



MTIG

MAGÍSTER EN TECNOLOGÍAS
DE INFORMACIÓN Y GESTIÓN

INF3591

Cloud Computing

Clase 8
Virtualización

Cristian Daniel Ruz Ruz - cruz@ing.puc.cl
Germán Leandro Contreras Sagredo - glcontreras@uc.cl

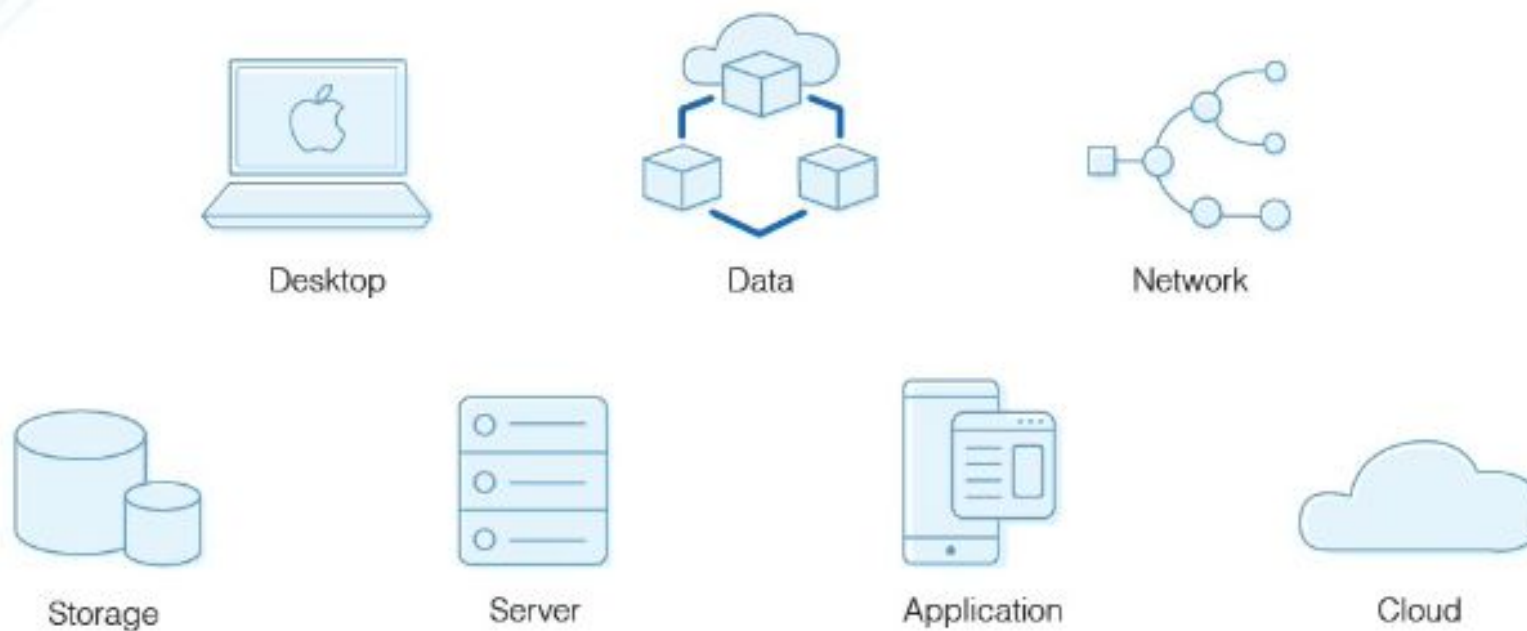
Planificación

Calendario actual

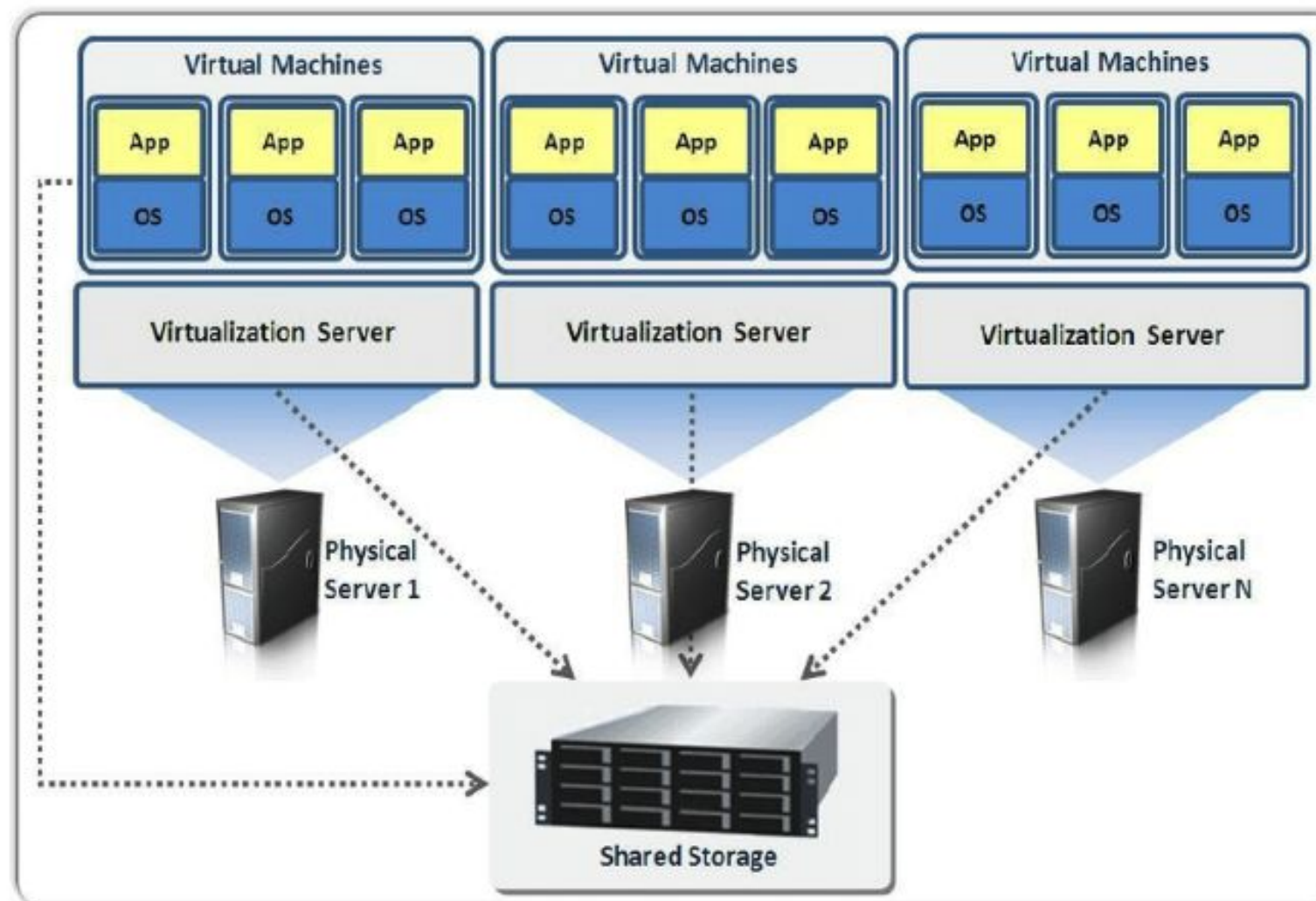
Fecha	Temario	Evaluación
Jueves 24 de marzo	Clase 1. Introducción; Conceptos básicos de Cloud Computing.	-
Jueves 31 de marzo	Clase 2. Casos de uso, seguridad, <i>billing</i> .	-
Jueves 7 de abril	Clase 3. Almacenamiento de datos en la nube.	-
Miércoles 13 de abril	Clase 4. Bases de datos en la nube.	-
Jueves 21 de abril	Clase 5. Repaso con casos reales.	Control 1 - Taller evaluado 1
Jueves 28 de abril	Clase 6. Cloud Computing con servidor.	Taller evaluado 2
Jueves 5 de mayo	Clase 7. Cloud Computing sin servidor.	Taller evaluado 3
Miércoles 18 de mayo	Clase 8. Virtualización. Vía Zoom.	Taller evaluado 4
Jueves 19 de mayo	Clase 9. Repaso con casos reales.	Control 2

Tipos de virtualización

Types of Virtualization

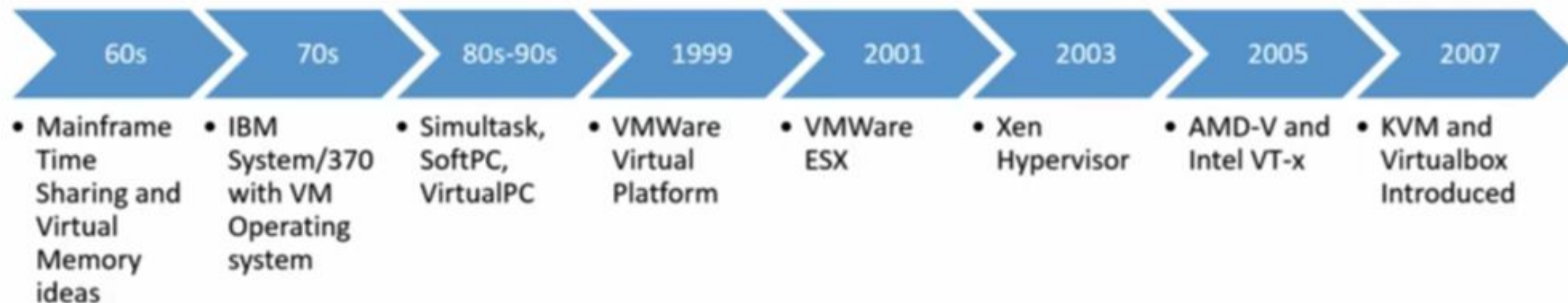


Virtualización



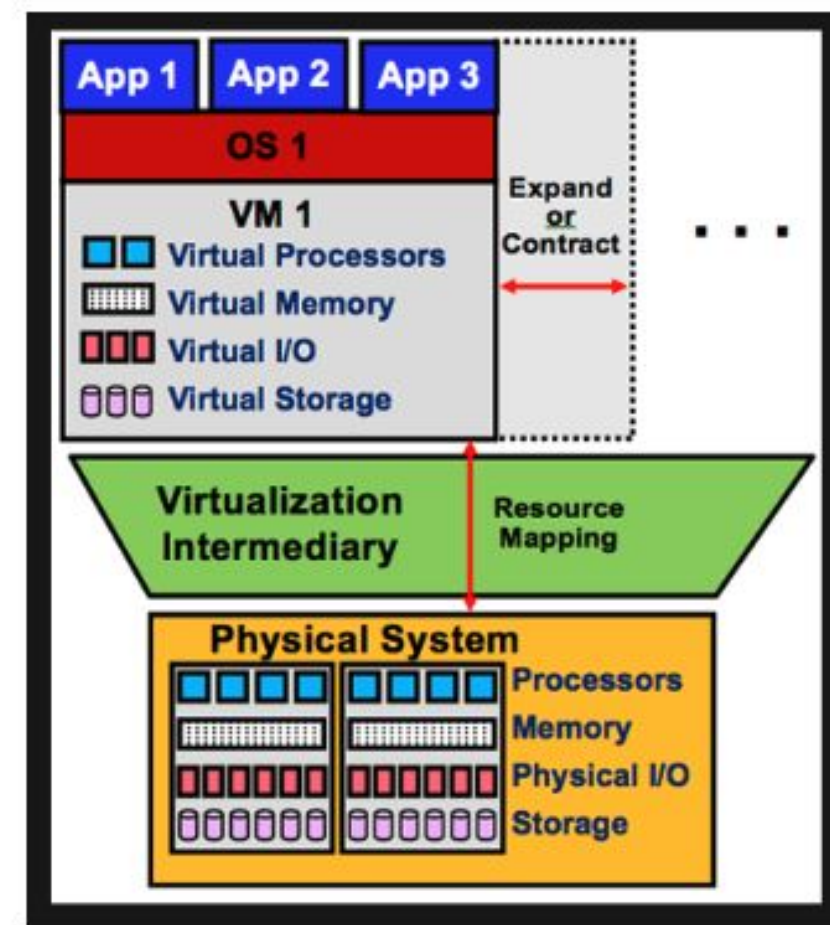
Virtualización

Permite compartir recursos físicos en la forma de recursos virtuales, manteniendo niveles de rendimiento, flexibilidad y aislamiento.

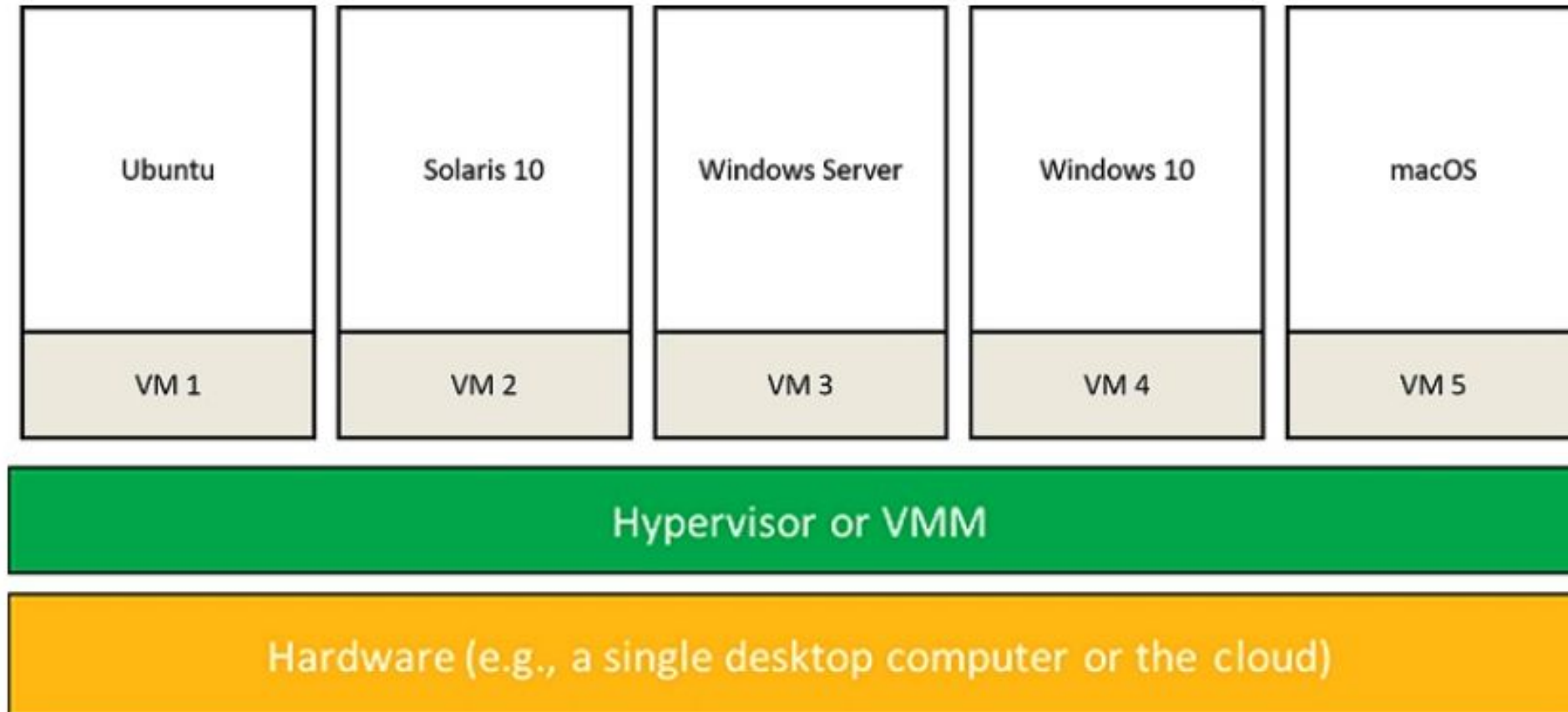


¿Cómo se ve una máquina virtual?

- Soporta el modelo Cloud
- Permite elasticidad
- Permite sandboxing de recursos
- Mejora la utilización de los sistemas y la reducción de energía
- Permite ambientes con distintos sistemas operativos



Múltiples sistemas operativos

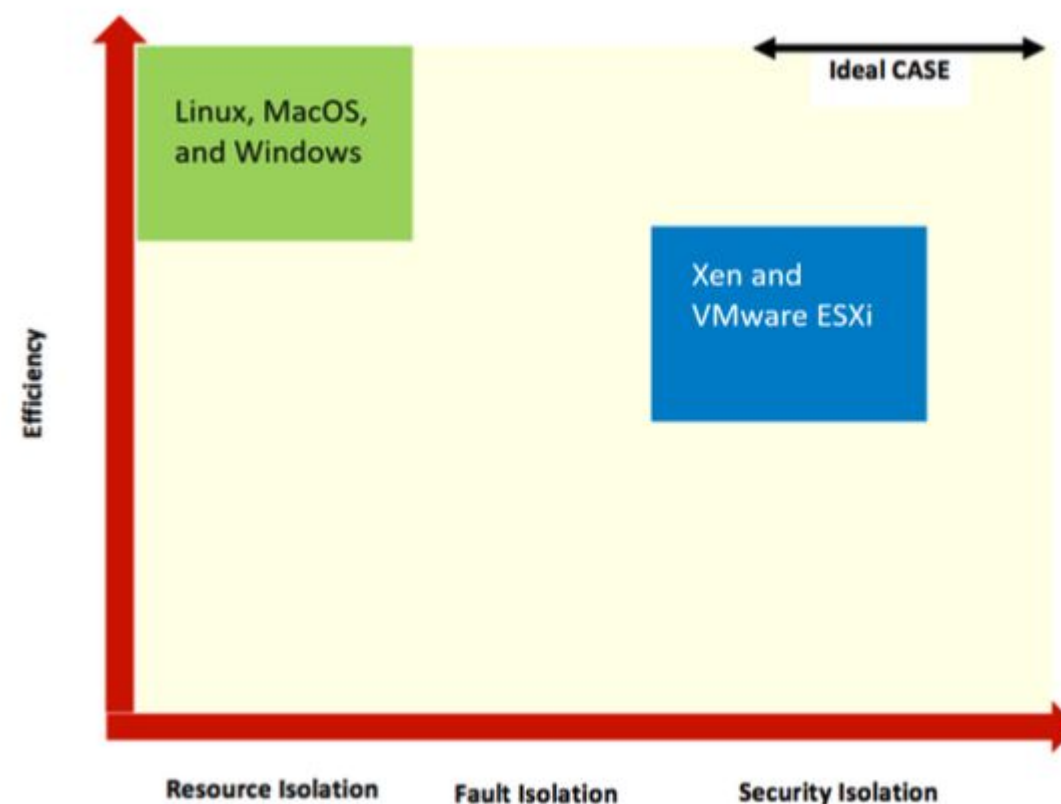


Tipos de aislamiento

Fault isolation. Una falla en un programa no debería afectar a otros programas.

Resource isolation. Capacidad de establecer o modificar los recursos asignados a los programas.

Security isolation. Una aplicación no debe poder acceder a datos (archivos, direcciones de memoria, puertos) de otra aplicación

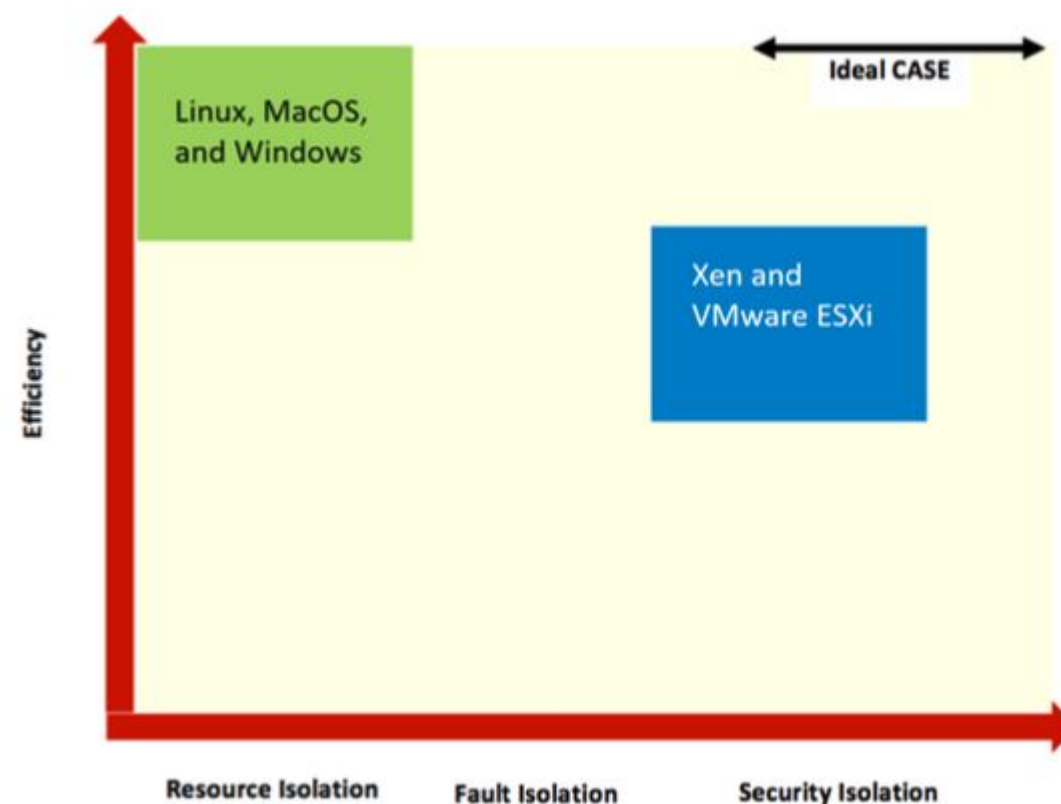


Tipos de aislamiento

SO tradicionales solo proveen resource isolation

Hypervisors y VMs pueden proveer las tres, pero el rendimiento de una VM es menor a una máquina física

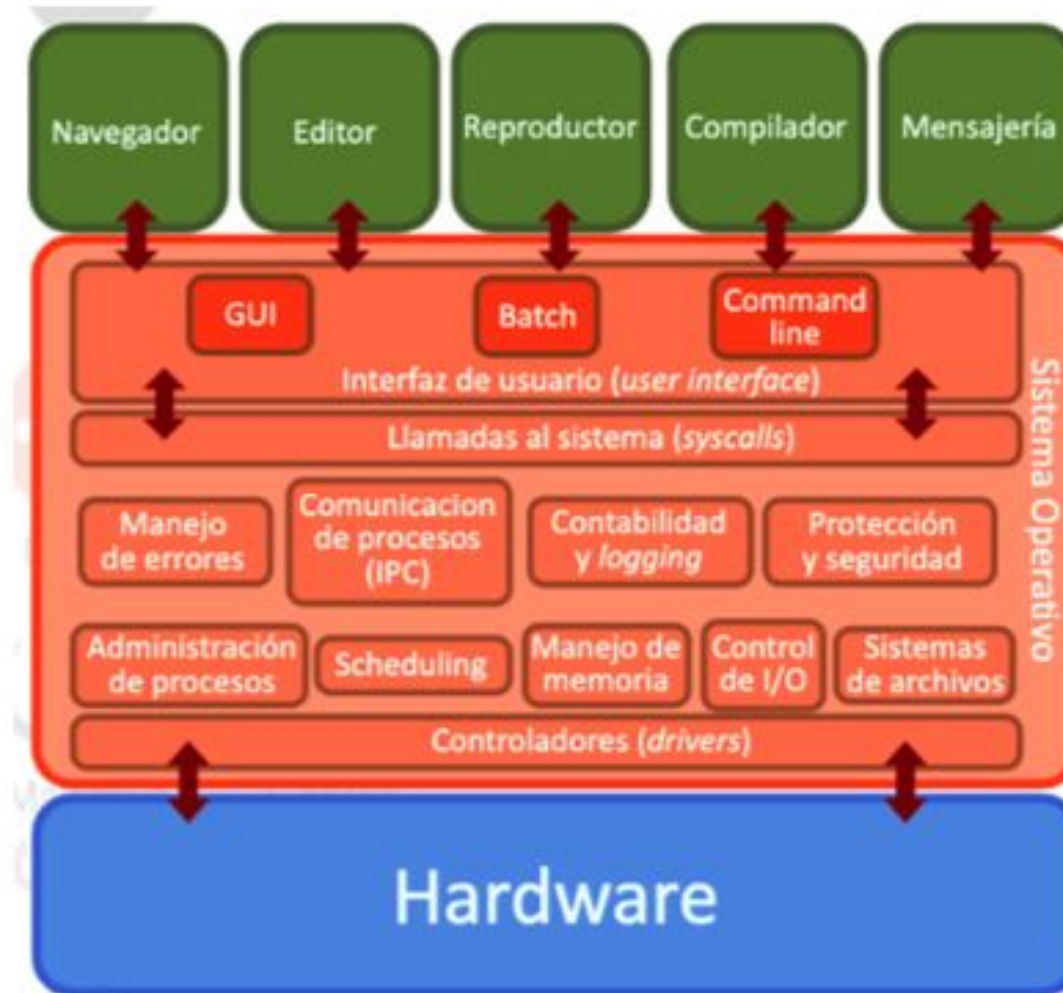
Acciones de un sistema operativo se ven replicadas en los *hypervisor*



Niveles de abstracción

SO proveen niveles de abstracción para proteger al usuario (y al sistema)

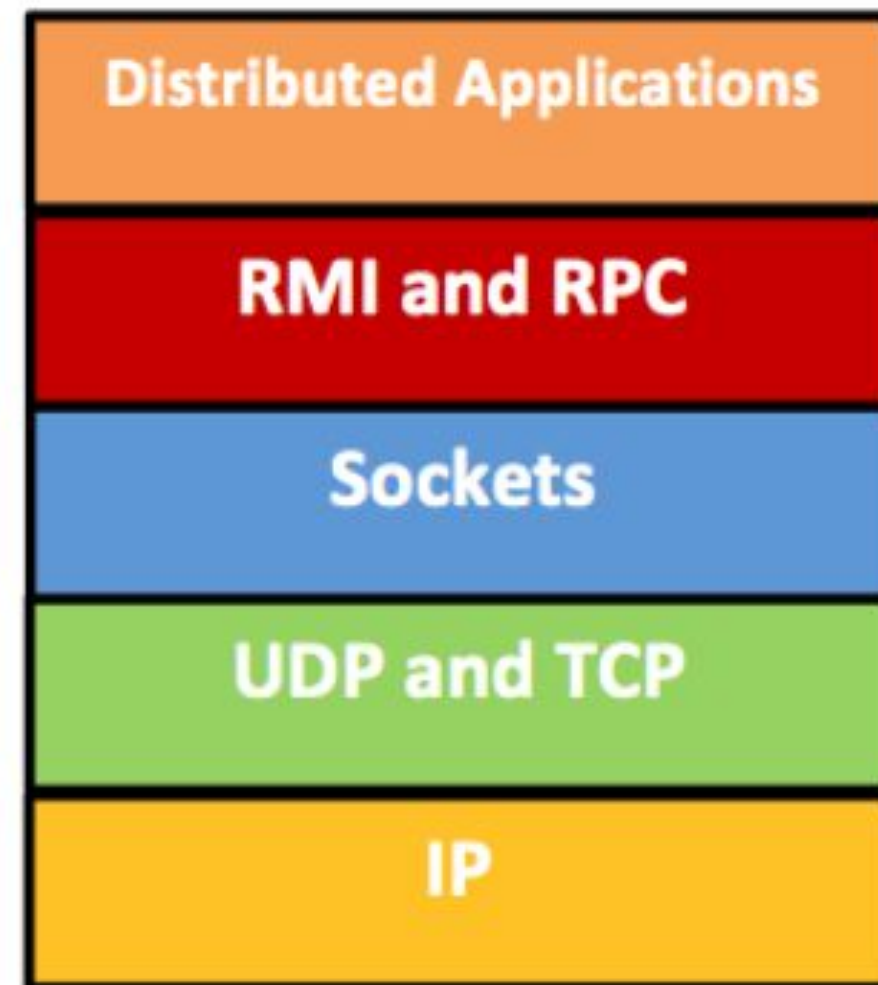
Abstracciones de un sistema operativo: procesos, archivos



Niveles de abstracción

En sistemas distribuidos también hay niveles de abstracción.

Programas se comunican a través de interfaces de alto nivel.



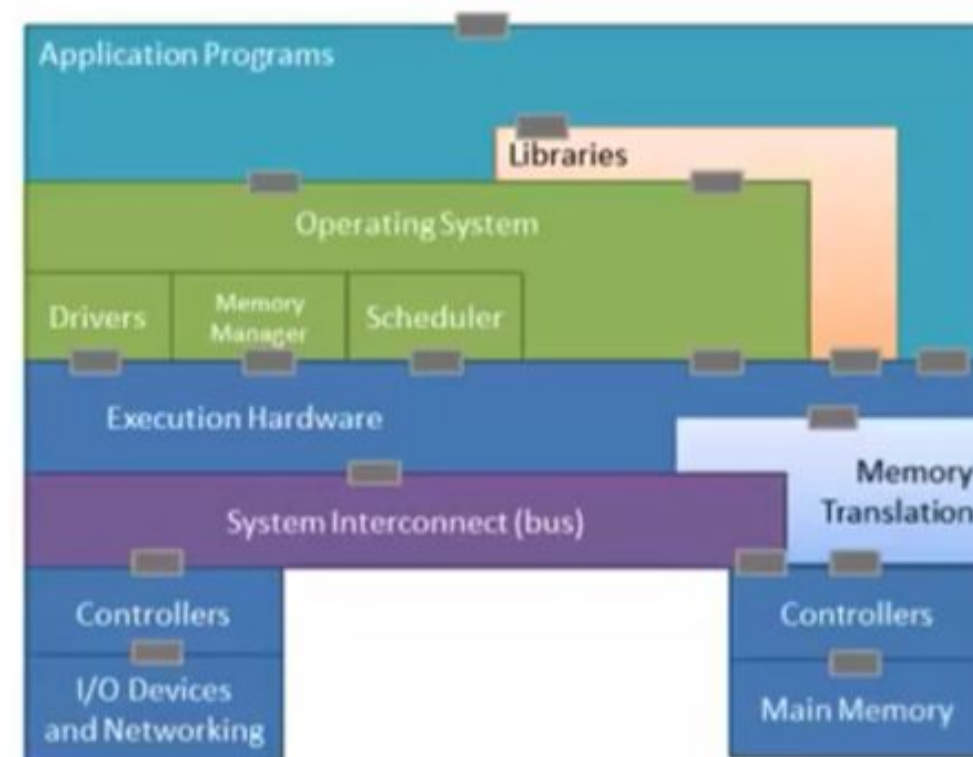
¿Por qué es posible?

Componentes de sistema se comunican mediante interfaces

API (Application Program Interface) es un ejemplo a nivel de software

ABI (Application Binary Interface) es como una API pero a nivel de código binario

ISA (Instruction Set Architecture) interfaz entre el hardware y el software



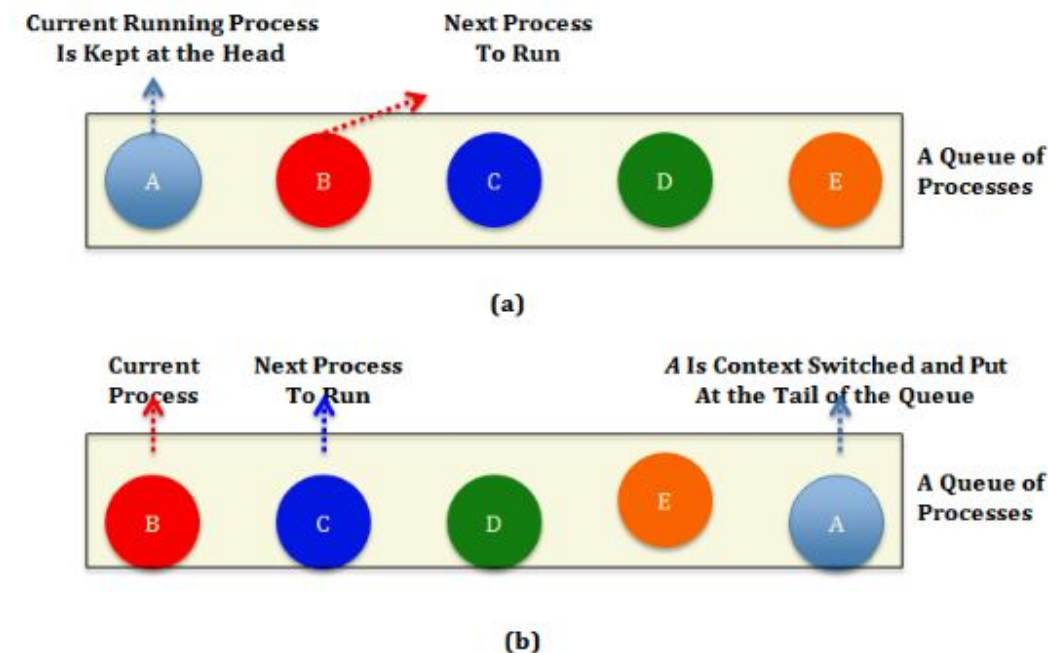
Compartiendo recursos de un procesador

Sistemas operativos tradicionales comparten recursos

Métodos de virtualización siguen la misma lógica

Compartir en tiempo: permitir que un componente tome turnos para usar un recurso. Ej: CPU (core)

Compartir en espacio: permitir que un recurso sea usados por dos o más elementos simultáneamente. Ej: memoria



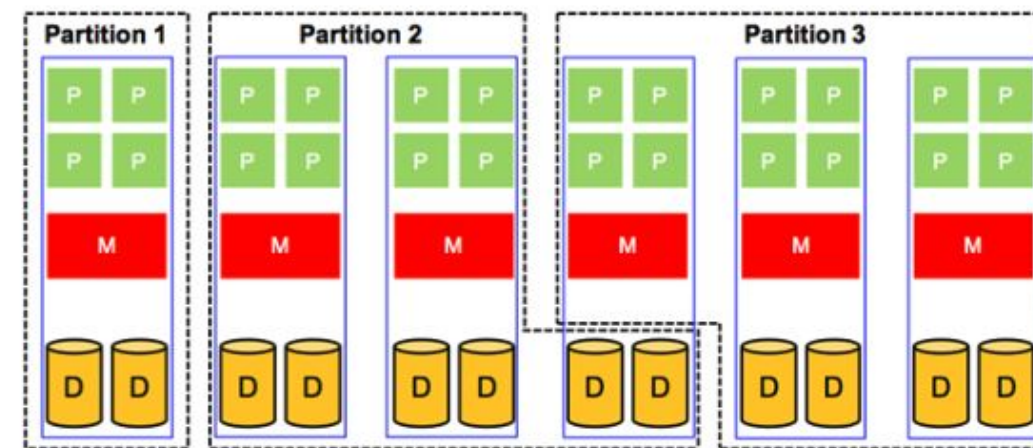
Compartiendo recursos de un multiprocesador

Sistemas modernos son multicore (multinúcleo)

Ejemplo 1: multiprocesador compartido en espacio (3 parti

Ventajas:

- cada partición posee sus recursos físicos y los accede rápidamente
- simple de implementar
- una partición no puede bloquear a otra

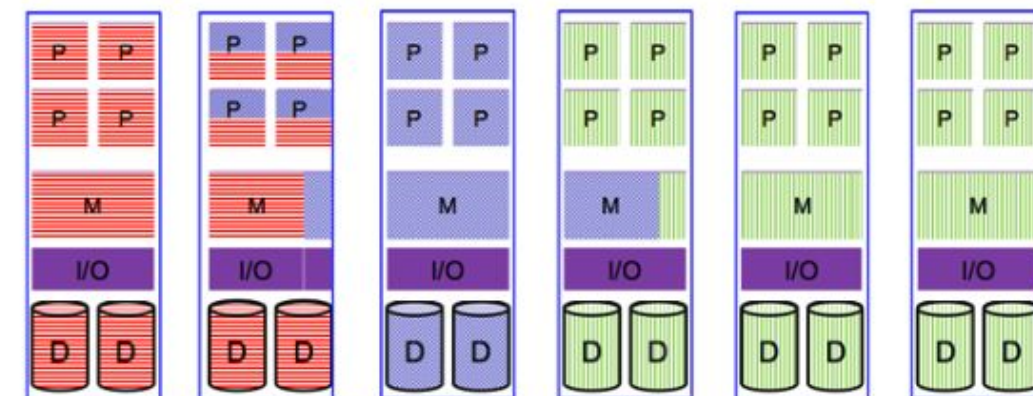


Desventajas:

- Solución subóptima de utilización

Ejemplo 2: multiprocesador compartido en tiempo.

- Mejor utilización
- Mayor complejidad para asignar y proteger recursos

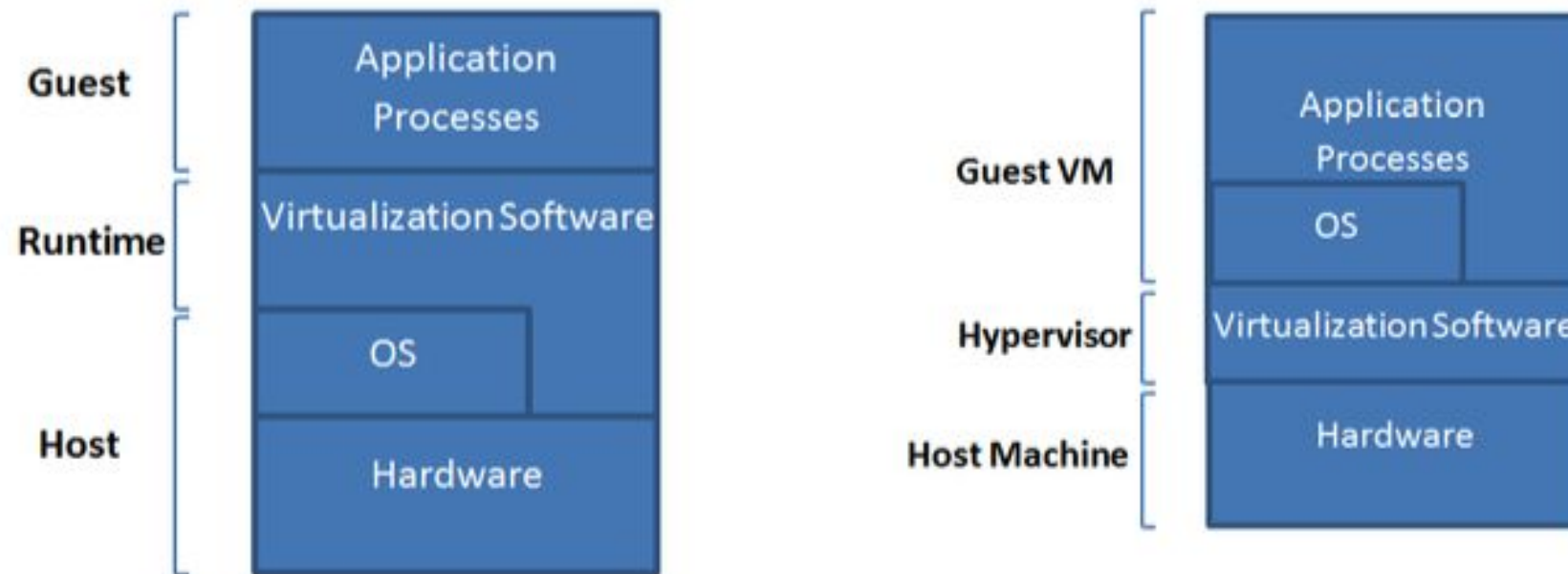


Entonces, ¿cómo se virtualiza?

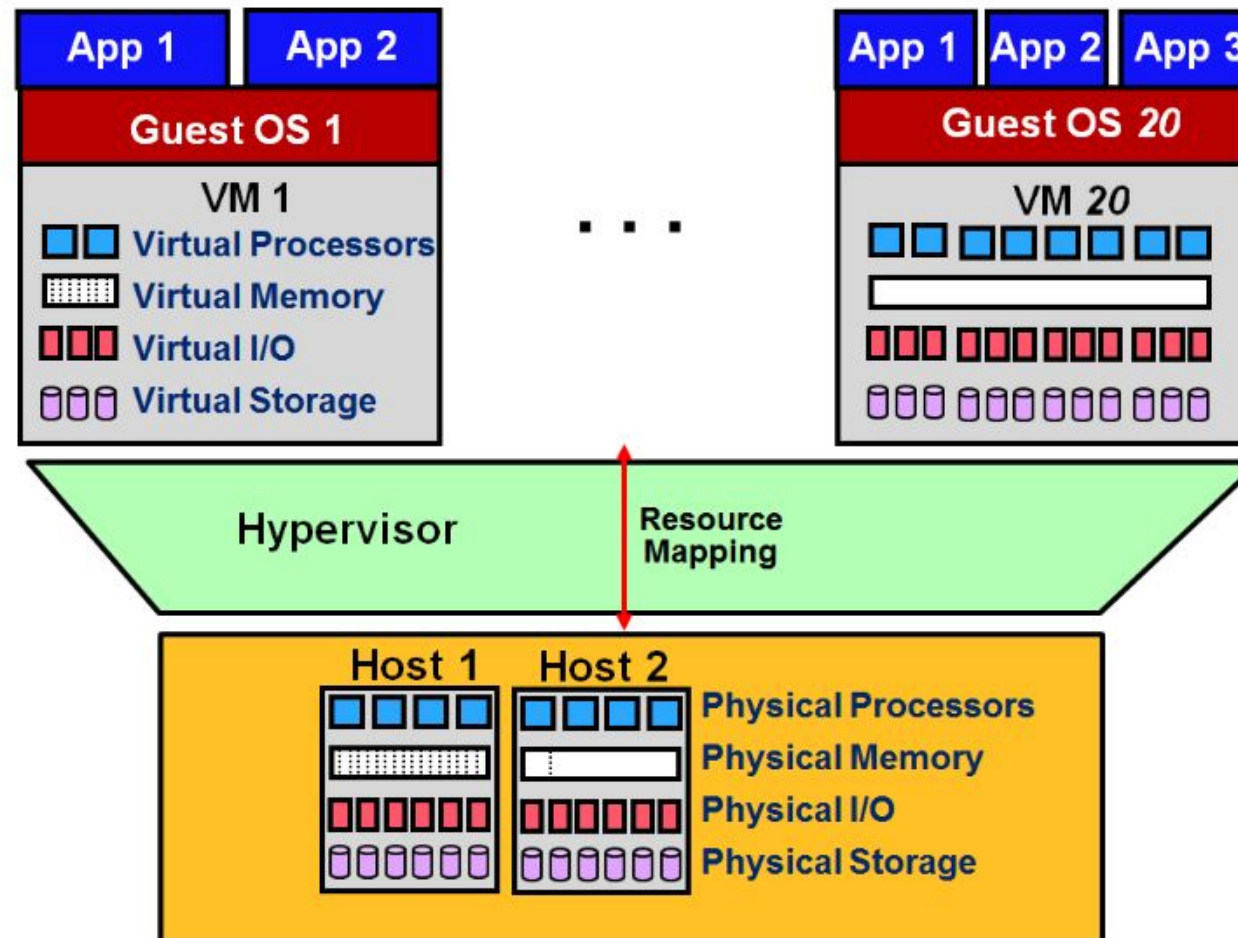
Versión a nivel de procesos: emulación

Versión a nivel de sistema: hypervisor Paralells, Hyper-V, VirtualBox

Un sistema podría tener más virtual CPUs (vCPU) que el sistema físico



Entonces, ¿cómo se virtualiza?



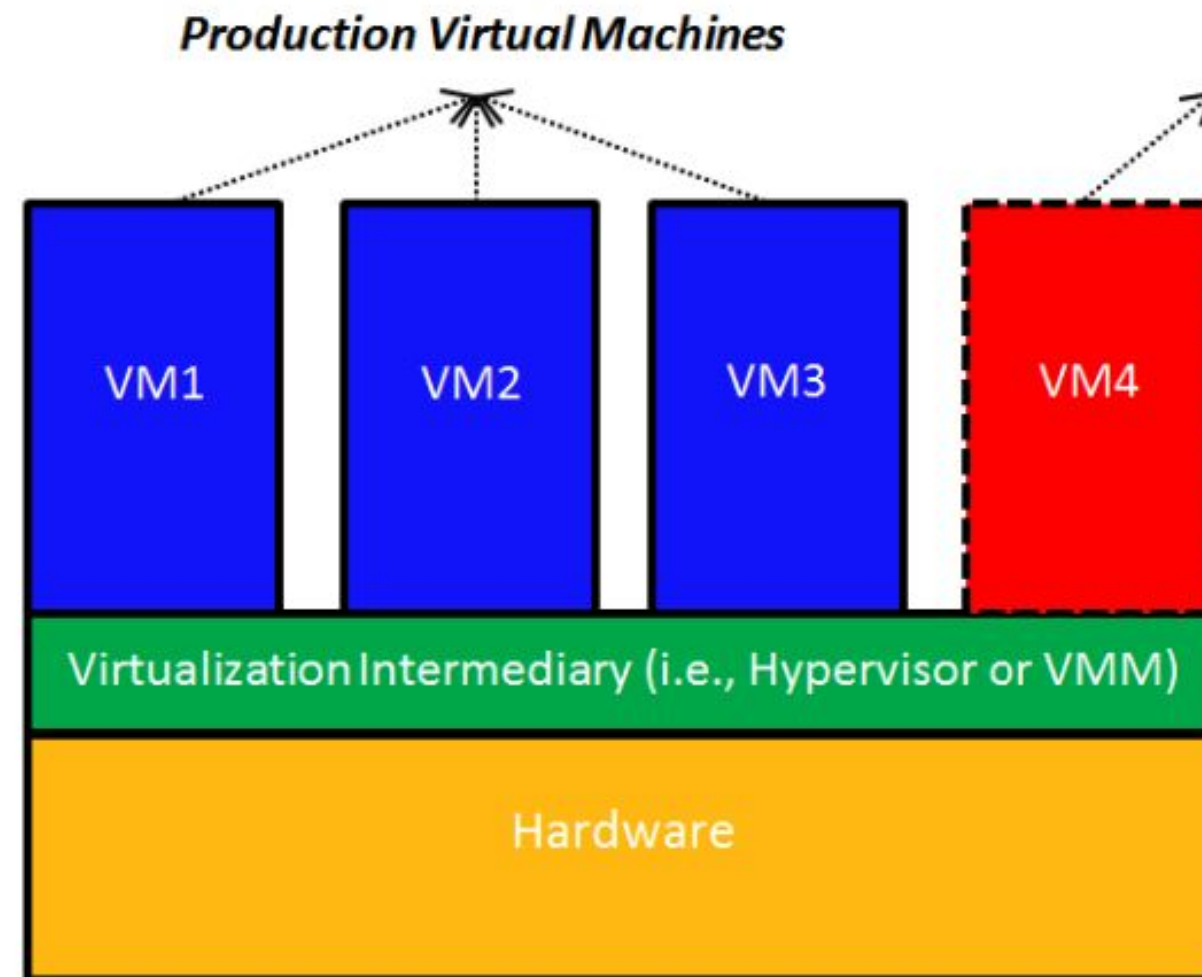
Aplicaciones

Es posible tener VMs en producción y otras en sandbox

Sandboxing

Monitoreo de ataques y preparación
antes de pasar a producción

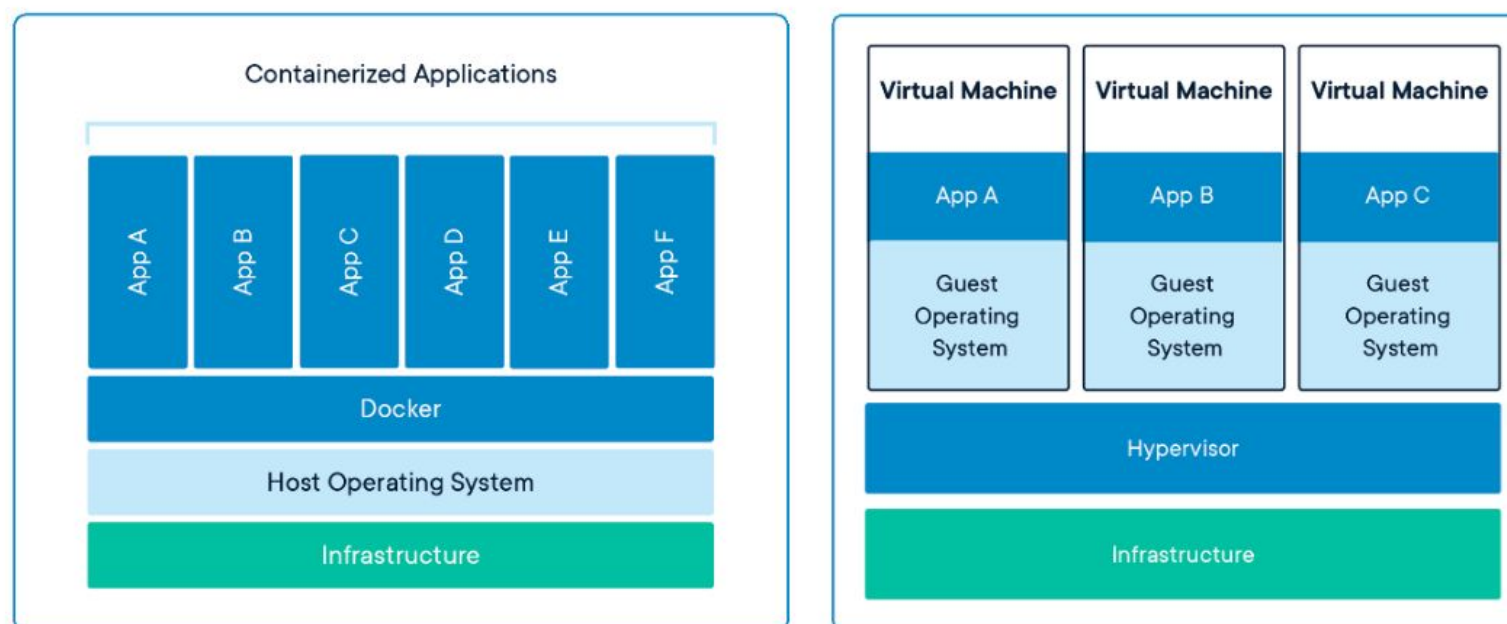
Se hace un clon de la máquina real para
inspeccionar paquetes



Containers y VMs

Container: unidad de software que empaqueta código y sus dependencias, de manera que la aplicación ejecute de manera rápida y confiable entre distintos ambientes. El contenedor contiene todo lo necesario para ejecutar una aplicación.

Sistema puede almacenar más contenedores que VMs

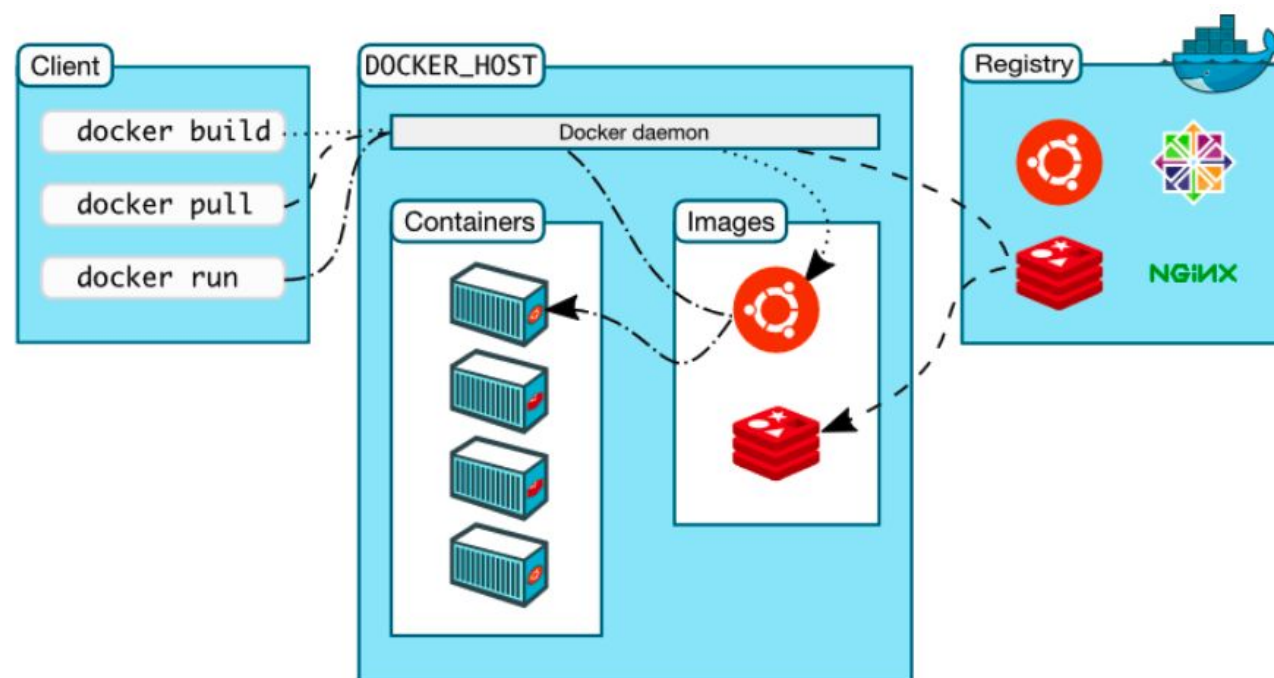


Containers y VMs

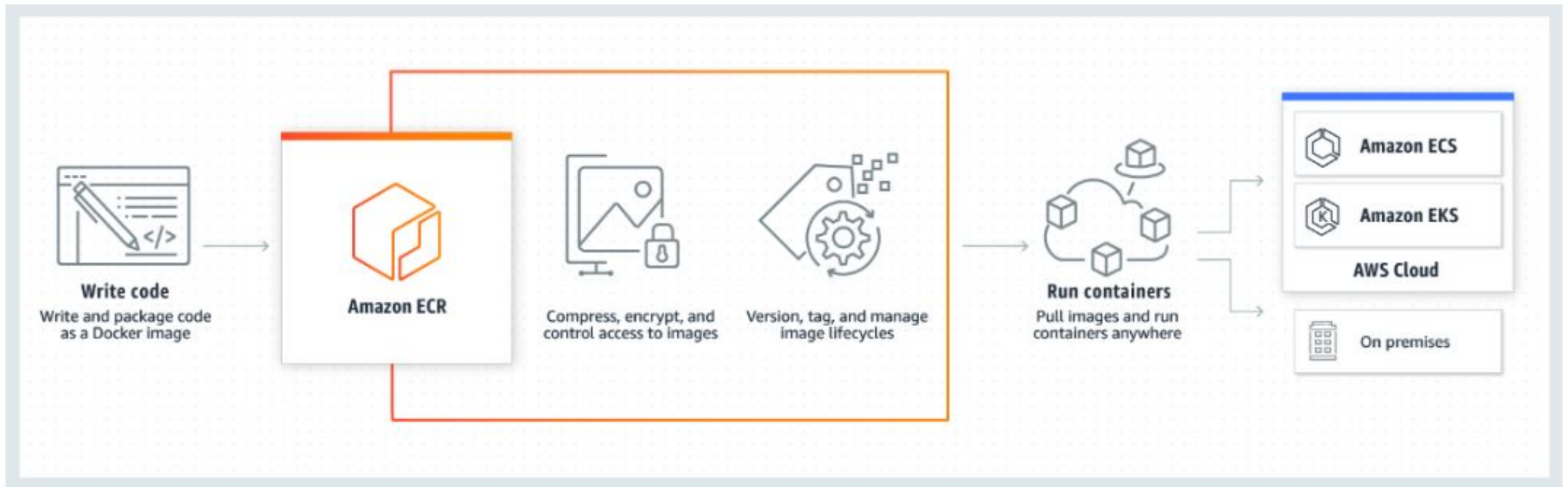
Container image: “plantilla” para crear un container

Docker client: controla ciclo de vida de containers

Registry: repositorio de imágenes (Ej: DockerHub, Amazon ECR, Azure Container Registry, Google Container Registry)



Containers y VMs



Virtualización para redes

Nodos, hosts, VMs se identifican en la red por IP y MAC

Pero también podemos separar enlaces (flow ID), y redes (VLAN, VPN) para aislamiento o monitoreo

Desafíos en un datacenter Cloud

La topología de red del datacenter debe ser capaz de soportar la demanda actual y futura Debe ser capaz de maximizar el rendimiento (respuestas a solicitudes), y minimizar el costo en hardware

Ofrecer garantías de disponibilidad e integridad en presencia de falla.

Permitir reducir costos de operación, y ser environment-friendly

¿Qué tecnología de red? (ethernet, infiniband, myrinet)

¿Cómo hago una asignación eficiente de IPs que facilite encontrar una dirección (routing)?

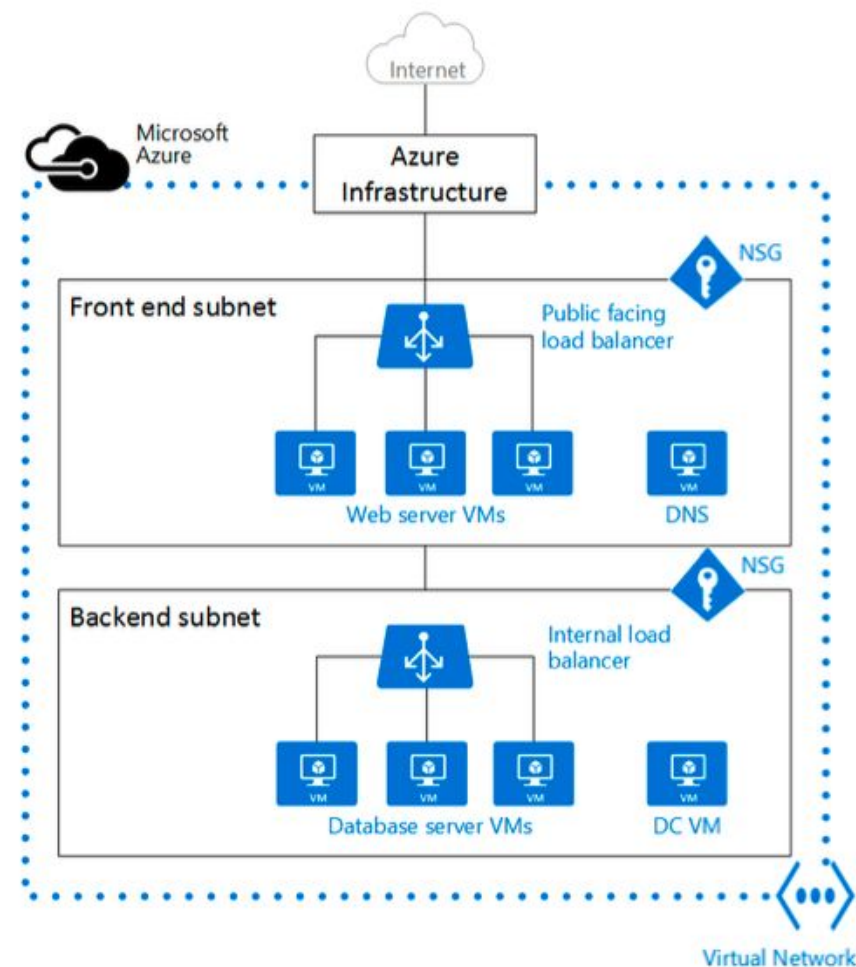
Virtualización para redes

Ejemplo: Virtual Network en Azure (VNet)

Virtual Load Balancer, con IP pública Load balancer y VMs en una subred

Backend: bases de datos y otros recurso en otra subred

Cada subred posee diferentes reglas de acceso de tráfico: Virtual Firewall





MTIG

MAGÍSTER EN TECNOLOGÍAS
DE INFORMACIÓN Y GESTIÓN

INF3591

Cloud Computing

Clase 8
Virtualización

Cristian Daniel Ruz - cruz@ing.puc.cl
Germán Leandro Contreras Sagredo - glcontreras@uc.cl