
  - Containers and Docker.

Containers

- OS-level virtualization also called containerization.
- A ***container is an isolated virtual environment*** which can run an application.
- Several containers can be created on each operating system, to each of which a ***subset of the computer's resources*** is allocated.
- Programs running inside a container **can only see the container's contents** and devices assigned to the container.

Docker

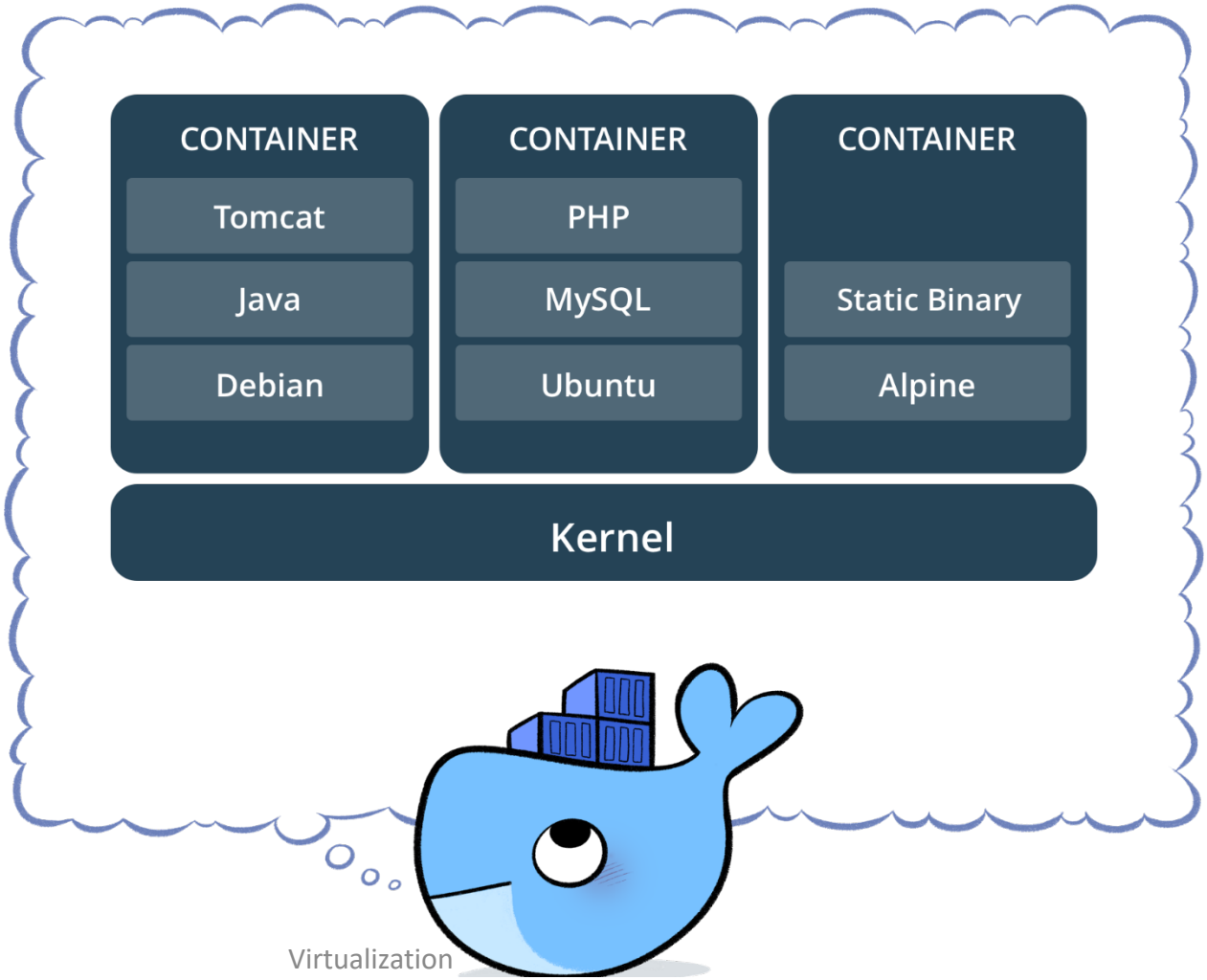


- Docker is the company driving the container movement .

- A container image is
  - A lightweight, stand-alone, executable package of a piece of software
  - It includes everything needed to run it: code, runtime, system tools, system libraries, settings.

- Available for both Linux and Windows based apps, containerized software ***will always run the same, regardless of the environment***.

Docker


CONTAINER
Tomcat
Java
Debian
CONTAINER
PHP
MySQL
Ubuntu
CONTAINER
Static Binary
Alpine
Kernel

Containers vs. Virtual Machines

- Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers.

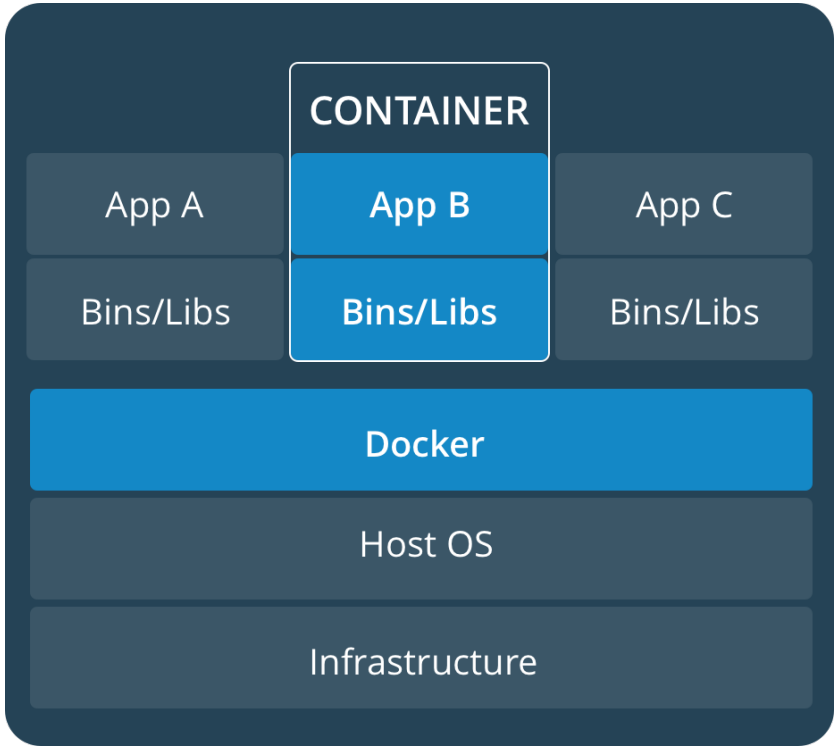
- A VM includes a full copy of an operating system, one or more apps, necessary binaries & libraries-taking up tens of GBs.

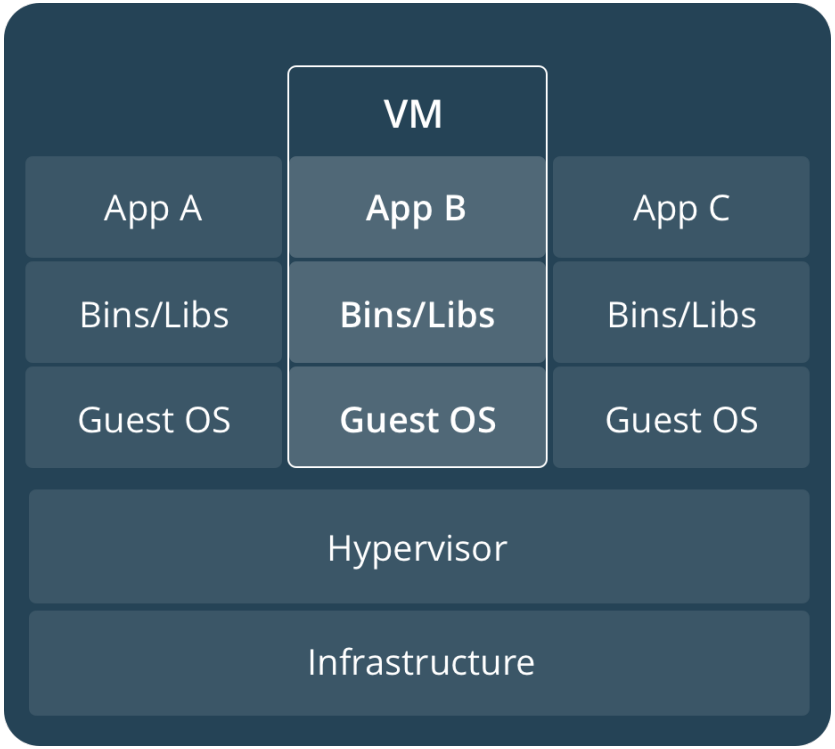
- VMs can also be ***slow to boot***.

Containers vs. Virtual Machines (Cont.)

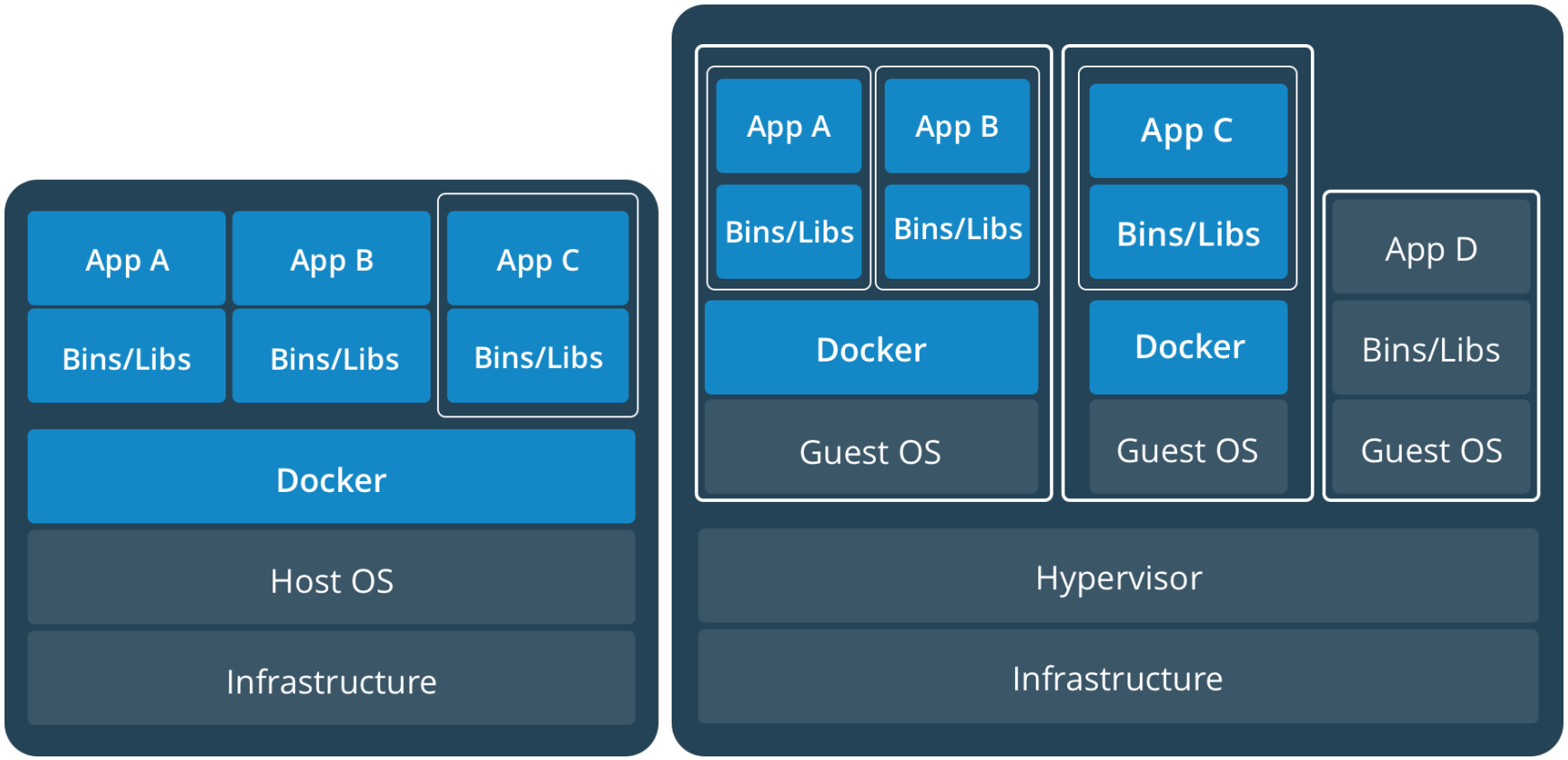
- Containers are an abstraction at the app layer that packages code and dependencies together.
- Multiple containers can run on the same machine **and share the OS kernel with other containers**, each running as isolated processes in user space.
- Containers take up less space than VMs
  - Container images are typically tens of MBs in size
  - ***Start almost instantly***.

Containers vs. Virtual Machines (Cont.)



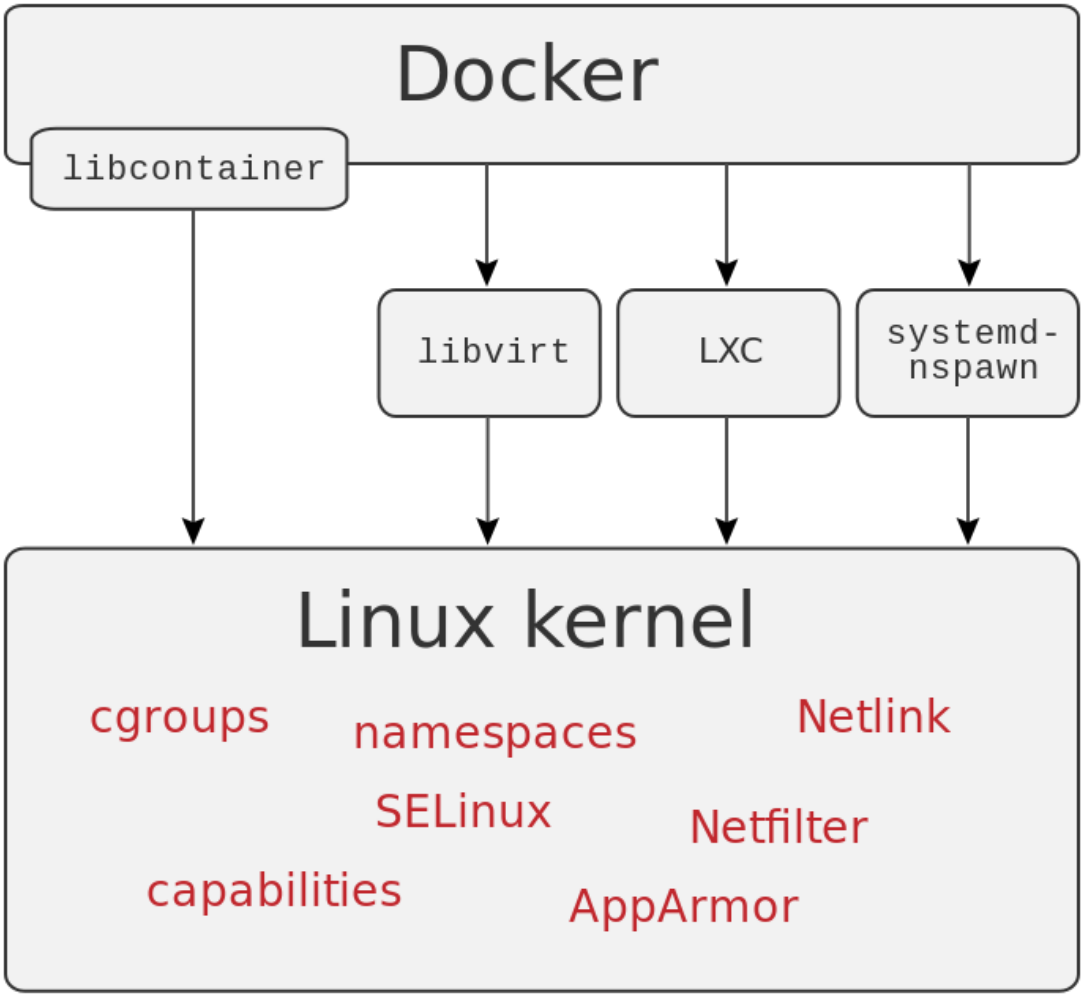


Containers vs. Virtual Machines (Cont.)



OS-level Virtualization

Docker Technology



Docker Technology

- Docker is developed **primarily for Linux**, where it uses the resource isolation features of the Linux kernel such as **cgroups and kernel namespaces**, and a **union-capable file system such as OverlayFS** and others.

Docker Technology- cgroups

- **cgroups** (abbreviated from **control groups**) is a Linux kernel feature that limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes.

Docker Technology- Namespaces

- **Namespaces** are a feature of the Linux kernel that partitions kernel resources such that one set of processes sees one set of resources while another set of processes sees a different set of resources.

Part 2

Programming Language-level Virtualization

- Programming language-level virtualization is mostly used to achieve **ease of deployment of applications**, **managed execution**, **and portability across different platforms** and operating systems.

- It consists of a virtual machine **executing the byte code of a program**, which is the result of the **compilation process**.

- Compilers implemented and used this technology to produce a binary format **representing the machine code for an abstract architecture**.

Execution Virtualization

Programming Language-level Virtualization (Cont.)

- Generally these virtual machines constitute **a simplification of the underlying hardware instruction set** and provide some high-level instructions that map some of the features of the languages compiled for them.

- At runtime**, the byte code can be either interpreted or compiled on the fly against the underlying hardware instruction set.**

Programming Language-level Virtualization (Cont.)

- Virtual machine programming languages become popular with Sun's introduction of the **Java platform** in 1996.
- The **Java virtual machine** was originally designed for the execution of programs written in the Java language, but other languages such as Python, Pascal, Groovy, and Ruby were made available.
- The ability to support multiple programming languages has been one of the key elements of the **Common Language Infrastructure (CLI)** which is the specification behind .NET Framework.

Programming Language-level Virtualization (Cont.)

- Both Java and the CLI are stack-based virtual machines
  - The reference model of the abstract architecture is based on an execution stack that is used to perform operations.
  - The byte code generated by compilers for these architectures contains a set of instructions that **load operands on the stack, perform some operations with them, and put the result on the stack.**

Additional review: https://andreabergia.com/stack-based-virtual-machines/

- Stack-based virtual machines possess the property of being easily interpreted and executed simply by **lexical analysis** and hence are **easily portable over different architectures**.

Programming Language-level Virtualization (Cont.)

- The main advantage of programming-level virtual machines, also called process virtual machines, **is the ability to provide a uniform execution environment across different platforms**.

- Programs compiled into byte code can be executed on any operating system and platform for which a virtual machine able to execute that code has been provided.

- The implementation of the virtual machine for different platforms is still a costly task, **but it is done once and not for any application.**

Programming Language-level Virtualization (Cont.)

- Moreover, process virtual machines allow for more control over the execution of programs **since they do not provide direct access to the memory.**

- Security is another advantage of managed programming languages; by filtering the I/O operations, the process virtual machine can easily support sandboxing of applications.

- All these advantages come with a price: **performance**
  - Virtual machine programming languages generally expose an inferior performance compared to languages compiled against the real architecture.

Application-level Virtualization

- Application-level virtualization is a technique allowing applications to be run in runtime environments that do not natively support all the features required by such applications.
- In this scenario, applications are not installed in the expected runtime environment but are run as though they were.
- In general, these techniques are mostly concerned with partial file systems, libraries, and operating system component emulation.

Application-level Virtualization

- Such emulation is performed by a thin layer - a program or an operating system component - that is in charge of executing the application.
- Emulation can also be used to execute program binaries compiled for different hardware architectures.
- One of the most popular solutions implementing application virtualization is Wine, which is a software application allowing Unix-like operating systems to execute programs written for the Microsoft Windows platform.

Application-level Virtualization

- Wine features a software application acting as a container for the guest application and a set of libraries, called Winelib, that developers can use to compile applications to be ported on Unix system.

- A similar solution for the Mac OS X environment is CrossOver, which allows running Windows applications directly on the Mac OS X operating system.

Part 3

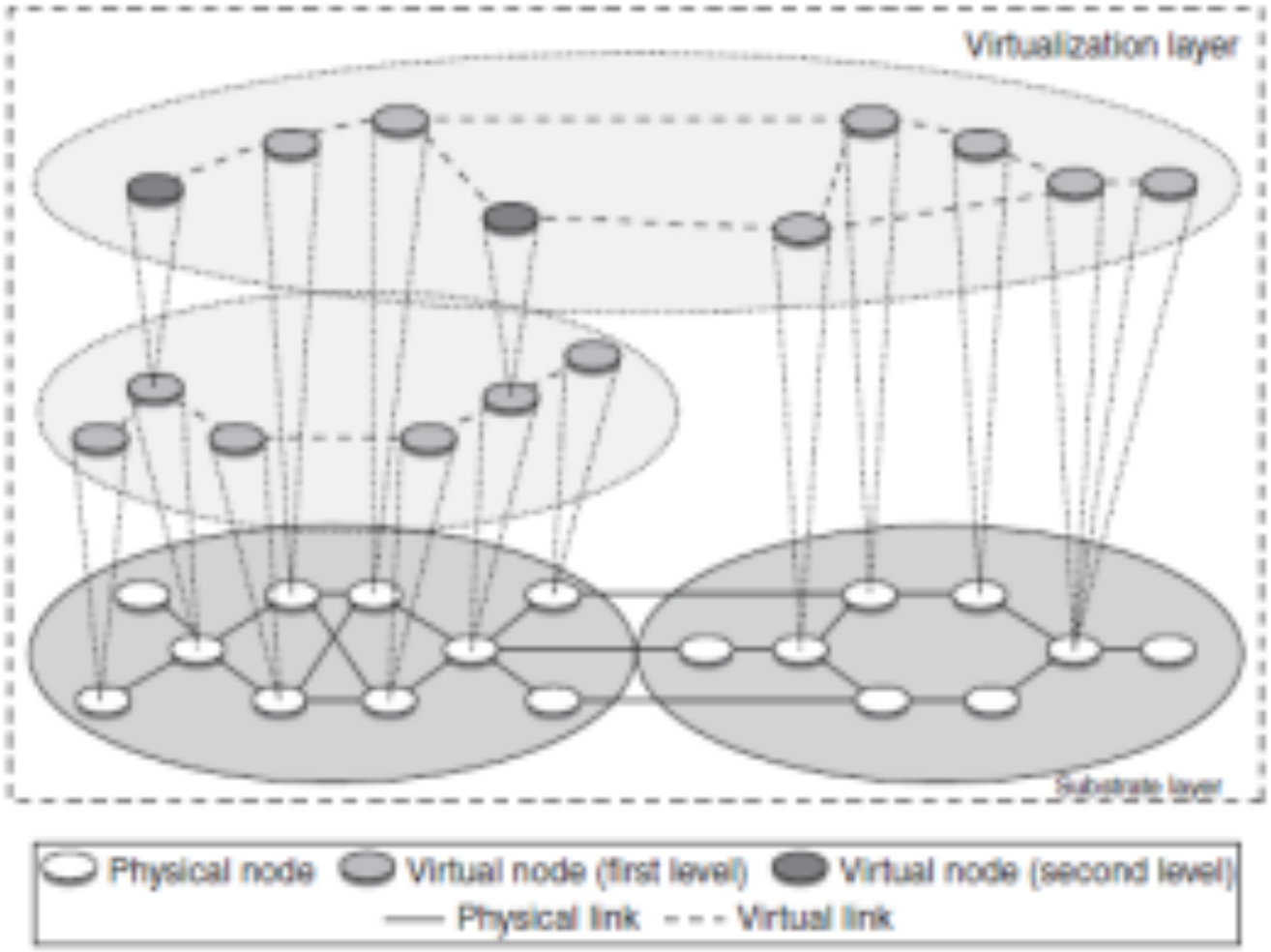
Network Virtualization

- Network virtualization is to segment the physical network resources in cloud data centers into smaller segmentations and lease it to cloud tenants, like leasing VMs in clouds enabled by host virtualization.

- In a virtualized network, multiple virtual networks (VNs) which may consist of virtual routers, virtual switches as well as virtual links that interconnect them, share a physical network and run isolated protocol stacks.

- Network virtualization allows users to design and deploy a network without requiring knowledge of the physical network.

Network Virtualization (Cont.)



Network Virtualization (Cont.)

- In the substrate layer, physical nodes and links from different network administrative domains serve as a substrate for the deployment of VNs.

- Physical nodes, at the core of the physical networks, represent network devices (e.g., switches and routers) that internally run virtual (or logical) routers instantiated to serve VNs' routing necessities.

- In the virtualization layer, virtual nodes and links are created on the top of the substrate and combined to build VNs.

Storage Virtualization

- Storage virtualization is a system administration practice that allows decoupling the physical organization of the hardware from its logical representation.

- Using this technique, users do not have to be worried about the specific location of their data, which can be identified using a logical path.

Storage Virtualization

- Storage virtualization consists of grouping multiple (possibly heterogeneous) storage devices that are seen as a single virtual storage space.

- There are two main abstractions to represent storage virtualization in clouds: virtual volumes and virtual data objects.

- The virtualization of storage devices as virtual volumes is important in the cloud computing context because it simplifies the task of assigning disks to VMs.

Storage Virtualization (Cont.)

- Cloud storage of virtual data objects enables scalable and redundant creation and retrieval of data objects directly into/from the cloud.

- With storage virtualization, cloud users do not need to know exactly where their data are stored.

- The details of which disks and partitions contain which objects or volumes are transparent to users, which also facilitates storage management for cloud providers.