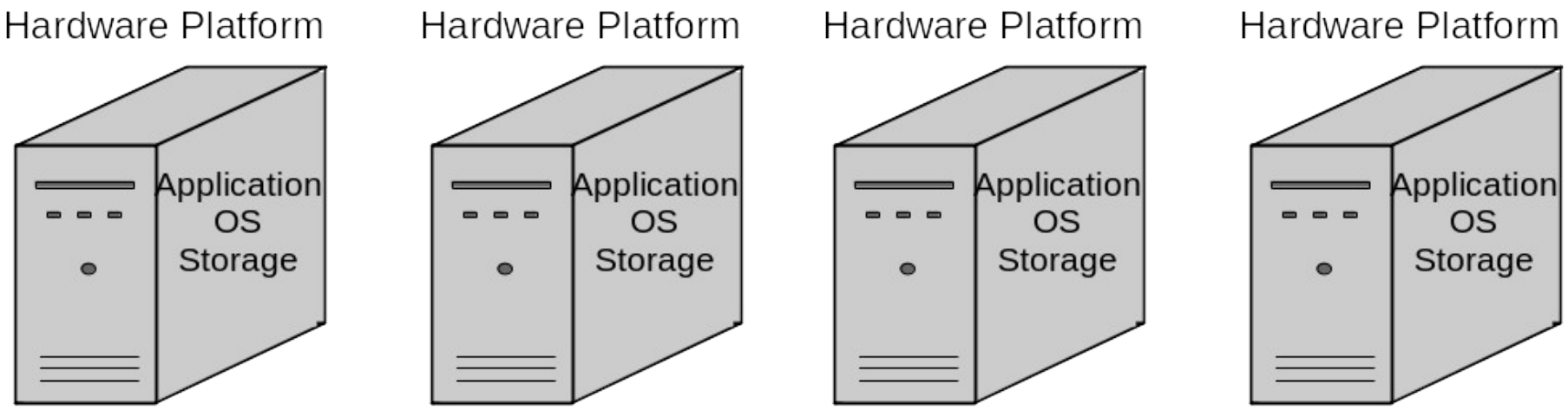
Virtualization

Open Data Management & the Cloud

(Data Science & Scientific Computing / UniTS – DMG)

Traditional service delivery



Web server
Windows
IIS

App server
Linux
Glassfish

E-mail
Windows
Exchange

Database
Linux
MariaDB

Problems

★ Multiple servers are deployed to serve different operational use cases

★ Increase the operational costs

★ Servers capacity is not fully exploited

⇒ Inefficiency and too high operating costs

Virtualization is abstraction

★ ability to simulate a hardware platform, such as a server, storage device or network resource. All of the functionality is separated (abstracted) from the hardware and simulated as a "virtual instance" with the ability to operate just like the hardware solution. **(Hardware or Server Virtualization)**

★ Ability to create an abstraction mechanism so that a logical address can be mapped to a physical resource. Example: load balancing. This kind of abstraction enables the key benefit of cloud computing: shared, ubiquitous access. **(Cloud computing)**

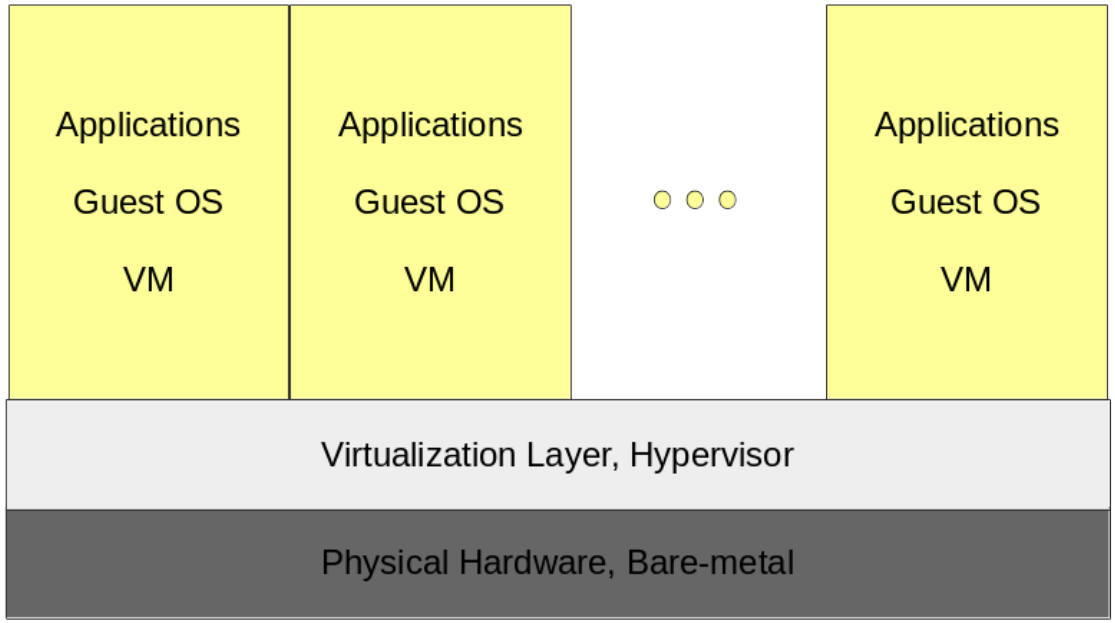
What is Hardware (or Server) Virtualization?

Hardware Virtualization, in computing, is the ability to simulate a hardware platform, such as a server, storage device or network resource. All of the functionality is *separated* (abstracted) from the hardware and simulated as a "virtual instance" with the ability to operate just like the hardware solution. A single hardware platform can be used to support multiple virtual devices or machines, which are easy to spin up or down as needed.

Virtual Machine is the software simulation of a computer. It is able to run an Operating Systems and applications interacting with the virtualized abstracted resources, not with the physical resources, of the actual host computer.

Hypervisor is a software tool installed on the physical host system to provide the thin software layer of abstraction that decouples the OS from the physical bare-metal. It allows to split a computer in different separate environment, the Virtual Machines, distributing them the computer resources

Virtual Server Architecture



Guests

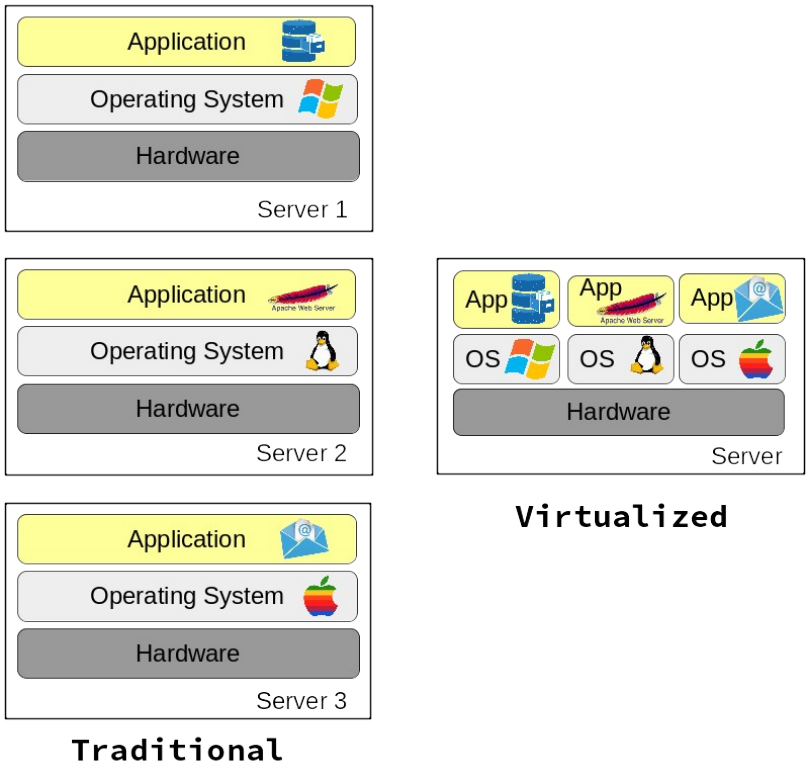
Host

Host machine is the actual machine on which the virtualization takes place

Guest machine is the virtual machine.

Hypervisor or **VM Manager** is the software or firmware that creates a virtualization layer on the host hardware

Traditional vs. Virtualized Server Architecture



Traditional architecture: add a server every time a new service/application is needed.

Virtualized architecture: different Operating Systems supporting different applications coexist on the same hardware

Virtual Server Architecture details

Virtual Machine is a virtual computing system.
It has tightly isolated software container with an operating system and applications inside.

Each Virtual Machine in a host is independent.

In a single physical server can be put multiple VMs enabling the run of multiple OSes and applications.

Virtual Machines features

⇒ Partitioning
- ➢ Run different OSes on the same physical host
- ➢ Partition physical resources between VMs

⇒ Isolation
- ➢ Fault and security isolation at the hardware level
- ➢ Preserve performance with advanced resource control

⇒ Encapsulation
- ➢ Save the VM state to files
- ➢ VMs can be moved and copied moving and copying files

⇒ Hardware Independence
- ➢ VMs can be copied, moved or migrated to different physical servers

Hypervisors

Hardware Virtualization Types

⇒ **Full Virtualization**: the hypervisor provides complete hardware abstraction creating simulated hardware devices. The guest OS don't know (or care) about the presence of a hypervisor and issue commands to what it thinks is actual hardware.

⇒ **Paravirtualization**: para means partial. The guest OS is aware that it is a guest, it recognizes the presence of an hypervisor and it has drivers to issue some commands, mainly I/O operations, directly to the host OS, more efficiently than inside a virtual environment. The guest OS must be modified

⇒ **Hardware assisted virtualization**: is a type of full virtualization where the microprocessor architecture has special instructions to aid the virtualization of the hardware. These hardware extensions help the hypervisor tackle complex tasks at the processor level rather than through software emulation

VirtualBox example

VirtualBox Demo

Desktop Virtualization

Desktop virtualization or **client virtualization**, is a virtualization technology abstracting (isolating) the computer desktop (client) environment from the physical computer.

Desktop virtualization is a type of **client-server computing** because the virtualized desktop is hosted on server and streamed to end user via different means.
The virtualized desktop is **served to the user on the network**, so the user can login remotely and access his desktop from any location. The user interacts with the virtual desktop in the same way he uses the physical desktop.

VDI (Virtual Desktop Infrastructure) is a method of desktop virtualization. VDI hosts the desktop environment in a virtual machine (VM) that runs on a centralized or remote server. In this desktop virtualization method it is a virtual machine per user. There are other methods where it is a session per user.

Remote Desktop example

- **Guacamole**

  web application that supports graphical access via remote desktop protocols (RDPs) directly in the browser

Network Virtualization

Is the ability to create logical, virtual networks that are decoupled from the underlying network hardware.

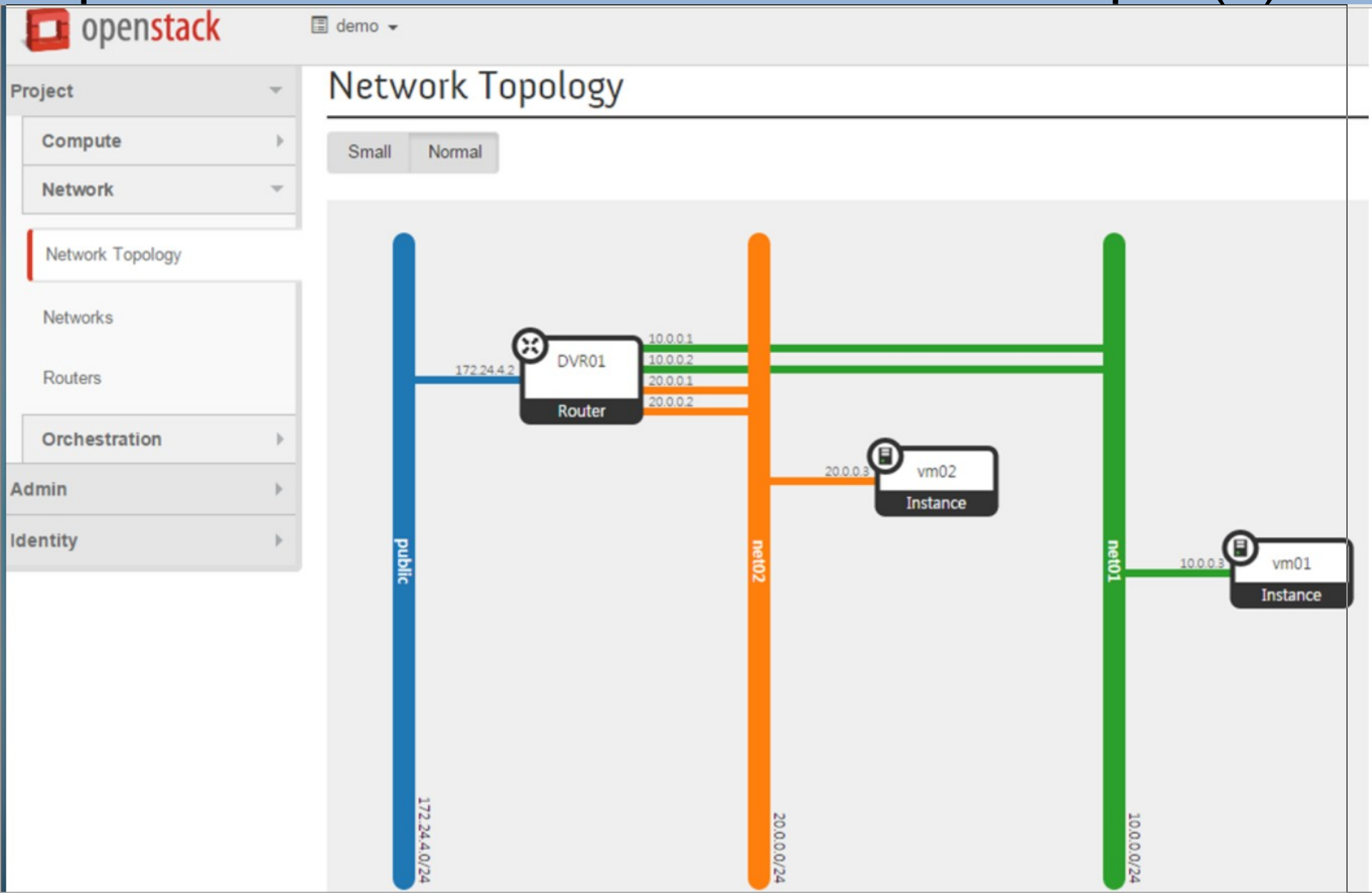
Virtualization applied to the network creates a logical software-based view of the hardware and software networking resources (switches, routers, ports, etc.). The physical networking devices are simply responsible for the forwarding of packets, while the virtual network (software) provides an intelligent abstraction that makes it easy to deploy and manage network services and underlying network resources.

Applications run on a virtual network as they where running on a physical network.

Virtual Network example

OpenStack Network Demo

OpenStack Network Virtualization Example (2)



Storage Virtualization

Storage virtualization is the process of grouping the physical storage from multiple network storage devices so that it looks like a single storage device.

⇒ **Types of storage virtualization:** they are mainly two

- **file-based storage**, which eliminates the dependencies between the data accessed at the file level and the location where the files are physically stored
- **block-based storage** which abstracts (separate) logical storage (partition) from physical storage, so that it may be accessed without regard to physical storage location or heterogeneous structure.

Virtualization vs. Cloud

Virtualization is software that manipulates hardware, while cloud computing refers to a service that results from that manipulation. You can't have cloud computing without virtualization.

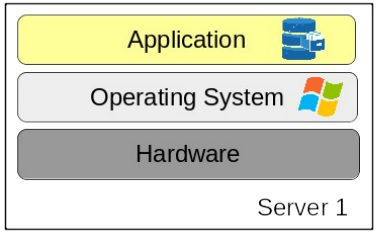
Cloud Computing and Virtualization relationship

Virtualization is a key enabler of the first four of five key attributes of cloud computing:

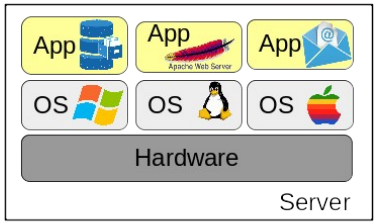
- **Service-based:** A service-based architecture is where clients are abstracted from service providers through service interfaces.
- **Scalable and elastic:** Services can be altered to affect capacity and performance on demand.
- **Shared services:** Resources are pooled in order to create greater efficiencies.
- Metered usage**:** Services are billed on a usage basis.
- **Internet delivery:** The services provided by cloud computing are based on Internet protocols and formats.

Gartner, “Server Virtualization: One Path that Leads to Cloud Computing” by Thomas J. Bittman, 10/29/2009, Research Note G00171730

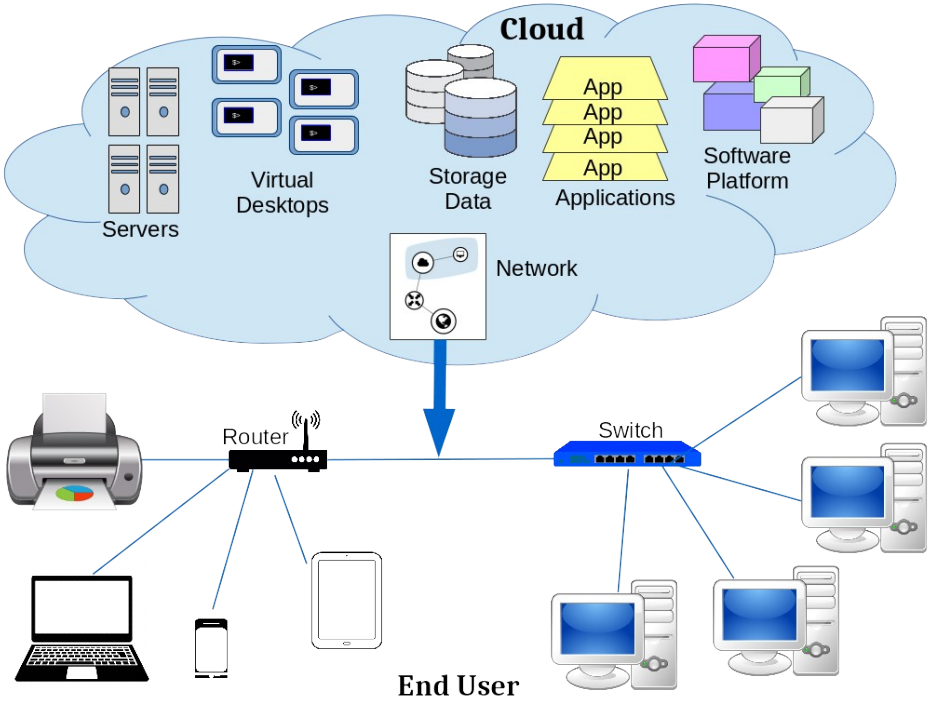
Traditional vs. Virtualization vs. Cloud arch.



Traditional



Virtualized



Cloud

Machine imaging and Virtual Appliances

Machine images are sometimes referred to as **“virtual appliances”**—systems that are meant to run on virtualization platforms.

Virtual appliance is a **virtual** machine image file consisting of a pre-configured operating system environment and a single **application**. The OS is minimized to the features needed to run the specific application hosted.

A **virtual application** is an **application** that has been optimized to run on **virtual** infrastructure. The **application** software along with just enough operating system (JeOS or "juice") is combined inside a **virtual** machine **container** in a manner that maximizes the performance of the **application**.
Most virtual appliances are configurable from a Web page.

Load balancing

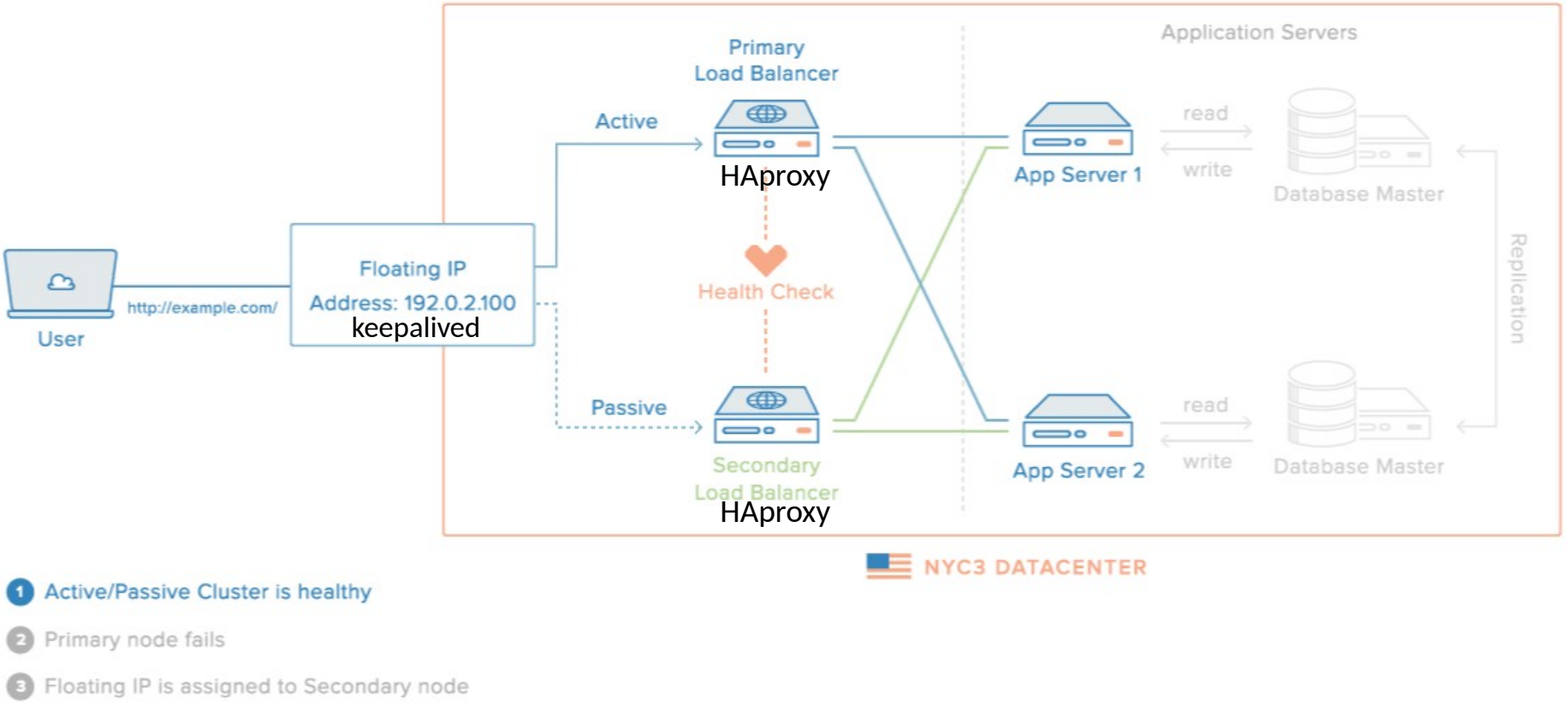
One characteristic of cloud computing is virtualized network access to a service. The user access the available resource, no matter where the resource is located.

The load balancing is a technique to distribute the workload across multiple computing resources, such as computers, a computer cluster, network links, central processing units, or disk drives.

The workload is distributed on the base of scheduling algorithms. Examples:

- Round robin - Requests are distributed across the group of servers sequentially: the next system in a list of systems gets the request
- Round robin DNS - IP addresses are assigned out of a pool of available IP addresses
- Least Connections - A new request is sent to the server with the fewest current connections to clients. The relative computing capacity of each server is factored into determining which one has the least connections
- Fastest response time - New connections are sent to the server that is currently providing the fastest response to new connections or requests

Load balancing example



Reference:
https://www.digitalocean.com/community/tutorials/what-is-load-balancing

Workload managers

The more sophisticated load balancers are workload managers. They **determine the current utilization of the resources in their pool**, the response time, the work queue length, connection latency and capacity, and other factors in order to assign tasks to each resource. Among the features you find in load balancers are polling resources for their health, the ability to bring standby servers online (priority activation), workload **weighting** based on a resource's capacity (asymmetric loading), HTTP traffic compression, TCP offload and **buffering**, **security** and **authentication**, and packet shaping using content filtering and priority queuing.