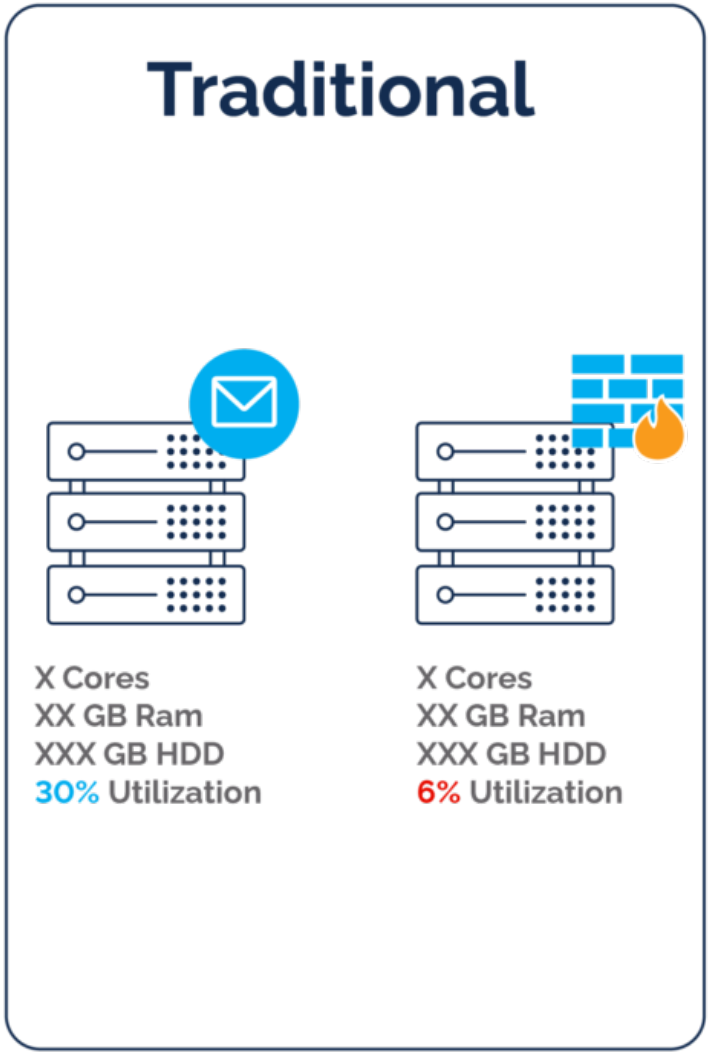
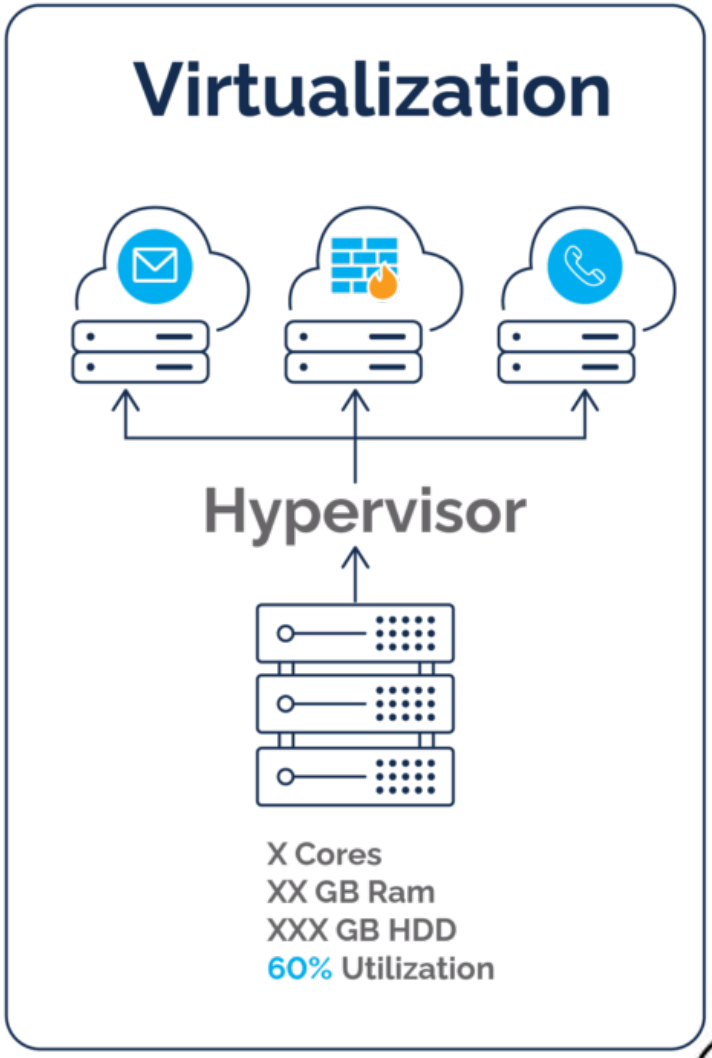
Hypervisors

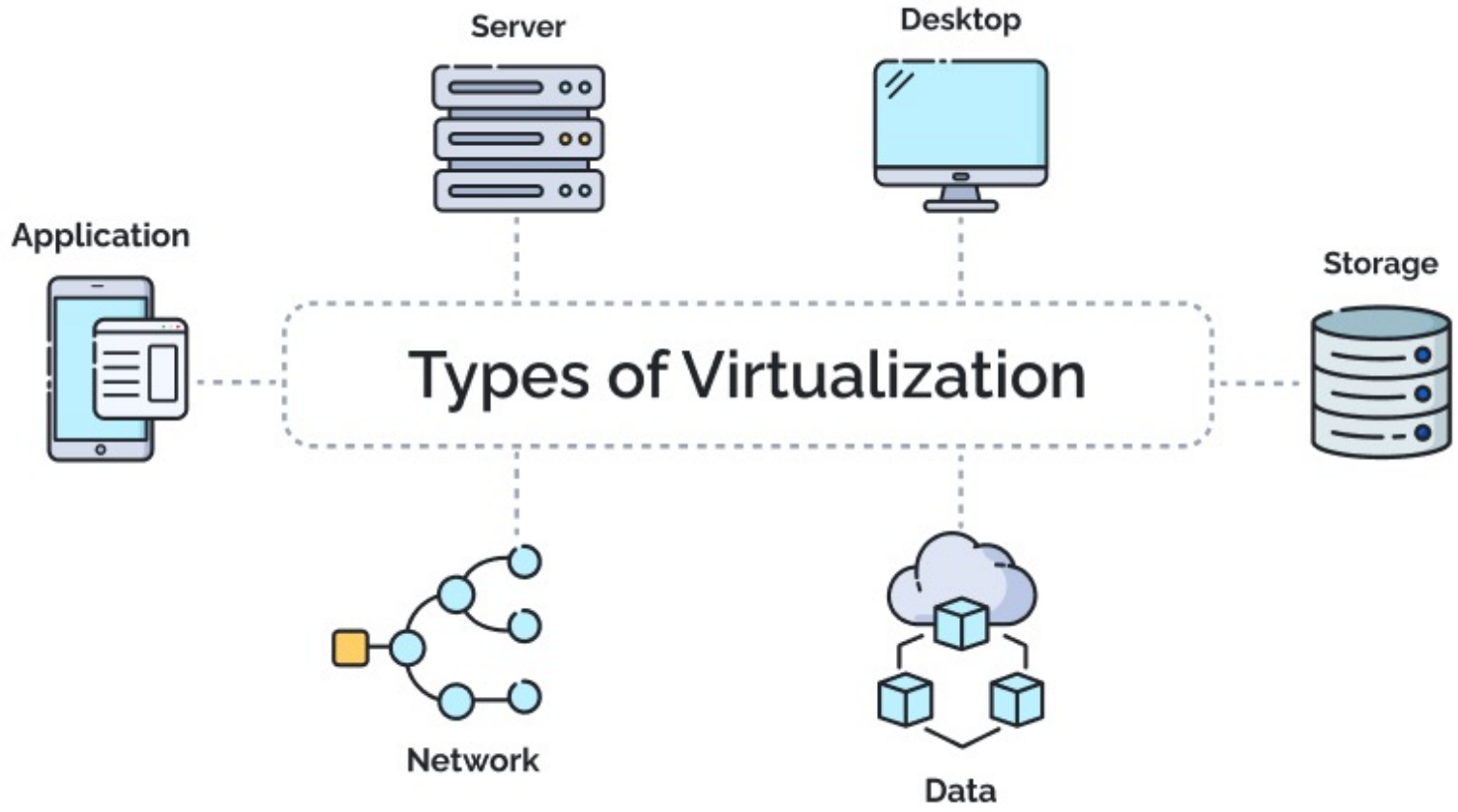
Traditional vs Virtualization





https://www.parkplacetechnologies.com/blog/traditional-data-center-vs-virtualization-differences/

Types of Virtualization



https://www.starwindsoftware.com/blog/virtualization/

Virtualization and Hypervisors

Virtualization works by **abstracting physical hardware** and devices **from the applications** running on that hardware

Hypervisors make virtualization possible by translating requests between the **physical and virtual resources**

Hypervisors support the creation and management of VMs by **abstracting** a computer's **software** from its **hardware**

Bare metal hypervisors are sometimes **embedded into the firmware** (BIOS) to enable the OS of a computer to **access and use virtualization software** (Intel-VT or AMD-V)

https://www.vmware.com/topics/hypervisor; https://www.techtarget.com/searchitoperations/tip/Whats-the-difference-between-Type-1-vs-Type-2-hypervisor

Hypervisor

Also known as a virtual machine monitor or VMM, is software that **creates** and **runs virtual machines**

A hypervisor allows a host computer to support multiple guest VMs by virtually **sharing its resources** (i.e. memory)

Hypervisors provide **greater IT mobility** since the guest VMs are independent of the host hardware

Multiple virtual machines can **run on one physical server**, **reducing space**, **energy**, and **maintenance** requirements

https://www.vmware.com/topics/hypervisor; images from Google Images

Benefits of Hypervisors

Speed

virtual machines can be **created instantly**, making it easier to provision resources as needed for **dynamic workloads**

Efficiency

running **several VMs on a physical machine is more efficient** than to run multiple underutilized physical machines

Flexibility

separates the OS from the underlying hardware, so the software no longer relies on specific hardware devices or drivers

Portability

multiple OS to reside on the **same physical server**; VMs in the hypervisor run **independently** from the physical machine; allows **shifting of workloads** and **allocate resources as needed**

https://www.vmware.com/topics/hypervisor

Types of Hypervisors

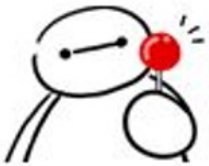
Type 1 – Bare Metal or Native Hypervisor

- **Lightweight OS** directly running on top of the host machine
- **Isolated** from the guest OS, results in **better security**
- **Better performance** than hosted hypervisors
- Mostly used by **enterprise** for **data center** computing needs

Type 2 – Hosted or Client Hypervisor

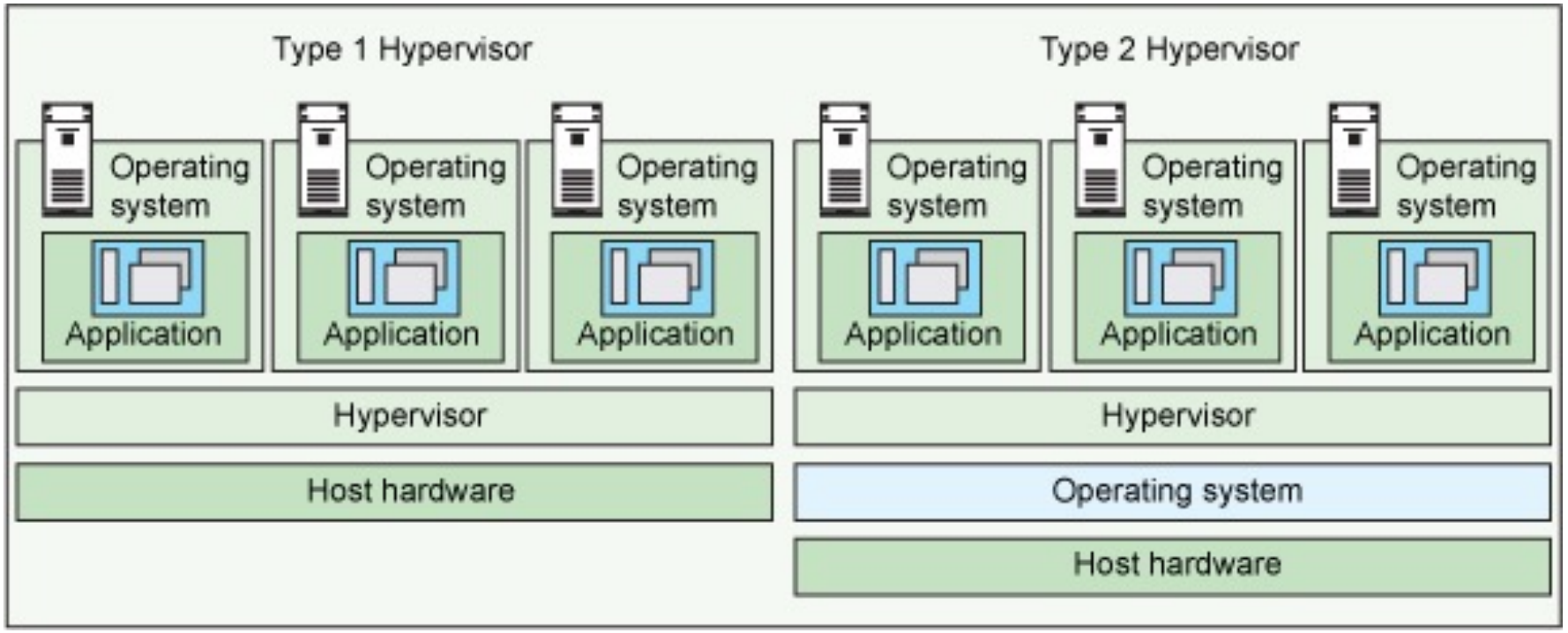
- Runs as a **software** on an OS, like other programs
- Can still run the **same** or a **different guest OS** from the host OS
- Have **higher latency** than bare metal hypervisors
- Mostly used by **end users** and **software testing**, where higher latency is less of a concern

*Note: A bare-metal server (traditional) will always provide higher performance than a virtual server sharing with other virtual servers



https://www.vmware.com/topics/hypervisor

A closer look in Hypervisors



Vmware vSphere (ESXi and vCenter)

Xen Hypervisor (Citrix Hypervisor)

KVM (Kernel—based VM)

Microsoft Hyper-V

Open Stack

VMware Workstation Player

QEMU (Quick EMUlator)

Parallels Desktop

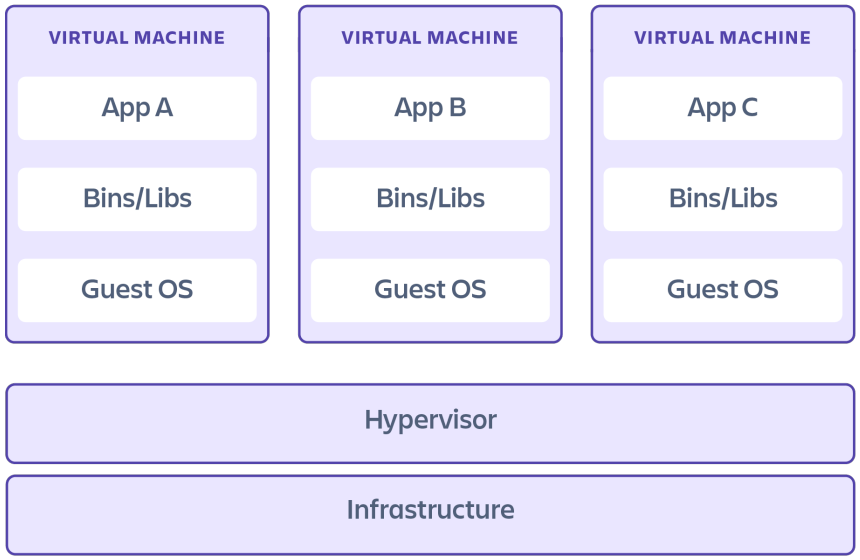
VMWare Fusion

Oracle Virtual Box

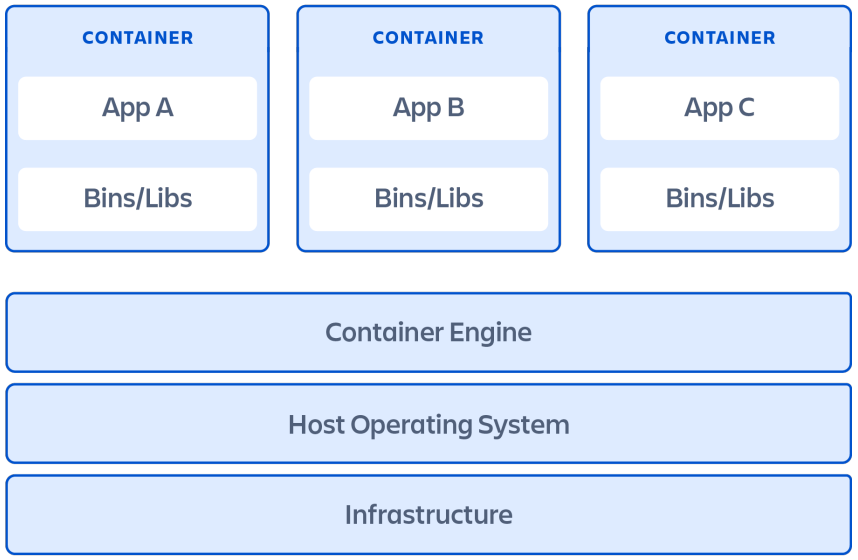
https://www.researchgate.net/publication/310620289_Performance_Analysis_of_Live_and_Offline_VM_Migration_Using_KVM; https://www.stackscale.com/blog/hypervisors/

Virtual Machines vs Containers

Virtual machines



Containers



VMWare Workstation Player

QEMU

Docker

LXC (Linux Containers)

https://www.atlassian.com/microservices/cloud-computing/containers-vs-vms

Virtual Machines vs Containers

Feature	Container	Virtual machine
Operating system	Shares the host operating system's kernel	Has its own kernel
Portability	More portable	Less portable
Speed	Faster to start up and shut down	Slower to start up and shut down
Resource usage	Uses fewer resources	Uses more resources
Use cases	Good for portable and scalable applications	Good for isolated applications

CT Use Cases:

- Web development
- Microservices architecture
- Continuous integration and delivery
- Cloud computing

VM Use Cases:

- Testing
- Development
- Isolation
- Cloud computing
- Disaster recovery

https://cloud.google.com/discover/containers-vs-vms

Container vs Hypervisor

Hypervisor

Allow an **operating system to run independently** from the underlying hardware using virtual machines

Can **run multiple OS** on top of a bare-metal hypervisor or installed on top of the OS of a hosted hypervisor

Share virtual computing, storage, memory, and network **resources**

Used to **create and run VMs**

More featured and is generally used by end users to enterprise organizations

Container

Allow **applications to run independently** of an operating system

Can **run on any operating system**, with only a container engine to run

Are **extremely portable** since in a container, an application has everything it needs to run.

Compared to VMs, containers **package an app** and its **services**

More **lightweight and portable than VMs** and used for fast and flexible app development and deployment

https://www.vmware.com/topics/hypervisor

Virtualization vs Cloud Computing

	Virtualization	Cloud
Definition	Technology	Methodology
Purpose	Create multiple simulated environments from 1 physical hardware system	Pool and automate virtual resources for on-demand use
Use	Deliver packaged resources to specific users for a specific purpose	Deliver variable resources to groups of users for a variety of purposes
Configuration	Image-based	Template-based
Lifespan	Years (long-term)	Hours to months (short-term)
Cost	High capital expenditures (CAPEX), low operating expenses (OPEX)	Private cloud: High CAPEX, low OPEX Public cloud: Low CAPEX, high OPEX
Scalability	Scale up	Scale out
Workload	Stateful	Stateless
Tenancy	Single tenant	Multiple tenants

https://www.redhat.com/en/topics/cloud-computing/cloud-vs-virtualization

Virtualization Enterprise Readiness Capabilities

Fast Provisioning, Ease of Deployment, and Agility

Manageability, Flexibility, and Automation

Support, Maintainability, and Serviceability

Performance and Scalability

Reliability, Availability, and Robustness

Security and Accountability

Partitionability, Isolation, and Hardware Independence

https://www.ibm.com/developerworks/cloud/library/cl-hypervisorcompare/
https://www.vmware.com/in/solutions/virtualization.html

Hypervisors